2025 IEEE 64th Conference on Decision and Control (CDC)
December 10-12, 2025. Rio de Janeiro, Brazil

Jointly Computation- and Communication-Efficient Distributed Learning

Xiaoxing Ren[1], Nicola Bastianello[2⋆], Karl H. Johansson[2], Thomas Parisini[3,4,5]

2025 IEEE 64th Conference on Decision and Control (CDC) | 979-8-3315-2627-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CDC57313.2025.11312358

***Abstract*— We address distributed learning problems over undirected networks. Specifically, we focus on designing a novel ADMM-based algorithm that is jointly computation- and communication-efficient. Our design guarantees computational efficiency by allowing agents to use stochastic gradients during local training. Moreover, communication efficiency is achieved as follows: i) the agents perform multiple training epochs between communication rounds, and ii) compressed transmissions are used. We prove *exact* linear convergence of the algorithm in the strongly convex setting. We corroborate our theoretical results by numerical comparisons with state of the art techniques on a classification task.**

I. Introduction

Smart devices equipped with computational and communications resources underwent, in recent years, a widespread adoption in many applications, including power grids, robotics, traffic and sensor networks [1], [2]. These devices then become the components of multi-agent systems that can collect data and cooperatively achieve learning tasks. However, their resources (CPU and communications) might be limited. Thus, in this paper we focus on the design of a *distributed learning algorithm that is jointly computation- and communication-efficient*.

Different techniques have been explored to design *computation-efficient* algorithms. The main solution is the use of stochastic gradients, which allow the agents to update their models using only a subset of their local data [3], [4]. However, stochastic gradients might cause inexact convergence, and *variance reduction* was proposed to solve this issue, see *e.g.*, [5], [6].

In terms of *communication-efficiency*, the main methods adopted in machine learning are *compression* and *local training*. Compression allows to reduce the size of communications exchanged by the agents, for example by transmitting only the most significant components of a model. Different distributed algorithms were designed to use compressed communications (see, for instance [7], [8], [9], [10]. Of these, [7], [8] also implement an *error feedback* mechanism, which ensures exact convergence even when using compression.

While compression reduces the size of communications, local training instead reduces their frequency. The idea is to allow the agents to perform multiple steps of training between each round of communications. This paradigm has been applied *e.g.* in [11], [12] to gradient tracking, in [4] to the distributed dual ascent method, and in our works [13], [14] that focus on redesigning the distributed ADMM.

It is worth noting that the algorithms we reviewed integrate only one or two computation- or communication-efficient design strategies, thus resulting in inexact convergence when dealing with both stochastic gradients and compression. Therefore, in this paper we devise an algorithm aiming at joint computational and communication efficiency. Our main contributions are:

- We propose a novel algorithm, LT-ADMM-CC (Local Training ADMM with Compressed Communication), based on [14], which employs stochastic gradients with variance reduction for computation-efficiency, and compression and local training for communication-efficiency. Additionally, our design integrates error feedback.
- We analyze the convergence of LT-ADMM-CC in a strongly convex setting, and prove its exact linear convergence to the optimal solution. This important result is enabled by double feedback loop of variance reduction and error feedback, which asymptotically reject the stochastic gradient and compression errors, respectively.
- We provide a numerical comparison of LT-ADMM-CC with state-of-the-art alternatives in the context of a classification task, thus highlighting its exact convergence and showcasing its superior performance.

II. Algorithm Design and Analysis

We start by formally introducing the problem, then present the proposed algorithm and analyze its convergence.

A. Problem Formulation

Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of N agents, each of which has access to a local dataset that defines the cost

$$f_i(x) = \frac{1}{m_i} \sum_{h=1}^{m_i} f_{i,h}(x)\,, \tag{1}$$

The work of X. R and T. P. was partially supported by European Union's Horizon 2020 research and innovation programme under grant agreement no. 739551 (KIOS CoE).

The work of N.B. and K.H.J. was partially supported by the European Union's Horizon Research and Innovation Actions programme under grant agreement No. 101070162, and partially by Swedish Research Council Distinguished Professor Grant 2017-01078 Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

[1]School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, United States.

[2]School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden.

[3]Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom.

[4]Department of Electronic Systems, Aalborg University, Denmark.

[5]Department of Engineering and Architecture, University of Trieste, Trieste, Italy.

⋆Corresponding author. Email: nicolba@kth.se.

with $f_{i,h} : \mathbb{R}^n \to \mathbb{R}$ being the loss function associated to data point $h \in \{1, \dots, m_i\}$. The goal is for the network to solve the following consensus optimization problem

$$\min_{x_i \in \mathbb{R}^n,\ i \in \mathcal{V}} \quad \frac{1}{N} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad x_1 = x_2 = \cdots = x_N \,, \tag{2}$$

where the objective sums the local costs (1), and the constraints enforce agreement on a shared trained model. In the following, we denote the (unique) optimal solution as $\mathbf{X}^* = \mathbf{1}_N \otimes x^*$, where $\otimes$ denotes Kronecker product, and $x^* = \arg\min_{x \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(x)$.

We characterize the problem via the following assumptions.

Assumption 1: The cost f_i of each agent $i \in \mathcal{V}$ is L-smooth and μ-strongly convex, with $0 < \mu \leq L < \infty$.[1]

Assumption 2: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected and undirected.

These assumptions are standard for distributed problems, with only strong convexity being somewhat restrictive. However, strong convexity is instrumental to prove linear convergence; and we remark that it can be relaxed to allow for nonconvex problems extending the analysis of [14].

B. Algorithm design

We recall that we focus on designing a distributed learning algorithm that is jointly computation- and communication-efficient. To this end, we start from the distributed ADMM of [15], and in the following steps we integrate some suitable design modifications. Therefore, our starting point is the algorithm characterized by the updates:

$$x_{i,k+1} = \operatorname{prox}_{f_i}^{1/\rho|\mathcal{N}_i|} \left(\sum\nolimits_{j \in \mathcal{N}_i} z_{ij,k} / \rho \, |\mathcal{N}_i| \right) \tag{3a}$$

$$z_{ij,k+1} = 0.5 \left(z_{ij,k} - z_{ji,k} + 2\rho x_{j,k+1} \right) \tag{3b}$$

where $\rho > 0$ is a penalty parameter, $\operatorname{prox}_{f_i}^{1/\rho|\mathcal{N}_i|}(z) = \arg\min_{x \in \mathbb{R}^n} \{ f_i(x) + (\rho \, |\mathcal{N}_i| / 2) \|x - z\|^2 \}$, and $z_{ij,k}$ and $z_{ji,k}$ are edge-wise auxiliary variables.

a) Communication-efficiency 1: The distributed implementation of (3) entails agent $j \in \mathcal{N}_i$ sending $-z_{ji,k} + 2\rho x_{j,k+1}$ to i [15]. However, in a learning setting where $n \gg 1$, this requires prohibitively large bandwidth, and as a solution compressed communications can be employed. Letting $\mathcal{C} : \mathbb{R}^n \to \mathbb{R}^n$ be a compression operator,[2] we could then allow agent j to transmit $\mathcal{C}\left(-z_{ji,k} + 2\rho x_{j,k+1}\right)$. This design choice would reduce the communication burden, but would also result in inexact convergence. Thus, we also integrate an *error feedback* mechanism that asymptotically reduces to zero the error induced by compression. In particular, we define the auxiliary variables $u_{i,k}$ and $s_{ij,k}$, and rewrite (3b) as:

$$z_{ij,k+1} = 0.5 \left(\hat{z}_{ij,k} - \hat{z}_{ji,k} \right) + r\rho x_{i,k+1} - r\rho (\hat{x}_{i,k+1} - \hat{x}_{j,k+1}) \tag{4}$$

where $r > 0$ and we define

$$\hat{x}_{i,k+1} = u_{i,k+1} + \mathcal{C}(x_{i,k+1} - u_{i,k+1}), \tag{5a}$$

$$\hat{z}_{ij,k+1} = s_{ij,k+1} + \mathcal{C}(z_{ij,k+1} - s_{ij,k+1}) \tag{5b}$$

and, setting $\eta \in (0, 1]$,

$$\begin{aligned} u_{i,k+1} &= (1 - \eta) u_{i,k} + \eta \hat{x}_{i,k} \\ s_{ij,k+1} &= \hat{z}_{ij,k}. \end{aligned} \tag{6}$$

It can be seen that the transmission of $\hat{x}_{j,k+1}$ involves the exact transmission of $u_{j,k+1}$ because of (5a). To overcome this, we let agent i keep a copy $\tilde{u}_{i,k}$ of $u_{j,k}$, according to (6), set $\tilde{u}_{i,0} = u_{j,0}$, agent i can maintain $\tilde{u}_{i,k+1} = u_{j,k+1}$ by only receiving $\mathcal{C}(x_{j,k} - u_{j,k})$ by mathematical induction. Specifically, given $\tilde{u}_{i,k} = u_{j,k}$, $\tilde{u}_{i,k+1} = (1 - \eta)\tilde{u}_{i,k} + \eta(\tilde{u}_{i,k} + \mathcal{C}(x_{j,k} - u_{j,k})) = u_{j,k+1}$, $\forall k \geq 0$. The same holds for $\hat{\mathbf{z}}_{ji,k}$, we let agent i keep a copy $\tilde{s}_{ij,k}$ of $s_{ji,k}$.

b) Communication efficiency 2: Besides reducing the required bandwidth via compression, in our design we also reduce the frequency of communications via local training. Following the idea in [14], we notice that update (3a) requires the solution of an optimization problem, which in general lacks a closed form. The idea then is to allow the agents to approximate its solution with a finite number τ of gradient-based steps:

$$\begin{aligned} &\phi_{i,k}^0 = x_{i,k} \\ &\phi_{i,k}^{t+1} = \phi_{i,k}^t + \qquad\qquad\qquad\qquad t = 0, \dots, \tau - 1 \\ &\quad - \gamma g_i(\phi_{i,k}^t) - \beta\rho \, |\mathcal{N}_i| \left(r^2 x_{i,k} - \frac{r}{\rho \, |\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} z_{ij,k} \right) \\ &x_{i,k+1} = \phi_{i,k}^\tau \end{aligned} \tag{7}$$

where $g_i(\phi_{i,k}^t) = \nabla f_i(\phi_{i,k}^t)$, $\gamma > 0$ is the step-size and $\beta > 0$ an additional regularization weight. As a consequence, one round of communication is performed every τ local updates.

c) Computational efficiency: The use of full gradient evaluations $g_i(\phi_{i,k}^t) = \nabla f_i(\phi_{i,k}^t)$ might be computationally expensive when $m_i \gg 1$ in (1). Thus, we allow the agents to use a variance reduced stochastic gradient estimator [16], where each agent maintains a table of component gradients $\{\nabla f_{i,h}(r_{i,h,k}^t)\}, h = 1, \dots, m_i$, where $r_{i,h,k}^t$ represents the most recent iterate at which the component gradient was computed. This table is reset at the start of each new local training, and the agents estimate their local gradients as

$$\begin{aligned} g_i\left(\phi_{i,k}^t\right) = &\frac{1}{|\mathcal{B}_i|} \sum_{h \in \mathcal{B}_i} \left(\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \nabla f_{i,h}\left(r_{i,h,k}^t\right) \right) \\ &+ \frac{1}{m_i} \sum_{h=1}^{m_i} \nabla f_{i,h}(r_{i,h,k}^t). \end{aligned} \tag{8}$$

where $\mathcal{B}_i$ represents a randomly selected subset of indices from $\{1, \dots, m_i\}$, with $|\mathcal{B}_i| < m_i$. The estimated gradient is then used to update $\phi_{i,k}^{t+1}$ according to (7); after each update, the agents refresh their local memory by setting $r_{i,h,k}^{t+1} = \phi_{i,k}^{t+1}$ if $h \in \mathcal{B}_i$, and $r_{i,h,k}^{t+1} = r_{i,h,k}^t$ otherwise. This update requires a full gradient computation at the beginning of each

[1] We recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if it is differentiable and $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathbb{R}^n$; moreover, f is μ-strongly convex if $\langle \nabla f_i(x) - \nabla f_i(y), x - y \rangle \geq \mu \|x - y\|^2$, $\forall x, y \in \mathbb{R}^n$.

[2] Precise assumptions will be introduced in section II-C.

local training step, in the following steps ($t > 0$), each agent only computes $|\mathcal{B}_i|$ component gradients.

The result of our design is reported in Algorithm 1.

Algorithm 1

Input: For each node i, initialize $x_{i,0} = z_{ij,0}$, $u_{i,0} = \tilde{u}_{i,0} = 0$, $s_{ij,0} = \tilde{s}_{ij,0} = 0$, $j \in \mathcal{N}_i$. Set the penalty parameter $\rho > 0$, the number of local training steps $\tau > 0$, and the parameters $\gamma, \beta, r > 0$, $0 < \eta \leq 1$.

for $k = 0, 1, \ldots$ every agent i **do**
  // local training
  $\phi^0_{i,k} = x_{i,k}$, $r^0_{i,h,k} = x_{i,k}$, for all $h \in \{1, \ldots . m_i\}$
  for $t = 0, 1, \ldots, \tau - 1$ **do**
    Draw the batch $\mathcal{B}_i$ uniformly at random
    Update the gradient estimator according to (8)
    Update $\phi_{i,k}$ according to (7)
    If $h \in \mathcal{B}_i$ update $r^{t+1}_{i,h,k} = \phi^{t+1}_{i,k}$, else $r^{t+1}_{i,h,k} = r^t_{i,h,k}$
  end for
  Set $x_{i,k+1} = \phi^\tau_{i,k}$, update $u_{i,k+1}$ and $s_{ij,k+1}$ according to (6), update $\hat{x}_{i,k+1}$ according to (5a)
  // communication
  Transmit $\mathcal{C}(z_{ij,k} - s_{ij,k})$ and $\mathcal{C}(x_{i,k+1} - u_{i,k+1})$ to each neighbor $j \in \mathcal{N}_i$, and receive the corresponding transmissions
  // auxiliary update
  Update the local copy $\tilde{u}_{i,k+1}$ and $\tilde{s}_{ij,k+1}$, update $z_{ij,k+1}$ according to (4).
  Update $\hat{z}_{ij,k+1}$ according to (5b)
end for

C. Convergence analysis

We start by introducing suitable assumptions on the compressor operator.

Assumption 3: The compression operator $\mathcal{C} : \mathbb{R}^n \to \mathbb{R}^n$ is unbiased $\mathbb{E}\big[\mathcal{C}(x)\big] = x$, and there exists a constant $p \geq 1$ such that $\mathbb{E}\big[\|\mathcal{C}(x) - x\|^2\big] \leq p\|x\|^2, \forall x \in \mathbb{R}^n$.

Assumption 4: We assume that all agents' compressors are mutually independent among the agents, *i.e.*, their outputs are mutually independent random variables.

We can now characterize the convergence of Algorithm 1. The result is proved in the Appendix.

Theorem 1: Let Assumptions 1, 2, 3 and 4 hold. Let $\{\mathbf{X}_k\}_{k\in\mathbb{N}}$ be the trajectory generated by LT-ADMM-CC. Then with sufficiently small γ, bounded p, there exist positive parameters β, τ, r, ρ such that the states $\mathbf{X}_k$ converge linearly to the optimal solution $\mathbf{X}^*$.

III. Numerical Results

In this section, we compare LT-ADMM-CC with state-of-the-art alternatives for a classification task characterized by

$$f_i(x) = \sum_{h=1}^{m_i} \log\left(1 + \exp(-b_i^h a_i^h x)\right) + (\epsilon/2)\,\|x\|^2 \quad (9)$$

with $a_i^h \in \mathbb{R}^n$, and $b_i^h \in \{-1, 1\}$ randomly generated. We choose a ring network with $N = 10$, set $n = 5$ and $m_i = 100$, and use stochastic gradients with a batch of $|\mathcal{B}| = 1$.

A. LT-ADMM-CC performance

We start by evaluating the performance of LT-ADMM-CC with the following unbiased compressors. The other parameters are always set as $\tau = 5$, $\rho = 0.1$, $\beta = 0.2$, $\gamma = 0.3$, $r = 1$.

b-bit quantizer The first compressor is defined as

$$\mathcal{C}_1(x) = \frac{\|x\|_\infty \operatorname{sign}(x)}{2^{b-1}} \circ \left\lfloor 2^{b-1}\,\frac{|x|}{\|x\|_\infty} + \kappa \right\rfloor$$

where $\operatorname{sign}(x)$, $|\cdot|$, $\lfloor\cdot\rfloor$ are applied element-wise, $\circ$ is the element-wise product; and $\kappa \sim \mathcal{U}[0,1]^n$ is a random perturbation.

Rand-k The second compressor is defined as

$$\mathcal{C}_2(x) = \frac{n}{k}\sum\nolimits_{i\in S} x_i\, e_i,$$

where $S \subset [n]$ is a subset of cardinality k chosen uniformly at random, and $\{e_1, \ldots, e_d\}$ is the standard basis in $\mathbb{R}^n$.

Figure 1 shows the evolution of $\|\nabla F(\bar{x}_k)\|^2$, with $\bar{x}_k = (1/N)\sum_{i=1}^N x_{i,k}$ over the iterations $k \in \mathbb{N}$. As predicted by

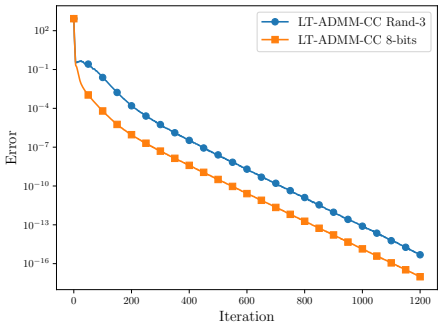

Fig. 1. LT-ADMM-CC with different compressors.

the theory, the algorithm achieves exact convergence for both $\mathcal{C}_1$ and $\mathcal{C}_2$, although the specific compressor might affect the speed of convergence.

B. Comparison

We compare now the performance of LT-ADMM-CC with alternative distributed optimization methods that employ compression: CEDAS [9], COLD [8], DPDC [7, Algorithm 1], and LEAD [10]. For all algorithms, we use an 8-bit quantizer and stochastic gradients with batch-size $|\mathcal{B}| = 1$; we also hand-tune the parameters of the algorithms to achieve optimal performance.

These algorithms have different computational and communication complexities. To account for this in our simulations, we assign a time cost of t_g for a component gradient evaluation ($\nabla f_{i,h}$), and of t_c for a round of communications. The total time-cost incurred by each algorithm is then reported in Table I.

Figure 2 reports the evolution of $\|\nabla F(\bar{x}_k)\|^2$ against time, with $t_c = 10t_g$ to represent a scenario where communication is expensive. We notice that LEAD, CEDAS, DPDC-sgd, COLD-sgd only converge to a neighborhood of the optimal solution. This is due to the fact that they employ stochastic

TABLE I

COMPUTATION TIME OF THE ALGORITHMS OVER τ ITERATIONS.

Algorithm [Ref.]	Time
LEAD [10]	$\tau(t_g + t_c)$
CEDAS [9]	$\tau(t_g + 2t_c)$
COLD [8]& DPDC [7]	$\tau\,(t_g + t_c)$ stochastic gradient
COLD [8]& DPDC [7]	$\tau\,(m_i t_g + t_c)$ full gradient
LT-ADMM-CC	$(m_i + \tau - 1)t_g + 2t_c$

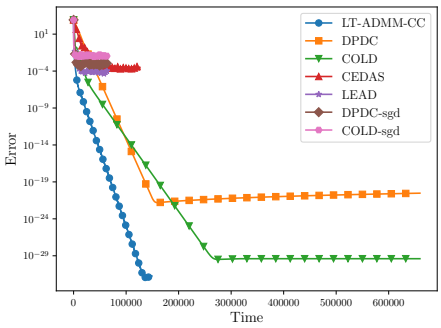

Fig. 2. Comparison of distributed optimization algorithms with compressed communication.

gradients, but no variance reduction. LT-ADMM-CC instead converges exactly owing to variance reduction and error feedback. We remark that DPDC and COLD also employ error feedback, hence they converge exactly when using full gradients. Notice, however, that using full gradients entails a higher time-cost, as demonstrated by their slower convergence as compared to LT-ADMM-CC.

IV. Concluding Remarks

In this paper, we proposed a novel distributed learning algorithm that is jointly computation- and communication-efficient. This is achieved by integrating communication compression, local training, and variance reduction. We proved the exact, linear convergence of the algorithm, and compared it numerically with state-of-the-art alternatives. Future research efforts will be devoted to weakening some assumptions, for example the strong convexity one, and to apply the proposed algorithm to real-world scenarios.

Appendix

In the following, we report the proof of Theorem 1, the detailed proof can be found in [17].

A. Preliminary transformation

Denote $\Phi_k^t = \mathrm{col}\{\phi_{1,k}^t, \phi_{2,k}^t, ..., \phi_{N,k}^t\}$, $G(\Phi_k^t) = \mathrm{col}\{g_1(\phi_{1,k}^t), g_2(\phi_{2,k}^t), ..., g_N(\phi_{N,k}^t)\}$, $\mathrm{F}(\mathbf{X}) = \mathrm{col}\{f_1(x_1), f_2(x_2), ..., f_N(x_N)\}$, $\mathbf{Z} = \mathrm{col}\{z_{ij}\}_{i,j\in\mathcal{E}}$. Define $\mathbf{A} = \mathrm{blk\,diag}\{\mathbf{1}_{d_i}\}_{i\in\mathcal{V}} \otimes \mathbf{I}_n \in \mathbb{R}^{Mn\times Nn}$, where $d_i = |\mathcal{N}_i|$ is the degree of node i, and $M = \sum_i |\mathcal{N}_i|$. $\mathbf{P} \in \mathbb{R}^{Mn\times Mn}$ is a permutation matrix that swaps e_{ij} with e_{ji}. $\mathbf{A}^T\mathbf{P}\mathbf{A} = \tilde{\mathbf{A}}$ is the adjacency matrix, $\mathbf{A}^T\mathbf{A} = \mathrm{diag}\{d_i\mathbf{I}_n\}_{i\in\mathcal{V}}$ is the degree matrix, denote d_u as the largest degree among the agents. Denote the largest and smallest nonzero eigenvalue of $\mathbf{L} = \mathbf{D} - \tilde{\mathbf{A}}$ as λ_u and λ_l, respectively.

The compact form of LT-ADMM-CC is:

$$\hat{\mathbf{X}}_k = \mathbf{U}_k + \mathcal{C}(\mathbf{X}_k - \mathbf{U}_k), \quad \mathbf{U}_{k+1} = (1-\eta)\mathbf{U}_k + \eta\hat{\mathbf{X}}_k,$$
$$\hat{\mathbf{Z}}_k = \mathbf{S}_k + \mathcal{C}(\mathbf{Z}_k - \mathbf{S}_k), \quad \mathbf{S}_{k+1} = \hat{\mathbf{Z}}_k \tag{10a}$$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \gamma\sum_{t=0}^{\tau-1}\nabla G(\Phi_k^t) - \beta r\sum_{t=0}^{\tau-1}(r\rho\mathbf{A}^T\mathbf{A}\mathbf{X}_k - \mathbf{A}^T\mathbf{Z}_k) \tag{10b}$$

$$\mathbf{Z}_{k+1} = \frac{1}{2}(\mathbf{I}-\mathbf{P})\hat{\mathbf{Z}}_k + r\rho\mathbf{A}\mathbf{X}_{k+1} - r\rho(\mathbf{I}-\mathbf{P})\mathbf{A}\hat{\mathbf{X}}_{k+1}. \tag{10c}$$

We introduce the following variables

$$\begin{aligned}\mathbf{Y}_k &= r\mathbf{A}^T\mathbf{Z}_k - \frac{\gamma}{\beta}\nabla F(\bar{\mathbf{X}}_k) - r^2\rho\mathbf{D}\mathbf{X}_k\\ \tilde{\mathbf{Y}}_k &= r\mathbf{A}^T\mathbf{P}\mathbf{Z}_k + \frac{\gamma}{\beta}\nabla\mathrm{F}(\bar{\mathbf{X}}_k) - r^2\rho\mathbf{D}\mathbf{X}_k,\end{aligned} \tag{11}$$

where $\bar{\mathbf{X}}_k = \mathbf{1}_N \otimes \bar{x}_k$, with $\bar{x}_k = \frac{1}{N}\mathbf{1}^T\mathbf{X}_k$. Multiplying both sides of (10c) by $r\mathbf{1}^T\mathbf{A}^T$, and using the initial condition, we obtain $r\mathbf{1}^T\mathbf{A}^T\mathbf{Z}_{k+1} = r^2\rho\mathbf{1}^T\mathbf{D}\mathbf{X}_{k+1}$ for all $k \geq 0$. As a consequence $\tilde{\mathbf{Y}}_k = \mathbf{1}\otimes\frac{1}{N}\mathbf{1}^T\nabla\mathrm{F}(\bar{\mathbf{X}}_k) = \frac{\gamma}{\beta}\mathbf{1}\otimes\frac{1}{N}\sum_i\nabla f_i(\bar{x}_k)$, and (10) can be rewritten as

$$\begin{bmatrix}\mathbf{X}_{k+1}\\ \mathbf{Y}_{k+1}\\ \tilde{\mathbf{Y}}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \beta\tau\mathbf{I} & \mathbf{0}\\ \rho\tilde{\mathbf{L}} & \rho\tilde{\mathbf{L}}\beta\tau + \frac{1}{2}\mathbf{I} & -\frac{1}{2}\mathbf{I}\\ \mathbf{0} & -\frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I}\end{bmatrix}\otimes\mathbf{I}_n\begin{bmatrix}\mathbf{X}_k\\ \mathbf{Y}_k\\ \tilde{\mathbf{Y}}_k\end{bmatrix} - \mathbf{h}_k, \tag{12}$$

where $\tilde{\mathbf{L}} = r^2(\tilde{\mathbf{A}} - \mathbf{D}) = -r^2\mathbf{L}$ and

$$\begin{aligned}\mathbf{h}_k = \Bigg[&\gamma\sum_{t=0}^{\tau-1}(\nabla G(\Phi_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k));\\ &\gamma\rho\tilde{\mathbf{L}}\sum_{t=0}^{\tau-1}(\nabla G(\Phi_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k))\\ &+\frac{\gamma}{\beta}(\nabla\mathrm{F}(\bar{\mathbf{X}}_{k+1}) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k)) + \tilde{\mathbf{L}}(\mathbf{X}_{k+1} - \hat{\mathbf{X}}_{k+1});\\ &\frac{\gamma}{\beta}(-\nabla\mathrm{F}(\bar{\mathbf{X}}_{k+1}) + \nabla\mathrm{F}(\bar{\mathbf{X}}_k)) - \tilde{\mathbf{L}}(\mathbf{X}_{k+1} - \hat{\mathbf{X}}_{k+1})\\ &-\frac{r\mathbf{A}^T(\mathbf{I}-\mathbf{P})}{2}(\mathbf{Z}_k - \hat{\mathbf{Z}}_k)\Bigg].\end{aligned}$$

Denote $\|\hat{\mathbf{\Phi}}_k\|^2 = \sum_{i=1}^N\sum_{t=0}^{\tau-1}\|\phi_{i,k}^t - \bar{x}_k\|^2 = \sum_{t=0}^{\tau-1}\|\mathbf{\Phi}_k^t - \bar{X}_k\|^2$, using Assumption 1 we derive that

$$\begin{aligned}&\|\sum_{t=0}^{\tau-1}(G(\Phi_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k))\|^2\\ &\leq 2\tau L^2\|\hat{\mathbf{\Phi}}_k\|^2 + 2\tau\sum_{t=0}^{\tau-1}\|G(\Phi_k^t) - \nabla F(\Phi_k^t)\|^2.\end{aligned}$$

Denote $\overline{G}(\Phi_k^t) = \frac{1}{N}\sum_{i=1}^N g_i(\phi_{i,k}^t)$, we have

$$\begin{aligned}\|\sum_{t=0}^{\tau-1}\overline{G}(\Phi_k^t)\|^2 &= \|\frac{1}{N}\sum_i\sum_t(\nabla f_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\bar{x}^k\right)\\ &+\nabla f_i\left(\bar{x}^k\right) + g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right))\|^2\\ &\leq \frac{3\tau L^2}{N}\|\hat{\mathbf{\Phi}}^k\|^2 + \frac{3\tau}{N}\sum_i\sum_t\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right)\|^2\\ &+3\tau^2\|\nabla F(\bar{x}_k)\|^2.\end{aligned} \tag{13}$$

We also have $\|\nabla F(\bar{\mathbf{X}}_{k+1}) - \nabla F(\bar{\mathbf{X}}_k)\|^2 = NL^2\|\bar{x}_{k+1} - \bar{x}_k\|^2 = NL^2\gamma^2\|\sum_t \overline{G}(\Phi_k^t)\|^2$, using Assumption 3, it further holds that:

$$\begin{aligned}\|\mathbf{h}_k\|^2 &\le \gamma^2(1+3\rho^2\|\tilde{\mathbf{L}}\|^2)(2\tau L^2\|\widehat{\mathbf{\Phi}}_k\|^2 \\ &+ 2\tau\sum_{t=0}^{\tau-1}\|G(\Phi_k^t) - \nabla F(\Phi_k^t)\|^2) \\ &+ 18L^2\frac{\gamma^4}{\beta^2}(\tau L^2\|\widehat{\Phi}^k\|^2 + \tau\sum_i\sum_t\left\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right)\right\|^2 \\ &+ N\tau^2\|\nabla F(\bar{x}_k)\|^2) + 6r^4(p-1)\lambda_u^2\|\mathbf{X}_{k+1} - \mathbf{U}_{k+1}\|^2 \\ &+ \frac{3r^2d_u}{4}\|(\mathbf{I}-\mathbf{P})(\mathbf{Z}_k - \hat{\mathbf{Z}}_k)\|^2.\end{aligned} \tag{14}$$

B. Key bounds

Lemma 1: Let Assumption 2 hold, when $\beta < \frac{2}{r^2\tau\lambda_u\rho}$,

$$\|\bar{\mathbf{X}}_k - \mathbf{X}_k\|^2 \le \frac{18\beta\tau}{r^2\lambda_l\rho}\|\widehat{\mathbf{d}}_k\|^2, \quad \|\bar{\mathbf{Y}}_k - \mathbf{Y}_k\|^2 \le 9\|\widehat{\mathbf{d}}_k\|^2, \tag{15}$$

$$\widehat{\mathbf{d}}_{k+1} = \mathbf{\Delta}\widehat{\mathbf{d}}_k - \widehat{\mathbf{h}}_k, \tag{16}$$

where $\widehat{\mathbf{d}}_k = \widehat{\mathbf{V}}^{-1}\left[\widehat{\mathbf{Q}}^T\mathbf{X}_k; \widehat{\mathbf{Q}}^T\mathbf{Y}_k; \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k\right]$, $\widehat{\mathbf{h}}_k = \widehat{\mathbf{V}}^{-1}\widehat{\mathbf{Q}}^T\mathbf{h}_k$. $\mathbf{\Delta}$ is a block-diagonal matrix satisfies $\delta = \|\mathbf{\Delta}\| = 1 - r^2\lambda_l\rho\tau\beta/2$, $\widehat{\mathbf{V}}^{-1}$ is a block orthogonal matrix.

Now we derive an upper bound for $\mathrm{E}[\|g_i(\phi_{i,k}^t) - \nabla f_i(\phi_i^k)\|^2]$, which is the variance of the gradient estimator. Define t_i^k as the averaged optimality gap of the auxiliary variables of $\left\{\mathbf{r}_{i,j}^k\right\}_{j=1}^{m_i}$ at node i as follows:

$$\begin{aligned}t_{i,k}^t &= \frac{1}{m_i}\sum_{h=1}^{m_i}\|r_{i,h,k}^t - \bar{x}_k\|^2, \\ t_k^t &= \sum_{i=1}^{N}t_{i,k}^t = \frac{1}{m_i}\sum_{h=1}^{m_i}\|\mathbf{r}_{h,k}^t - \bar{\mathbf{X}}_k\|^2, \\ t_k &= \sum_{t=0}^{\tau-1}t_k^t = \sum_{t=0}^{\tau-1}\sum_{i=1}^{N}t_{i,k}^t.\end{aligned}$$

Then using $\mathbb{E}[\|a-\mathbb{E}[a]\|^2] \le \mathbb{E}[\|a\|^2]$ with $a = \nabla f_{i,h}(\phi_{i,k}^t) - \nabla f_{i,h}(r_{i,h,k}^t)$,

$$\begin{aligned}&\mathrm{E}\left[\|g_i(\phi_{i,k}^t) - \nabla f_i\left(\phi_{i,k}^t\right)\|^2\right] \\ &\le \mathrm{E}\left[\left\|\nabla f_{i,s_i^k}(\phi_{i,k}^t) - \nabla f_{i,s_i^k}\left(\mathbf{r}_{i,s_i^k}^k\right)\right\|^2\right] \\ &= \frac{1}{m_i}\sum_{j=1}^{m_i}\|(\nabla f_{i,j}(\phi_{i,k}^t) - \nabla f_{i,j}(\bar{x}_k)) \\ &+ (\nabla f_{i,j}(\bar{x}_k) - \nabla f_{i,j}(\mathbf{r}_{i,j}^k))\|^2 \\ &\le 2L^2\left\|\phi_{i,k}^t - \bar{x}_k\right\|^2 + 2L^2t_i^k,\end{aligned} \tag{17}$$

it follows that

$$\sum_i\sum_t\left\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i(\phi_{i,k}^t)\right\|^2 \le 2L^2\|\widehat{\mathbf{\Phi}}_k\|^2 + 2L^2t_k. \tag{18}$$

Lemma 2: Let Assumptions hold; when $\beta < \frac{2}{r^2\tau\lambda_u\rho}$ and $4\gamma^2\tau(2L^2+L^2) \le \frac{1/4}{\tau-1}$, we have

$$\begin{aligned}\mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] &\le \left(\frac{72\beta\tau^2}{r^2\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\ &+ 16\tau^3\gamma^2N\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + 32\tau^2\gamma^2L^2\mathbb{E}[t_k].\end{aligned} \tag{19}$$

The following lemma provides the bound on t_k.

Lemma 3: Let Assumptions 1 and 2 hold, $\{t_k\}$ be the iterates generated by LT-ADMM-CC. If γ satisfies $4\gamma^2\tau(2L^2+L^2) \le \frac{1/4}{\tau-1}$, $\frac{16\gamma^2L^2}{m_l} < \frac{1}{2m_u}$, $24\gamma^2L^2 < 2$ and $\frac{8m_u\tau}{m_l}32\tau^2\gamma^2L^2 < \frac{1}{2}$, $\beta < \frac{2}{r^2\tau\lambda_u\rho}$, we have for all $k \in \mathbb{N}$:

$$\mathbb{E}[t_k] \le 2(s_0+s_1)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + 2s_2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2], \tag{20}$$

where $s_0 = \frac{36\beta\tau^2m_u}{\lambda_l\rho} + \frac{144\tau^2m_u}{m_l}\beta^2$, $s_1 = \left(\frac{72\beta\tau^2}{r^2\lambda_l\rho} + 144\tau^3\beta^2\right)\frac{8m_u\tau}{m_l}$, $s_2 = \frac{16N\gamma^2m_u\tau^2}{m_l} + \frac{8m_u\tau}{m_l}16\tau^3\gamma^2N$.

Lemma 4: Let Assumptions 1, 2, 3 and 4 hold, set $r^2\lambda_u\rho\tau\beta = 1$, γ satisfies the conditions in Lemma 3, it holds that

$$\begin{aligned}\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] &\le (\delta + \frac{q_0\|\widehat{\mathbf{V}}^{-1}\|^2}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\ &+ \frac{q_1\|\widehat{\mathbf{V}}^{-1}\|^2}{1-\delta}\mathbb{E}[\|\bar{x}_k - x^*\|^2] \\ &+ \frac{a_5\|\widehat{\mathbf{V}}^{-1}\|^2}{1-\delta}\mathbb{E}[\|\mathbf{X}_k - \mathbf{U}_k\|^2] \\ &+ \frac{a_6\|\widehat{\mathbf{V}}^{-1}\|^2}{1-\delta}\mathbb{E}[\|(\mathbf{I}+\mathbf{P})(\mathbf{Z}_k - \mathbf{S}_k)\|^2]\end{aligned} \tag{21}$$

where

$$\begin{aligned}a_0 &= \gamma^2\bigg((1+3\rho^2\|\tilde{\mathbf{L}}\|^2)6\tau L^2 + 54L^4\frac{\gamma^2\tau}{\beta^2} \\ &+ 72r^4(p-1)\lambda_u^2\frac{\tau L^2}{\eta}\bigg) \\ a_1 &= \gamma^2\bigg((1+3\rho^2\|\tilde{\mathbf{L}}\|^2)4\tau L^2 + 36L^4\frac{\gamma^2}{\beta^2}\tau \\ &+ 48r^4(p-1)\lambda_u^2\frac{\tau L^2}{\eta}\bigg) \\ a_3 &= \gamma^2\left(18L^2\frac{\gamma^2}{\beta^2}N\tau^2 + 24r^4(p-1)\lambda_u^2\tau^2N\right) \\ a_4 &= \frac{8(p-1)}{3\rho^2\eta} + \frac{15d_u(p-1)}{\lambda_l\rho} \\ a_5 &= 6r^4(p-1)\lambda_u^2(1-\eta+\eta^2(p-1)) \\ &+ \frac{3r^2d_u}{4}\frac{5}{2}(p-1)8\rho^2r^2\lambda_u^2(p-1) \\ a_6 &= \frac{3r^2d_u}{8}(p-1) \\ q_0 &= (\frac{8\tau\lambda_u}{\lambda_lK^2} + \frac{16\tau}{9K^2})a_0 + 64\tau^2\gamma^2L^2a_0(s_0+s_1) \\ &+ 2a_1(s_0+s_1) + a_4 \\ q_1 &= 16\tau^3\gamma^2NL^2a_0 + 64\tau^2\gamma^2L^4a_0s_2 + 2a_1s_2L^2 + a_3, \\ \delta &= 1 - \frac{\lambda_l}{2\lambda_u}, \quad K = r^2\rho\lambda_u\end{aligned} \tag{22}$$

C. Deviation of $\bar{x}_k$ from the optimal solution x^*

From (10b), we can derive that

$$\begin{aligned}\mathrm{E}\|\bar{x}_{k+1}-x^*\|^2 &\le \mathrm{E}\|\bar{x}_k-x^*\|^2+\gamma^2\mathrm{E}\|\sum_{t=0}^{\tau-1}\overline{G}(\Phi_k^t)\|^2\\ &-2\frac{\gamma}{N}\langle\bar{x}_k-x^*,\sum_{t=0}^{\tau-1}\sum_{i=1}^{N}\nabla f_i(\phi_{i,k}^t)\rangle\end{aligned} \tag{23}$$

Since $\langle(z-y),\nabla g(x)\rangle \ge g(z)-g(y)+\frac{\mu}{4}\|y-z\|^2-L\|z-x\|^2, \forall x,y,z\in R^n$ for any L-smooth and μ-strongly convex function g [18], we have

$$\begin{aligned}&-\frac{2\gamma}{N}\sum_i\sum_t\left\langle\left(\bar{x}_k-x^\star\right),\nabla f_i\left(\phi_{i,k}^t\right)\right\rangle\\ &\le\frac{2\gamma}{N}\sum_i\sum_t(f_i\left(x^\star\right)-f_i\left(\bar{x}_k\right)-\frac{\mu}{4}\left\|\bar{x}_k-x^\star\right\|^2\\ &+L\left\|\bar{x}_k-\phi_{i,k}^t\right\|^2)\\ &=-2\gamma\tau(F\left(\bar{x}_k\right)-F\left(x^\star\right)+\frac{\mu}{4}\left\|\bar{x}_k-x^\star\right\|^2)+\frac{2\gamma L}{N}\|\widehat{\mathbf{\Phi}}^k\|^2,\end{aligned} \tag{24}$$

it follows that

$$\begin{aligned}&\mathrm{E}[\|\bar{x}^{k+1}-x^\star\|^2]\le\left(1-\frac{\mu\tau\gamma}{2}\right)\mathbb{E}\left\|\bar{x}^k-x^\star\right\|^2\\ &+\frac{2\gamma L}{N}\mathbb{E}\|\widehat{\Phi}^k\|^2-2\gamma\tau\left(F\left(\bar{x}^k\right)-F\left(x^\star\right)\right)\\ &+\gamma^2\|\sum_{t=0}^{\tau-1}\overline{G}(\Phi_k^t)\|^2\\ &\le\Bigg(1-\frac{\mu\tau\gamma}{2}+(\frac{2\gamma L}{N}+\frac{9\gamma^2\tau L^2}{N})16\tau^3\gamma^2NL^2\\ &+(\frac{2\gamma L}{N}+\frac{9\gamma^2\tau L^2}{N})64\tau^2\gamma^2L^2s_2L^2\\ &+\gamma^2\frac{12\tau L^2}{N}s_2L^2+3\gamma^2\tau^2L^2\Bigg)\mathrm{E}[\|\bar{x}^k-x^\star\|^2]\\ &+\Bigg((\frac{2\gamma L}{N}+\frac{9\gamma^2\tau L^2}{N})(\frac{72\beta\tau^2}{\lambda_l\rho}+144\tau^3\beta^2)\\ &+(\frac{2\gamma L}{N}+\frac{9\gamma^2\tau L^2}{N})64\tau^2\gamma^2L^2(s_0+s_1)\\ &+\gamma^2\frac{12\tau L^2}{N}(s_0+s_1)\Bigg)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2]\\ &=w_0\mathrm{E}[\|\bar{x}^k-x^\star\|^2]+w_1\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2].\end{aligned} \tag{25}$$

D. Proof of Theorem 1

Let Assumptions 1, 2, 3 and 4 hold, γ satisfies the conditions in Lemma 3, $r^2\lambda_u\rho\tau\beta=1$. Based on the above relations, we can derive that

$$\mathbf{T}_{k+1}\le\mathbf{\Xi}\mathbf{T}_k, \tag{26}$$

where $\mathbf{T}_k=[\mathbb{E}[\|\bar{x}^{k+1}-x^\star\|^2];\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2];\mathbb{E}[\|\mathbf{X}_{k+1}-\mathbf{U}_{k+1}\|^2];\mathbb{E}[\|(\mathbf{I}+\mathbf{P})(\mathbf{Z}_{k+1}-\mathbf{S}_{k+1})\|^2]]$. The diagonal elements of $\mathbf{\Xi}$ are $\mathbf{\Xi}_{11}=w_0$, $\mathbf{\Xi}_{22}=\delta+\frac{q_0\|\widehat{\mathbf{V}}^{-1}\|^2}{1-\delta}$, $\mathbf{\Xi}_{33}=1-\eta+\eta^2(p-1)$, $\mathbf{\Xi}_{44}=\frac{5(p-1)}{2}$. When the spectral radius of $\mathbf{\Xi}$ verifies $\mathrm{sr}(\mathbf{\Xi})<1$, then $\mathbf{X}_k$ generated by LT-ADMM-CC converges linearly to the optimal solution $\mathbf{X}^*$. This condition can be verified by a suitable choice of parameters $\beta,\tau,r,\rho,\gamma,\eta,p$.

REFERENCES

[1] D. K. Molzahn, F. Dorfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.

[2] A. Nedić and J. Liu, "Distributed Optimization for Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, May 2018.

[3] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International conference on machine learning*. PMLR, 2019, pp. 3478–3487.

[4] S. A. Alghunaim, "Local Exact-Diffusion for Decentralized Optimization and Learning," Feb. 2023.

[5] H. Li, Z. Lin, and Y. Fang, "Variance Reduced EXTRA and DIGing and Their Optimal Acceleration for Strongly Convex Decentralized Optimization," *Journal of Machine Learning Research*, vol. 23, no. 222, pp. 1–41, 2022.

[6] X. Jiang, X. Zeng, J. Sun, and J. Chen, "Distributed Stochastic Gradient Tracking Algorithm With Variance Reduction for Non-Convex Optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5310–5321, Sep. 2023.

[7] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Communication compression for distributed nonconvex optimization," *IEEE Transactions on Automatic Control*, vol. 68, no. 9, pp. 5477–5492, 2022.

[8] J. Zhang, K. You, and L. Xie, "Innovation compression for communication-efficient distributed optimization with linear convergence," *IEEE Transactions on Automatic Control*, vol. 68, no. 11, pp. 6899–6906, 2023.

[9] K. Huang and S. Pu, "Cedas: A compressed decentralized stochastic gradient method with improved convergence," *IEEE Transactions on Automatic Control*, 2024.

[10] X. Liu, Y. Li, R. Wang, J. Tang, and M. Yan, "Linear convergent decentralized optimization with compression," in *International Conference on Learning Representations*, 2021.

[11] E. D. Hien Nguyen, S. A. Alghunaim, K. Yuan, and C. A. Uribe, "On the Performance of Gradient Tracking with Local Updates," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. Singapore, Singapore: IEEE, Dec. 2023, pp. 4309–4313.

[12] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized Gradient Tracking with Local Steps," Jan. 2023.

[13] X. Ren, N. Bastianello, K. H. Johansson, and T. Parisini, "Distributed learning by local training admm," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 7124–7129.

[14] ——, "Communication-efficient stochastic distributed learning," *arXiv preprint arXiv:2501.13516*, 2025.

[15] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous Distributed Optimization Over Lossy Networks via Relaxed ADMM: Stability and Linear Convergence," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2620–2635, Jun. 2021.

[16] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," *Advances in neural information processing systems*, vol. 27, 2014.

[17] X. Ren, N. Bastianello, K. H. Johansson, and T. Parisini, "Jointly computation-and communication-efficient distributed learning," *arXiv preprint arXiv:2508.15509*, 2025.

[18] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.