Distributed optimization with quantization for computing Wasserstein barycenters

Roman Krawtschenko[1], César A. Uribe[2], Alexander Gasnikov[3], Pavel

Dvurechensky[4]

[1] Humboldt University
    Rudower Chaussee 25
    12489 Berlin
    Germany
    E-Mail: r.kravchenko90@gmail.com

[2] Massachusetts Institute of Technology
    77 Massachusetts Ave
    Cambridge
    MA 02139
    USA
    E-Mail: cauribe@mit.edu

[3] Moscow Institute of Physics and Technology
    Institutskiy Pereulok, 9
    Dolgoprudny, Moscow Region, 141701
    Russian Federation
    E-Mail: gasnikov@yandex.ru

[4] Weierstrass Institute
    Mohrenstr. 39
    10117 Berlin
    Germany
    E-Mail: pavel.dvurechensky@wias-berlin.de

Distributed optimization with quantization for computing Wasserstein barycenters

Roman Krawtschenko, César A. Uribe, Alexander Gasnikov, Pavel Dvurechensky

Abstract

We study the problem of the decentralized computation of entropy-regularized semi-discrete Wasserstein barycenters over a network. Building upon recent primal-dual approaches, we propose a sampling gradient quantization scheme that allows efficient communication and computation of approximate barycenters where the factor distributions are stored distributedly on arbitrary networks. The communication and algorithmic complexity of the proposed algorithm are shown, with explicit dependency on the size of the support, the number of distributions, and the desired accuracy. Numerical results validate our algorithmic analysis.

1 Introduction

Optimal transport (OT) has become an important part of the modern machine learning for its ability to take into account the geometry of the data into the computations. Applications range from image retrieval [54] and image classification [16] to Generative Adversarial Networks [3]. An immediate use of Wasserstein distances is the definition of Frechet means of distributions [1, 52], which is usually called the Wasserstein barycenter (WB) [8, 28]. Informally, WB allows us to define for example, an average image when interpreted as a discrete probability distribution. Such flexibility along side the geometric and statistical properties of WB has been a large number of applications, e.g., image morphing and image interpolation of natural images [57], averaging atmospheric gas concentration data [6], graph representation learning [58], fairness in ML [13], geometric clustering [48], Bayesian Learning [4], stain normalization and augmentation [49], probability and density forecast combination [15], multimedia analysis and fusion [36], unsupervised multilingual alignment [44], clustering patterns for COVID-19 dynamics [51], channel pruning [56], and many others.

As OT and, in particular, the WB framework is used in machine learning applications, the scale of the problem increases as well. Thus, the geometric and statistical advantages of WB come with a high computational cost. For example, calculating the WB of two images of $1M$ pixels translates into a large-scale optimization problem, where the definition of the Wasserstein distance itself contains a minimization problem with approximately 10^{12} variables. Approximating WB with high accuracy requires processing a large number of samples to get good statistical estimates, and dependencies on the distribution support, number of distribution, and desired accuracy are usually high [10]. Therefore, the complexity and scalability of approximating WB is also a main research thrust within the ML community [29, 45, 53, 45], where some heuristics [11], strongly-polynomial 2-approximation [9], fast computation [34, 32], saddle-point [61], and proximal methods [60, 65] have been proposed, as well as approaches based on multimarginal optimal transport [46, 62].

In addition to scale, modern ML applications require other considerations. In particular, decentralized approaches [59] have recently emerged over centralized ones due to its ability to take into account data ownership, privacy, fault tolerance, and scalability, e.g., parameter server [17], federated

learning [40, 47, 37], among others. In this paper, we are focused on: • *Communication-efficient*, • Stochastic (*semi-discrete*), and • *Decentralized* computation of Wasserstein barycenters. *Decentralized* because we assume it is not possible to store the whole dataset into one machine, and probability distributions are stored locally on a number of agents or workers connected over some arbitrary network. The network structure defines some limited communication capabilities between the nodes. *Semi-discrete* because in contrast with most of the available WB computation approaches, we assume the base probability distributions are continuous, while we want to recover their best discrete (finite) barycenter. Moreover, the semi-discrete setup implies that agents or machines are oblivious to their local probability distribution and can learn about it from the realization of an associated random variable. Finally, *Communication-efficient* because we exploit the fact that the gradients of the associated dual formulation [55, 64] of the WB lie in the probability simplex, i.e., is a discrete distribution. Thus, we propose a quantized communication approach that drastically reduces the network's communication load by sampling from such a discrete distribution, creating a sparse approximation of the stochastic gradient. Applied to the WB problem, this means that instead of sending the whole distribution, e.g., an image, in each iteration, agents share a histogram from a finite number of samples drawn from its current WB estimate.

The main **contributions** of this paper are:

- ■ We propose a general quantized communication-based algorithm for the distributed computation of semi-discrete WB over networks.
- ■ We analyze the communication and communication complexity of the proposed algorithm in two setups: 1) The number of samples from the distribution and the samples from the gradient are allowed to increase with time. 2) The number of samples is fixed to some predefined constant.
- ■ Our specific results for the computation of WB are based on a new general primal-dual analysis of the accelerated stochastic gradient descent method, for which we provide a new convergence rate and sample complexity analysis.
- ■ We provide numerical experiments that empirically support our theoretical finding on the convergence of the proposed algorithm.

Related works: Efficient gradient quantization was studied in [35, 39, 38]. However, only non-accelerated stochastic gradient approaches are available. A similar approach to ours was recently studied in [5], which uses SGD and sampling scheme in the discrete setting. In contrast, we use AGM (Accelerated gradient method) instead of SGD (Stochastic gradient descent), and we study the WB problem in the semi-discrete setting and give precise computation and communication complexity in terms of the dimension n of the barycenter space, the number of considered distributions m, and the network architecture. Centralized Semi-discrete WB [14] was studied in [59], where the convergence of the discrete approximations of the continuous measures was studied. Decentralized discrete WB communication and computation complexity was studied in [63], and their corresponding semi-discrete analysis is available in [21], both approaches with no quantization for communication efficiency. Empirical studies of the approximation of the fully continuous barycenter are recently available in [43]. To the best of the authors' knowledge, the best estimate of the complexity of computing the WB in the centralized discrete setup is available in [20].

This paper is organized as follows. Section 2 introduces the problem of distributed WB computation. Section 4 presents the general primal-dual accelerated gradient method. Section 3 analyzes the primal-dual properties of the WB problem. Section 5 states our proposed algorithm and main results. Section 6 shows the numerical results. Finally, Section 7 provides conclusions and future work.

Notations: We denote $\mathcal{M}^1_+(\mathcal{X})$ as the set of positive Radon probability measures on a metric space $\mathcal{X}$. $Y \sim \mu$ means that random variable Y is distributed according to measure μ. Let e_i denote the
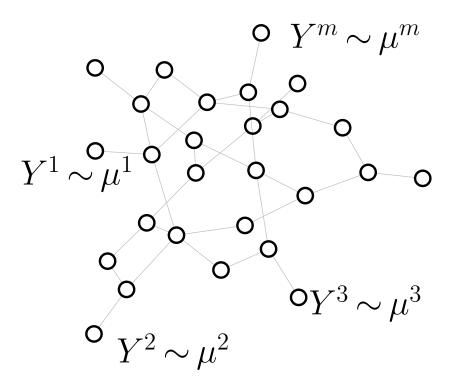
Figure 1: A network of agents, where each agent is able to sample from a different distribution, i.e., agent $i \in V$ samples from μ^i.

i-th unit vector. Let W be a positive semi-definite matrix with maximal and minimal nonzero eigenvalues denoted by $\lambda_{\max}(W)$ and $\lambda_{\min}^+(W)$. We denote the condition number of a matrix W by $\chi(W) \triangleq \lambda_{\max}(W)/\lambda_{\min}^+(W)$. $S_1(n) \triangleq \{p \in \mathbb{R}^n : \sum_{l=1}^n p_l = 1\}$ denotes the probability simplex of dimension n. $\tilde{O}(\cdot)$ denotes the complexity up to polylogarithms. δ denotes the Dirac mass and $[p]_i$ denote the i-th component of the vector p. $\otimes$ denotes the Kronecker product of matrices and $\odot$ denotes the component-wise vector multiplication. $\mathrm{nnz}(x)$ denotes the number of non-zero elements of vector x.

Given a matrix $\overline{W}$ and its rows $\overline{W}_i$ we write $\kappa(\overline{W}) := \sum_{i=1}^m \mathrm{nnz}(\overline{W}_i)$.

2 Problem Statement and Results

In this section, we describe the problem of the distributed computation of the entropy-regularized semi-discrete Wasserstein barycenter. Moreover, we present a summary of the main results of this paper regarding the communication, sample, and arithmetic complexity of the proposed algorithm.

Consider a network of m agents defined over a connected, fixed, and undirected graph $\mathcal{G} = (V, E)$, where $V = \{1, \cdots, m\}$ is the set of agents and E is the set of edges, such that $(i, j) \in E$ if agent i and agent j can communicate with each other. We assume that the graph $\mathcal{G}$ does not have self-loops. Moreover, at each time $k \geq 0$, each agent has access to *realizations* or *samples* (the number of samples will be specified later) from a stationary random process $\{Y_k^i\}_{k \geq 0}$ such that $Y_k^i \sim \mu^i$, where $\mu^i \in \mathcal{M}_+^1(\mathcal{Y})$ is a continuous probability distribution with with density $q^i(y)$ on a metric space $\mathcal{Y}$, see Figure 1.

The group of agents tries to collaboratively compute the entropy-regularized semi-discrete Wasserstein barycenter, defined as the discrete probability distribution ν, on a fixed support $z_1, \ldots, z_n \in \mathcal{Z}$,

i.e., $\nu = \sum_{i=1}^{n}[p]_i\delta(z_i)$, and $p \in S_1(n)$ such that

$$p^* = \underset{p\in S_1(n)}{\operatorname{argmin}} \sum_{i=1}^{m} \omega_i \mathcal{W}_{\gamma,\mu_i}(p), \tag{1}$$

where $\{\omega_i\}_{i=1}^{m}$ is a set of non-negative weights with $\sum_{i=1}^{m}\omega_i = 1$. Without loss of generality we set $\omega_i = 1/m$. Moreover, denote $\mathcal{W}_{\gamma,\mu}(p) \triangleq \mathcal{W}_\gamma(\mu, p)$ for fixed probability measure μ, where $\mathcal{W}_\gamma(\mu, p)$ is the entropy-regularized semi-discrete Wasserstein distance:

$$\mathcal{W}_\gamma(\mu, p) = \min_{\pi\in\Pi(\mu,p)} \left\{ \sum_{i=1}^{n} \int_{\mathcal{Y}} c_i(y)\pi_i(y)dy + \right.$$
$$\left. \gamma \sum_{i=1}^{n} \int_{\mathcal{Y}} \pi_i(y)\log\left(\frac{\pi_i(y)}{\xi}\right)dy \right\}, \tag{2}$$

where ξ is the density of uniform distribution on $\mathcal{Y} \times \mathcal{Z}$, $c_i(y) = c(z_i, y)$ denotes the cost of transporting a unit of mass from point $z_i \in \mathcal{Z}$ to point $y \in \mathcal{Y}$, and $\Pi(\mu, p)$ is the set of admissible coupling measures π defined as

$$\Pi(\mu, p) = \left\{ \pi \in \mathcal{M}_+^1(\mathcal{Y}) \times S_1(n) : \right.$$
$$\left. \sum_{i=1}^{n} \pi_i(y) = q(y), y \in \mathcal{Y}, \int_{\mathcal{Y}} \pi_i(y)dy = p_i \right\}$$

and γ is the regularization parameter.

Note that unlike [31], we regularize the problem by the Kullback-Leibler divergence from the uniform distribution ξ, which allows us to find explicitly the Fenchel conjugate for $\mathcal{W}_\gamma(\mu, p)$, to be defined later in Lemma 3.

We are interested in the distributed computation of approximate solutions of (1), such that, each agent finds an approximate distribution $\tilde{p}_i$, where

$$\frac{1}{m}\sum_{i=1}^{m}\mathcal{W}_\gamma(\mu_i, \tilde{p}_i) - \frac{1}{m}\sum_{i=1}^{m}\mathcal{W}_\gamma(\mu_i, p^*) \le \varepsilon, \tag{3}$$

and $\tilde{p}_1, \cdots, \tilde{p}_m$ are close to each other in a sense to be defined soon. Due to the distributed nature of the problem, we impose the constraint $\tilde{p}_1 = \cdots = \tilde{p}_m$. Also, note that with an appropriate selection of the regularization parameter γ, one can also compute an approximate solution to the non-regularized Wasserstein barycenter.

Proposition 1. *Assume that the space $\mathcal{Y} \times \mathcal{Z}$ is compact with the volume $\Omega \triangleq \operatorname{Vol}(\mathcal{Y} \times \mathcal{Z})$ and there exist $\tilde{p}$ satisfying (3) with $\gamma = \hat{\gamma} \triangleq \varepsilon/(4\log(\Omega))$. Then*

$$\frac{1}{m}\sum_{i=1}^{m}\mathcal{W}_0(\mu_i, \tilde{p}) - \frac{1}{m}\sum_{i=1}^{m}\mathcal{W}_0(\mu_i, p^*) \le 2\varepsilon. \tag{4}$$

Remark 1. *Formally, the ρ-Wasserstein distance for $\rho \ge 1$ is $(\mathcal{W}_0(\mu, \nu))^{\frac{1}{\rho}}$ if $\mathcal{Y} = \mathcal{Z}$ and $c_i(y) = d^\rho(z_i, y)$, d being a distance on $\mathcal{Y}$. For simplicity, we refer to (2) as regularized Wasserstein distance. However, we want to emphasize that our algorithm does not rely on any specific choice of cost $c_i(y)$ and can be used for the Monge-Kantorovich distances where the transportation costs do not have to satisfy the metric properties.*

Table 1: A summary of complexity bounds for **Q-DecPDSAG**. Constant and logarithmic terms are hidden for notation convenience.

26emComplexity	DecPDSAG [21]	Q-DecPDSAG Increasing Batch	Constant Batch
Communication Rounds	$\dfrac{\sqrt{\chi(W)n}}{\varepsilon}$	$\dfrac{\sqrt{\chi(W)n}}{\varepsilon}$	$\dfrac{\chi(W)n}{\varepsilon^2}$
Total Arithmetic Operations	$\chi(W)\max\{\dfrac{mn^2}{\varepsilon^2},\dfrac{n\kappa(W)}{\varepsilon^2}\}$	$\chi(W)\max\{\dfrac{mn^2}{\varepsilon^2},\dfrac{n\kappa(W)}{\varepsilon^2}\}$	$\dfrac{\chi(W)n^2\kappa(W)}{\varepsilon^2}$
Total Number of bits sent	$\dfrac{\sqrt{\chi(W)}mn^{3/2}}{\varepsilon}$	$\sqrt{\chi(W)}\kappa(W)\max\{\dfrac{n^{3/2}}{\varepsilon},\dfrac{n}{\varepsilon^2}\}$	$\dfrac{\chi(W)n\kappa(W)}{\varepsilon^2}$
Total Arithmetic Operations expander graph	$\dfrac{mn^2}{\varepsilon^2}$	$\dfrac{mn^2}{\varepsilon^2}$	$\dfrac{mn^2}{\varepsilon^2}$
Total Number of bits sent expander graph	$\dfrac{mn^{3/2}}{\varepsilon}$	$\max\{\dfrac{mn^{3/2}}{\varepsilon},\dfrac{mn}{\varepsilon^2}\}$	$\dfrac{mn}{\varepsilon^2}$
Number of bits sent at iteration k	n	$\max\{n,k\}$	M
Total Number of Samples	$\dfrac{\sqrt{\chi(W)}mn}{\varepsilon^2}$	$\dfrac{\sqrt{\chi(W)}mn}{\varepsilon^2}$	$\dfrac{\chi(W)mn}{\varepsilon^2}$

The objective of the group of agents is to solve (1). However, the distributed structure induced by the network $\mathcal{G}$ imposes communication constraints we need to consider. We assumed that at each time $t \geq 0$, each agent $i \in V$ has access to the samples from the distribution μ^i only, i.e., agent i has no information about μ^j for $j \neq i$. Therefore, cooperation is needed. Such cooperation is modeled as the ability of the agents to share their gradients using the set of edges E.

In Table 1, we informally summarize the main results of this paper. We also describe the communication and the computational complexity of the proposed algorithm, quantized distributed primal-dual stochastic accelerated gradient method **Q-DecPDSAG**. We compare the obtained bounds with the Non-quantized version denoted **DecPDSAG** [21]. The specifics of the proposed quantization scheme will be described in Section 3.2. Moreover, we compare two different regimes for **Q-DecPDSAG**, namely, having an increasing number of samples per iteration and having a constant number of samples per interaction. We analyze the computational complexity of **Q-DecPDSAG** in terms of arithmetic operations and the total communication complexity, as the total number of coordinates sent per node over all the iterations of the algorithm for the general graph as well as for the specific architecture where we use expander graphs.

In the next section, we study the primal-dual properties of the WB problem. We specifically exploit the dual problem structure and propose an effective sampling strategy for the construction of estimates of the stochastic gradient. This will be the main building block of our algorithm.

3 Duality and Quantization

In this Section, we build upon the dual properties of the entropy-regularized Wasserstein distance to develop a distributed algorithm that allows for the computation of a Wasserstein barycenter.

3.1 Duality of the WB problem

A commonly used method to introduce the distributed structure of the problem is by reformulating (1) to take into account the constraints imposed by the graph $\mathcal{G}$ explicitly. To do so, we define the Laplacian matrix $\bar{W} \in \mathbb{R}^{m \times m}$ of the graph $\mathcal{G}$ as $[\bar{W}]_{ij} \triangleq -1$, $(i,j) \in E$, $[\bar{W}]_{ij} = \deg(i)$ if $i = j$, and 0 otherwise, where $\deg(i)$ is the number of neighbors of node i, i.e., the degree of the node i. Similarly, we define the auxiliary matrix $W \triangleq \bar{W} \otimes I_n$, which takes into account the dimension n of the decision variable. Therefore, assuming that the graph $\mathcal{G}$ is undirected and connected, $\bar{W}$ is symmetric and positive semidefinite. Moreover, the matrix W inherits the properties of $\bar{W}$. More importantly, the Laplacian matrix has the property: $W\mathrm{p} = 0$ if and only if $p_1 = \cdots = p_m$, where $\mathrm{p} = [p_1^T, \cdots, p_m^T]^T \in \mathbb{R}^{mn}$. The relation $\sqrt{W}\mathrm{p} = 0$ if and only if $p_1 = \cdots = p_m$ also holds [21]. Thus, we equivalently rewrite problem (1) as

$$\max_{\substack{p_1, \ldots, p_m \in S_1(n) \\ \sqrt{W}\mathrm{p}=0}} -\frac{1}{m} \sum_{i=1}^{m} \mathcal{W}_{\gamma,\mu_i}(p_i). \tag{5}$$

Given that (5) is an optimization problem with linear constraints, we introduce a vector of dual variables $\boldsymbol{\lambda} = [\lambda_1^T, \cdots, \lambda_m^T]^T \in \mathbb{R}^{mn}$ for the constraints $\sqrt{W}\mathrm{p} = 0$. Then, the Lagrangian dual problem for (1) can be written as

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^{mn}} \mathcal{W}_{\gamma}^*(\boldsymbol{\lambda}) \triangleq \frac{1}{m} \sum_{i=1}^{m} \mathcal{W}_{\gamma,\mu_i}^*(m[\sqrt{W}\boldsymbol{\lambda}]_i), \tag{6}$$

where $\mathcal{W}_{\gamma,\mu_i}^*(\cdot)$ is the Fenchel-Legendre transform of $\mathcal{W}_{\gamma,\mu_i}(p_i)$ defined as

$$\mathcal{W}_{\gamma,\mu_i}^*(m[\sqrt{W}\boldsymbol{\lambda}]_i) = \max_{p_i \in S_1(n)} \left\{ \langle \lambda_i, [\sqrt{W}\mathrm{p}]_i \rangle - \frac{1}{m} \mathcal{W}_{\gamma,\mu_i}(p_i) \right\},$$

and $[\sqrt{W}\mathrm{p}]_i$ and $[\sqrt{W}\boldsymbol{\lambda}]_i$ denote the i-th n-dimensional block of vectors $\sqrt{W}\mathrm{p}$ and $\sqrt{W}\boldsymbol{\lambda}$ respectively.

The next two auxiliary lemmas state properties of $\mathcal{W}_{\gamma,\mu_i}^*(\cdot)$ what will be useful for our analysis in [21]. In particular, $\mathcal{W}_{\gamma,\mu_i}^*(\cdot)$ is a smooth function with Lipschitz-continuous gradient and can be expressed as an expectation of a function of additional random argument.

Lemma 1 (Lemma 1 in [21]). *Given a positive Radon probability measure $\mu \in \mathcal{M}_+^1(\mathcal{Y})$ with density $q(y)$ on a metric space $\mathcal{Y}$, the Fenchel-Legendre dual function of $\mathcal{W}_{\gamma,\mu}(p)$ can be written as*

$$\mathcal{W}_{\gamma,\mu}^*(\bar{\lambda}) = \mathbb{E}_{Y \sim \mu} \left[\gamma \log \left(\frac{1}{q(Y)} \sum_{\ell=1}^{n} \exp \left(\frac{[\bar{\lambda}]_\ell - c_\ell(Y)}{\gamma} \right) \right) \right].$$

Moreover, $\mathcal{W}_{\gamma,\mu}^*(\bar{\lambda})$ has m/γ-Lipschitz gradients w.r.t. 2-norm, and its l-th coordiante, for $l = 1, \ldots, n$, is defined as

$$[\nabla \mathcal{W}_{\gamma,\mu}^*(\bar{\lambda})]_l = \mathbb{E}_{Y \sim \mu} \left[\frac{\exp(([\bar{\lambda}]_l - c_l(Y))/\gamma)}{\sum_{\ell=1}^{n} \exp(([\bar{\lambda}]_\ell - c_\ell(Y))/\gamma)} \right].$$

Lemma 2 (Lemma 2 in [21]). *The function $\mathcal{W}_{\gamma}^*(\boldsymbol{\lambda})$ in (6) has $m\lambda_{\max}(W)/\gamma$-Lipschitz gradients w.r.t. 2-norm. Moreover, for $l = 1, \ldots, n$, it holds that*

$$\left[\nabla \mathcal{W}_{\gamma}^*(\boldsymbol{\lambda}) \right]_l = \sum_{j=1}^{m} \sqrt{W}_{lj} \nabla \mathcal{W}_{\gamma,\mu_j}^*(\bar{\lambda}_j), \tag{7}$$

where we denoted $\bar{\lambda}_j = m[\sqrt{W}\boldsymbol{\lambda}]_j$.

Lemma 2 states that the *dual* problem (6) is a smooth stochastic convex optimization problem. This is in contrast, to [42], where the primal problem is a stochastic optimization problem. This will be the main observation that will allow us to propose an efficient sampling and communication strategy to find an approximate solution of (6), and (1). More importantly, the gradient of the dual function can be computed in a distributed manner over a network, where each agent $j \in V$ computes $\nabla \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j)$ using local information only. Then, the gradient is shared with neighbors following the graph topology, and a full gradient step can be taken.

Next, we describe our sampling and quantization approach. We take advantage of the smooth stochastic form of the dual problem, for which each agent can obtain an estimate of the gradient-based on samples from the random variables Y^i. Moreover, such an approximate gradient will be a probability distribution itself. Thus, instead of communicating the full gradient, an agent can share a histogram of samples of the approximate gradient itself.

3.2 Quantized stochastic gradients

We build our main result based on the idea of double sampling of the gradients. In [21], the authors propose to use sampling from measures μ_i to approximate the gradients of the dual problem. Given that each agent i can obtain at each iteration M_1 samples of the random variable $Y^i_k \sim \mu_i$, one can define an approximate gradient as

$$\widehat{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j) = \frac{1}{M_1} \sum_{r=1}^{M_1} p_j(\bar{\lambda}_j, Y^j_r), \tag{8}$$

where, for all $l = 1, \cdots, n$,

$$[p_j(\bar{\lambda}_j, Y^j_r)]_l = \frac{\exp(([\bar{\lambda}_j]_l - c_l(Y^j_r))/\gamma)}{\sum_{k=1}^n \exp(([\bar{\lambda}_j]_k - c_k(Y^j_r))/\gamma)}. \tag{9}$$

However, such an approximate gradient will be a vector of dimension n. For the case where n is large, this might be prohibitively expensive in terms of communications. Thus, we take advantage of the fact that $\widehat{\nabla} \mathcal{W}^*_{\gamma,\mu}(\bar{\lambda}) \in S_1(n)$, to propose a strategy that allows lower communication costs. We propose to use the approximate stochastic gradient defined in the next lemma.

Lemma 3. *Let each agent $i \in V$, at iteration k, take M_1 samples of the random variable Y^i, and build a stochastic gradient as defined in* (8). *Now, let each agent $i \in V$ take M_2 samples of a discrete random variable ξ on $l \in \{1, ..., n\}$ with $\xi = l$ with probability $[\widehat{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j)]_l$, and construct a histogram, or approximate quantized sampled gradient of its local dual function as $\widetilde{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j) = \frac{1}{M_2} \sum_{r=1}^{M_2} e_{\xi_r}$, which are then combined in the stochastic approximation for the whole dual objective* (6) *defined as*

$$\left[\widetilde{\nabla} \mathcal{W}^*_{\gamma}(\boldsymbol{\lambda}) \right]_l = \sum_{j=1}^m \sqrt{W}_{lj} \widetilde{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j).$$

Then, the quantized stochastic gradient is unbiased, i.e.,

$$\mathbb{E} \widetilde{\nabla} \mathcal{W}^*_{\gamma}(\boldsymbol{\lambda}) = \nabla \mathcal{W}^*_{\gamma}(\boldsymbol{\lambda}). \tag{10}$$

Furthermore, its variance is bounded by

$$\mathbb{E} \| \widetilde{\nabla} \mathcal{W}^*_{\gamma}(\boldsymbol{\lambda}) - \nabla \mathcal{W}^*_{\gamma}(\boldsymbol{\lambda}) \|_2^2 \leq 2\lambda_{\max}(W) \sum_{i=1}^m \left(\frac{1}{M_{i,1}} + \frac{1}{M_{i,2}} \right).$$

There are two main advantages in the approach described in Lemma 3. First, the number of communications between the nodes diminishes with a smaller second batch, while the double-batched algorithm preserves roughly the same complexity just as the Accelerated Distributed Computation of Wasserstein barycenter in [21]. The second advantage is the quantization aspect such that agents exchange sparse integer vectors instead of dense vectors of float variables.

The main observation of this Section is that the dual problem to the WB setup is a stochastic optimization problem. Moreover, its stochastic gradients live in a probability simplex, which defines a discrete probability distribution. Such a unique structure allows for effective sampling strategies, which translate into effective communication due to the primal-dual properties of the problem. In the next section, we propose a new analysis of stochastic optimization problems with such a structure. Later we will specialize it for the WB problem, but our general results could be of independent interest.

4 General Primal-Dual Accelerated Stochastic Gradient Method

For any finite-dimensional real vector space E, we denote by E^* its dual, by $\langle \lambda, x \rangle$ the value of a linear function $\lambda \in E^*$ at $x \in E$. Let $\| \cdot \|$ denote some norm on E and $\| \cdot \|_*$ denote the norm on E^* which is dual to $\| \cdot \|$, i.e. $\|\lambda\|_* = \max\{\langle \lambda, x \rangle : \|x\| \leq 1\}$. For a linear operator $A : E_1 \to E_2$, we define the adjoint operator $A^T : E_2^* \to E_1^*$ in the following way $\langle u, Ax \rangle = \langle A^T u, x \rangle, \quad \forall\, u \in E_2^*, x \in E_1$. We say that a function $f : E \to \mathbb{R}$ has a L-Lipschitz continuous gradient w.r.t. norm $\| \cdot \|_*$ if it is continuously differentiable and its gradient satisfies Lipschitz condition $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|, \quad \forall\, x, y \in E$.
The main problem we consider in this section is

$$\min_{x \in Q}\{f(x) \quad : \quad Ax = b\}, \tag{11}$$

where Q is a simple closed convex set, $A : E \to H$ is given linear operator, $b \in H$ is given, $\Lambda = H^*$. The dual problem for (11) consists in minimization of the function

$$\varphi(\lambda) \triangleq \langle \lambda, b \rangle + \max_{x \in Q}\{-f(x) - \langle A^T \lambda, x \rangle\}$$
$$= \langle \lambda, b \rangle + f^*(-A^T \lambda). \tag{12}$$

Here f^* is the Fenchel-Legendre conjugate function for f. We say that a (random) point $\hat{x}$ is an $\varepsilon-$solution to Problem (11) if

$$f(\mathbb{E}\hat{x}) - f^* \leq \varepsilon, \quad \|A\mathbb{E}\hat{x} - b\|_2 \leq \frac{\varepsilon}{R}, \tag{13}$$

where R is such that the dual problem (12) has a solution λ^* satisfying $\|\lambda^*\|_2 \leq R < +\infty$.

We assume that $f^*(-A^T \lambda) = \mathbb{E}_\xi F^*(-A^T \lambda, \xi)$, where ξ is random vector and F^* is the Fenchel-Legendre conjugate function to some function $F(x, \xi)$, i.e. it satisfies $F^*(-A^T \lambda, \xi) = \max_{x \in Q}\{ \langle -A^T \lambda, x \rangle - F(x, \xi)\}$. $F^*(\bar{\lambda}, \xi)$ is assumed to be smooth and, hence $\nabla_{\bar{\lambda}} F^*(\bar{\lambda}, \xi) = x(\bar{\lambda}, \xi)$, where $x(\bar{\lambda}, \xi)$ is the solution of the maximization problem

$$x(\bar{\lambda}, \xi) = \arg\max_{x \in Q}\{\langle \bar{\lambda}, x \rangle - F(x, \xi)\}.$$

Further, we assume that the dual problem (12) can be accessed by a stochastic oracle $(\Phi(\lambda, \xi), \nabla\Phi(\lambda, \xi))$ with $\Phi(\lambda, \xi) = \langle \lambda, b \rangle + F^*(-A^T \lambda, \xi)$ and $\nabla\Phi(\lambda, \xi) = b - A\nabla F^*(-A^T \lambda, \xi)$. Let us summarize our main assumptions as follows.

Assumptions 1. ■ *The gradient of the objective function φ is L-Lipschitz continuous.*

■ *The stochastic oracle pair $(\Phi(\lambda, \xi), \nabla\Phi(\lambda, \xi))$ for the dual problem satisfies*

$$\mathbb{E}_\xi \Phi(\lambda, \xi) = \varphi(\lambda) \tag{14}$$
$$\mathbb{E}_\xi \nabla\Phi(\lambda, \xi) = \nabla\varphi(\lambda). \tag{15}$$

■ *There is a random variable $\widetilde{\xi}$, which is independent of ξ, and a stochastic approximation $\widetilde{\nabla}\Phi(\lambda, \xi, \widetilde{\xi})$ for $\nabla\Phi(\lambda, \xi)$ satisfying*

$$\mathbb{E}_{\widetilde{\xi}} \widetilde{\nabla}\Phi(\lambda, \xi, \widetilde{\xi}) = \nabla\Phi(\lambda, \xi), \quad \forall \lambda, \xi \tag{16}$$
$$\mathbb{E}_{\xi, \widetilde{\xi}} \| \widetilde{\nabla}\Phi(\lambda, \xi, \widetilde{\xi}) - \nabla\varphi(\lambda)\|_2^2 \leq \sigma^2, \quad \forall \lambda. \tag{17}$$

■ *The dual problem* (12) *has a solution λ^* and there exists some $R > 0$ such that $\|\lambda^*\|_2 \leq R < +\infty$.*

We propose Algorithm 1 for solving (11) using the stochastic oracle described above. Contrary to [21], we extend the accelerated stochastic gradient method from [18] and propose its primal-dual version. The main benefit is flexibility in the choice of the batch size, which is assumed fixed in [21]. Also, the algorithm from [18] cannot be directly used since it does not allow us to reconstruct the solution of the primal problem, i.e., the barycenter in our application. We believe that this general primal-dual accelerated stochastic gradient method with flexible batch size can be of independent interest. The next theorem presents the convergence properties of Algorithm 1.

Theorem 1. *Let Algorithm 1 be applied to solve Problem* (11) *under the above assumptions.* **Case A** *Assume also that at each iteration the variance σ_k^2 of the stochastic gradient satisfies $\sigma_k^2 \leq \varepsilon L\alpha_k/A_k$. Then, Algorithm 1 outputs an $\varepsilon-$ solution to Problem* (11) *after $\sqrt{8LR^2/\varepsilon}$ iterations.* **Case B** *Assume the variance σ^2 is constant. Then, Algorithm 1 outputs an $\varepsilon-$ solution to Problem* (11) *after $\max\{\sqrt{4LR^2/\varepsilon}, 9\sigma^2 R^2/\varepsilon^2\}$ iterations.*

5 Proposed Algorithm and Analysis

In this Section, we adjust the main primal-dual algorithm to the Wasserstein Barycenter problem. The algorithm has two different options for the batch size, one for the increasing batch size similar to the main algorithm in [21] and one for the constant batch size. We adapt the structure of the general Algorithm 1 for the specific properties of the WB. This allows us to propose Algorithm 2 for the distributed computation of WB in the semi-discrete setting with quantized communications. Finally, the following theorems establish the complexity results in Table 1.

Theorem 2. *Let $\mathcal{G}$ be a connected, undirected, fixed graph, and let $\{\mu^i\}_{i \in V}$ be a set of probability distributions with $\mu_i \in \mathcal{M}^1_+(\mathcal{Z})$. Moreover, set $N = \widetilde{O}\left(\sqrt{\chi(\overline{W})}n/\varepsilon\right)$ (communication rounds), and pick the increasing batch size strategy with $M^k_{i,1} = M^k_{i,2} = \max\{1, \lceil(k+2)/\log(\Omega)\rceil\}$ for all $i \in V$ and all $k \geq 0$. Then, the output of Algorithm 2, i.e., $\hat{p}^N$, is an $\varepsilon-$ solution of* (1). *Moreover, the total number of samples drawn from $\{\mu^i\}_{i \in V}$ is $\widetilde{O}\left(\sqrt{\chi(\overline{W})}mn/\varepsilon^2\right)$, the total number of arithmetic operations is $\widetilde{O}\left(\max\{\chi(\overline{W})mn^2/\varepsilon^2, \chi(\overline{W})\kappa(\overline{W})n/\varepsilon^2\}\right)$ a, and the total number of bits sent per node is $\widetilde{O}\left(\sqrt{\chi(\overline{W})}\kappa(\overline{W})\min\{n^{3/2}/\varepsilon, n/\varepsilon^2\}\right)$.*

Algorithm 1 Primal-Dual Accelerated Stochastic Gradient Method

1: **Initialization**
   Set $\eta_0 = \zeta_0 = z_0 = \lambda_0 = 0$ and choose a number of iterations N.
2: **for** $k = 0, \ldots, N-1$ **do**
3:    Select the coefficients:
      Case A)
      $A_k = \dfrac{1}{2}(k+1)(k+2), \quad \alpha_k = \dfrac{k+1}{2}$ with $\beta_k = 2L$.
      Case B)
      $\alpha_k = \dfrac{k+1}{2\sqrt{2}}, \quad \beta_k = L + \dfrac{\sigma(k+2)^{3/2}}{2^{1/4}\sqrt{3}R}, \quad A_k = \dfrac{(k+1)(k+2)}{4\sqrt{2}}.$
4:    $\tau_k = \alpha_{k+1}/A_{k+1}.$
5:    $z_k = -\dfrac{1}{\beta_k} \sum_{l=0}^{k} \alpha_l \widetilde{\nabla}\, \Phi(\lambda_l, \xi_l, \widetilde{\xi_l})$
6:    $\lambda_{k+1} = \tau_k z_k + (1 - \tau_k)\eta_k.$
7:    $\zeta_{k+1} = z_k - \dfrac{\alpha_{k+1}}{\beta_k} \widetilde{\nabla}\, \Phi(\lambda_{k+1}, \xi_{k+1}, \widetilde{\xi}_{k+1}).$
8:    $\eta_{k+1} = \tau_k \zeta_{k+1} + (1 - \tau_k)\eta_k.$
9: **end for**
10: $\hat{x}_{k+1} = \frac{1}{A_{k+1}} \sum_{l=0}^{k+1} \alpha_l x(-A^T \lambda_{k+1}, \xi_{k+1}).$
Output: The point $\hat{x}_N$.

The results in Theorem 2 are two-fold. On the one hand, it provides the number of communication rounds required for Algorithm 2 to obtain an arbitrary approximation of the non-regularized WB when sampling scheme Case A) is selected. Moreover, we provide an explicit sample, arithmetic, and communication complexities for generating such an approximation. In particular, if one uses expander graphs, which are in general well-connected, the total number of arithmetic operations is $\widetilde{O}\left(mn^2/\varepsilon^2\right)$, and the total number of bits sent per node is $\widetilde{O}(\min\{n^{3/2}/\varepsilon, n/\varepsilon^2\})$.

Next, we present a result for the sampling scheme Case B) in Algorithm 2.

Theorem 3. *Let $\mathcal{G}$ be a connected, undirected, fixed graph, and let $\{\mu^i\}_{i \in V}$ be a set of probability distributions where $\mu_i \in \mathcal{M}_+^1(\mathcal{Z})$. Moreover, set $N = \widetilde{O}\left(\chi(\overline{W})n/\varepsilon^2\right)$ (communication rounds), and pick the constant batch strategy with $M_{i,1}^k = M_{i,2}^k = M$ for all $i \in V$ and all $k \geq 0$. Then, the output of Algorithm 2, i.e., $\hat{p}^N$, is an $\varepsilon-$ solution of* (1). *Moreover, Ner of samples from the set of distributions $\{\mu^i\}_{i \in V}$ is $\widetilde{O}\left(\chi(\overline{W})mn/\varepsilon^2\right)$. The total number of arithmetic operations is $\widetilde{O}\left(\chi(\overline{W})\kappa(\overline{W})n^2/\varepsilon^2\right)$. The total number of bits sent per node for the general graph is $\widetilde{O}\left(\chi(\overline{W})\kappa(\overline{W})n/\varepsilon^2\right)$.*

Similarly as in Theorem 2, when expander graphs are used, the total number of arithmetic operations is $\widetilde{O}\left(mn^2/\varepsilon^2\right)$, and the total number of bits sent per node is $\widetilde{O}(mn/\varepsilon^2)$.

6 Numerical Results

We present numerical experiments to support Theorem 3 and Theorem 2. Figure 2 shows the consensus gap and the dual function value of the WB problem generated by Algorithm 2 for $m = 10$ and $m = 100$ agents. The set of distributions are set to be Gaussian with means and variances randomly selected and assigned. Moreover, we compare the performance for the Non-quantized AGM for WB computation proposed in [21]. We test the algorithm for various combinations of sampling rates M_1 and M_2. As expected, the algorithm from [21] has the best results because it is a non-quantized communication approach, and at each iteration, each agent communicates its full gradient with its neigh-

---

Algorithm 2 Q-DecPDSAG: Decentralized Quantized and Stochastic Computation of the Semi-Discrete Entropy-Regularized Wasserstein Barycenter

---

1: **Initialization**
2: Each agent $i \in V$ is assigned a distribution μ_i.
3: All agents $i \in V$ set $\eta_i^0 = \zeta_i^0 = z_i^0 = \zeta_i^0 = \bar{\lambda}_i^0 = \hat{p}_i^0 = \mathbf{0} \in \mathbb{R}^n$, and $\gamma = \varepsilon/(4\ln(\Omega))$.
4: Set the number of iterations N
5: Select the sampling scheme:
   A) Increasing batch size.
   Set $M_i^{1,k} = M_i^{2,k} = \max\{1, \lceil (k+2)/\ln(\Omega) \rceil\}$, $A_k = \frac{1}{2}(k+1)(k+2)$, $\alpha_k = \frac{k+1}{2}$ with $\beta_k = 2L$.
   B) Constant batch size.
   Set $M_i^{1,k} = M_i^{2,k} = M$, $\alpha_k = \frac{k+1}{2\sqrt{2}}$ and $\beta_k = L + \frac{\sigma(k+2)^{3/2}}{2^{1/4}\sqrt{3}R}$, $A_k = \frac{(k+1)(k+2)}{4\sqrt{2}}$.
6: Compute $\widetilde{\nabla}\mathcal{W}_{\gamma,\mu_i}^*(\bar{\lambda}_i^0)$ according to Lemma 3.
7: Send $\widetilde{\nabla}\mathcal{W}_{\gamma,\mu_i}^*(\bar{\lambda}_i^0)$ to neighbors, i.e., $\{j \mid (i,j) \in E\}$.
8: **Initialization ends.**
9: **For each agent $i \in V$:**
10: **for** $k = 0, \ldots, N-1$ **do**
11:    $\tau^k = \alpha_{k+1}/A_{k+1}$
12:    $z_i^k = -\frac{1}{\beta_k} \sum_{l=0}^{k} \alpha_l \sum_{j=1}^{m} W_{ij} \widetilde{\nabla}\mathcal{W}_{\gamma,\mu_j}^*(\bar{\lambda}_i^k)$.
13:    $\bar{\lambda}_i^{k+1} = \tau_k z_i^k + (1-\tau_k)\eta_i^k$.
14:    Compute $\widetilde{\nabla}\mathcal{W}_{\gamma,\mu_i}^*(\bar{\lambda}_i^{k+1})$ according to Lemma 3.
15:    Send $\widetilde{\nabla}\mathcal{W}_{\gamma,\mu_i}^*(\bar{\lambda}_i^{k+1})$ to neighbors.
16:    $\zeta_i^{k+1} = z_i^k - \frac{\alpha_{k+1}}{\beta_k} \sum_{j=1}^{m} W_{ij} \widetilde{\nabla}\mathcal{W}_{\gamma,\mu_j}^*(\bar{\lambda}_i^{k+1})$.
17:    $\eta_i^{k+1} = \tau_k \zeta_i^{k+1} + (1-\tau_k)\eta_i^k$.
18:    $\hat{p}_i^{k+1} = \frac{\alpha_{k+1}p^{k+1}(\bar{\lambda}_i^{k+1},\cdot)+A_k\hat{p}_i^k}{A_{k+1}}$.
19: **end for**
Output: The point $\hat{p}^N$.

---

bors. However, the experiments also show that even with a small number of samples like $M_1 = 1$ and $M_2 = 10$ Algorithm 2 can rapidly converge to the barycenter with significantly less communication overhead. Figure 3 shows the iterates generated by Algorithm 2 for $k = \{10, 100, 200, 500\}$ iterations. Again, we compare the performance with AGM from [21] and various sampling schemes M_1 and M_2.

7 Discussion and Future Work

We presented an algorithm for the computation of approximate semi-discrete WB over networks. We analyzed the primal-dual structure of the WB associated optimization problem and proposed a novel accelerated stochastic method to minimize functions with stochastic dual structure. The proposed method's main advantage is its flexibility in the sampling scheme of the dual stochastic gradient. Two sampling approaches were studied, increasing batch sized and constant batch size. The constant batch is particularly useful when a specific sample size is available at each iteration step. These sampling schemes drastically reduce the communication complexity of the WB computation. Moreover, we explicitly analyze the communication, sample, and arithmetic complexity of the proposed methods. Future work should focus on studying whether the total arithmetic operations can be improved further
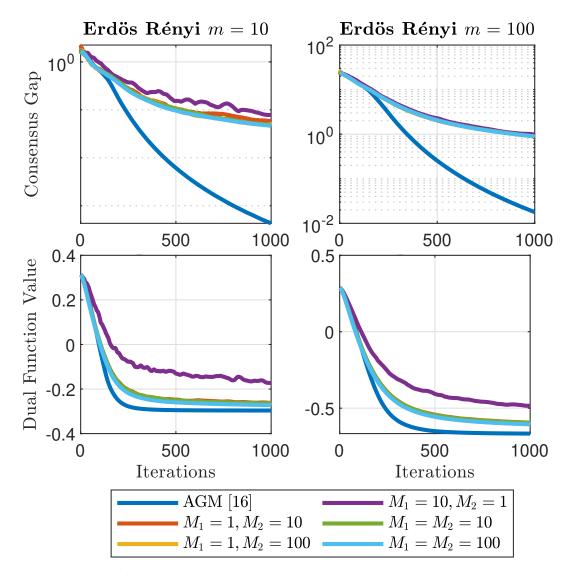
Figure 2: Consensus gap and dual function values generated by Algorithm 2.

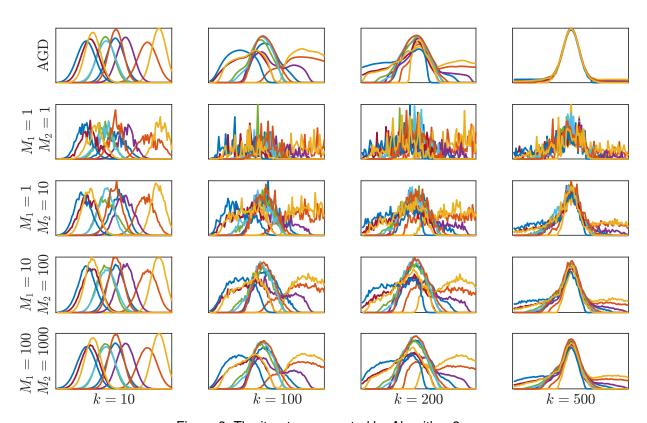than nm^2/ε^2 as in the discrete case [41, 45].

Acknowledgements

Figure 3: The iterates generated by Algorithm 2.

References

[1] M. Agueh and G. Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] A. S. Anikin, A. V. Gasnikov, P. E. Dvurechensky, A. I. Tyurin, and A. V. Chernov. Dual approaches to the minimization of strongly convex functionals with a simple structure under affine constraints. *Computational Mathematics and Mathematical Physics*, 57(8):1262–1276, Aug 2017.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv:1701.07875*, 2017.

[4] J. Backhoff-Veraguas, J. Fontbona, G. Rios, and F. Tobar. Bayesian learning with wasserstein barycenters. *arXiv:1805.10833*, 2018.

[5] M. Ballu, Q. Berthet, and F. Bach. Stochastic optimization for regularized wasserstein estimators. *arXiv:2002.08695*, 2020.

[6] M. Barré, C. Giron, M. Mazzolini, and A. d'Aspremont. Averaging atmospheric gas concentration data using wasserstein barycenters. *arXiv:2010.02762*, 2020.

[7] A. Bayandina, P. Dvurechensky, A. Gasnikov, F. Stonyakin, and A. Titov. Mirror descent and convex optimization problems with non-smooth inequality constraints. In P. Giselsson and A. Rantzer, editors, *Large-Scale and Distributed Optimization*, chapter 8, pages 181–215. Springer International Publishing, 2018. arXiv:1710.06612.

[8] E. Boissard, T. Le Gouic, J.-M. Loubes, et al. Distribution's template estimate with wasserstein metrics. *Bernoulli*, 21(2):740–759, 2015.

[9] S. Borgwardt. An lp-based, strongly-polynomial 2-approximation algorithm for sparse wasserstein barycenters. *Operational Research*, pages 1–41, 2020.

[10] S. Borgwardt and S. Patterson. On the computational complexity of finding a sparse wasserstein barycenter. *arXiv:1910.07568*, 2019.

[11] P.-Y. Bouchet, S. Gualandi, and L.-M. Rousseau. Primal heuristics for wasserstein barycenters. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 239–255. Springer, 2020.

[12] A. Chernov, P. Dvurechensky, and A. Gasnikov. Fast primal-dual gradient method for strongly convex minimization problems with linear constraints. In Y. Kochetov, M. Khachay, V. Beresnev, E. Nurminski, and P. Pardalos, editors, *Discrete Optimization and Operations Research: 9th International Conference, DOOR 2016, Vladivostok, Russia, September 19-23, 2016, Proceedings*, pages 391–403. Springer International Publishing, 2016.

[13] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil. Fair regression with wasserstein barycenters. *arXiv:2006.07286*, 2020.

[14] S. Claici, E. Chien, and J. Solomon. Stochastic Wasserstein barycenters. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 999–1008. PMLR, 2018.

[15] R. Cumings-Menon and M. Shin. Probability forecast combination via entropy regularized wasserstein distance. *Entropy*, 22(9):929, 2020.

[16] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

[17] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

[18] O. Devolder. Stochastic first order methods in smooth convex optimization. *CORE Discussion Paper 2011/70*, 2011. `https://cdn.uclouvain.be/public/Exports%20reddot/core/documents/coredp2011_70web.pdf`.

[19] D. Dvinskikh, E. Gorbunov, A. Gasnikov, P. Dvurechensky, and C. A. Uribe. On primal and dual approaches for distributed stochastic convex optimization over networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7435–7440, 2019. arXiv:1903.09844.

[20] D. Dvinskikh and D. Tiapkin. Improved complexity bounds in wasserstein barycenter problem. *arXiv:2010.04677*, 2020.

[21] P. Dvurechenskii, D. Dvinskikh, A. Gasnikov, C. Uribe, and A. Nedich. Decentralize and randomize: Faster algorithm for wasserstein barycenters. In *Advances in Neural Information Processing Systems*, pages 10760–10770, 2018.

[22] P. Dvurechensky and A. Gasnikov. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.

[23] P. Dvurechensky, A. Gasnikov, E. Gasnikova, S. Matsievsky, A. Rodomanov, and I. Usik. Primal-dual method for searching equilibrium in hierarchical congestion population games. In *Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016) Vladivostok, Russia, September 19 - 23, 2016*, pages 584–595, 2016. arXiv:1606.08988.

[24] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376, 2018. arXiv:1802.04367.

[25] P. Dvurechensky, A. Gasnikov, S. Omelchenko, and A. Tiurin. A stable alternative to Sinkhorn's algorithm for regularized optimal transport. In A. Kononov, M. Khachay, V. A. Kalyagin, and P. Pardalos, editors, *Mathematical Optimization Theory and Operations Research*, pages 406–423, Cham, 2020. Springer International Publishing.

[26] P. Dvurechensky, Y. Nesterov, and V. Spokoiny. Primal-dual methods for solving infinite-dimensional games. *Journal of Optimization Theory and Applications*, 166(1):23–51, Jul 2015.

[27] P. E. Dvurechensky, A. V. Gasnikov, and A. A. Lagunovskaya. Parallel algorithms and probability of large deviation for stochastic convex optimization problems. *Numerical Analysis and Applications*, 11(1):33–37, Jan 2018. arXiv:1701.01830.

[28] J. Ebert, V. Spokoiny, and A. Suvorikova. Construction of non-asymptotic confidence sets in 2-Wasserstein space. *arXiv:1703.03658*, 2017.

[29] J. Fan, A. Taghvaei, and Y. Chen. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv:2007.04462*, 2020.

[30] A. V. Gasnikov and P. E. Dvurechensky. Stochastic intermediate gradient method for convex optimization problems. *Doklady Mathematics*, 93(2):148–151, Mar 2016.

[31] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3440–3448. Curran Associates, Inc., 2016.

[32] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov. Accelerated alternating minimization, accelerated Sinkhorn's algorithm and accelerated Iterative Bregman Projections. *arXiv:1906.03622*, 2019.

[33] S. V. Guminov, Y. E. Nesterov, P. E. Dvurechensky, and A. V. Gasnikov. Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems. *Doklady Mathematics*, 99(2):125–128, Mar 2019.

[34] W. Guo, N. Ho, and M. Jordan. Fast algorithms for computational optimal transport and wasserstein barycenter. In *International Conference on Artificial Intelligence and Statistics*, pages 2088–2097, 2020.

[35] S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv:1904.05115*, 2019.

[36] C. Jin, J. Wang, J. Wei, L. Tan, S. Liu, W. Zhao, S. Liu, and X. Lv. Multimedia analysis and fusion via wasserstein barycenter. *International Journal of Networked and Distributed Computing*, 8(2):58–66, 2020.

[37] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[38] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi. Decentralized deep learning with arbitrary communication compression. *arXiv:1907.09356*, 2019.

[39] A. Koloskova, S. U. Stich, and M. Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv:1902.00340*, 2019.

[40] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[41] A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. Uribe. On the complexity of approximating Wasserstein barycenters. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3530–3540, Long Beach, California, USA, 09–15 Jun 2019. PMLR. arXiv:1901.08686.

[42] G. Lan, S. Lee, and Y. Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pages 1–48, 2018.

[43] L. Li, A. Genevay, M. Yurochkin, and J. Solomon. Continuous regularized wasserstein barycenters. *arXiv:2008.12534*, 2020.

[44] X. Lian, K. Jain, J. Truszkowski, P. Poupart, and Y. Yu. Unsupervised multilingual alignment using wasserstein barycenter. *arXiv:2002.00743*, 2020.

[45] T. Lin, N. Ho, X. Chen, M. Cuturi, and M. I. Jordan. Computational hardness and fast algorithm for fixed-support wasserstein barycenter. *arXiv:2002.04783*, 2020.

[46] T. Lin, N. Ho, M. Cuturi, and M. I. Jordan. On the Complexity of Approximating Multimarginal Optimal Transport. *arXiv e-prints*, 2019. arXiv:1910.00152.

[47] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas. Federated learning of deep networks using model averaging. corr abs/1602.05629 (2016). *arXiv preprint arXiv:1602.05629*, 2016.

[48] L. Mi, T. Yu, J. Bento, W. Zhang, B. Li, and Y. Wang. Variational wasserstein barycenters for geometric clustering. *arXiv:2002.10543*, 2020.

[49] S. Nadeem, T. Hollmann, and A. Tannenbaum. Multimarginal wasserstein barycenter for stain normalization and augmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 362–371. Springer, 2020.

[50] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, pages 1–28, 2020. arXiv:1809.05895.

[51] F. Nielsen, G. Marti, S. Ray, and S. Pyne. Clustering patterns connecting covid-19 dynamics and human mobility using optimal transport. *arXiv:2007.10677*, 2020.

[52] V. M. Panaretos and Y. Zemel. *An Invitation to Statistics in Wasserstein Space: Fréchet Means in the Wasserstein Space W2*, pages 59–74. Springer International Publishing, Cham, 2020.

[53] G. Puccetti, L. Rüschendorf, and S. Vanduffel. On the computation of Wasserstein barycenters. *Journal of Multivariate Analysis*, 176(C), 2020.

[54] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[55] K. Scaman, F. R. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3027–3036, 2017.

[56] Y. Shen, L. Shen, H.-Z. Huang, X. Wang, and W. Liu. Cpot: Channel pruning via optimal transport. *arXiv:2005.10451*, 2020.

[57] D. Simon and A. Aberdam. Barycenters of natural images constrained wasserstein barycenters for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7910–7919, 2020.

[58] E. Simou, D. Thanou, and P. Frossard. node2coords: Graph representation learning with wasserstein barycenters. *arXiv:2007.16056*, 2020.

[59] M. Staib, S. Claici, J. M. Solomon, and S. Jegelka. Parallel streaming wasserstein barycenters. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2647–2658. Curran Associates, Inc., 2017.

[60] F. S. Stonyakin, D. Dvinskikh, P. Dvurechensky, A. Kroshnin, O. Kuznetsova, A. Agafonov, A. Gas-nikov, A. Tyurin, C. A. Uribe, D. Pasechnyuk, and S. Artamonov. Gradient methods for problems with inexact model of the objective. In M. Khachay, Y. Kochetov, and P. Pardalos, editors, *Mathematical Optimization Theory and Operations Research*, pages 97–114, Cham, 2019. Springer International Publishing. arXiv:1902.09001.

[61] D. Tiapkin, A. Gasnikov, and P. Dvurechensky. Stochastic saddle-point optimization for wasserstein barycenters. *arXiv:2006.06763*, 2020.

[62] N. Tupitsa, P. Dvurechensky, A. Gasnikov, and C. A. Uribe. Multimarginal optimal transport by accelerated alternating minimization. In *2020 IEEE 59th Conference on Decision and Control (CDC)*, 2020. (accepted), arXiv:2004.02294.

[63] C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedić. Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*, pages 6544–6549, Dec 2018.

[64] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić. A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pages 1–40, 2020.

[65] Y. Xie, X. Wang, R. Wang, and H. Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pages 433–453. PMLR, 2020.

[66] A. Yurtsever, Q. Tran-Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3150–3158, Cambridge, MA, USA, 2015. MIT Press.

A Proof of Proposition 1

It is clear that the density ξ of the uniform distribution on $\mathcal{Y} \times \mathcal{Z}$ is $1/\Omega$, and that the entropy term in (2) is bounded from below by zero and from above by the value achieved when when π is a Dirac mass. In the latter case the maximum value is $\gamma \log \Omega$, which is no greater than $\varepsilon/4$ by our choice of $\gamma = \hat{\gamma} = \varepsilon/(4 \log(\Omega))$. Thus, for any μ, p, $\mathcal{W}_0(\mu, p) \in [\mathcal{W}_{\hat{\gamma}}(\mu, p), \mathcal{W}_{\hat{\gamma}}(\mu, p) + \varepsilon/4]$. Substituting this in (3), we obtain the desired result.

B Proof of Lemma 3

By the definition of ξ_r, $\mathbb{E}_{\xi_r}[e_{\xi_r}] = \widehat{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j)$ and it is easy to verify that the stochastic approximation $\widetilde{\nabla} \mathcal{W}^*_{\gamma,\mu_j}(\bar{\lambda}_j)$ is unbiased, and (10) holds. Let us now estimate the variance. Denoting $\mathbb{E} = \mathbb{E}_{Y_l^i \sim \mu_j, \xi_r^i, i=1,\dots,m, l=1,\dots,M_{i,1}, r=1,\dots,M_{i,2}}$ where ξ_r^i is distributed as described in Lemma 3 we have the following chain of estimates.

$$
\begin{aligned}
\mathbb{E}\|\widetilde{\nabla}\mathcal{W}^*_\gamma(\boldsymbol{\lambda}) - \nabla\mathcal{W}^*_\gamma(\boldsymbol{\lambda})\|_2^2 &= \mathbb{E}\left\| \sqrt{W}\begin{pmatrix} \widetilde{\nabla}\mathcal{W}^*_{\gamma,\mu_1}([\bar{\boldsymbol{\lambda}}]_1) \\ \dots \\ \widetilde{\nabla}\mathcal{W}^*_{\gamma,\mu_m}([\bar{\boldsymbol{\lambda}}]_m) \end{pmatrix} - \sqrt{W}\begin{pmatrix} \nabla\mathcal{W}^*_{\gamma,\mu_1}([\bar{\boldsymbol{\lambda}}]_1) \\ \dots \\ \nabla\mathcal{W}^*_{\gamma,\mu_m}([\bar{\boldsymbol{\lambda}}]_m) \end{pmatrix} \right\|_2^2 \\
&\leq (\lambda_{\max}(\sqrt{W}))^2 \mathbb{E}\left\| \begin{matrix} \widetilde{\nabla}\mathcal{W}^*_{\gamma,\mu_1}([\bar{\boldsymbol{\lambda}}]_1) - \nabla\mathcal{W}^*_{\gamma,\mu_1}([\bar{\boldsymbol{\lambda}}]_1) \\ \dots \\ \widetilde{\nabla}\mathcal{W}^*_{\gamma,\mu_m}([\bar{\boldsymbol{\lambda}}]_m) - \nabla\mathcal{W}^*_{\gamma,\mu_m}([\bar{\boldsymbol{\lambda}}]_m) \end{matrix} \right\|_2^2 \\
&= (\lambda_{\max}(\sqrt{W}))^2 \mathbb{E}\sum_{i=1}^m \left\| \widetilde{\nabla}\mathcal{W}^*_{\gamma,\mu_i}([\bar{\boldsymbol{\lambda}}]_i) - \nabla\mathcal{W}^*_{\gamma,\mu_i}([\bar{\boldsymbol{\lambda}}]_i) \right\|_2^2 \\
&= (\lambda_{\max}(W))\sum_{i=1}^m \mathbb{E}\left\| \frac{1}{M_{i,2}}\sum_{r=1}^{M_{i,2}} e_{\xi_r}(\bar{\boldsymbol{\lambda}}_i, Y_1^i, \dots, Y_{M_{i,1}}^i) - \mathbb{E}p_i(\bar{\boldsymbol{\lambda}}_i, Y^i) \right\|_2^2.
\end{aligned}
$$

We define $\bar{p} = \frac{1}{M_1}\sum_{l=1}^{M_1} p\left(\bar{\boldsymbol{\lambda}}_i, Y_l^i\right)$. Using the triangle inequality for the square of the norm we get:

$$
\mathbb{E}\|e_\xi - \mathbb{E}(p)\|_2^2 \leq 2\mathbb{E}\|e_\xi - \bar{p}\|_2^2 + 2\mathbb{E}\|\bar{p} - E(p)\|_2^2. \tag{18}
$$

Moreover, using the [21, Lemma 2], we have $\mathbb{E}\|\bar{p} - \mathbb{E}(p)\|_2^2 \leq \frac{1}{M_1} - \frac{1}{M_1}\|\mathbb{E}(p)\|_2^2$. For a fixed λ and $p(Y_r) := p(\lambda, Y_r)$, the first term in inequality (18) can be computed as follows:

$$
\begin{aligned}
\mathbb{E}\|e_\xi - \bar{p}\|_2^2 = \mathbb{E}\left[\mathbb{E}[\|e_\xi - \bar{p}\|_2^2|Y]\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^n ([e_\xi]_i - [\bar{p}]_i)|Y]\right]\right] = \\
&\mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^n [e_\xi]_i|Y\right]\right] - \sum_{i=1}^n \mathbb{E}[\bar{p}]_i^2 = \\
&= \mathbb{E}\left[\sum_{i=1}^n [\bar{p}]_i\right] - \frac{1}{M_1^2}\sum_{i=1}^n \mathbb{E}\sum_{r=1}^{M_1}[p(Y_r)]_i^2 = 1 - \frac{1}{M_1}\mathbb{E}\|p\|_2^2.
\end{aligned}
$$

Using the above procedure for the sum of e_ξ we get

$$\mathbb{E}\|\frac{1}{M_2}\sum_{r=1}^{n}e_{\xi_r} - \mathbb{E}p\|_2^2 = \frac{1}{M_2^2}\mathbb{E}\|\sum_{r=1}^{M_2}(e_{\xi_r} - \bar{p} + \bar{p} - \mathbb{E}p)\|_2^2$$

$$\leq \mathbb{E}\left[\frac{2}{M_2^2}\|\sum_{r=1}^{M_2}(e_{\xi_r} - \bar{p})\|_2^2 + 2\|\bar{p} - \mathbb{E}p\|_2^2\right]. \tag{19}$$

Since ξ_r are conditionally independent, we can bound the first term by:

$$\mathbb{E}\frac{2}{M_2^2}\|\sum_{r=1}^{M_2}(e_{\xi_r} - \sum_{r=1}^{M_2}\bar{p})\|_2^2 = \frac{2}{M_2^2}\left(\mathbb{E}\left[\mathbb{E}\left[\sum_{r=1}^{M_2}\|e_{\xi_r}\|_1|Y\right]\right] - \mathbb{E}\left[\left(\mathbb{E}\left[\sum_{r=1}^{M_2}\|e_{\xi_r}\|_1|Y\right]\right)^2\right]\right)$$

$$= \frac{2}{M_2^2}\mathbb{E}\left[M_2\sum_{i=1}^{n}[\bar{p}]_i\right] - \frac{2\mathbb{E}\|\bar{p}\|_2^2}{M_2} = \frac{2}{M_2}(1 - E\|\bar{p}\|_2^2).$$

Thus, the inequality (19) becomes:

$$\mathbb{E}\|\frac{1}{M_2}\sum_{r=1}^{n}e_{\xi_r} - \mathbb{E}p\|_2^2 \leq \frac{2}{M_2}(1 - \frac{1}{M_1}\mathbb{E}\|p\|_2^2) - \frac{2}{M_1}\|E[p]\|_2^2 + \frac{2}{M_1}$$

$$= \frac{2}{M_2} + \frac{2}{M_1}(1 - \|E[p]\|_2^2 - \frac{2}{M_2}\mathbb{E}\|p\|_2^2) \leq 2\left(\frac{1}{M_2} + \frac{1}{M_1}\right).$$

This leads to:

$$\mathbb{E}\|\widetilde{\nabla}\mathcal{W}_\gamma^*(\boldsymbol{\lambda}) - \nabla\mathcal{W}_\gamma^*(\boldsymbol{\lambda})\|_2^2 = (\lambda_{\max}(W))\sum_{i=1}^{m}2\left(\frac{1}{M_{i,2}} + \frac{1}{M_{i,1}}\right) \tag{20}$$

and the desired result follows.

C Proof of Theorem 1

As mentioned in Section 4, we consider primal-dual pair of problems of the form:

$$\min_{x\in Q}\{f(x) : Ax = b\} \tag{21}$$

$$\min_{\lambda\in\Lambda}\varphi(\lambda) = \min_{\lambda\in\Lambda}\langle\lambda, b\rangle + \max_{x\in Q}\{-f(x) - \langle A^T\lambda, x\rangle\} = \min_{\lambda\in\Lambda}\langle\lambda, b\rangle + f^*(-A^T\lambda). \tag{22}$$

As before we assume that the conditions 1 are satisfied.

Algorithm 1 generalizes the Algorithm 6.1 of [18] (see also [22, 30]), here we present the Algorithm 6.1 to give a reader a better overview and its convergence theorem. Then we analyze the convergence rate of our primal-dual version of the Algorithm 1. Note that the primal-dual analysis of the existing accelerated methods [66, 2, 7, 12, 23, 24, 25, 26, 33, 50] does not apply since the dual problem is a stochastic optimization problem and we use additional randomization.

Algorithm 6.1 of [18] applied to the dual problem (22) with stochastic inexact oracle $\widetilde{\nabla}\Phi(\lambda, \xi, \widetilde{\xi})$ is listed as Algorithm C3. Since the objective $\varphi(\lambda)$ has Lipschitz-continuous gradient, in the setting of [18] we have $\delta = 0$.

The following condition for the convergence analysis of Algorithm C3 is stated at the beginning of Section 6.1 in [18].

Algorithm C3 Stochastic Fast Gradient Method ([18, Algorithm 6.1])

Input: Sequences $\alpha_k, \beta_k, A_k = \sum_{i=0}^{k} \alpha_k$ satisfying Assumptions 2. Distance generating function $d(\lambda)$ and corresponding Bregman divergence $V[z_k](\lambda)$.

1: **Initialization**

Set $\eta_0 = \zeta_0 = z_0 = \lambda_0$ and choose a number of iterations N.

2: **for** $k = 0, \dots, N-1$ **do**

3:    $\tau_k = \alpha_{k+1}/A_{k+1}$.

4:
$$z_k = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \{ \beta_k d(\lambda) + \sum_{l=0}^{k} \alpha_l \langle \widetilde{\nabla} \, \Phi(\lambda_l, \xi_l, \widetilde{\xi}_l), \lambda - \lambda_l \rangle \} \tag{23}$$

5:    $\lambda_{k+1} = \tau_k z_k + (1 - \tau_k)\eta_k$.

6:    Compute $\widetilde{\nabla} \, \Phi(\lambda_{k+1}, \xi_{k+1}, \widetilde{\xi}_{k+1})$.

7:
$$\zeta_{k+1} = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \{ \langle \beta_k V[z_k](\lambda) + \alpha_{k+1} \langle \widetilde{\nabla} \, \Phi(\lambda_{k+1}, \xi_{k+1}, \widetilde{\xi}_{k+1}), \lambda - z_k \rangle \}. \tag{24}$$

8:    $\eta_{k+1} = \tau_k \zeta_{k+1} + (1 - \tau_k)\eta_k$.

9: **end for**

Output: The point η_N.

Assumptions 2. *The following conditions hold for the sequences $\{\alpha_k\}$ and $\{\beta_k\}$, for all $k \geq 0$.*

$$\alpha_0 \in]0,1] \text{ and } \beta_{k+1} \geq \beta_k > L \text{ for all } k \geq 0,$$

$$\text{Coupling condition: } \alpha_k^2 \beta_k \leq \Big(\sum_{\ell=0}^{k} \alpha_\ell \Big) \beta_{k-1}. \tag{25}$$

We denote, for any $N \geq 0$,

$$\widetilde{\Psi}_N(\lambda) = \beta_N d(\lambda) + \sum_{k=0}^{N} \alpha_k \big(\Phi(\lambda_k, \xi_k) + \langle \widetilde{\nabla} \, \Phi(\lambda_k, \xi_k, \widetilde{\xi}_k), \lambda - \lambda_k \rangle \big), \quad \widetilde{\Psi}_N^* = \min_{\lambda \in \Lambda} \widetilde{\Psi}_N(\lambda).$$

Theorem C1 (Lemma 2 in [18])**.** *Assume that Conditions 1, 2 are satisfied. Then, the output of Algorithm C3 for all $N \geq 0$ has the following property:*

$$A_N \varphi(\eta_N) \leq \widetilde{\Psi}_N^* + \sum_{k=0}^{N} \alpha_k (\varphi(\lambda_k) - \Phi(\lambda_k, \xi_k)) + \sum_{k=1}^{N} A_{k-1} \langle \nabla \varphi(\lambda_k) - \widetilde{\nabla} \, \Phi(\lambda_k, \xi_k, \widetilde{\xi}_k),$$

$$\lambda_k - \eta_{k-1} \rangle + \sum_{k=0}^{N} \frac{A_k}{(\beta_k - L)} \| \widetilde{\nabla} \, \Phi_k(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla \varphi(\lambda_k) \|_2^2. \tag{26}$$

Let us now move to our generalization of Algorithm C3. In our setting, we take Algorithm C3 with a particular choice of the prox-function $d(\lambda) = \frac{1}{2}\|\lambda\|_2^2$ and the corresponding Bregman divergence $V[\widetilde{\lambda}](\lambda) = \frac{1}{2}\|\lambda - \widetilde{\lambda}\|_2^2$. Then, the steps (23) and (24) of Algorithm C3 can be computed analytically. Indeed, the optimality conditions result in equations $\beta_k \lambda + \sum_{l=0}^{k} \alpha_l \widetilde{\nabla} \, \Phi(\lambda_l, \xi_l, \widetilde{\xi}_l) = 0$ and $\beta_k(\lambda - $

$z_k) + \alpha_{k+1} \widetilde{\nabla} \Phi(\lambda_{k+1}, \xi_{k+1}, \widetilde{\xi}_{k+1}) = 0$. The steps therefore read as $z_k = -\frac{1}{\beta_k} \sum_{l=0}^{k} \alpha_l \nabla \Phi(\lambda_l, \xi_l)$ and $\zeta_{k+1} = z_k - \frac{\alpha_{k+1}}{\beta_k} \nabla \Phi(\lambda_{k+1}, \xi_{k+1})$. Our generalization consists in adding an update for the primal variable, such that not only the dual, but also the primal problem (21) is solved with an optimal rate. Thus, in our primal-dual algorithm, we add the step $\hat{x}_{k+1} = \frac{1}{A_{k+1}} \sum_{i=0}^{k+1} \alpha_i x(-A^T \lambda_i, \xi_i) = \frac{\alpha_{k+1} x(-A^T \lambda_{k+1}, \xi_{k+1}) + A_k \hat{x}_k}{A_{k+1}}$ to the algorithm for the update of the primal variable. Here the vector $x(-A^T \lambda, \xi)$ is defined as

$$x(-A^T \lambda, \xi) = \arg \max_{x \in Q} \{\langle -A^T \lambda, x \rangle - F(x, \xi)\}.$$

We also specify two choices of sequences $\alpha_k, \beta_k, A_k = \sum_{i=0}^{k} \alpha_k$. Having made these adjustments we arrive at Algorithm 1.

The following Lemma is needed for the proof of Theorem C2.

Lemma C1. *Let the assumptions of Section 4 in the main paper hold. Then*

$$\mathbb{E}_{\xi_k}(\varphi(\lambda_k) + \langle \nabla \Phi(\lambda_k, \xi_k), \lambda - \lambda_k \rangle)$$
$$\leq -f(\mathbb{E}_{\xi_k} x(-A^T \lambda_k, \xi_k)) + \langle b - A \mathbb{E}_{\xi_k} x(-A^T \lambda_k, \xi_k), \lambda \rangle. \quad (27)$$

Proof. The proof can be found in [21, Theorem 2]. □

Theorem C2 gives a convergence result for Algorithm 1 which is a primal-dual version of the Algorithm C3.

Theorem C2. *Let Assumptions 1 hold and the variance in each iteration be bounded* $\mathbb{E}_{\xi_k, \widetilde{\xi}_k} \| \widetilde{\nabla} \Phi(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla \varphi(\lambda_k) \|_2^2 \leq \sigma_k^2$. *Let accuracy $\varepsilon > 0$ and the number of steps $N \geq 0$ be given. Then, for the primal sequence* $\hat{x}_{k+1} = \frac{1}{A_{k+1}} \sum_{i=0}^{k+1} \alpha_i x(-A^T \lambda_i, \xi_i) = \frac{\alpha_{k+1} x(-A^T \lambda_{k+1}, \xi_{k+1}) + A_k \hat{x}_k}{A_{k+1}}$ *its expected value $\mathbb{E}\hat{x}_N$ satisfies*

$$f(\mathbb{E}\hat{x}_N) - f^* \leq \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N} \sum_{k=0}^{N} \frac{A_k}{(\beta_k - L)} \sigma_k^2, \quad (28)$$

$$\|A\mathbb{E}\hat{x}_N - b\|_2 \leq \frac{\beta_N R}{2A_N} + \frac{1}{RA_N} \sum_{k=0}^{N} \frac{A_k}{(\beta_k - L)} \sigma_k^2. \quad (29)$$

where the expectation is taken w.r.t. all the randomness $\xi_1, \ldots, \xi_N, \widetilde{\xi}_1, \ldots, \widetilde{\xi}_N$.

Proof. The dualization result relies on the weak duality and convexity of the considered problem. Taking the expectation in the inequality (26) in Theorem C1 for the unbiased stochastic approximation cancels out certain terms and we obtain

$$A_N \varphi(\eta_N) \leq \mathbb{E}\widetilde{\Psi}_N^* + \sum_{k=0}^{N} \frac{A_k}{(\beta_k - L)} \mathbb{E}\| \widetilde{\nabla} \Phi_k(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla \varphi(\lambda_k) \|_2^2. \quad (30)$$

Let us first estimate $\mathbb{E}\widetilde{\Psi}_N^*$. We introduce the feasible set $\Lambda_R := \{\lambda \in H^* : \|\lambda\|_2 \leq 2R\}$. Then,

$$\mathbb{E}\min_{\lambda\in\Lambda}\widetilde{\Psi}_N(\lambda) = \mathbb{E}\Big[\min_{\lambda\in\Lambda}d(\lambda) + \sum_{k=0}^{N}\alpha_k(\varphi(\lambda_k) + \langle\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k),\lambda-\lambda_k\rangle)\Big]$$

$$\leq \min_{\lambda\in\Lambda_R}\beta_N d(\lambda) + \mathbb{E}\Big[\sum_{k=0}^{N}\alpha_k(\varphi(\lambda_k) + \langle\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k),\lambda-\lambda_k\rangle)\Big]$$

$$= \min_{\lambda\in\Lambda_R}\beta_N d(\lambda) + \sum_{k=0}^{N}\alpha_k\mathbb{E}\Big[(\varphi(\lambda_k) + \langle\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k),\lambda-\lambda_k\rangle)\Big] \qquad (31)$$

Applying Lemma C1 above and using $\mathbb{E}_{\xi_k,\widetilde{\xi}_k}\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k) = \mathbb{E}_{\xi_k}\nabla\Phi(\lambda_k,\xi_k)$, we get:

$$\min_{\lambda\in\Lambda_R}\sum_{k=0}^{N}\alpha_k\mathbb{E}\Big[(\varphi(\lambda_k) + \langle\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k),\lambda-\lambda_k\rangle)\Big]$$

$$= \min_{\lambda\in\Lambda_R}\sum_{k=0}^{N}\alpha_k\mathbb{E}\Big[(\varphi(\lambda_k) + \langle\nabla\Phi(\lambda_k,\xi_k),\lambda-\lambda_k\rangle)\Big]$$

$$\leq \min_{\lambda\in\Lambda_R}\left\{\sum_{k=0}^{N}\alpha_k(-f(\mathbb{E}x(-A^T\lambda_k,\xi_k)) + \langle b - A\mathbb{E}x(-A^T\lambda_k,\xi_k),\lambda\rangle)\right\}$$

$$\leq A_N\min_{\lambda\in\Lambda_R}\left\{-f(\mathbb{E}\hat{x}_N) + \langle b - A\mathbb{E}\hat{x}_N,\lambda\rangle\right\}$$

$$= -A_N f(\mathbb{E}\hat{x}_N) + A_N\min_{\lambda\in\Lambda_R}\langle b - A\mathbb{E}\hat{x}_N,\lambda\rangle$$

$$= -A_N f(\mathbb{E}\hat{x}_N) + A_N\min_{\lambda\in\Lambda_R}\langle b - A\mathbb{E}\hat{x}_N,\frac{A\mathbb{E}\hat{x}_N - b}{\|A\mathbb{E}\hat{x}_N - b\|}\lambda\rangle$$

$$= -A_N f(\mathbb{E}\hat{x}_N) - 2A_N R\|b - A\mathbb{E}\hat{x}_N\|_2.$$

In the inequalities above, we also used the convexity of f, the property of A_N being a linear combination $A_N = \sum_{k=0}^{N}\alpha_k$, and definitions of $\hat{x}_N$ and Λ_R.

Thus, $\mathbb{E}\widetilde{\Psi}_N^*$ is bounded by

$$\mathbb{E}\widetilde{\Psi}_N^* \leq \beta_N d(\lambda) - A_N f(\mathbb{E}\hat{x}_N) - 2A_N R\|b - A\mathbb{E}\hat{x}_N\|_2. \qquad (32)$$

Moreover, using (30), and (32), the fact that $d(\lambda)\leq R^2/2$, and the unbiasedness of the stochastic approximation, we get the following bound on the iteration term $A_N\varphi(\eta_N)$:

$$A_N\mathbb{E}\varphi(\eta_N) \leq \beta_N R^2/2 - A_N f(\mathbb{E}\hat{x}_N) - 2A_N R\|b - A\mathbb{E}\hat{x}_N\|_2$$

$$+ \sum_{k=0}^{N}\frac{A_k}{(\beta_k - L)}\mathbb{E}\|\widetilde{\nabla}\,\Phi(\lambda_k,\xi_k,\widetilde{\xi}_k) - \nabla\varphi(\lambda_k)\|_2^2. \qquad (33)$$

Now, let us estimate the convergence of the primal objective $f(\mathbb{E}\hat{x}_N)$.

Using the weak duality $-f(x^*)\leq\varphi(\lambda^*)$ we obtain:

$$f(\mathbb{E}\hat{x}_N) - f(x^*) \leq f(\mathbb{E}\hat{x}_N) + \varphi(\eta^*) \leq f(\mathbb{E}\hat{x}_N) + \mathbb{E}\varphi(\eta_N) \qquad (34)$$

Dividing all terms by A_N in inequality 33 and using it we obtain

$$f(\mathbb{E}\hat{x}_N) - f(x^*) \leq f(\mathbb{E}\hat{x}_N) + \mathbb{E}\varphi(\eta_N)$$

$$\leq \frac{\beta_N R^2}{2A_N} - 2R\|b - A\mathbb{E}\hat{x}_N\|_2 + \frac{1}{A_N}\sum_{k=0}^{N}\frac{A_k}{(\beta_k - L)}\mathbb{E}\|\widetilde{\nabla}\Phi(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla\varphi(\lambda_k)\|_2^2$$

$$= \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N}\sum_{k=0}^{N}\frac{A_k}{(\beta_k - L)}\mathbb{E}\|\widetilde{\nabla}\Phi(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla\varphi(\lambda_k)\|_2^2. \quad (35)$$

Regarding the convergence of the argument $\|A\mathbb{E}\hat{x}_N - b\|_2$, we know

$$2R\|b - A\mathbb{E}\hat{x}_N\|_2 \leq f(x^*) - f(\mathbb{E}\hat{x}_N)$$

$$+ \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N}\sum_{k=0}^{N}\frac{A_k}{(\beta_k - L)}\mathbb{E}\|\widetilde{\nabla}\Phi(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla\varphi(\lambda_k)\|_2^2. \quad (36)$$

Therefore, we have for any $x \in Q$, $f(x^*) \leq f(x) + \langle\lambda^*, Ax - b\rangle$. Using the bound $\|\lambda^*\|_2 \leq R$, and choosing $x = \mathbb{E}\hat{x}_N$ translates to $f(x^*) \leq f(\mathbb{E}\hat{x}_N) + \langle\lambda^*, A\mathbb{E}\hat{x}_N - b\rangle \leq f(\mathbb{E}\hat{x}_N) + R\|A\mathbb{E}\hat{x}_N - b\|_2$. Finally, using this estimate in the inequality (36) gives us:

$$2R\|b - A\mathbb{E}\hat{x}_N\|_2 \leq R\|A\mathbb{E}\hat{x}_N - b\|_2$$

$$+ \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N}\sum_{k=0}^{N}\frac{A_k}{(\beta_k - L)}\mathbb{E}\|\widetilde{\nabla}\Phi(\lambda_k, \xi_k, \widetilde{\xi}_k) - \nabla\varphi(\lambda_k)\|_2^2, \quad (37)$$

and the desired result follows.

$\square$

Using the techniques developed in [27, 19], these convergence rate guarantees can be extended to the bounds in terms of probability of large deviations.

We next analyze Case A) of Algorithm 1 and prove Theorem 1 case A).

Theorem C3 (Proof of Theorem 1 case A). *Select Case A in Algorithm 1 by choosing $A_k = \frac{1}{2}(k + 1)(k + 2)$, $\alpha_k = \frac{k+1}{2}$ with $\beta_k = 2L$. Assume that the variance σ_k satisfies $\sigma_k^2 \leq \frac{\varepsilon L \alpha_k}{A_k}$. Then, Algorithm 1 gives an ε-solution after $\sqrt{\frac{8LR^2}{\varepsilon}}$ iterations.*

Proof. By (28), we obtain

$$f(\mathbb{E}\hat{x}_N) - f^* = \varphi(\eta_N) - \varphi(\lambda^*) \le \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N} \sum_{k=0}^{N} \frac{A_k}{\beta_k - L} \sigma_k^2$$

$$\le \frac{4LR^2}{(N+1)(N+2)} + \frac{4\varepsilon}{L(N+1)(N+2)} \sum_{k=0}^{N} \frac{A_k}{L} \frac{\varepsilon L \alpha_k}{A_k}$$

$$= \frac{4LR^2}{(N+1)(N+2)} + \frac{4\varepsilon}{L(N+1)(N+2)} \sum_{k=0}^{N} L \frac{k+1}{4}$$

$$= \frac{4LR^2}{(N+1)(N+2)} + \frac{2\varepsilon}{(N+1)(N+2)} \frac{(N+1)(N+2)}{4}$$

$$= \frac{4LR^2}{(N+1)(N+2)} + \frac{\varepsilon}{2}. \tag{38}$$

We can see that after $N = \sqrt{\frac{8LR^2}{\varepsilon}}$ the r.h.s. becomes smaller than $\frac{\varepsilon}{2}$. The bound for $\|A\mathbb{E}\hat{x}_N - b\|_2$ is obtained in the same way using (29).

$\square$

We next analyze Case B) of Algorithm 1 and prove Theorem 1 case B).

Theorem C4 (Proof of Theorem 1 case B). *Assume that we have some constant variance $\sigma_k^2 = \sigma^2$. Given $c \ge 1, a = 2^c, b > 0$ let the coefficients be defined by $\alpha_k = \frac{k+1}{2^c}$ and $\beta_k = L + \frac{\sigma}{R} b(k+2)^c$, $A_k = \sum_{\ell=0}^{k} \alpha_\ell = \frac{1}{2a}(k+1)(k+2)$. Select case B of Algorithm 1 by setting $c = 3/2$ with the resulting step sizes $\alpha_k = \frac{k+1}{2\sqrt{2}}$ and $\beta_k = L + \frac{\sigma}{2^{1/4}\sqrt{3}R}(k+2)^{3/2}$. Then Algorithm 1 gives an ε-solution after $\sqrt{\frac{4LR^2}{\varepsilon}} \bigvee \frac{9\sigma^2 R^2}{\varepsilon^2}$ iterations.*

Proof. Using (28), we get

$$f(\mathbb{E}\hat{x}_N) - f^* \le \frac{\beta_N R^2}{2A_N} + \frac{1}{A_N} \sum_{k=0}^{N} \frac{A_k}{\beta_k - L} \sigma_k^2$$

$$\le \frac{2^{3/2} LR^2}{(N+1)(N+2)} + \frac{2^{9/4}(N+3)^{3/2}\sigma R}{\sqrt{3}(N+1)(N+2)}$$

$$= \frac{4LR^2}{N^2} + \frac{\sqrt{3}(3^{1/3}N)^{3/2}\sigma R}{N^2}$$

$$= \frac{4LR^2}{N^2} + \frac{3\sigma R}{N^{1/2}}$$

$$= \Theta\Big(\frac{LR^2}{N^2} + \frac{\sigma R}{\sqrt{N}}\Big). \tag{39}$$

By the choice of the number of iterations $N = \sqrt{\frac{4LR^2}{\varepsilon}} \bigvee \frac{9\sigma^2 R^2}{\varepsilon^2}$, we obtain that the r.h.s. is smaller than ε. The bound for $\|A\mathbb{E}\hat{x}_N - b\|_2$ is obtained in the same way using (29).

$\square$

D   Proof of Theorem 2

Proof. For the Wasserstein barycenter problem, the dual objective function is $\varphi(\boldsymbol{\lambda}) = \mathcal{W}_\gamma^*(\boldsymbol{\lambda})$, the primal objective is $f(p) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{W}_{\gamma, \mu_i}(p_i)$, and the linear operator is $A = \sqrt{W}$. We set the approximate gradient $\widetilde{\nabla} \, \Phi(\boldsymbol{\lambda}, Y_1, \dots, Y_{M_1}, \xi_1, \dots, \xi_{M_2}) := \widetilde{\nabla} \mathcal{W}_\gamma^*(\boldsymbol{\lambda}, Y_1, \dots, Y_{M_1}, \xi_1, \dots, \xi_{M_2})$. With this setting we can apply the Theorem C3. We set $\gamma = \varepsilon/(4\ln(\Omega))$ and the weights to $\omega_l = \frac{1}{m}$. According to Theorem C3 the variance has to satisfy the condition

$$\sigma_k^2 \le \frac{\varepsilon L \alpha_k}{A_k} = \frac{\varepsilon m \lambda_{\max}(W)}{(k+2)\gamma} = \frac{\lambda_{\max}(W) 4m \ln(\Omega)}{(k+2)}, \tag{40}$$

where we used also Lemma 2. According to Lemma 3 the total variance is bounded by

$$\mathbb{E}\|\widetilde{\nabla}\mathcal{W}_\gamma^*(\boldsymbol{\lambda}) - \nabla\mathcal{W}_\gamma^*(\boldsymbol{\lambda})\|_2^2 \le 2\lambda_{\max}(W) \sum_{i=1}^{m} \Big(\frac{1}{M_{i,1}} + \frac{1}{M_{i,2}} \Big), \; \boldsymbol{\lambda} \in \mathbb{R}^{mn}. \tag{41}$$

Setting the batch sizes $M_k = 1 \bigvee \left\lceil \frac{k+2}{\ln(\Omega)} \right\rceil \le 1 \bigvee \left\lceil \frac{k}{\ln(\Omega)} + 1 \right\rceil$ which is the Case A of Algorithm 2 satisfies the condition (40), and thus we can apply Theorem C3. Since the decreasing variance condition is satisfied the algorithm converges after $N = \sqrt{\frac{8LR^2}{\varepsilon}}$ iterations according to Theorem C3. Using the estimate $R \le \sqrt{\frac{2n}{m\lambda_{\min}^+(W)}}$ from (49), and $L = m\lambda_{\max}(W)/\gamma$ from Lemma 2 we have that Algorithm 2 converges after $\widetilde{O}\Big(\frac{\sqrt{\chi(\overline{W})n}}{\varepsilon}\Big)$ iterations.

Moreover, the total number of samples is

$$m \sum_{k=1}^{N} M_k \le m \sum_{k=1}^{N} k + 2 = m\Big(\frac{N(N+1)}{2} + 2N\Big)$$

$$= \widetilde{O}\left(\chi(\overline{W})mN^2\right) = \widetilde{O}\left(\frac{\chi(\overline{W})mn}{\varepsilon^2}\right). \tag{42}$$

Let us compute the computational complexity for a particular node i. On each iteration we have to batch M_k samples. Sampling individual vector sample $p_i(\bar{\lambda}_i, Y_r^i)$ costs n with $r = 1, \dots, M_k$. Therefor the overall sampling cost is $M_k n$ at the k-th iteration.

At iteration k the cost of summation of individual gradients (Equations (12) and (16) in Algorithm 2) is $\mathrm{nnz}(\overline{W}_i)M_k$.

Obtaining realizations costs n. Other operations cost $O(n)$, thus the computational complexity is $\widetilde{O}(M_k n)$ at each iteration. Summing the complexity up over the iterations results in

$$\sum_{k=1}^{N} (n\log_2 n + M_k(n + \mathrm{nnz}(\overline{W}_i))) = Nn\log_2 n + (n + \mathrm{nnz}(\overline{W}_i)) \sum_{k=1}^{N} \frac{k+2}{\ln(\Omega)}$$

$$\le Nn\log_2 n + (n + \mathrm{nnz}(\overline{W}_i)) \sum_{k=1}^{N} k + 2$$

$$= Nn\log_2 n + (n + \mathrm{nnz}(\overline{W}_i))\Big(\frac{N(N+1)}{2} + 2N\Big) \tag{43}$$

computational complexity per node. Summing this up over all the nodes results in

$$\sum_{i=1}^{m} Nn \log_2 n + (n + \mathrm{nnz}(\overline{W}_i))\Big(\frac{N(N+1)}{2} + 2N\Big)$$

$$= mNn \log_2 n + \Big(mn + \sum_{i=1}^{m} \mathrm{nnz}(\overline{W}_i)\Big)\Big(\frac{N(N+1)}{2} + 2N\Big)$$

$$= \widetilde{O}\left(mnN^2 \bigvee \sum_{i=1}^{m} \mathrm{nnz}(\overline{W}_i)N^2\right) = \widetilde{O}\left(\frac{\chi(\overline{W})mn^2}{\varepsilon^2} \bigvee \frac{\chi(\overline{W})n}{\varepsilon^2}\kappa(W)\right). \tag{44}$$

Compared to the non-quantized version the number of bit communications per node per round for the quantized version is now $\mathrm{nnz}(\overline{W}_i)(M_k \bigwedge n)$ instead of $\mathrm{nnz}(\overline{W}_i) \cdot n$ because we send M_k coordinates and at most the full dimensional vector. Summing this over the number of iterations and over the nodes gives us

$$\sum_{i=1}^{m}\sum_{k=1}^{N} \mathrm{nnz}(\overline{W}_i)(n \wedge M_k) \leq \sum_{i=1}^{m} \mathrm{nnz}(\overline{W}_i) \sum_{k=1}^{N} n \bigwedge \sum_{k=1}^{N} M_k$$

$$= \widetilde{O}\left(\sqrt{\chi(\overline{W})}\kappa(\overline{W})\left(\frac{n^{3/2}}{\varepsilon} \bigwedge \frac{n}{\varepsilon^2}\right)\right) \tag{45}$$

total number of sent communication bits.

$\square$

Remark 2. *The complexity bounds for the expander graph are easily derived from the general results of Theorem C3 and C4 using the equality $\sum_{i=1}^{m} \mathrm{nnz}(\overline{W}_i) = \widetilde{O}(m)$.*

E Proof of Theorem 3

Proof. We choose the dual, primal objective and approximate gradient as in proof of Theorem 2. According to Theorem C4, Algorithm 2 case B) converges after $N \geq \sqrt{\frac{4LR^2}{\varepsilon}} \bigvee \frac{9\sigma^2 R^2}{\varepsilon^2}$ iterations. We choose case B of Algorithm 2 with constant batch size $M = M_{i,1}^k = M_{i,2}^k$. Then, by Lemma 3

$$\sigma^2 \leq 2\lambda_{\max}(W) \sum_{i=1}^{m} \frac{2}{M} = \frac{4\lambda_{\max}(W)m}{M}. \tag{46}$$

Using again the estimate for R and L the algorithm converges after $N = \sqrt{\frac{32\chi(\overline{W})n}{\varepsilon}} \bigvee \frac{72\chi(\overline{W})n}{\varepsilon^2 M}$

$= \widetilde{O}\Big(\frac{\chi(\overline{W})n}{\varepsilon^2}\Big)$ iterations for a small constant batch size.

The total number of samples is $m \sum_{k=1}^{N} M = mNM = \widetilde{O}\left(\frac{\chi(\overline{W})mn}{\varepsilon^2}\right).$

Sampling individually $p_i(\bar{\lambda}_i, Y_r^i), \quad r = 1, \ldots, M$ costs n. Therefor the overall sampling cost is Mn for each iteration.

At iteration k the cost of summation of individual gradients (Equations (12) and (16) in Algorithm 2) is $\mathrm{nnz}(\overline{W}_i) \cdot M$. Sampling M type 2 samples costs Mn. Afterwards the cost of sampling M type 1 samples is also Mn. Other operations cost $O(n)$, thus the computational complexity is $\widetilde{O}(Mn)$ at each iteration. Summing the complexity up over the overall computational complexity per node is

$$\sum_{k=1}^{N} Mn = \widetilde{O}\left(\frac{\chi(\overline{W})n^2}{\varepsilon^2}\right) \tag{47}$$

resulting in $\sum_{i=1}^{m} \sum_{k=1}^{N} Mn \cdot \mathrm{nnz}(\overline{W}_i) = \widetilde{O}\left(\frac{\chi(\overline{W})n^2}{\varepsilon^2}\right)$ total arithmetic operations.

The number of bits sent per iteration for the quantized version is now $\mathrm{nnz}(\overline{W}_i) \cdot (M \wedge n) = \mathrm{nnz}(\overline{W}_i) \cdot M$ per node for $M \leq n$. Summing these estimates over the number of iterations gives us

$$\sum_{i=1}^{m} \sum_{k=1}^{N} \mathrm{nnz}(\overline{W}_i) \cdot M$$

$$= \sum_{i=1}^{m} \mathrm{nnz}(\overline{W}_i) M \left(\sqrt{\frac{32n\chi(\overline{W})}{\varepsilon}} \bigvee \frac{72n\chi(\overline{W})}{\varepsilon^2 M}\right) = \widetilde{O}\left(\frac{\kappa(\overline{W})\chi(\overline{W})n}{\varepsilon^2}\right) \tag{48}$$

number of communication bits sent in total.

$\square$

Remark 3. *After sampling M type 2 samples which costs Mn we can first create a binary tree which costs $n \log_2 n$. Afterwards the cost of sampling of one coordinate (type 1 samples) is just $M \log_2 n$.*

F Additional Results

This useful lemma is used to give an estimation of the area where we have to search for the dual solution:

Lemma F2. *We can upper bound the norm of the dual solution by $\|\lambda^*\|_2 \leq R$ with*

$R^2 = \dfrac{2n \sum\limits_{l=1}^{m} w_l^2 \|C_l\|_\infty^2}{\lambda_{\min}^+(W)}$ *for γ proportional to ε and c is a constant close to 2 (see [42] and Lemma 8 in [41]).*

We normalize the cost matrix $\|C\|_\infty^2 \leq 1$. For the weights $\omega_i = \dfrac{1}{m}$ the estimate becomes

$$R \leq \sqrt{\frac{2n}{m\lambda_{\min}^+(W)}}. \tag{49}$$

G Additional Simulation Results

In this section, we show additional simulation results for the proposed Algorithm 2. In each case, we assume there is a network with 30 nodes, and randomly generate 30 different Gaussian distributions
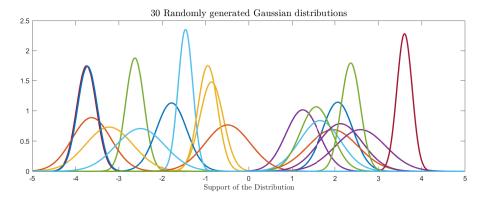
Figure 4: The randomly generated Gaussian distributions. Each distribution is assigned to each agent.

shown in Figure 4. Each node is assigned one of those Gaussian distributions, with uniformly sampled means and variances. Agents have the ability to query samples from those distributions, but are oblivious to the distribution itself. We test the proposed algorithm on five different graph classes: path, cycle, star, Erdős-Renywe test three different setups for number of samples, namely, $M_1 = 1$ and $M_2 = 100$, $M_1 = 10$ and $M_2 = 10$, and $M_1 = 100$ and $M_2 = 1$. A video of the resulting barycenter estimates for the path graph can be found in `https://youtu.be/aSwNZvCkrCw`.
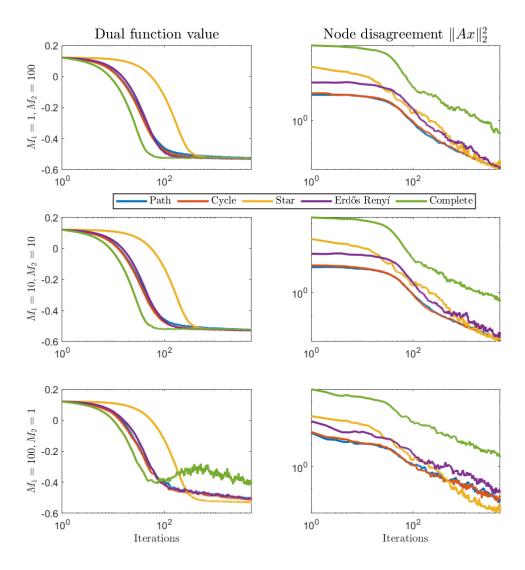
Figure 5: The dual function value and node disagreement of the iterates generated by Algorithm 2 on five different graphs, namely: path, cycle, star, Erdős-Renyí, and complete graphs, with 30 nodes each. Moreover, we show the effects of different number of samples per iterations. For each network, we test $M_1 = 1$ and $M_2 = 100$, $M_1 = 10$ and $M_2 = 10$, and $M_1 = 100$ and $M_2 = 1$.