

Distributed Markov Chain Monte Carlo Sampling based on the Alternating Direction Method of Multipliers

Alexandros E. Tzikas

*Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, USA*

ALEXTZIK@STANFORD.EDU

Licio Romao

*Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, USA*

LICIO@STANFORD.EDU

Mert Pilanci

*Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA*

PILANCI@STANFORD.EDU

Alessandro Abate

*Department of Computer Science
University of Oxford
Oxford, OX1 3QD, United Kingdom*

AABATE@CS.OX.AC.UK

Mykel J. Kochenderfer

*Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, USA*

MYKEL@STANFORD.EDU

Abstract

Many machine learning applications require operating on a spatially distributed dataset. Despite technological advances, privacy considerations and communication constraints may prevent gathering the entire dataset in a central unit. In this paper, we propose a distributed sampling scheme based on the alternating direction method of multipliers, which is commonly used in the optimization literature due to its fast convergence. In contrast to distributed optimization, distributed sampling allows for uncertainty quantification in Bayesian inference tasks. We provide both theoretical guarantees of our algorithm's convergence and experimental evidence of its superiority to the state-of-the-art. For our theoretical results, we use convex optimization tools to establish a fundamental inequality on the generated local sample iterates. This inequality enables us to show convergence of the distribution associated with these iterates to the underlying target distribution in Wasserstein distance. In simulation, we deploy our algorithm on linear and logistic regression tasks and illustrate its fast convergence compared to existing gradient-based methods.

Keywords: Markov chain Monte Carlo, distributed algorithms, sampling, alternating direction method of multipliers, proximal operator

1 Introduction

Since the 1950s, with the foundational works by Dantzig (1963) and later developments in the 1980s (Bertsekas and Tsitsiklis, 2015), various research communities have recognized the importance of distributing computation to improve scalability. For example, the robotics community has explored multi-robot simultaneous localization and planning, multi-robot target tracking, and multi-robot task assignment (Shorinwa et al., 2023a). The machine learning community has explored federated learning (Li et al., 2019) and distributed training of neural networks (Yu et al., 2022).

In order to select parameters in statistical models, optimization methods can be used to generate point estimates that aim at maximizing a given performance metric. A different approach for selecting parameters relies on a Bayesian treatment, whose goal is to obtain samples from the posterior distribution on the parameter space (Andrieu et al., 2003; Sekkat, 2022). Sampling methods constitute an important module in such a Bayesian paradigm, because they allow us to retain a full probabilistic framework of the uncertainty affecting our statistical model. Such Bayesian approaches are usually employed to avoid overfitting (Bhar et al., 2023; Andrieu et al., 2003) and perform uncertainty quantification (Bhar et al., 2023; Andrieu et al., 2003). Techniques based on Markov chain Monte Carlo (MCMC) (Andrieu et al., 2003), variational inference (Blei et al., 2017), and Langevin dynamics (Welling and Teh, 2011) have been proposed to perform sampling.

In this paper, we seek a distributed sampling mechanism for two reasons. First, there are applications in which the available data is spatially distributed, and collecting such data in a central processing unit is not feasible due to privacy issues, communication constraints or simply the size of the data. Second, the agents that hold the local, private data are usually equipped with computational power, thus enabling local computations using a (small) subset of the entire dataset.

Our proposed sampling scheme leverages the consensus alternating direction method of multipliers (ADMM), termed C-ADMM, presented by Mateos et al. (2010); Shorinwa et al. (2023a,b), and Shi et al. (2014). A unique and crucial feature of the proposed sampling scheme, which we refer to as the distributed ADMM-based sampler (D-ADMMS), is the addition of a noise term in the proximal step of C-ADMM. As opposed to existing literature (Gürbüzbalaban et al., 2021) that relies on gradient computations for the parameter updates, the added noise and proximal updates of D-ADMMS are essential to its superior convergence behavior. In summary, our main contributions are as follows:

- We show how to adapt C-ADMM to perform sampling in a distributed manner, by designing a new distributed sampling algorithm, which is termed D-ADMMS.
- We develop a new analysis to show convergence of the distribution of the generated iterates of D-ADMMS to the target distribution.
- We study the performance of the proposed scheme on regression tasks and discuss advantages with respect to standard Langevin dynamics.

The remainder of the paper is structured as follows. In section 2, we review related work. In section 3, we formulate the problem, while in section 4 we describe our proposed approach. In section 5, we detail the convergence analysis of our proposed algorithm.

Section 6 contains the numerical experiments. Finally, we conclude with closing remarks and a discussion of future work in section 7.

2 Related Work

Langevin and Hamiltonian Gradient MCMC. When the goal is to sample from a target distribution in Gibbs form ($\propto \exp -U(x)$), discretizations of stochastic differential equations (SDEs) are an attractive sampling technique because the stationary distribution of these discretizations is usually close to the target distribution. Langevin algorithms are MCMC methods based on the discretization of the overdamped Langevin diffusion, while Hamiltonian algorithms are MCMC methods based on the underdamped Langevin diffusion. Both Langevin and Hamiltonian methods scale well with high-dimensional sampling spaces (Kungurtsev et al., 2023). They use first-order information of the target distribution to guide the dynamics towards the relevant regions of the parameter space. The stationary distribution of such algorithms, for example the unadjusted Langevin algorithm (ULA) (Parayil et al., 2021), may contain a bias with respect to the stationary distribution of the underlying SDE. To mitigate this bias, a Metropolis-Hastings correction can be introduced at the expense of increasing computation. Stochastic optimization tools were combined with Langevin dynamics by Welling and Teh (2011) to obtain a *single-agent* MCMC algorithm that uses mini-batches of data at every iteration (to obtain *gradient estimates*), along with a variable step-size and noise variance to guide convergence to the desired distribution.

These methods have recently been extended to the distributed setting, where the goal is to sample from a distribution, which is proportional to $\exp -\sum_{i \in \mathcal{V}} f_i(x)$, in a network of a set \mathcal{V} of agents. The function f_i , for $i \in \mathcal{V}$, is only known to the corresponding agent i . Agents can communicate with their direct neighbors, as defined by the set of edges in the communication graph. The methods that have been studied include: a modified version of *distributed stochastic gradient descent* (Nedic and Ozdaglar, 2009), namely distributed SGLD (D-SGLD) (Gürbüzbalaban et al., 2021), and a method called *distributed stochastic gradient Hamiltonian Monte Carlo* (D-SGHMC) (Gürbüzbalaban et al., 2021), which is an adaptation of the SGHMC method to the distributed setting. SGHMC can be faster than SGLD, because it is based on the discretization of the underdamped inertial Langevin diffusion, which converges to the stationary distribution faster than its overdamped counterpart due to a momentum-based accelerating step. Gürbüzbalaban et al. (2021) provide convergence guarantees of the probability distribution of the local iterates of each agent to the target distribution in terms of Wasserstein distance.

A *distributed Hamiltonian Monte-Carlo* algorithm with a Metropolis acceptance step was recently derived (Kungurtsev et al., 2023). Each agent estimates *both first-order and second-order information* of the global potential function of the target distribution, making this approach different from ours. The ULA, *which is gradient-based*, has also been modified for the *distributed* case, giving rise to the D-ULA scheme (Parayil et al., 2021). Assuming conditional independence among the data, the posterior is given as a product of local posteriors that are used to define the local dynamics. *The distributed gradient-based ULA* (Parayil et al., 2021) has been modified to reduce the communication requirements between the agents by Bhar et al. (2023). The difference is that the local iterate is not shared at

every iteration but only asynchronously by a triggering mechanism, which is based on the iterate's variation.

It should be evident from this discussion that *most distributed sampling algorithms are gradient-based*. The optimization literature suggests that stochastic first-order methods are usually not the fastest to converge (Ryu and Yin, 2022), and can be sensitive to the choice of hyperparameters (Toulis et al., 2021). At the same time, C-ADMM offers fast convergence in distributed optimization tasks (Shorinwa et al., 2023b). Motivated by this, we focus on developing an ADMM-inspired distributed sampling scheme.

Distributed MCMC. In the case of large datasets, data is divided among agents. Sampling from the global posterior in this case can be done using parallelized MCMC. Neiswanger et al. (2014) develop an MCMC sampling framework where each processing unit contains part of the dataset. Each agent deploys an MCMC method independently (without communicating) to sample from the product term of the global posterior related to its dataset. Then, by appropriately combining these individual samples, samples from the global posterior can be obtained, in the spirit of a divide-and-conquer scheme. *The combination procedure nevertheless requires a central coordinator.* An alternative approach is to use several Markov chains. Ahn et al. (2014) propose such a *distributed sampling algorithm based on stochastic gradient Langevin dynamics (SGLD)*. However, *it requires a central coordinator.*

MCMC and Federated Learning. *The paradigm of SGLD has been used in the field of federated learning (Deng et al., 2022) to convert optimization algorithms into sampling algorithms.* In federated learning, multiple agents aim to jointly optimize an objective without sharing their private data. The agents containing the private data can communicate with a centralized unit. Every iteration consists of a broadcast step, multiple local gradient steps by each agent using its local function, and finally a consensus step. Inspired by SGLD, the federated averaging optimization algorithm was modified into a *gradient-based sampling algorithm by adding noise in the local gradient updates (Deng et al., 2022)*. A combination of *gradient-based Langevin dynamics* and compression techniques to reduce the communication cost has been studied (Karagulyan and Richtárik, 2023). *Nevertheless, the algorithm is not applicable in a distributed setting because it assumes a centralized processing unit.*

A generalization of the federated averaging algorithm to compute the mode of the posterior distribution has been explored (Al-Shedivat et al., 2020). *The centralized processing unit performs gradient steps on a suitable objective, while the agents employ a variant of stochastic gradient MCMC in order to compute local covariances and expectations.* Finally, *a distributed method* that minimizes the Kullback-Leibler (KL) divergence with the data likelihood function extends federated learning (Lalitha et al., 2019). It consists of a local Bayesian update, *a projection onto the allowed family of posteriors*, and a consensus step.

Proximal Langevin Algorithms. ADMM has a proximal update at each iteration. The literature suggests that proximal operators are more stable than subgradients (Bauschke and Combettes, 2011). Proximal optimization algorithms can be viewed as discretizations of gradient flow differential equations, whose equilibria are the minimizers of the considered function (Parikh and Boyd, 2014). Analogously, although common forms of Langevin MCMC methods employ subgradients, proximal MCMC methods possess favorable convergence and efficiency properties (Durmus et al., 2018; Salim et al., 2019; Pereyra, 2016). While the classical ULA is based on a forward Euler approximation of the Langevin SDE

that has the target distribution as the stationary distribution, proximal Langevin algorithms are based on discretizing an SDE whose stationary distribution equals the target distribution's Moreau approximation (Pereyra, 2016). The regularity properties of the Moreau approximation function lead to discrete approximations of the SDE with favourable stability and convergence qualities. To correct for the incurred error, a Metropolis-hastings accept-reject step has been proposed (Pereyra, 2016). However, *this algorithm requires knowing the full energy function of the desired distribution at every iteration and therefore it is not suitable for distributed computation.*

A proximal stochastic Langevin algorithm has been proposed to sample from a distribution $\propto \exp -U(x)$ with $U(x) = F(x) + \sum_i G_i(x)$, where F is smooth convex and G_i are (possibly non-smooth) convex functions (Salim et al., 2019). The authors assume that $F(x)$ and $G_i(x)$ can be written as expectations of functions $f(x, \xi)$ and $g_i(x, \xi)$, where ξ is a random variable. This allows the use of stochastic information on $F(x)$ and $G_i(x)$ when designing the algorithm. At every iteration, a stochastic gradient step is taken with respect to $F(x)$ and Gaussian noise is added. Then, stochastic proximal operators of each G_i are deployed sequentially. *Although our problem fits in this problem formulation, the algorithm by Salim et al. (2019) is not suitable for deployment in a distributed network setting, as it requires the sequential deployment of the proximal operators.*

Connections between ADMM, Sampling, and SDEs. ADMM is an optimization method that has been developed to combine the robustness and convergence of the method of multipliers and the decomposability of dual ascent (Boyd et al., 2011). Vono et al. (2019) propose fast and efficient variations of the Gibbs sampler, based on the idea of variable splitting and variable augmentation, in order to sample from $\exp -\sum_i f_i(x)$. The methods employ surrogate distributions, which converge in the limit to the target distribution, and consist of sequentially sampling a variable from its conditional distribution on the remaining ones. If, instead of sampling, we perform MAP optimization at each step of their proposed algorithm, we recover the ADMM optimization method. *Although this method is closely related to our proposed scheme, it requires centralized communication at every step, as discussed in Vono et al. (2019, appendix B).* Other distributed sampling methods have been proposed that are inspired by ADMM and the variable-splitting idea in optimization. Rendell et al. (2020) introduce auxiliary variables and construct an MCMC algorithm on an extended state space. Their algorithm can be partly deployed in a distributed manner among agents, because the auxiliary variables are conditionally independent given the variable of interest. The auxiliary variables can be independently sampled at each agent given the variable of interest. *The algorithm requires a centralized machine to sample the variable of interest because the method consists of alternating between sampling the agent auxiliary variables given the variable of interest and vice versa.* An instance of this approach, the split Gibbs sampler, has been studied by Vono et al. (2022).

Connections between ADMM and SDEs are made for the case of a stochastic optimization problem where the sum of a function $g(x)$ and the expected value of another function $f(x, \xi)$, where ξ is a random variable, is to be minimized (Zhou et al., 2020). At every iteration, the primal variables are updated using a random sample of f , namely $f(x, \xi_i)$. Zhou et al. (2020) prove that the primal iterate converges as the step size decreases, in some sense, to the stochastic process satisfying a given SDE. *Different from our formulation, the*

only noise in the setting of Zhou et al. (2020) stems from the noisy sample of fat every iteration.

3 Problem Formulation

We consider a network of agents. Each agent i possesses a local function $f_i(x)$, where $x \in \mathbb{R}^d$. In machine learning applications, $f_i(x)$ could pertain to the loss, as a function of the model's parameter x , on the agent i 's dataset. As such, $f_i(x)$ is considered unknown to all agents other than agent i . This is due to privacy considerations and the high cost of transmitting agent i 's data across the network. The overall goal is to sample in a distributed manner from the distribution

$$\mu^*(x) \propto \exp -F(x), \quad F(x) = \sum_{i \in \mathcal{V}} f_i(x). \quad (1)$$

Such a log-concave function arises as the posterior distribution in various Bayesian inference problems, such as distributed Bayesian linear and logistic regression (Gürbüzbalaban et al., 2021).

The communication topology of the network is characterized by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$, for some integer N , is the set of agents, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of communication links, i.e., $(i, j) \in \mathcal{E}$ if and only if $i \neq j$ and node i can communicate directly with node j . The neighborhood of agent i is denoted $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$. The cardinality of \mathcal{N}_i is denoted N_i . Complementary to the undirected graph \mathcal{G} , we also describe the existing communication topology via a directed graph, $\mathcal{G}_d = (\mathcal{V}, \mathcal{A})$. Every edge $e \in \mathcal{E}$ is associated with two directed links in \mathcal{A} that connect the same nodes as e . Therefore the cardinality of \mathcal{A} is twice that of \mathcal{E} , i.e., $|\mathcal{A}| = 2|\mathcal{E}|$, and \mathcal{G}_d describes the same topology as \mathcal{G} .

4 Description of the Proposed Method

In this section, we introduce our proposed method, D-ADMMS. We start by providing background material on distributed optimization and C-ADMM. We then describe how we modify C-ADMM, which is used for distributed optimization, in order to obtain D-ADMMS, which performs distributed sampling.

4.1 Background on C-ADMM for Distributed Optimization

In distributed optimization problems, we consider the set-up introduced in section 3. However, in distributed optimization we aim to solve the optimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} f_i(x), \quad (2)$$

instead of sample from eq. (1). We may introduce a local optimization variable, x_i for each agent i , and consensus constraints in order to obtain the optimization problem

$$\begin{aligned} & \underset{\substack{\{x_i\}_{i \in \mathcal{V}}, \\ \{z_{i,j}\}_{(i,j) \in \mathcal{E}}}}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} f_i(x_i) \\ & \text{subject to} \quad x_i = x_j \quad \forall (i, j) \in \mathcal{E}. \end{aligned} \quad (3)$$

If \mathcal{G} is connected, x^* is an optimal point of problem (2) if and only if $x_i = x^*, \forall i \in \mathcal{V}$, is an optimal point of problem (3). Problem (3) lends itself to a distributed treatment.

C-ADMM is a distributed optimization algorithm inspired by the method of multipliers, which computes a primal-dual solution pair for the optimization problem via the augmented Lagrangian: the primal variables x_i are updated as the minimizers of the augmented Lagrangian and the dual variables are updated via (dual) gradient ascent on the augmented Lagrangian Shorinwa et al. (2023b). C-ADMM introduces auxiliary optimization variables to problem (3) for each consensus constraint, which allows for distributed update steps. At step $(k+1)$, the primal variable of agent i , $x_i^{(k+1)}$, is updated according to

$$x_i^{(k+1)} \leftarrow \operatorname{argmin}_x \left\{ f_i(x) + p_i^{(k)^T} x + \rho \sum_{j \in \mathcal{N}_i} \left\| x - \frac{x_i^{(k)} + x_j^{(k)}}{2} \right\|_2^2 \right\}, \quad (4)$$

while the dual variable of agent i at step $(k+1)$, $p_i^{(k+1)}$, which corresponds to the consensus constraints involving agent i and its neighbors, is updated according to

$$p_i^{(k+1)} \leftarrow p_i^{(k)} + \rho \sum_{j \in \mathcal{N}_i} (x_i^{(k+1)} - x_j^{(k+1)}), \quad (5)$$

with initialization at zero.

4.2 The Proposed Method: D-ADMMS

Our distributed MCMC algorithm, D-ADMMS, is a modified version of the C-ADMM optimization algorithm (Shorinwa et al., 2023b). In contrast to C-ADMM, which is a distributed optimization algorithm, D-ADMMS is a distributed sampling algorithm. D-ADMMS is given in Algorithm 1. A key feature of the proposed sampling scheme is the added noise in the update of the primal variables. The MCMC sample corresponds to the local primal variable $x_i^{(k)}$. At every iteration, each agent updates its local primal iterate by solving a proximal problem. The primal update of D-ADMMS can equivalently be written as

$$x_i^{(k+1)} = \operatorname{prox}_{\gamma_i f_i} \left\{ \sum_{j \in \mathcal{N}_i} \frac{x_i^{(k)} + x_j^{(k)}}{2N_i} + \frac{\sqrt{2}}{2\rho} w_i^{(k+1)} + \frac{p_i^{(k)}}{2\rho N_i} \right\}, \quad (6)$$

where $\gamma_i = 2 / (\rho N_i)$. Each agent then communicates its primal variable to its neighbors and updates its dual variable $p_i^{(k)}$ based on the disagreement of the primal variables of the neighboring agents. The inspiration of the added noise in the proximal step is derived from the algorithm by Salim et al. (2019). The first step at each iteration of the algorithm by Salim et al. (2019) consists of a noiseless gradient step and then a proximal update with added noise. In D-ADMMS, the noiseless gradient step corresponds to the update of the dual variables, while the primal variables are updated with a noisy proximal step. The scaling of the noise involved is however different between the two algorithms.

5 Theoretical Analysis of the Proposed Method

We study the convergence of the distribution associated with the primal iterates, $x_i^{(k)}$, of the proposed algorithm, D-ADMMS, to the target distribution $\mu^*(x)$, in terms of 2-Wasserstein

Algorithm 1: Proposed Algorithm (D-ADMMS)

Initialization: $k \leftarrow 0$, $x_i^{(k)} \in \mathbb{R}^d$, $p_i^{(k)} = \mathbf{0}$ $\forall i \in \mathcal{V}$

Parameters: $\rho > 0$

Output: samples $x_i^{(k+1)}$ $\forall i \in \mathcal{V}$

do in parallel $\forall i \in \mathcal{V}$

$w_i^{(k+1)} \sim \mathcal{N}(0, I)$

$$x_i^{(k+1)} \leftarrow \operatorname{argmin}_x \left\{ f_i(x) + p_i^{(k)T} x + \rho \sum_{j \in \mathcal{N}_i} \left\| x - \frac{x_i^{(k)} + x_j^{(k)}}{2} + \frac{\sqrt{2}}{2\rho} w_i^{(k+1)} \right\|_2^2 \right\}$$

Communicate $x_i^{(k+1)}$ to neighbors $j \in \mathcal{N}_i$

Receive $x_j^{(k+1)}$ from neighbors $j \in \mathcal{N}_i$

$$p_i^{(k+1)} \leftarrow p_i^{(k)} + \rho \sum_{j \in \mathcal{N}_i} (x_i^{(k+1)} - x_j^{(k+1)})$$

$k \leftarrow k + 1$

distance. The 2-Wasserstein distance between two probability measures μ and ν with finite second moments is defined as

$$W(\mu, \nu) = \left(\inf_{\tau \in \Gamma(\mu, \nu)} \mathbb{E}_{(x,y) \sim \tau} \|x - y\|^2 \right)^{1/2}, \quad (7)$$

where $\Gamma(\mu, \nu)$ is the set of all couplings between μ and ν (Villani and Villani, 2009). We adopt a convex analysis perspective. We base our analysis of the convergence rate (with respect to the iterates' Wasserstein distance to the target distribution) of D-ADMMS on the analysis of the convergence rate (with respect to the iterates' Euclidian distance to the optimal point) of C-ADMM by Shi et al. (2014). Modifying the analysis by Shi et al. (2014) for our purposes is not trivial for two reasons: i) the added noise in the primal update of D-ADMMS gives rise to terms that do not exist in C-ADMM, and ii) it is not straight forward how to obtain a Wasserstein distance bound on the distribution of the iterates from a Euclidian distance bound of the iterates. We use $\|\cdot\|$ for the standard Euclidean norm and $\|\cdot\|_G^2$ for the G -matrix norm $(\cdot)^T G (\cdot)$.

This section is organized as follows. We start the first subsection by stating our assumptions and introducing helpful quantities. We finish it with a lemma that shows the equivalence of the D-ADMMS updates and a different set of updates, which involve the same primal variables. This second set of updates is used in our analysis. In the second subsection, we prove a recursive inequality for the Wasserstein distance between the distribution of the primal iterates of D-ADMMS and the target distribution of eq. (1). Our main result is Theorem 3, which is included in the third subsection. Theorem 3 states that there exists a decreasing upper-bound on the Wasserstein distance between the distribution of the primal iterates of D-ADMMS and the target distribution of eq. (1), as the iterations evolve.

5.1 Assumptions, Definitions, and an Equivalent Expression of D-ADMMS

Assumption 1 *The local objective functions $f_i(x)$ are strongly convex: $\forall x_a, x_b \in \mathbb{R}^d, \forall i \in \mathcal{V}$, it holds that*

$$\langle \nabla f_i(x_a) - \nabla f_i(x_b), x_a - x_b \rangle \geq m_{f_i} \|x_a - x_b\|^2, \quad m_{f_i} > 0.$$

Assumption 2 *The gradients of the local objective functions are Lipschitz continuous: $\forall x_a, x_b \in \mathbb{R}^d, \forall i \in \mathcal{V}$, it holds that*

$$\|\nabla f_i(x_a) - \nabla f_i(x_b)\| \leq M_{f_i} \|x_a - x_b\|, \quad M_{f_i} > 0.$$

Assumption 3 *The graph topology \mathcal{G} is connected.*

We further define the consensus convex optimization problem associated with μ^* as

$$\begin{aligned} & \underset{\substack{\{x_i\}_{i \in \mathcal{V}}, \\ \{z_{i,j}\}_{(i,j) \in \mathcal{A}}}}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} f_i(x_i) \\ & \text{subject to} \quad x_i = z_{i,j}, \quad x_j = z_{i,j} \quad \forall (i, j) \in \mathcal{A}. \end{aligned} \tag{8}$$

Concatenating each x_i in $X \in \mathbb{R}^{Nd}$ and all $z_{i,j}$ in $Z \in \mathbb{R}^{|\mathcal{A}|d}$, we may write the constraint of eq. (8) as

$$AX + BZ = 0. \tag{9}$$

Here, $A = [A_1; A_2]$, where $A_1, A_2 \in \mathbb{R}^{|\mathcal{A}|d \times Nd}$. If $(i, j) \in \mathcal{A}$ and $z_{i,j}$ is the q -th block of Z , then the (q, i) block of A_1 and the (q, j) block of A_2 are the $d \times d$ identity matrices. All other blocks of A_1, A_2 contain zero entries. Also $B = [-I_{|\mathcal{A}|d}; -I_{|\mathcal{A}|d}]$, where $I_{|\mathcal{A}|d}$ is the $|\mathcal{A}|d \times |\mathcal{A}|d$ identity matrix. We define $f(X) = \sum_{i \in \mathcal{V}} f_i(x_i)$. By Assumptions 1 and 2, $f(X)$ is strongly convex with constant $m_f = \min_i m_{f_i}$ and its gradients are Lipschitz continuous with constant $M_f = \max_i M_{f_i}$.

Assumption 4 $\sum_{i \in \mathcal{V}} f_i(x)$ admits a (unique) minimizer.

Problem (8) then admits a unique solution (X^*, Z^*) , because \mathcal{G} is connected and $\sum_{i \in \mathcal{V}} f_i(x)$ admits a unique minimizer. It is evident that the optimization problem (8) is related to our problem of interest, which is to sample from distribution (1) in a distributed manner, but solving problem (8) is not our objective. In our analysis, we use the minimizer of problem (8) as a fixed point to which the Euclidian distance of the primal iterates of D-ADMMS can be bounded. By assigning a point mass distribution to this fixed point, we are then able to obtain relations for the Wasserstein distance of the primal iterates of D-ADMMS to the target distribution.

We now introduce matrices M_-, M_+, L_-, L_+ , and D , based on the network topology \mathcal{G} . M_+ and M_- are the extended unoriented and oriented incidence matrices of \mathcal{G}_d , respectively. L_+ and L_- are the extended signless and signed Laplacian matrices of \mathcal{G} , respectively. D is the extended degree matrix of \mathcal{G} . By ‘‘extended’’ we mean the Kronecker product with I_d . We also denote $w^{(k)} = (w_1^{(k)}, \dots, w_N^{(k)}) \in \mathbb{R}^{Nd}$, and $p^{(k)} = (p_1^{(k)}, \dots, p_N^{(k)}) \in \mathbb{R}^{Nd}$.

Lemma 1 Define $\beta \in \mathbb{R}^{|\mathcal{A}|d}$. The update equations of D-ADMMS in Algorithm 1 can be derived from the iterates

$$\nabla f(X^{(k+1)}) + M_- \beta^{(k+1)} + \sqrt{2}Dw^{(k+1)} = \rho M_+ (Z^{(k)} - Z^{(k+1)}), \quad (10)$$

$$\beta^{(k+1)} - \beta^{(k)} - \frac{\rho}{2} M_-^T X^{(k+1)} = 0, \quad (11)$$

$$\frac{1}{2} M_+^T X^{(k)} - Z^{(k)} = 0, \quad (12)$$

where $X^{(k)}$ is the concatenation of the $x_i^{(k)}$ from Algorithm 1. ■

Proof The proof is included in Appendix A.

By the lemma above, we may analyze the primal iterates of D-ADMMS using eq. (10-12).

The Karush-Kuhn-Tucker (KKT) conditions for problem (8) are

$$\nabla f(X^*) + M_- \beta^* = 0, \quad (13)$$

$$M_-^T X^* = 0, \quad (14)$$

$$\frac{1}{2} M_+^T X^* - Z^* = 0, \quad (15)$$

as described by Shi et al. (2014), where β^* denotes the unique optimal multiplier that exists in the column space of M_-^T . Since the equality constraints of problem (8) are feasible, by Slater's condition (Boyd and Vandenberghe, 2004), there exists an optimal multiplier $\hat{\beta}$ that satisfies the KKT conditions. Its projection onto the column space of M_-^T is β^* , as analyzed by Shi et al. (2014).

Assumption 5 $\beta^{(0)}$ is in the column space of M_-^T .

By inspection of eq. (11), we observe that under Assumption 5, $\beta^{(k)}$ is in the column space of M_-^T for all $k \geq 0$.

5.2 A Recursive Inequality of Convergence for D-ADMMS

We define $U = (Z, \beta)$, $G = \text{diag}\{\rho I_{2|\mathcal{E}|d}, \frac{1}{\rho} I_{2|\mathcal{E}|d}\}$, and $U^* = (Z^*, \beta^*)$. We also denote the largest singular value and the smallest nonzero singular value of a matrix M , as $\sigma_{\max}(M)$, and $\sigma_{\min}(M)$ respectively. We may now compute recursive bounds on $U^{(k)}$ and $X^{(k)}$. Then we can obtain recursive bounds on the Wasserstein distance of the iterate. We denote $\mu_{(\cdot)}$ as the probability distribution of random variable (\cdot) . Also $\boldsymbol{\mu}^*(X) = \prod_{i=1}^N \mu^*(x_i)$. Finally $W_G^2(\mu_{U^{(k)}}, \mu_{U^*}) = \rho W^2(\mu_Z, \mu_{Z^*}) + (1/\rho) W^2(\mu_\beta, \mu_{\beta^*})$.

Lemma 2 Under Assumptions (1-5), for any $\kappa > 1$, there exists a $\delta > 0$, such that the distribution of the iterates of D-ADMMS satisfies the relation

$$\begin{aligned} W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*) &\leq \frac{1}{\sqrt{m_f}} W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \\ &\quad \frac{1}{\sqrt{2m_f}} \sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)} + W(\mu_{X^* - \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}}, \boldsymbol{\mu}^*), \end{aligned} \quad (16)$$

and $W_G(\mu_{U^{(k)}}, \mu_{U^*})$ is recursively upper-bounded by

$$W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) \leq \sqrt{a} W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}} + \sqrt{|\mathbb{E}(r^{(k+1)}) - \left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2|}, \quad (17)$$

where

$$y^{(k+1)} = 2\frac{b}{\sqrt{m_f}}\bar{w}^{(k+1)} + c\sigma_{\max}(M_-)\sqrt{\rho}\|Dw^{(k+1)}\| + \frac{cd}{\sqrt{m_f}}\|Dw^{(k+1)}\|, \quad (18)$$

$$\begin{aligned} r^{(k+1)} &= \frac{\sqrt{2}b}{m_f}\bar{w}^{(k+1)}\|Dw^{(k+1)}\| + b\left(\bar{w}^{(k+1)}\right)^2 + \frac{b}{2m_f^2}\|Dw^{(k+1)}\|^2 \\ &\quad + \frac{\sqrt{2}cd}{m_f}\|Dw^{(k+1)}\|^2 - e\|Dw^{(k+1)}\|^2, \end{aligned} \quad (19)$$

$$\bar{w}^{(k+1)} = \left\| \frac{1}{\sqrt{2}m_f}Dw^{(k+1)} \right\| + \left\| \sqrt{2}Dw^{(k+1)} \right\|,$$

$$\delta = \min \left\{ \frac{(\kappa - 1)\sigma_{\min}^2(M_-)}{\kappa\sigma_{\max}^2(M_+)}, \frac{m_f}{\frac{\rho}{4}\sigma_{\max}^2(M_+) + \frac{\kappa M_f^2}{\rho\sigma_{\min}^2(M_-)}} \right\} > 0, \quad (20)$$

and

$$\begin{aligned} a &= \frac{2m_f + 1}{2m_f(1 + \delta)}, \quad b = \frac{1}{2(1 + \delta)}, \quad c = \frac{2\sqrt{2}\delta}{(1 + \delta)\sigma_{\min}^2(M_-)}, \\ d &= \frac{\rho\sigma_{\max}^2(M_-)}{2}, \quad e = \frac{2\delta}{(1 + \delta)\rho\sigma_{\min}^2(M_-)}. \end{aligned} \quad (21)$$

Proof The proof is included in Appendix B. ■

5.3 Our Main Result

By inspection of eq. (17-21), if there exists a $\rho > 0$ such that δ satisfies $2m_f\delta > 1$, then $a < 1$ and we have convergence in terms of Wasserstein distance for the primal iterates $x_i^{(k)}$. This idea is formalized in the following theorem, whose result depends on the condition number of $f(X)$, $\tau_f = \frac{M_f}{m_f}$, and the condition number of the graph topology, $\tau_G = \frac{\sigma_{\max}(M_+)}{\sigma_{\min}(M_-)}$.

Theorem 3 Assume Assumptions (1-5) hold and $\tau_f^{-1}\sqrt{\tau_f^{-2} + 4\tau_G^{-2}} - \tau_f^{-2} > m_f^{-1}$ is true. Then, there exists a $\rho > 0$ such that $a < 1$ and

$$\begin{aligned} W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*) &\leq \frac{1}{\sqrt{m_f}} (\sqrt{a})^k W_G(\mu_{U^0}, \mu_{U^*}) + \frac{1}{\sqrt{am_f}} \frac{Y}{1 - \sqrt{a}} + \\ &\quad \frac{1}{\sqrt{m_f}} \frac{\sqrt{R}}{1 - \sqrt{a}} + \frac{1}{\sqrt{2m_f}} \sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)} + W(\mu_{X^* - \frac{1}{\sqrt{2m_f}}Dw^{(k+1)}}, \boldsymbol{\mu}^*), \end{aligned}$$

Y and R are upper bounds on the terms $\mathbb{E}(y^{(l)})$ and $\mathbb{E}(r^{(l)})$ respectively, and $Y, R \geq 0$ holds.

Proof The proof is included in Appendix C. ■

Because of the definitions $L_+ = \frac{1}{2}M_+M_+^T$ and $L_- = \frac{1}{2}M_-M_-^T$, we have that $\tau_G = \sqrt{\frac{\sigma_{\max}(L_+)}{\sigma_{\min}(L_-)}}$. $\sigma_{\min}(L_-)$ is known as the graph's algebraic connectivity (Shi et al., 2014), while $\sigma_{\max}(L_+)$ is related to the node degrees of \mathcal{G} (Cvetković et al., 2007). Assuming a ring-cyclic graph topology of 5 agents, we have $\tau_G = 1.7$. If we further assume $m_f = 2$, we get the sufficient condition for convergence: $\tau_f < 1.23$. Increasing the connectedness of the graph, for a fully connected graph of 5 agents, we have $\tau_G = 1.26$. Then, for $m_f = 2$, the sufficient condition becomes $\tau_f < \sqrt{6}$. We observe that as the graph becomes more connected, τ_G decreases. Then, fixing τ_f and m_f , Theorem 1 implies that D-ADMMS converges if the connectivity is above a certain threshold. In addition, a decreases with increasing connectivity. The convergence upper bound however also contains terms, including Y and R , which depend on noise and topology characteristics and can increase as the graph becomes more connected (D has larger values in its diagonal and a decreases).

We also note that the sufficient convergence condition $m_f^{-1} < \tau_f^{-1} \sqrt{\tau_f^{-2} + 4\tau_G^{-2}} - \tau_f^{-2}$ is not symmetric with respect to the condition number τ_f . In other words, scaling the functions f_i by a common constant, which does not change τ_f , modifies the convergence condition. This is because the scaling constant affects the solution of the proximal update for $x_i^{(k+1)}$.

Since ADMM is practically stable when the f_i are not strongly convex, e.g., for indicator functions, our algorithm is also applicable in this case. However, the theoretical analysis above assumes strong convexity.

6 Simulation Results

In this section, we test D-ADMMS in simulation. We first discuss the existing baselines from the literature and then provide simulation results for two different scenarios: Bayesian linear regression and Bayesian logistic regression.¹

6.1 Description of Baseline Algorithms

We compare our proposed algorithm against D-SGLD (Gürbüzbalaban et al., 2021), D-SGHMC (Gürbüzbalaban et al., 2021), and D-ULA (Parayil et al., 2021). We let S be a doubly stochastic matrix associated with the network's communication topology, i.e., a matrix whose (i, j) -entry, which we denote by S_{ij} , is non-negative and less than one, and is different from zero if and only if the entry $j \in \mathcal{N}_i$; whenever $j = i$, we let S_{ii} be defined as $1 - \sum_{j \in \mathcal{N}_i} S_{ij}$.

D-SGLD is characterized by

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} S_{ij} x_j^{(k)} - \eta_{DSGLD} \nabla f_i(x_i^{(k)}) + \sqrt{2\eta_{DSGLD}} w_i^{(k+1)}, \quad w_i^{(k+1)} \sim \mathcal{N}(0, I). \quad (22)$$

1. The source code for the presented experiments can be found in the following repository: https://github.com/sisl/Distributed_ADMM_Sampler

D-SGHMC proceeds with

$$v_i^{(k+1)} = v_i^{(k)} - \eta_{\text{DSGHMC}} \left(\gamma v_i^{(k)} + \nabla f_i(x_i^{(k)}) \right) + \sqrt{2\gamma\eta_{\text{DSGHMC}}} w_i^{(k+1)}, \quad w_i^{(k+1)} \sim \mathcal{N}(0, I), \quad (23)$$

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} S_{ij} x_j^{(k)} + \eta_{\text{DSGHMC}} v_i^{(k+1)}. \quad (24)$$

Finally, D-ULA evolves through the update

$$\begin{aligned} x_i^{(k+1)} &= x_i^{(k)} - \zeta^{(k)} \sum_{j \in \mathcal{N}_i} \left(x_i^{(k)} - x_j^{(k)} \right) - \alpha^{(k)} N \nabla f_i(x_i^{(k)}) \\ &\quad + \sqrt{2\alpha^{(k)}} w_i^{(k+1)}, \quad w_i^{(k+1)} \sim \mathcal{N}(0, NI). \end{aligned} \quad (25)$$

6.2 Bayesian Linear Regression

6.2.1 PROBLEM

We study the distributed Bayesian linear regression setting. We consider a varying number of agents ($N = 5, 20, 100$) and a varying network topology (fully connected, ring-cyclic, and no-edge) in our results. We assume that the model parameter is $x \in \mathbb{R}^2$, and we simulate IID data points (z^l, y^l) for each agent through

$$\delta^l \sim \mathcal{N}(0, \xi^2 I), \quad z^l \sim \mathcal{N}(0, I), \quad y^l = x^T z^l + \delta^l. \quad (26)$$

We assign $\xi = 4$. The prior on x is $\mathcal{N}(0, \lambda I)$ with $\lambda = 10$. Each agent is assigned n_i independent samples. The global posterior, from which we aim to sample, by a simple application of Bayes' rule, is of the form $\pi(x) \propto \exp - \sum_{i \in \mathcal{V}} f_i(x)$, where

$$f_i(x) = \frac{1}{2\xi^2} \sum_{l=1}^{n_i} \left(y_i^l - x^T z_i^l \right)^2 + \frac{1}{2\lambda N} \|x\|^2. \quad (27)$$

6.2.2 ALGORITHM

We do not perform hyper-parameter tuning and assume $\eta_{\text{DSGLD}} = 0.009$, $\eta_{\text{DSGHMC}} = 0.1$, $\gamma = 7$, as done by Gürbüzbalaban et al. (2021), while $\rho = 5$. For D-ULA, we follow the logistic regression example by Parayil et al. (2021) with modifications to avoid divergence. We assume $\alpha^{(k)} = 0.00082/(230 + k)^{\chi_2}$, $\zeta^{(k)} = 0.48/(230 + k)^{\chi_1}$. We assume $\chi_1 = \chi_2 = 0.05$ for the cyclic and no-edge topologies. For the case of the fully connected graph: when $N = 5, 20$ we assume $\chi_1 = 0.55$ and $\chi_2 = 0.05$. Note that the selection of parameters for D-ULA does not satisfy the conditions presented in the original paper (Parayil et al., 2021), but the selected parameter values perform well experimentally. The algorithm by Parayil et al. (2021) required the most testing to determine suitable parameters. Note that $x_i^{(0)} \sim \mathcal{N}(0, I)$, $\forall i \in \mathcal{V}$ for all algorithms. We also test two cases for n_i equal to 50 and 200. We tune the parameters independently for both values of n_i .

6.2.3 RESULTS

We present results with respect to the 2-Wasserstein distance between the empirical distribution of the iterates $x_i^{(k)}$ and the true posterior in Figures 1 and 2 for $n_i = 50$ and $n_i = 200$ respectively. The true posterior and the iterate distributions are Gaussian (Gürbüzbalaban et al., 2021). Furthermore, the true posterior is known in closed form (Gürbüzbalaban et al., 2021). The Gaussian distribution of each agent’s iterate is estimated by empirically computing the expectation and covariance through 100 independent trials. The same holds for the average agent iterate $\sum_{i \in \mathcal{V}} x_i^{(k)} / N$. The closed form of the 2-Wasserstein distance between Gaussians, in the case of non-singular covariances, only involves the covariances and means of the distributions (Givens and Shortt, 1984).

Figures 1 and 2 include the Wasserstein distance between the average iterate (avg) and the target distribution and the Wasserstein distance between the iterate of an agent (ag) and the target distribution. We observe that our proposed algorithm converges faster than the other presented schemes (D-SGLD, D-ULA, D-SGHMC) in the cases of sparsely connected network topology (ring-cyclic graph), while it can become slower for more connected graphs (fully connected). This agrees with results in the optimization literature, which show that C-ADMM can be slower as the communication topology becomes more connected (Bof et al., 2018). In the case of a highly connected graph, the dual variables take a longer time to converge because of the larger number of inter-agent disagreement terms involved, which slows convergence. Although a in Theorem 1 decreases with increasing connectedness, the theoretical upper bound need not decrease, because it includes terms that increase with increased connectivity. Therefore, the experimental behavior is not necessarily misaligned with the theoretical results.

We have also included the performance of ADMM (i.e., $w_i^{(k)} = 0$ in Algorithm 1) in Figures 1 and 2. ADMM is an optimization scheme and hence it drives $x_i^{(k)}$ to a point with an optimal value for the associated optimization problem, for any initialization of $x_i^{(k)}$. We observe that in the no-edge topology, ADMM performs exactly the same as D-ADMMS. This is attributed to the last term of the primal update, which vanishes in the case of a no-edge topology because no agent has any neighbors. The superior performance (in Wasserstein distance) of ADMM compared to the sampling schemes for the case $N = 100$ can be attributed to the very small ($\leq 10^{-3}$) entries of the true posterior covariance in this case, because of the large number of samples present. This means that simply finding the MAP (which is also the mean because of Gaussian structure) with ADMM is enough for a small Wasserstein distance to the true posterior distribution. For $N = 100$, the inability of CVXPY (Diamond and Boyd, 2016) to converge justifies the exclusion of the fully connected case.

The fast convergence (in cases) of D-ADMMS is intuitively attributed to the proximal primal update of the algorithm. D-ADMMS is able to take large steps in terms of Wasserstein distance in the initial iterations because its primal update consists of completely solving an optimization problem. Even in the case of no-edge topology, we observe the large reduction of Wasserstein distance in the early iterations. Nevertheless, because each agent does not gather information from other agents in the update of its dual variables, the trajectory in the primal space remains uncorrected in this case. This leads D-ADMMS to converge farther from the target distribution.

DISTRIBUTED MCMC SAMPLING VIA ADMM

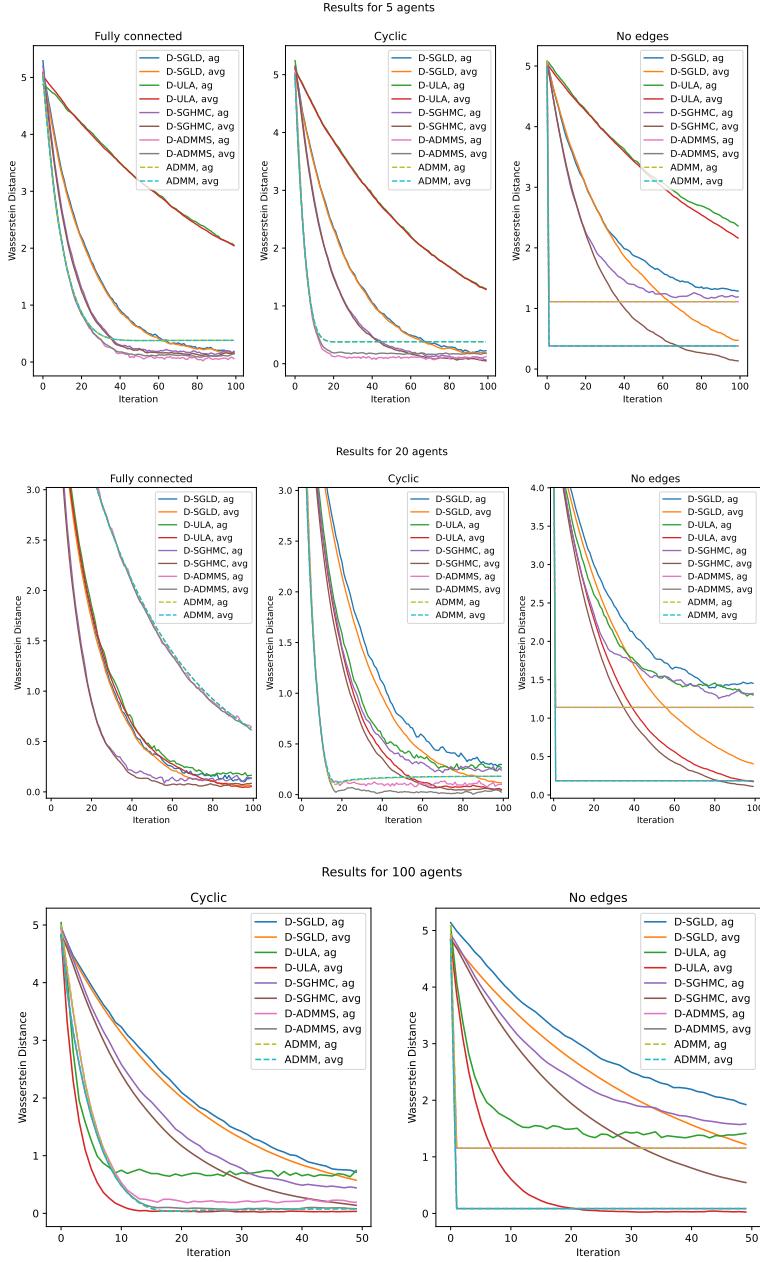


Figure 1: 2-Wasserstein distance to target distribution vs iteration for $n_i = 50$. Both the distance to the target distribution of the average iterate (avg) and a specific agent iterate (ag) are provided for each method. For the sparsely connected (cyclic) graph topology, our proposed algorithm (D-ADMMS) outperforms the baselines (D-SGLD, D-ULA, D-SGHMC) in terms of Wasserstein distance between the distribution of the agent iterate and the target distribution.

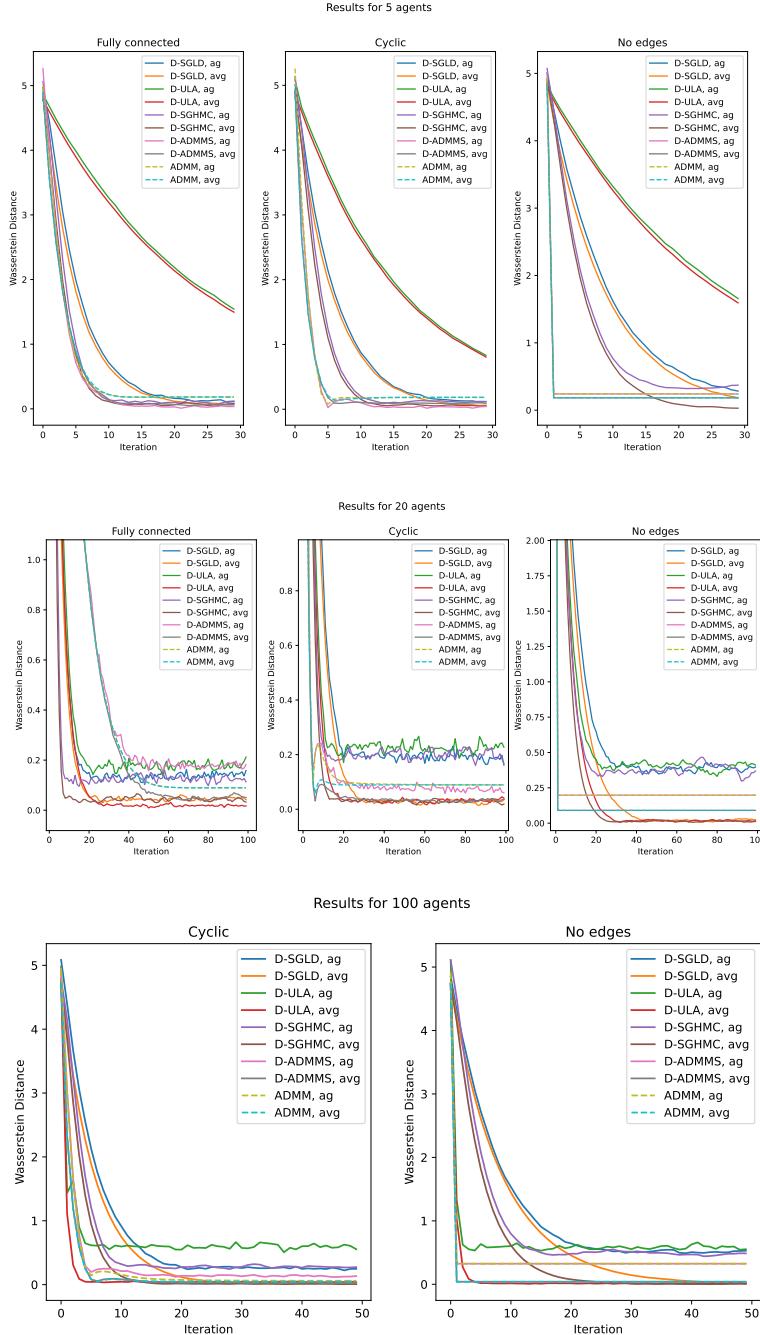


Figure 2: 2-Wasserstein distance to target distribution vs iteration for $n_i = 200$. Both the distance to the target distribution of the average iterate (avg) and a specific agent iterate (ag) are provided for each method. For the sparsely connected (cyclic) graph topology, our proposed algorithm (D-ADMMS) outperforms the baselines (D-SGLD, D-ULA, D-SGHMC) in terms of Wasserstein distance between the distribution of the agent iterate and the target distribution.

6.2.4 ABLATION

We study the robustness of the proposed scheme with respect to its hyper-parameter ρ . We note that D-ADMMS only requires setting the hyper-parameter ρ . In Figure 3, we assume a network of 20 agents with $n_i = 50$ and we compute the 2-Wasserstein distance between the empirical distribution of an agent's iterate, $x_i^{(k)}$, when D-ADMMS is deployed, and the true posterior. We deploy D-ADMMS with different values of ρ in order to study the algorithm's robustness to the choice of hyper-parameter. We use 100 trials in order to determine the iterate's empirical distribution.

Figure 4 shows the convergence in terms of 2-Wasserstein distance between an agent's iterate, when D-ADMMS is deployed, and the true posterior for a cyclic network of 20 agents with $n_i = 50$. In this case, however, we fix $\rho = 5$ and vary the initial probability distribution of the 100 sample iterates. We perform 10 experiments. For the first experiment, we assume that the initial distribution for each agent iterate is the standard normal ($x_i^{(0)} \sim \mathcal{N}(0, I)$, $\forall i \in \mathcal{V}$). In each remaining experiment q , the 100 sample iterates are initially drawn from the distribution $\mathcal{N}((-1, 2), \Sigma_q)$, where $\Sigma_q = A_q A_q^T$, and the entries of A_q are drawn from the uniform $\mathcal{U}(0, 10)$. Figure 5 demonstrates the evolution of samples in D-ADMMS for a cyclic topology of 5 agents with $n_i = 50$, $x_i(0) \sim \mathcal{N}(0, I)$, $\rho = 5$, and $x \in \mathbb{R}^1$.

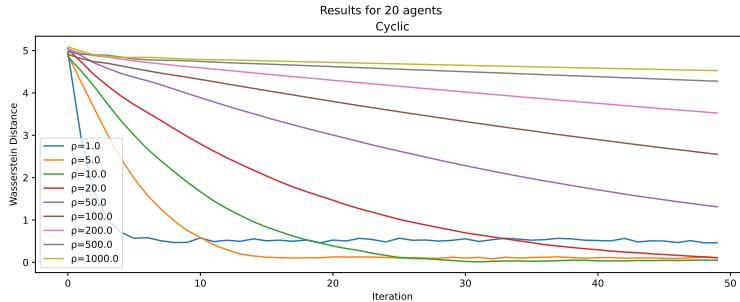


Figure 3: 2-Wasserstein distance of an agent's iterate to the target distribution for varying ρ in D-ADMMS.

6.3 Bayesian Logistic Regression

6.3.1 PROBLEM

We consider the distributed Bayesian logistic regression setting. We consider a varying number of agents ($N = 5, 20, 50$) and a varying network topology (fully connected, ring-cyclic, and no-edge) in our results. We first create a dataset of IID (z^l, y^l) pairs indexed by l , where y^l is a binary label (0 or 1). The likelihood function of a given sample for model parameters x is

$$\mathbb{P}(y = 1 | z; x) = (1 + \exp -x^T z)^{-1}. \quad (28)$$

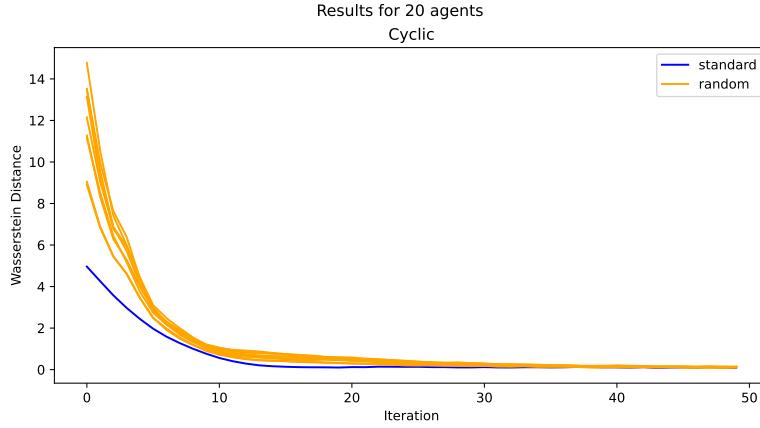


Figure 4: 2-Wasserstein distance of an agent's iterate to the target distribution for varying initial sample distribution in D-ADMMS. Standard refers to $x_i(0) \sim \mathcal{N}(0, I)$.

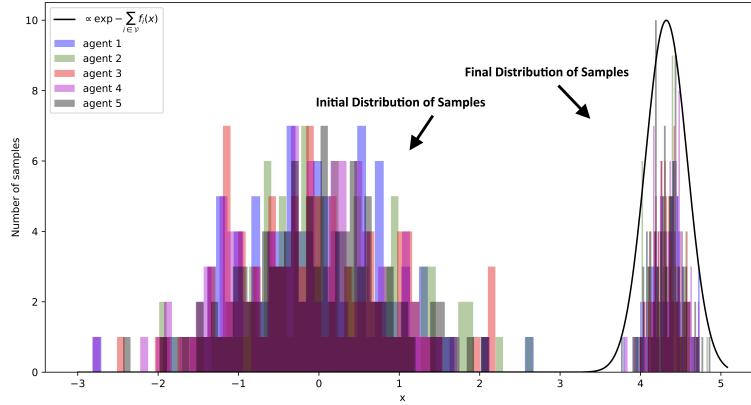


Figure 5: Evolution of the agents' sample distributions in D-ADMMS for a cyclic network of five agents. Each color corresponds to the samples of a different agent. We also include the true global posterior distribution up to a scaling factor.

We assume $x \in \mathbb{R}^3$. The prior distribution over the model parameters is $\mathcal{N}(0, \lambda I)$, where $\lambda = 10$. A data point is created as follows:

$$z^l \sim \mathcal{N}(0, 20I), \quad p^l \sim \mathcal{U}(0, 1), \quad (29)$$

while the label y^l is 1 if $p^l \leq (1 + \exp -x^T z^l)^{-1}$ and 0 otherwise. The parameter x is sampled from its prior distribution and $\mathcal{U}(0, 1)$ denotes the uniform distribution between 0 and 1. Assume that each agent i possesses n_i independent data points (z_i^l, y_i^l) , where the first \tilde{n}_i data points are those with label $y_i^l = 1$. Then the goal in Bayesian logistic regression

is to sample from the global posterior, which is proportional to $\exp - \sum_{i \in \mathcal{V}} f_i(x)$, where

$$f_i(x) = \sum_{l=1}^{n_i} \log \left(1 + \exp \psi_l x^T z_i^l \right) + \frac{\|x\|^2}{2\lambda N}, \quad (30)$$

and $\psi_l = -1$ if $1 \leq l \leq \tilde{n}_i$, while $\psi_l = 1$ otherwise.

6.3.2 ALGORITHM

We use $n_i = 50$, $\rho = 5$, $\eta_{\text{DSGLD}} = 0.0003$, $\eta_{\text{DSGHMC}} = 0.02$, and $\gamma = 30$ (Gürbüzbalaban et al., 2021). For D-ULA, we assume $\alpha^{(k)}$ and $\zeta^{(k)}$ follow the same equations as in Section 6.2. We assume $\chi_1 = \chi_2 = 0.05$ for the cyclic and no-edge topologies. For the case of the fully connected graph: when $N = 5, 20$ we assume $\chi_1 = 0.55, \chi_2 = 0.05$, while for $N = 50$ we set $\chi_1 = \chi_2 = 0.9$.

6.3.3 DISCUSSION

Because the posterior does not admit a Gaussian explicit formula, instead of 2-Wasserstein distance, we provide results for prediction accuracy. In Figure 6, we show the mean prediction accuracy on the total dataset, along with the ± 1 standard deviation interval, of each agent, based on its iterate $x_i^{(k)}$, as a function of the iteration. If $\mathbb{P}(y = 1 | z^l; x_i^{(k)}) \geq 0.5$, the agent assigns label 1 to the data-point. The mean and standard deviation are computed through 100 independent trials. We also include ADMM (i.e., $w_i^{(k)} = 0$ in Algorithm 1).

We observe that D-ADMMS is able to obtain the highest accuracy out of all the sampling methods presented, in a smaller number of iterations for the cyclic topology. Its performance is similar to that of ADMM. ADMM possesses superior performance because it converges to the MAP model parameter, which is expected to have the highest accuracy. In addition, the deviation from the mean accuracy is smaller for D-ADMMS in the cyclic topology. For the no-edge topology, as in Bayesian linear regression, ADMM is identical to D-ADMMS. In the fully connected case for N equal to 20 and 50 the decline in performance of D-ADMMS is similar to that found in the Bayesian linear regression task. Overall, D-ADMMS performs worse as the number of agents and the connectivity of the graph increase. This behavior matches the observed performance in the Bayesian linear regression task.

7 Conclusions and Future Work

We proposed a novel distributed sampling algorithm based on ADMM. By using proximal updates to generate samples in a distributed manner for a log-concave distribution, our approach outperforms existing gradient-based algorithms. We have shown convergence of the distribution of the iterates of our algorithm to the target distribution and have demonstrated practical advantages for our method in regression problems. Despite these promising results, limitations of the proposed algorithm include synchronous and lossless communication among the agents of the network.

Future directions include: i) analyzing the proposed scheme as the discretization of a stochastic differential equation to improve the convergence guarantees, ii) exploring connections between gradient flows and MCMC algorithms in distributed settings, iii) designing

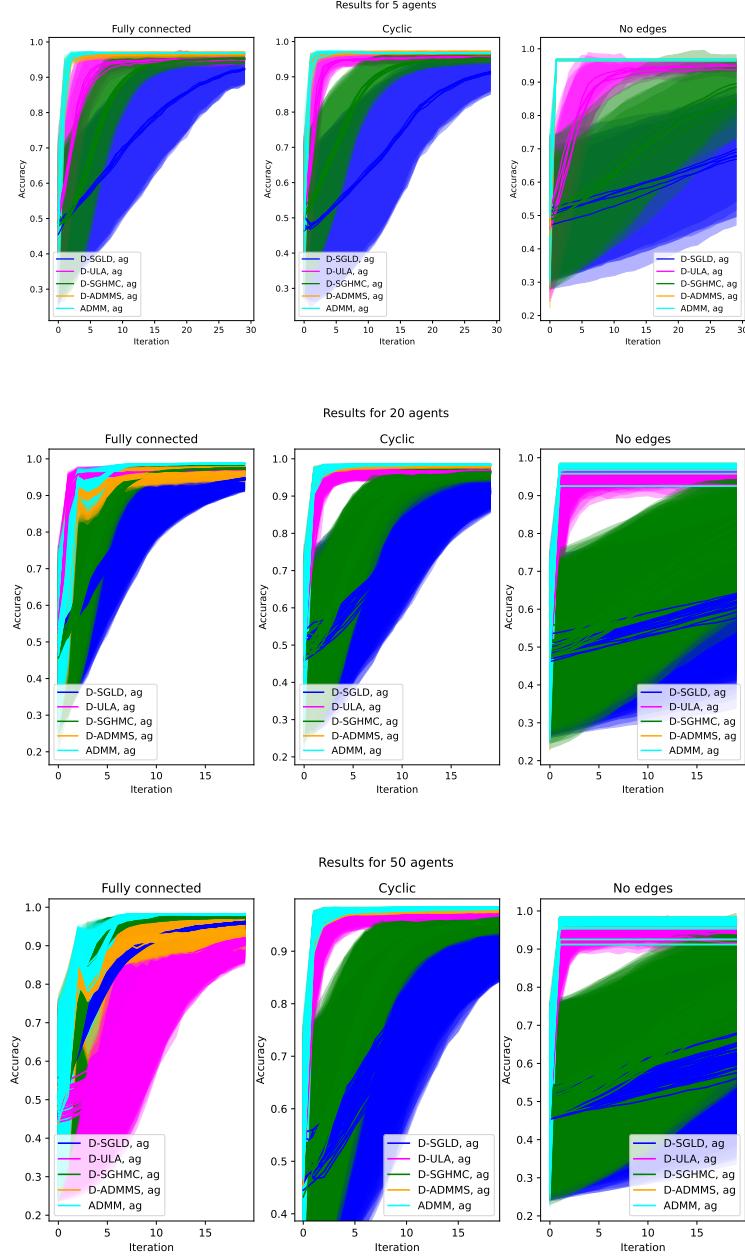


Figure 6: Prediction accuracy per iteration on complete dataset. We depict the accuracy per iteration on the complete dataset for each agent along with a 1-std interval over 100 independent trials for varying network topology. For the sparsely connected (cyclic) graph topology, our proposed algorithm (D-ADMMS) outperforms the baselines (D-SGLD, D-ULA, D-SGHMC) in terms of prediction accuracy.

distributed sampling algorithms based on accelerated first-order or higher-order optimization methods, iv) considering distributed sampling with constrained support.

Acknowledgments and Disclosure of Funding

The NASA University Leadership Initiative (grant # 80NSSC20M0163) provided funds to assist the first author with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity. For the first author, this work was also partially funded through the Alexander S. Onassis Foundation Scholarship program.

Appendix A. Proof of Lemma 1

We substitute

$$\frac{1}{2}M_+^T X^{(k)} - Z^{(k)} = 0 \quad (31)$$

into

$$\nabla f(X^{(k+1)}) + M_- \beta^{(k+1)} - \rho M_+ (Z^{(k)} - Z^{(k+1)}) + \sqrt{2} D w^{(k+1)} = 0, \quad (32)$$

$$\beta^{(k+1)} - \beta^{(k)} - \frac{\rho}{2} M_-^T X^{(k+1)} = 0, \quad (33)$$

to obtain

$$\nabla f(X^{(k+1)}) + M_- \beta^{(k+1)} - \frac{\rho}{2} M_+ M_+^T (X^{(k)} - X^{(k+1)}) + \sqrt{2} D w^{(k+1)} = 0, \quad (34)$$

$$\beta^{(k+1)} - \beta^{(k)} - \frac{\rho}{2} M_-^T X^{(k+1)} = 0, \quad (35)$$

We observe that eq. (34) depends on $M_- \beta^{(k+1)}$ rather than $\beta^{(k+1)}$. We thus multiply eq. (35) by M_- and obtain

$$M_- \beta^{(k+1)} - M_- \beta^{(k)} - \frac{\rho}{2} M_- M_-^T X^{(k+1)} = 0. \quad (36)$$

We substitute eq. (36) into eq. (34) and get

$$\nabla f(X^{(k+1)}) + M_- \beta^{(k)} - \frac{\rho}{2} M_+ M_+^T X^{(k)} + \left(\frac{\rho}{2} M_+ M_+^T + \frac{\rho}{2} M_- M_-^T \right) X^{(k+1)} + \sqrt{2} D w^{(k+1)} = 0. \quad (37)$$

We define $p^{(k)} = M_- \beta^{(k)}$ and from eq. (36, 37) we obtain the equivalent updates

$$\nabla f(X^{(k+1)}) + p^{(k)} - \frac{\rho}{2} M_+ M_+^T X^{(k)} + \left(\frac{\rho}{2} M_+ M_+^T + \frac{\rho}{2} M_- M_-^T \right) X^{(k+1)} + \sqrt{2} D w^{(k+1)} = 0, \quad (38)$$

$$p^{(k+1)} - p^{(k)} - \frac{\rho}{2} M_- M_-^T X^{(k+1)} = 0. \quad (39)$$

Notice that $L_+ = \frac{1}{2}M_+M_+^T$, $L_- = \frac{1}{2}M_-M_-^T$ and $D = \frac{1}{2}(L_+ + L_-)$. Hence eq. (38-39) can be written as

$$\nabla f(X^{(k+1)}) + p^{(k)} - \rho L_+ X^{(k)} + 2\rho D X^{(k+1)} + \sqrt{2}Dw^{(k+1)} = 0, \quad (40)$$

$$p^{(k+1)} - p^{(k)} - \rho L_- X^{(k+1)} = 0. \quad (41)$$

We have thus established that eq. (10-12) imply eq. (40-41). By a simple inspection, we can observe that eq. (40-41) are equivalent to the update equations of D-ADMMS in Algorithm 1, where $X^{(k)}$ is the concatenation of the $x_i^{(k)}$ from Algorithm 1.

Appendix B. Proof of Lemma 2

Let us start with the following auxiliary lemma.

Lemma 4 *Assume $x, y \in \mathbb{R}^n$. Then the inequality*

$$\|x + y\|^2 + (\kappa - 1)\|x\|^2 \geq \left(1 - \frac{1}{\kappa}\right)\|y\|^2 \quad (42)$$

holds for any $\kappa > 1$.

Proof By expanding the left-hand side (LHS) of (42), we obtain

$$\|x\|^2 + 2\langle x, y \rangle + \|y\|^2 + (\kappa - 1)\|x\|^2 \geq \left(1 - \frac{1}{\kappa}\right)\|y\|^2.$$

By bringing all terms to the LHS, we get

$$\kappa\|x\|^2 + 2\langle x, y \rangle + \frac{1}{\kappa}\|y\|^2 \geq 0.$$

By multiplying both sides by κ and noticing that $\kappa > 1 > 0$, we get

$$\|\kappa x\|^2 + 2\langle \kappa x, y \rangle + \|y\|^2 \geq 0,$$

which can be written as

$$\|\kappa x + y\|^2 \geq 0,$$

this concluding the proof of the lemma. ■

Our proof strategy leverages equations (10-12) and the KKT conditions in (13-15), which yield the relations:

$$\nabla f(X^{(k+1)}) - \nabla f(X^*) = \rho M_+ (Z^{(k)} - Z^{(k+1)}) - M_- (\beta^{(k+1)} - \beta^*) - \sqrt{2}Dw^{(k+1)}, \quad (43)$$

$$\beta^{(k+1)} - \beta^{(k)} = \frac{\rho}{2} M_-^T (X^{(k+1)} - X^*), \quad (44)$$

$$Z^{(k+1)} - Z^* = \frac{1}{2} M_+^T (X^{(k+1)} - X^*). \quad (45)$$

We now exploit the relations in eq. (43-45) to show inequalities that hold for the iterates generated by Algorithm 1. To this end, we split the analysis into five steps, which are presented in the sequel.

Step 1: A basic inequality

Since function $f(X)$ is strongly convex (under Assumption 1), we obtain

$$\begin{aligned}
 m_f \|X^{(k+1)} - X^*\|^2 &\leq \langle X^{(k+1)} - X^*, \nabla f(X^{(k+1)}) - \nabla f(X^*) \rangle = \\
 &\langle X^{(k+1)} - X^*, \rho M_+ (Z^{(k)} - Z^{(k+1)}) - M_- (\beta^{(k+1)} - \beta^*) - \sqrt{2}Dw^{(k+1)} \rangle = \\
 &\langle X^{(k+1)} - X^*, \rho M_+ (Z^{(k)} - Z^{(k+1)}) \rangle \\
 &+ \langle X^{(k+1)} - X^*, -M_- (\beta^{(k+1)} - \beta^*) \rangle + \langle X^{(k+1)} - X^*, -\sqrt{2}Dw^{(k+1)} \rangle = \\
 &\rho \langle M_+^T (X^{(k+1)} - X^*), Z^{(k)} - Z^{(k+1)} \rangle \\
 &- \langle M_-^T (X^{(k+1)} - X^*), \beta^{(k+1)} - \beta^* \rangle - \langle X^{(k+1)} - X^*, \sqrt{2}Dw^{(k+1)} \rangle. \quad (46)
 \end{aligned}$$

The first equality holds due to eq. (43), the second and third ones hold by simply expanding the terms and performing trivial algebraic manipulations. We substitute eq. (44-45) into the last equation of (46) to obtain

$$\begin{aligned}
 m_f \|X^{(k+1)} - X^*\|^2 &\leq \\
 2\rho \langle Z^{(k+1)} - Z^*, Z^{(k)} - Z^{(k+1)} \rangle + \frac{2}{\rho} \langle \beta^{(k)} - \beta^{(k+1)}, \beta^{(k+1)} - \beta^* \rangle - \langle X^{(k+1)} - X^*, \sqrt{2}Dw^{(k+1)} \rangle = \\
 2 \left(U^{(k)} - U^{(k+1)} \right)^T G \left(U^{(k+1)} - U^* \right) - \langle X^{(k+1)} - X^*, \sqrt{2}Dw^{(k+1)} \rangle, \quad (47)
 \end{aligned}$$

where we recall that $U^{(k)} = (Z^{(k)}, \beta^{(k)})$ and $G = \text{diag} \left\{ \rho I_{2|\mathcal{E}|d}, \frac{1}{\rho} I_{2|\mathcal{E}|d} \right\}$. Using the relation $\langle a - b, b - c \rangle_G = \|a - c\|_G^2 - \|a - b\|_G^2 - \|b - c\|_G^2$ for the first term on the right-hand side of eq. (47), with $a = U^{(k)}$, $b = U^{(k+1)}$, and $c = U^*$, we obtain

$$\begin{aligned}
 m_f \|X^{(k+1)} - X^*\|^2 &\leq \\
 \|U^{(k)} - U^*\|_G^2 - \|U^{(k+1)} - U^*\|_G^2 - \|U^{(k)} - U^{(k+1)}\|_G^2 - \langle X^{(k+1)} - X^*, \sqrt{2}Dw^{(k+1)} \rangle. \quad (48)
 \end{aligned}$$

We now upper bound the last term

$$\begin{aligned}
 - \langle X^{(k+1)} - X^*, \sqrt{2}Dw^{(k+1)} \rangle &\leq \|X^{(k+1)} - X^*\| \|\sqrt{2}Dw^{(k+1)}\| \\
 &\leq \frac{1}{2} \left(\|X^{(k+1)} - X^*\| + \|\sqrt{2}Dw^{(k+1)}\| \right)^2, \quad (49)
 \end{aligned}$$

where we apply the inequality $ab \leq \frac{1}{2}(a+b)^2$ in the last step. This yields

$$\begin{aligned}
 m_f \|X^{(k+1)} - X^*\|^2 &\leq \|U^{(k)} - U^*\|_G^2 \\
 &- \|U^{(k+1)} - U^*\|_G^2 - \|U^{(k)} - U^{(k+1)}\|_G^2 + \frac{1}{2} \left(\|X^{(k+1)} - X^*\| + \|\sqrt{2}Dw^{(k+1)}\| \right)^2. \quad (50)
 \end{aligned}$$

Eq. (50) is the basic inequality of our analysis. The next steps of the proof constitute in further developing such inequality.

Step 2: Dealing with the term $\|U^{(k)} - U^{(k+1)}\|_G^2 + m_f \|X^{(k+1)} - X^*\|^2$

Observe that

$$\begin{aligned} \|U^{(k)} - U^{(k+1)}\|_G^2 + m_f \|X^{(k+1)} - X^*\|^2 &= \\ \rho \|Z^{(k)} - Z^{(k+1)}\|^2 + \frac{1}{\rho} \|\beta^{(k)} - \beta^{(k+1)}\|^2 + m_f \|X^{(k+1)} - X^*\|^2. \end{aligned} \quad (51)$$

We now focus on obtaining a lower bound for the right-hand side (RHS) term in eq. (51), which will be done in two steps as described below.

STEP 2.1: AN INTERMEDIATE INEQUALITY

We show that the inequality

$$\begin{aligned} \frac{\rho \kappa \sigma_{\max}^2(M_+)}{(\kappa - 1) \sigma_{\min}^2(M_-)} \|Z^{(k+1)} - Z^{(k)}\|^2 + \frac{\kappa M_f^2}{\rho \sigma_{\min}^2(M_-)} \|X^{(k+1)} - X^*\|^2 &\geq \\ \frac{1}{\rho} \|\beta^{(k+1)} - \beta^*\|^2 - \frac{2\sqrt{2}}{\rho \sigma_{\min}^2(M_-)} \|M_- (\beta^{(k+1)} - \beta^*)\| \|Dw^{(k+1)}\| + \\ \frac{2}{\rho \sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\|^2, \end{aligned} \quad (52)$$

holds, for any $\kappa > 1$. The first step to obtain inequality (52) is to manipulate the relation in (43) by means of the inequality $\|a + b\|^2 + (\kappa - 1)\|a\|^2 \geq (1 - \frac{1}{\kappa})\|b\|^2$ (see Lemma 2 for a proof of this inequality), which holds for $\kappa > 1$. Indeed, setting $a = \nabla f(X^{(k+1)}) - \nabla f(X^*)$ and $b = M_- (\beta^{(k+1)} - \beta^*) + \sqrt{2}Dw^{(k+1)}$, we obtain

$$\begin{aligned} \left(1 - \frac{1}{\kappa}\right) \|M_- (\beta^{(k+1)} - \beta^*) + \sqrt{2}Dw^{(k+1)}\|^2 &\leq \\ \leq \|\rho M_+ (Z^{(k)} - Z^{(k+1)})\|^2 + (\kappa - 1) \|\nabla f(X^{(k+1)}) - \nabla f(X^*)\|^2 &\leq \\ \leq \rho^2 \sigma_{\max}^2(M_+) \|Z^{(k+1)} - Z^{(k)}\|^2 + (\kappa - 1) M_f^2 \|X^{(k+1)} - X^*\|^2, \end{aligned} \quad (53)$$

where the first inequality follows from $\|a + b\| = \|\rho M_+ (Z^{(k)} - Z^{(k+1)})\|$ due to (43), and the second inequality follows by Lipschitz continuity of $\nabla f(X)$ and the fact that the largest singular value of M_+ equals the induced 2-norm $\|M_+\| = \max_{x \neq 0} \frac{\|M_+ x\|}{\|x\|}$. By expanding the squares in the LHS of (53) and using the inequality $\langle a, b \rangle \geq -\|a\| \|b\|$, we have that

$$\begin{aligned} \left(1 - \frac{1}{\kappa}\right) \left(\|M_- (\beta^{(k+1)} - \beta^*)\|^2 - 2 \|M_- (\beta^{(k+1)} - \beta^*)\| \|\sqrt{2}Dw^{(k+1)}\| + \|\sqrt{2}Dw^{(k+1)}\|^2 \right) &\leq \\ \leq \rho^2 \sigma_{\max}^2(M_+) \|Z^{(k+1)} - Z^{(k)}\|^2 + (\kappa - 1) M_f^2 \|X^{(k+1)} - X^*\|^2. \end{aligned} \quad (54)$$

We now use the fact that $\|M_- (\beta^{(k+1)} - \beta^*)\|^2 \geq \sigma_{\min}^2(M_-) \|\beta^{(k+1)} - \beta^*\|^2$ (because both β^* and $\beta^{(k+1)}$ lie in the column space of M_-^T) and then multiply the resulting inequality by $\frac{\kappa}{\rho(\kappa-1)\sigma_{\min}^2(M_-)}$ to obtain eq. (52).

STEP 2.2: A TRIVIAL INEQUALITY

Notice that from eq. (45)

$$\|Z^{(k+1)} - Z^*\|^2 = \frac{1}{4} \|M_+^T (X^{(k+1)} - X^*)\|^2 \leq \frac{1}{4} \sigma_{\max}^2(M_+) \|X^{(k+1)} - X^*\|^2. \quad (55)$$

By simple multiplication of both sides by ρ , we obtain

$$\rho \|Z^{(k+1)} - Z^*\|^2 \leq \frac{\rho \sigma_{\max}^2(M_+)}{4} \|X^{(k+1)} - X^*\|^2. \quad (56)$$

STEP 2.3: COMBINING THE RESULTS FROM STEPS 2.1 AND 2.2 TO OBTAIN (52)

We add inequality (56) into (52) to obtain

$$\begin{aligned} & \frac{\rho \kappa \sigma_{\max}^2(M_+)}{(\kappa - 1) \sigma_{\min}^2(M_-)} \|Z^{(k+1)} - Z^{(k)}\|^2 + \left(\frac{\kappa M_f^2}{\rho \sigma_{\min}^2(M_-)} + \frac{\rho}{4} \sigma_{\max}^2(M_+) \right) \|X^{(k+1)} - X^*\|^2 \geq \\ & \rho \|Z^{(k+1)} - Z^*\|^2 + \frac{1}{\rho} \|\beta^{(k+1)} - \beta^*\|^2 - \frac{2\sqrt{2}}{\sigma_{\min}^2(M_-)} \|M_- (\beta^{(k+1)} - \beta^*)\| \|Dw^{(k+1)}\| + \\ & \frac{2}{\rho \sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\|^2. \end{aligned} \quad (57)$$

We now let

$$\delta = \min \left\{ \frac{(\kappa - 1) \sigma_{\min}^2(M_-)}{\kappa \sigma_{\max}^2(M_+)}, \frac{m_f}{\frac{\rho}{4} \sigma_{\max}^2(M_+) + \frac{\kappa M_f^2}{\rho \sigma_{\min}^2(M_-)}} \right\} > 0, \quad (58)$$

and notice that if $\delta = \frac{(\kappa - 1) \sigma_{\min}^2(M_-)}{\kappa \sigma_{\max}^2(M_+)}$, then $\delta \left(\frac{\kappa M_f^2}{\rho \sigma_{\min}^2(M_-)} + \frac{\rho}{4} \sigma_{\max}^2(M_+) \right) \leq m_f$. Similarly, if $\delta = \frac{m_f}{\frac{\rho}{4} \sigma_{\max}^2(M_+) + \frac{\kappa M_f^2}{\rho \sigma_{\min}^2(M_-)}}$, then $\delta \frac{\rho \kappa \sigma_{\max}^2(M_+)}{(\kappa - 1) \sigma_{\min}^2(M_-)} \leq \rho$. Therefore, by multiplying eq. (57) with δ , we obtain

$$\begin{aligned} & \rho \|Z^{(k+1)} - Z^{(k)}\|^2 + m_f \|X^{(k+1)} - X^*\|^2 \geq \\ & \rho \delta \|Z^{(k+1)} - Z^*\|^2 + \frac{\delta}{\rho} \|\beta^{(k+1)} - \beta^*\|^2 - \frac{2\sqrt{2}\delta}{\sigma_{\min}^2(M_-)} \|M_- (\beta^{(k+1)} - \beta^*)\| \|Dw^{(k+1)}\| + \\ & \frac{2\delta}{\rho \sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\|^2. \end{aligned} \quad (59)$$

We add the positive term $\frac{1}{\rho} \|\beta^{(k+1)} - \beta^{(k)}\|^2$ to the LHS of the last equation and apply the definition of $\|U^{(k+1)} - U^{(k)}\|_G^2$ to get

$$\begin{aligned} & \|U^{(k+1)} - U^{(k)}\|_G^2 + m_f \|X^{(k+1)} - X^*\|^2 \geq \\ & \delta \|U^{(k+1)} - U^*\|_G^2 - \frac{2\sqrt{2}\delta}{\sigma_{\min}^2(M_-)} \|M_- (\beta^{(k+1)} - \beta^*)\| \|Dw^{(k+1)}\| + \\ & \frac{2\delta}{\rho \sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\|^2. \end{aligned} \quad (60)$$

The last equation is a lower bound for the term $\|U^{(k)} - U^{(k+1)}\|_G^2 + m_f\|X^{(k+1)} - X^*\|^2$, thus concluding step 2 of the proof.

Step 3: Manipulating the lower bound (60) of $\|U^{(k)} - U^{(k+1)}\|_G^2 + m_f\|X^{(k+1)} - X^*\|^2$

We combine inequality (60) with eq. (50) and obtain

$$\begin{aligned} \|U^{(k+1)} - U^*\|_G^2 &\leq \frac{1}{1+\delta}\|U^{(k)} - U^*\|_G^2 + \frac{1}{2(1+\delta)}\left(\|X^{(k+1)} - X^*\| + \|\sqrt{2}Dw^{(k+1)}\|\right)^2 + \\ &\quad \frac{2\sqrt{2}\delta}{(1+\delta)\sigma_{\min}^2(M_-)}\|M_-\left(\beta^{(k+1)} - \beta^*\right)\|\|Dw^{(k+1)}\| - \frac{2\delta}{(1+\delta)\rho\sigma_{\min}^2(M_-)}\|Dw^{(k+1)}\|^2, \end{aligned} \quad (61)$$

by using $a \leq b \leq c \Rightarrow a \leq c$, where b stands for $\|U^{(k+1)} - U^{(k)}\|_G^2 + m_f\|X^{(k+1)} - X^*\|^2$, and some algebraic manipulations.

STEP 3.1: AN INTERMEDIATE TRICK

Eq. (48) also gives us the upper bound

$$\|X^{(k+1)} - X^*\|^2 \leq \frac{1}{m_f}\|U^{(k)} - U^*\|_G^2 + \frac{1}{m_f}\langle X^{(k+1)} - X^*, -\sqrt{2}Dw^{(k+1)} \rangle, \quad (62)$$

which, by completing the square, is equivalently written as

$$\|X^{(k+1)} - X^* + \frac{1}{\sqrt{2}m_f}Dw^{(k+1)}\|^2 \leq \frac{1}{m_f}\|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2}\|Dw^{(k+1)}\|^2. \quad (63)$$

We now work to reform the second and third terms on the (RHS) of eq. (61).

STEP 3.2: MANIPULATING THE THIRD TERM ON THE RHS OF EQ. (61)

We first manipulate the third term of eq. (61). From eq. (44) we have that

$$\beta^{(k+1)} - \beta^* = \beta^{(k)} - \beta^* + \frac{\rho}{2}M_-^T(X^{(k+1)} - X^*). \quad (64)$$

which gives

$$\begin{aligned}
 \|M_-(\beta^{(k+1)} - \beta^*)\| &\leq \sigma_{\max}(M_-) \|\beta^{(k+1)} - \beta^*\| \\
 &\leq \sigma_{\max}(M_-) \left(\|\beta^{(k)} - \beta^*\| + \frac{\rho \sigma_{\max}(M_-)}{2} \|X^{(k+1)} - X^*\| \right) \\
 &\leq \sigma_{\max}(M_-) \cdot \\
 &\quad \left(\|\beta^{(k)} - \beta^*\| + \frac{\rho \sigma_{\max}(M_-)}{2} \left(\|X^{(k+1)} - X^* + \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}\| + \left\| \frac{1}{\sqrt{2m_f}} Dw^{(k+1)} \right\| \right) \right) \\
 &\leq \sigma_{\max}(M_-) \cdot \\
 &\quad \left(\|\beta^{(k)} - \beta^*\| + \frac{\rho \sigma_{\max}(M_-)}{2} \left(\sqrt{\frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2} + \left\| \frac{1}{\sqrt{2m_f}} Dw^{(k+1)} \right\| \right) \right) \\
 &\leq \sigma_{\max}(M_-) \sqrt{\rho} \sqrt{\|U^{(k)} - U^*\|_G^2} + \\
 &\quad \frac{\rho \sigma_{\max}^2(M_-)}{2} \left(\sqrt{\frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2} + \left\| \frac{1}{\sqrt{2m_f}} Dw^{(k+1)} \right\| \right). \quad (65)
 \end{aligned}$$

The first step in the reasoning above applies the Euclidian norm and uses the properties of the singular values of a matrix. The second step uses the triangle inequality property of the Euclidian norm, eq. (64), and the properties of the singular values of a matrix. The third step uses the triangle inequality as well. The fourth step uses eq. (63). Finally, the fifth step uses the fact that

$$\|\beta^{(k)} - \beta^*\| = \sqrt{\rho} \sqrt{\frac{1}{\rho} \|\beta^{(k)} - \beta^*\|^2} \leq \sqrt{\rho} \sqrt{\frac{1}{\rho} \|\beta^{(k)} - \beta^*\|^2 + \rho \|Z^{(k)} - Z^*\|^2}, \quad (66)$$

and the definition of $\|U^{(k)} - U^*\|_G^2$.

STEP 3.3: MANIPULATING THE SECOND TERM ON THE RHS OF EQ. (61)

For the second term of eq. (61), we get

$$\begin{aligned}
 &\left(\|X^{(k+1)} - X^*\| + \|\sqrt{2} Dw^{(k+1)}\| \right)^2 \leq \\
 &\quad \left(\|X^{(k+1)} - X^* + \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}\| + \underbrace{\left\| \frac{1}{\sqrt{2m_f}} Dw^{(k+1)} \right\|}_{\bar{w}^{(k+1)} \geq 0} + \|\sqrt{2} Dw^{(k+1)}\| \right)^2 \leq \\
 &\quad \frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + 2\bar{w}^{(k+1)} \|X^{(k+1)} - X^* + \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}\| + (\bar{w}^{(k+1)})^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2 \leq \\
 &\quad \frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + 2\bar{w}^{(k+1)} \sqrt{\frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2} + (\bar{w}^{(k+1)})^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2,
 \end{aligned} \quad (67)$$

by using the triangle inequality of the Euclidian norm, and then simply developing the square and using eq. (63).

Step 4: Simplifying the recursive relationship (61) for $\|U^{(k+1)} - U^*\|_G^2$

By replacing the third and second term on the RHS of eq. (61), with their upper bounds, eq. (65) and eq. (67) respectively, and by combining like terms, we obtain

$$\begin{aligned} \|U^{(k+1)} - U^*\|_G^2 &\leq \underbrace{\frac{2m_f + 1}{2m_f(1 + \delta)}}_a \|U^{(k)} - U^*\|_G^2 + \\ &\quad \underbrace{\frac{1}{2(1 + \delta)} \left(2\bar{w}^{(k+1)} \sqrt{\frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2} + (\bar{w}^{(k+1)})^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2 \right)}_b + \\ &\quad \underbrace{\frac{2\sqrt{2}\delta}{(1 + \delta)\sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\| \left(\sigma_{\max}(M_-) \sqrt{\rho} \sqrt{\|U^{(k)} - U^*\|_G^2} + \right)}_c + \\ &\quad \underbrace{\frac{\rho\sigma_{\max}^2(M_-)}{2} \left(\sqrt{\frac{1}{m_f} \|U^{(k)} - U^*\|_G^2 + \frac{1}{2m_f^2} \|Dw^{(k+1)}\|^2} + \left\| \frac{1}{\sqrt{2}m_f} Dw^{(k+1)} \right\| \right)}_d \\ &\quad - \underbrace{\frac{2\delta}{(1 + \delta)\rho\sigma_{\min}^2(M_-)} \|Dw^{(k+1)}\|^2}_e. \end{aligned} \quad (68)$$

We will now use the fact $\sqrt{a + b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$. Eq. (68) then gives us

$$\begin{aligned} \|U^{(k+1)} - U^*\|_G^2 &\leq a \|U^{(k)} - U^*\|_G^2 + 2 \frac{b}{\sqrt{m_f}} \bar{w}^{(k+1)} \|U^{(k)} - U^*\|_G + 2 \frac{b}{\sqrt{2}m_f} \bar{w}^{(k+1)} \|Dw^{(k+1)}\| \\ &\quad + b (\bar{w}^{(k+1)})^2 + \frac{b}{2m_f^2} \|Dw^{(k+1)}\|^2 + c\sigma_{\max}(M_-) \sqrt{\rho} \|Dw^{(k+1)}\| \|U^{(k)} - U^*\|_G \\ &\quad + \frac{cd}{\sqrt{m_f}} \|Dw^{(k+1)}\| \|U^{(k)} - U^*\|_G + \frac{cd}{\sqrt{2}m_f} \|Dw^{(k+1)}\|^2 + \frac{cd}{\sqrt{2m_f}} \|Dw^{(k+1)}\|^2 - e \|Dw^{(k+1)}\|^2 \\ &= a \|U^{(k)} - U^*\|_G^2 + \\ &\quad \underbrace{\left(2 \frac{b}{\sqrt{m_f}} \bar{w}^{(k+1)} + c\sigma_{\max}(M_-) \sqrt{\rho} \|Dw^{(k+1)}\| + \frac{cd}{\sqrt{m_f}} \|Dw^{(k+1)}\| \right)}_{y^{(k+1)}} \|U^{(k)} - U^*\|_G \\ &\quad + \underbrace{\left(\frac{\sqrt{2}b}{m_f} \bar{w}^{(k+1)} \|Dw^{(k+1)}\| + b (\bar{w}^{(k+1)})^2 + \frac{b}{2m_f^2} \|Dw^{(k+1)}\|^2 + \frac{\sqrt{2}cd}{m_f} \|Dw^{(k+1)}\|^2 - e \|Dw^{(k+1)}\|^2 \right)}_{r^{(k+1)}}. \end{aligned} \quad (69)$$

By simple substitution of the labeled quantities, we have

$$\|U^{(k+1)} - U^*\|_G^2 \leq a\|U^{(k)} - U^*\|_G^2 + y^{(k+1)}\sqrt{\|U^{(k)} - U^*\|_G^2} + r^{(k+1)}. \quad (70)$$

An important fact is that $y^{(k+1)}, r^{(k+1)}$ only depend on the noise at iteration $(k+1)$, i.e., these terms are a function of $w^{(k+1)}$.

Step 5: Obtaining bounds for the Wasserstein distances

We now choose the coupling² between the marginal probability distribution of $Z^{(k)}$ and Z^* (which is seen as a random variable with point mass at the value Z^*) to be that for which their normed distance squared is minimized. We do the same for the coupling between $\beta^{(k+1)}$ and β^* (which is also seen as a random variable with point mass at value β^*). Using Jensen's inequality for concave functions and the independence between $w^{(k+1)}$ and $U^{(k)}$, we get

$$\begin{aligned} W_G^2(\mu_{U^{(k+1)}}, \mu_{U^*}) &\leq \|U^{(k+1)} - U^*\|_G^2 \leq aW_G^2(\mu_{U^{(k)}}, \mu_{U^*}) + \\ &\quad \mathbb{E}(y^{(k+1)})\sqrt{W_G^2(\mu_{U^{(k)}}, \mu_{U^*})} + \mathbb{E}(r^{(k+1)}) \\ &\leq \left(\sqrt{a}W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2 + \mathbb{E}(r^{(k+1)}) - \left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2. \end{aligned} \quad (71)$$

We can now bound the Wasserstein distance, by using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$,

$$W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) \leq \sqrt{a}W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \underbrace{\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}} + \sqrt{|\mathbb{E}(r^{(k+1)}) - \left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2|}}_{h^{(k+1)}}. \quad (72)$$

From eq. (63) and by applying the same reasoning as before, we obtain

$$W^2(\mu_{X^{(k+1)}}, \mu_{X^* - \frac{1}{\sqrt{2m_f}}Dw^{(k+1)}}) \leq \frac{1}{m_f}W_G^2(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{1}{2m_f^2}\mathbb{E}(\|Dw^{(k+1)}\|^2). \quad (73)$$

By using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$, we finally get

$$W(\mu_{X^{(k+1)}}, \mu_{X^* - \frac{1}{\sqrt{2m_f}}Dw^{(k+1)}}) \leq \frac{1}{\sqrt{m_f}}W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{1}{\sqrt{2m_f}}\sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)}. \quad (74)$$

The triangle inequality of the 2-Wasserstein distance yields

$$W(\mu_{X^{(k+1)}}, \mu^*) \leq W(\mu_{X^{(k+1)}}, \mu_{X^* - \frac{1}{\sqrt{2m_f}}Dw^{(k+1)}}) + W(\mu_{X^* - \frac{1}{\sqrt{2m_f}}Dw^{(k+1)}}, \mu^*), \quad (75)$$

2. Assume that x and y are two random variables with marginal distributions $p(x)$ and $p(y)$ respectively. Then the coupling between x and y is given by the joint distribution $p(x, y)$, such that $p(x) = \sum_y p(x, y)$ and $p(y) = \sum_x p(x, y)$. The coupling is determined by the conditional distribution $p(x | y)$ such that $p(x, y) = p(x | y)p(y)$.

and by eq. (71,74), we obtain the final bound

$$\begin{aligned} W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*) &\leq \frac{1}{\sqrt{m_f}} W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{1}{\sqrt{2m_f}} \sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)} + \\ &\quad W(\mu_{X^* - \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}}, \boldsymbol{\mu}^*), \end{aligned} \quad (76)$$

where the last two terms in the RHS are constants.

Appendix C. Proof of Theorem 3

The proof of Theorem 3 is based on two steps: i) we telescopically expand the inequality of eq. (17), and ii) we find sufficient conditions for the telescopic sum to be decreasing with increasing iteration number.

Step 1: Telescopically expanding eq. (13)

We start from eq. (17) of the main body, which is also given below for convenience,

$$W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) \leq \sqrt{a} W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}} + \sqrt{|\mathbb{E}(r^{(k+1)}) - \left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2|}. \quad (77)$$

We first manipulate the last term on the RHS. By the triangle inequality of the absolute value, we have that

$$\sqrt{|\mathbb{E}(r^{(k+1)}) - \left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2|} \leq \sqrt{|\mathbb{E}(r^{(k+1)})|} + \left|\left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2\right|. \quad (78)$$

By the property $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$, we further have

$$\sqrt{|\mathbb{E}(r^{(k+1)})|} + \left|\left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2\right| \leq \sqrt{|\mathbb{E}(r^{(k+1)})|} + \sqrt{|\left(\frac{\mathbb{E}(y^{(k+1)})}{2\sqrt{a}}\right)^2|}. \quad (79)$$

Therefore eq. (17) can be written as

$$W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) \leq \sqrt{a} W_G(\mu_{U^{(k)}}, \mu_{U^*}) + \frac{\mathbb{E}(y^{(k+1)})}{\sqrt{a}} + \sqrt{|\mathbb{E}(r^{(k+1)})|}, \quad (80)$$

since $y^{(k+1)} \geq 0$. We recursively apply the inequality above to obtain

$$\begin{aligned} W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) &\leq (\sqrt{a})^{k+1} W_G(\mu_{U^0}, \mu_{U^*}) + \\ &\quad \sum_{l=1}^{k+1} (\sqrt{a})^{k-l} \mathbb{E}(y^{(l)}) + \sum_{l=1}^{k+1} (\sqrt{a})^{k+1-l} \sqrt{|\mathbb{E}(r^{(l)})|}. \end{aligned} \quad (81)$$

From eq. (18-21), we note that for given a, b, c, d , and e , the terms $\mathbb{E}(y^{(l)})$ and $\mathbb{E}(r^{(l)})$ are bounded, as they only depend on the noise at iteration l and the Euclidian norm, as well as the square of the Euclidian norm, of a Gaussian random variable are bounded. Assume the terms $\mathbb{E}(y^{(l)})$ and $\mathbb{E}(r^{(l)})$ are bounded by $Y \geq 0$ and $R \geq 0$ respectively. Then, eq. (81) becomes

$$W_G(\mu_{U^{(k+1)}}, \mu_{U^*}) \leq (\sqrt{a})^{k+1} W_G(\mu_{U^0}, \mu_{U^*}) + \sum_{l=1}^{k+1} (\sqrt{a})^{k-l} Y + \sum_{l=1}^{k+1} (\sqrt{a})^{k+1-l} \sqrt{R}. \quad (82)$$

Combining eq. (82) with eq. (16) in the main body, we obtain

$$\begin{aligned} W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*) &\leq \frac{1}{\sqrt{m_f}} (\sqrt{a})^k W_G(\mu_{U^0}, \mu_{U^*}) + \\ &\quad \frac{1}{\sqrt{m_f}} \sum_{l=1}^k (\sqrt{a})^{k-l} Y + \frac{1}{\sqrt{m_f}} \sum_{l=1}^k (\sqrt{a})^{k-l} \sqrt{R} + \\ &\quad \frac{1}{\sqrt{2m_f}} \sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)} + W(\mu_{X^* - \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}}, \boldsymbol{\mu}^*), \end{aligned} \quad (83)$$

where the last two terms on the RHS are constants. Assuming that $a < 1$, then, since $a > 0$, we have $\sum_{l=0}^{\infty} a^k = \frac{1}{1-a}$, which leads to the inequality

$$\begin{aligned} W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*) &\leq \frac{1}{\sqrt{m_f}} (\sqrt{a})^k W_G(\mu_{U^0}, \mu_{U^*}) + \\ &\quad \frac{1}{\sqrt{am_f}} \frac{Y}{1-\sqrt{a}} + \frac{1}{\sqrt{m_f}} \frac{\sqrt{R}}{1-\sqrt{a}} + \\ &\quad \frac{1}{\sqrt{2m_f}} \sqrt{\mathbb{E}(\|Dw^{(k+1)}\|^2)} + W(\mu_{X^* - \frac{1}{\sqrt{2m_f}} Dw^{(k+1)}}, \boldsymbol{\mu}^*). \end{aligned} \quad (84)$$

From the last equation, we observe that if $a < 1$, the upper bound for the Wasserstein distance $W(\mu_{X^{(k+1)}}, \boldsymbol{\mu}^*)$ decreases as the iteration number increases. This is because the first term on the RHS of eq. (84) decreases as k increases, while the remaining terms on the RHS of eq. (84) are constants for all iterations $k \geq 0$.

Step 2: Finding sufficient conditions for $a < 1$

We start with the definition of δ from eq. (20) of the main body. We observe that δ is a function of both κ and ρ , given by

$$\delta(\kappa, \rho) = \min \left\{ \frac{(\kappa - 1)\sigma_{\min}^2(M_-)}{\kappa\sigma_{\max}^2(M_+)}, \frac{m_f}{\frac{\rho}{4}\sigma_{\max}^2(M_+) + \frac{\kappa M_f^2}{\rho\sigma_{\min}^2(M_-)}} \right\}. \quad (85)$$

We observe that only the second argument in the definition of $\delta(\kappa, \rho)$ depends on ρ . Assuming a $\kappa > 1$ is selected, then the value

$$\rho(\kappa) = \frac{2\kappa^{\frac{1}{2}}M_f}{\sigma_{\min}(M_-)\sigma_{\max}(M_+)} \quad (86)$$

maximizes the second term in the min of eq. (85) and therefore δ for the given κ . In other words, $\rho(\kappa)$ from eq. (86) maximizes $\delta(\kappa, \rho)$ for the given κ . The maximum δ as a function of κ , termed $\delta_{\max}(\kappa)$, is hence given by

$$\delta_{\max}(\kappa) = \min \left\{ \frac{(\kappa - 1)\sigma_{\min}^2(M_-)}{\kappa\sigma_{\max}^2(M_+)}, \frac{m_f\sigma_{\min}(M_-)}{\kappa^{\frac{1}{2}}M_f\sigma_{\max}(M_+)} \right\}. \quad (87)$$

In order to find the maximum $\delta(\kappa, \rho)$ we need to maximize $\delta_{\max}(\kappa)$ in eq. (87) with respect to κ . We observe that the first term in eq. (87) is monotonically increasing as a function κ , while the second term in eq. (87) is monotonically decreasing as a function κ . Therefore, to obtain the maximum δ , we choose κ such that the two terms are equal. Such a κ exists and comes out to be

$$\kappa = 1 + \frac{1}{2} \sqrt{4 \frac{\tau_G^2}{\tau_f^2} + \frac{\tau_G^4}{\tau_f^4} + \frac{\tau_G^2}{2\tau_f^2}} > 1, \quad (88)$$

where

$$\tau_G = \frac{\sigma_{\max}(M_+)}{\sigma_{\min}(M_-)}, \quad \tau_f = \frac{M_f}{m_f}. \quad (89)$$

By plugging in κ from eq. (88) to eq. (87), we get the maximum possible value of $\delta(\kappa, \rho)$ to be

$$\delta_{\max} = \frac{1}{2\tau_f} \sqrt{\frac{1}{\tau_f^2} + \frac{4}{\tau_G^2}} - \frac{1}{2\tau_f^2}. \quad (90)$$

We turn our attention to the definition of a in eq. (21). $a < 1$ if and only if $2m_f\delta > 1$. Therefore, for convergence we need

$$\delta_{\max} > \frac{1}{2m_f}. \quad (91)$$

A sufficient condition for convergence is thus the following

$$\frac{m_f}{M_f} \sqrt{\frac{m_f^2}{M_f^2} + \frac{4\sigma_{\min}^2(M_-)}{\sigma_{\max}^2(M_+)} - \frac{m_f^2}{M_f^2}} > \frac{1}{m_f}. \quad (92)$$

The last equation can equivalently be written as

$$\tau_f^{-1} \sqrt{\tau_f^{-2} + 4\tau_G^{-2}} - \tau_f^{-2} > m_f^{-1}. \quad (93)$$

References

- Sungjin Ahn, Babak Shahbaba, and Max Welling. Distributed Stochastic Gradient MCMC. In *International Conference on Machine Learning*, 2014.
- Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated Learning via Posterior Averaging: A New Perspective and Practical Algorithms. In *International Conference on Learning Representations*, 2020.

- Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43, 2003.
- Heinz Bauschke and Patrick Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.
- Dimitri Bertsekas and John Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 2015.
- Kinjal Bhar, He Bai, Jemin George, and Carl Busart. Distributed Event-Triggered Unadjusted Langevin Algorithm for Bayesian Learning. *Automatica*, 156:111221, 2023.
- David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- Nicoletta Bo, Ruggero Carli, and Luca Schenato. Is ADMM Always Faster than Average Consensus? *Automatica*, 91:311–315, 2018.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, volume 3. Now Publishers, Inc., 2011.
- Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- Dragoš Cvetković, Peter Rowlinson, and Slobodan K Simić. Signless Laplacians of Finite Graphs. *Linear Algebra and its Applications*, 423(1):155–171, 2007.
- George Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.
- Wei Deng, Qian Zhang, Yi-An Ma, Zhao Song, and Guang Lin. On Convergence of Federated Averaging Langevin Dynamics. *arXiv preprint arXiv:2112.05120*, 2022.
- Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo: When Langevin Meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.
- Clark R Givens and Rae Michael Shortt. A Class of Wasserstein Metrics for Probability Distributions. *Michigan Mathematical Journal*, 31(2):231–240, 1984.
- Mert Gürbüzbalaban, Xuefeng Gao, Yuanhan Hu, and Lingjiong Zhu. Decentralized Stochastic Gradient Langevin Dynamics and Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 22(1):10804–10872, 2021.
- Avetik Karagulyan and Peter Richtárik. ELF: Federated Langevin Algorithms with Primal, Dual and Bidirectional Compression. *arXiv preprint arXiv:2303.04622*, 2023.

- Vyacheslav Kungurtsev, Adam Cobb, Tara Javidi, and Brian Jalaian. Decentralized Bayesian Learning with Metropolis-Adjusted Hamiltonian Monte Carlo. *Machine Learning*, pages 1–29, 2023.
- Anusha Lalitha, Xinghan Wang, Osman Kilinc, Yongxi Lu, Tara Javidi, and Farnaz Koushanfar. Decentralized Bayesian Learning over Graphs. *arXiv preprint arXiv:1905.10466*, 2019.
- Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*, 2019.
- Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B Giannakis. Distributed Sparse Linear Regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, 2010.
- Angelia Nedic and Asuman Ozdaglar. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- Willie Neiswanger, Chong Wang, and Eric P Xing. Asymptotically Exact, Embarrassingly Parallel MCMC. In *Conference on Uncertainty in Artificial Intelligence*, pages 623–632, 2014.
- Anjaly Parayil, He Bai, Jemin George, and Prudhvi Gurram. A Decentralized Approach to Bayesian Learning. *arXiv preprint arXiv:2007.06799*, 2021.
- Neal Parikh and Stephen Boyd. *Proximal Algorithms*, volume 1. Now Publishers, Inc., 2014.
- Marcelo Pereyra. Proximal Markov Chain Monte Carlo Algorithms. *Statistics and Computing*, 26:745–760, 2016.
- Lewis J Rendell, Adam M Johansen, Anthony Lee, and Nick Whiteley. Global Consensus Monte Carlo. *Journal of Computational and Graphical Statistics*, 30(2):249–259, 2020.
- Ernest K Ryu and Wotao Yin. *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators*. Cambridge University Press, 2022.
- Adil Salim, Dmitry Kovalev, and Peter Richtárik. Stochastic Proximal Langevin Algorithm: Potential Splitting and Nonasymptotic Rates. *Advances in Neural Information Processing Systems*, 32, 2019.
- Inass Sekkat. *Large Scale Bayesian Inference*. PhD thesis, École des Ponts ParisTech, 2022.
- Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.
- Ola Shorinwa, Trevor Halsted, Javier Yu, and Mac Schwager. Distributed Optimization Methods for Multi-Robot Systems: Part I – A Tutorial. *arXiv preprint arXiv:2301.11313*, 2023a.

- Ola Shorinwa, Trevor Halsted, Javier Yu, and Mac Schwager. Distributed Optimization Methods for Multi-Robot Systems: Part II – A Survey. *arXiv preprint arXiv:2301.11361*, 2023b.
- Panos Toulis, Thibaut Horel, and Edoardo M Airoldi. The Proximal Robbins–Monro Method. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(1):188–212, 2021.
- Cédric Villani and Cédric Villani. The Wasserstein Distances. *Optimal Transport: Old and New*, pages 93–111, 2009.
- Maxime Vono, Nicolas Dobigeon, and Pierre Chainais. Split-and-Augmented Gibbs Sampler—Application to Large-Scale Inference Problems. *IEEE Transactions on Signal Processing*, 67(6):1648–1661, 2019.
- Maxime Vono, Daniel Paulin, and Arnaud Doucet. Efficient MCMC Sampling with Dimension-Free Convergence Rate using ADMM-type Splitting. *Journal of Machine Learning Research*, 23(1):1100–1168, 2022.
- Max Welling and Yee W Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *International Conference on Machine Learning*, pages 681–688, 2011.
- Javier Yu, Joseph A Vincent, and Mac Schwager. DINNO: Distributed Neural Network Optimization for Multi-Robot Collaborative Learning. *IEEE Robotics and Automation Letters*, 7(2):1896–1903, 2022.
- Xiang Zhou, Huizhuo Yuan, Chris Junchi Li, and Qingyun Sun. Stochastic Modified Equations for Continuous Limit of Stochastic ADMM. *arXiv preprint arXiv:2003.03532*, 2020.