Communication-Efficient Stochastic Distributed Learning

Xiaoxing Ren, Nicola Bastianello, Karl H. Johansson, Thomas Parisini

***Abstract*— We address distributed learning problems, both nonconvex and convex, over undirected networks. In particular, we design a novel algorithm based on the distributed *Alternating Direction Method of Multipliers* (ADMM) to tackle the challenges of high communication costs, and large datasets. Our design deals with these challenges i) by enabling the agents to perform multiple local training steps between each round of communications; and ii) by allowing the agents to employ stochastic gradients while carrying out local computations. We show that the proposed algorithm converges to a neighborhood of a stationary point, for nonconvex problems, and of an optimal point, for convex problems. We also propose a variant of the algorithm to incorporate variance reduction thus achieving exact convergence. We show that the resulting algorithm indeed converges to a stationary (or optimal) point, and moreover that local training accelerates convergence. We thoroughly compare the proposed algorithms with the state of the art, both theoretically and through numerical results.**

***Index Terms*— Distributed learning; Stochastic optimization; Variance reduction; Local training.**

I. Introduction

Recent technological advances have enabled the widespread adoption of devices with computational and communication capabilities in many fields, for instance, power grids [1], robotics [2], [3], transportation networks [4], and sensor networks [5]. These devices connect with each other, forming multi-agent systems that cooperate to collect and process data [6]. As a result, there is a growing need for algorithms that enable efficient and accurate cooperative learning.

In specific terms, the objective in distributed learning is to train a model (*e.g.*, a neural network) with parameters $x \in \mathbb{R}^n$ *cooperatively* across a network of N agents. Each agent i has access to a local dataset which defines the local cost as

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

with $f_{i,h} : \mathbb{R}^n \to \mathbb{R}$ being the loss function associated to data point $h \in \{1, \dots, m_i\}$. Thus, the goal is for the agents to solve the following constrained problem [7], [8]:

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

where the objective is the sum of local costs (1) to pool together the agents' data. Moreover, each agent is assigned a set x_i of local model parameters, and the consensus constraints $x_1 = x_2 = \ldots = x_N$ ensure that the agents will asymptotically agree on a shared trained model.

To effectively tackle this problem, especially when dealing with large datasets that involve sensitive information, distributed methods have become increasingly important. These techniques offer significant robustness advantages over federated learning algorithms [6], as they do not rely on a central coordinator and thus, for example, have not a single point of failure. In particular, both distributed gradient-based algorithms [17], [18], [19], [20], and distributed Alternating Direction Method of Multipliers (ADMM) [21], [22], [23], [24] have proven to be effective strategies for solving such problems. ADMM-based algorithms have demonstrated strong robustness to practical constraints (see *e.g.* [22] and references therein), although this often comes at the cost of higher computational complexity compared to gradient-based methods. In this work, we propose novel ADMM-based algorithms that retain the computational efficiency characteristic of gradient methods.

However, many learning applications face the challenges of: high communication costs, especially when training large models, and large datasets. In this paper, we jointly address these challenges with the following approach. First, we guarantee the communication efficiency of our algorithm by adopting the paradigm of *local training*, which reduces the frequency of communications. In other terms, the agents perform multiple training steps between communication rounds. We tackle the second challenge by locally incorporating *stochastic gradients*. The idea is to allow the agents to estimate local gradients by employing only a (random) subset of the available data, thus avoiding the computational burden of full gradient evaluations on large datasets.

The work of X.R. and T.P. was 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 supported by the European Union's Horizon Research and Innovation Actions programme under grant agreement No. 101070162, and by Swedish Research Council Distinguished Professor Grant 2017-01078 Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

X. Ren is with the School of Civil and Environmental Engineering, Systems Engineering Field, Cornell University, Ithaca, NY, USA.

N. Bastianello and K. H. Johansson are with the School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden.

T. Parisini is with the Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom; with the Department of Electronic Systems, Aalborg University, Denmark; with the Department of Engineering and Architecture, University of Trieste, Trieste, Italy.

Corresponding author: N. Bastianello (email: nicolba@kth.se)

TABLE I
COMPARISON WITH THE STATE OF THE ART IN STOCHASTIC DISTRIBUTED OPTIMIZATION.

Algorithm [Ref.]	variance reduction	grad. steps ÷ comm.	# stored variables†	comm. size‡	# $\nabla f_{i,j}$ evaluations per iteration	assumpt.⋆	convergence
K-GT [9]	✗	$\tau \div 1$	2	$2\|\mathcal{N}_i\|$	1	n.c.	sub-linear, $\propto \sigma^2$
LED [10]	✗	$\tau \div 1$	2	$\|\mathcal{N}_i\|$	1	n.c. s.c.	sub-linear, $\propto \sigma^2$ linear, $\propto \sigma^2$
RandCom [11]	✗	$\left\lceil \frac{1}{p} \right\rceil \div 1$ (in mean)	2	$\|\mathcal{N}_i\|$	1	n.c. s.c.	sub-linear, $\propto \sigma^2$ linear, $\propto \sigma^2$
VR-EXTRA/DIGing [12], GT-VR [13]	✓	$1 \div 1$	3	$2\|\mathcal{N}_i\|$	$\|\mathcal{B}\|$, m_i every $\left\lceil \frac{1}{p} \right\rceil$	n.c. s.c.	sub-linear, $\to 0$ linear, $\to 0$
GT-SAGA [14], [15]	✓	$1 \div 1$	3	$2\|\mathcal{N}_i\|$	1	s.c. n.c.	sub-linear, $\to 0$ linear, $\to 0$
GT-SARAH [16]	✓	$1 \div 1$	3	$2\|\mathcal{N}_i\|$	$\|\mathcal{B}\|$, m_i every τ	n.c.	sub-linear, $\to 0$
GT-SVRG [14]	✓	$1 \div 1$	3	$2\|\mathcal{N}_i\|$	1, m_i every τ	s.c.	linear, $\to 0$
LT-ADMM [this work]	✗	$\tau \div 1$	$\|\mathcal{N}_i\| + 1$	$\|\mathcal{N}_i\|$	$\|\mathcal{B}\|$	n.c.	sub-linear, $\propto \sigma^2$
LT-ADMM-VR [this work]	✓	$\tau \div 1$	$\|\mathcal{N}_i\| + 1$	$\|\mathcal{N}_i\|$	$\|\mathcal{B}\|$, m_i every τ	n.c.	sub-linear, $\to 0$

† number of vectors in $\mathbb{R}^n$ stored by each agent between iterations (disregarding temporary variables)
‡ number of messages sent by each agent during a communication round
⋆ n.c. and s.c. stand for (non)convex and strongly convex

Our main contributions are as follows:

- We propose two algorithms based on distributed ADMM, with one round of communication between multiple local update steps. The first algorithm, Local Training ADMM (LT-ADMM), uses stochastic gradient descent (SGD) for the local updates, while the second algorithm, LT-ADMM with Variance Reduction (LT-ADMM-VR), uses a variance-reduced SGD method [25].
- We establish the convergence properties of LT-ADMM for both nonconvex and convex (not strongly convex) learning problems. In particular, we show almost-sure and mean-squared convergence of LT-ADMM to a neighborhood of the stationary point in the nonconvex case, and to a neighborhood of an optimum in the convex case. The radius of the neighborhood depends on specific properties of the problem and on tunable parameters. We prove that the algorithm achieves a convergence rate of $\mathcal{O}(\frac{1}{K\tau})$, where K is the number of iterations, and τ the number of local training steps.
- For LT-ADMM-VR, we prove *exact convergence* to a stationary point in the nonconvex case, and to an optimum in the convex case. The algorithm has a $\mathcal{O}(\frac{1}{K\tau})$ rate of convergence, which is faster than $\mathcal{O}(\frac{1}{K})$ obtained by related algorithms [16], [15], [13].
- We provide extensive numerical evaluations comparing the proposed algorithms with the state of the art. The results validate the communication efficiency of the algorithms. Indeed, LT-ADMM and LT-ADMM-VR outperform alternative methods when communications are expensive.

A. Comparison with the state of the art

We compare our proposed algorithms – LT-ADMM and LT-ADMM-VR– with the state of the art. The comparison is holistically summarized in Table I.

Decentralized learning algorithms, as first highlighted in the seminal paper [26] on federated learning, face the fundamental challenge of high communication costs. The authors of [26] address this challenge by designing a communication-efficient algorithms which allows the agents to perform multiple local training steps before each round of communication with the coordinator. However, the accuracy of the algorithm in [26] degrades significantly when the agents have heterogeneous data. Since then, alternative federated learning algorithms, *e.g.*, [27], [28], [29], [30], have been designed to employ local training without compromising accuracy. The interest for communication-efficient algorithms has more recently extended to the distributed set-up, where agents rely on peer-to-peer communications rather than on a coordinator as in federated learning. Distributed algorithms with local training have been proposed in [31], [9], [10], [11]. In particular, [31], [9], [10] present gradient tracking methods which allow each agent to perform a fixed number of local updates between each communication round. The algorithm in [11], which builds on [28], instead triggers communication rounds according to a given probability distribution, resulting in a time-varying number of local training steps. Another related algorithm is that of [32], which allows for both multiple consensus and gradient steps in each iteration. However, this algorithm requires a monotonically increasing number of communication rounds in order to guarantee exact convergence. A stochastic version of [32] was then studied in [33]. The algorithm has inexact gradient evaluations, but only allows for multiple consensus steps. An alternative approach to reducing the frequency of communications is to employ event-triggering, see *e.g.* [34], where messages are exchanged only when a certain condition is met.

When the agents employ stochastic gradients in the algorithms of [9], [10], [11], they only converge to a neighborhood of a stationary point, whose radius is proportional to the stochastic gradient variance. Different *variance reduction* tech-

niques are available to improve the accuracy of (centralized) algorithms relying on stochastic gradients, *e.g.*, [35], [25], [36]. Then, these methods have been applied to distributed optimization by combining them with widely used gradient tracking algorithms [12], [13], [14], [15], [16]. The resulting algorithms succeed in guaranteeing exact convergence to a stationary point despite the presence of gradient noise. However, they are not communication-efficient, as they only allow one gradient update per communication round.

We conclude by providing in Table I a summary of the key features of the algorithms discussed above. This table focuses on methods that employ the mechanisms of primary interest in this work – local training and variance reduction. First, we classify them based on whether they use or not variance reduction and local training. For the latter, we report the ratio of gradient steps to communication rounds that characterizes each algorithm, with a ratio of $1 \div 1$ signifying that no local training is used. Notice that LT-ADMM-VR is the only algorithm to use both variance reduction and local training, while the other only use one technique. We then compare the number of variables stored by the agents when they deploy each algorithm (disregarding temporary variables). We notice that the variable storage of LT-ADMM and LT-ADMM-VR, differently from the alternatives, scales with the size of an agent's neighborhood; this is due to the use of distributed ADMM as the foundation of our proposed algorithms [21]. We see that [10], [11], LT-ADMM, and LT-ADMM-VR require one communication per neighbor, while the other methods require two communications per neighbor. We also compare the algorithms by the computational complexity of the gradient estimators they employ, namely, the number of component gradient evaluations needed per local training iteration. The algorithms of [9], [10], [11], [14] use a single data point to estimate the gradient, while [12], [16], LT-ADMM, LT-ADMM-VR can apply mini-batch estimators that use a subset $\mathcal{B}$ of the local data points. The use of mini-batches yields more precise gradient estimates and increased flexibility. However, we remark that the gradient estimators used in [12], [16], [14], LT-ADMM, LT-ADMM-VR require a registry of component gradient evaluations, which needs to be refreshed entirely at fixed intervals. This coincides with the evaluation of a full gradient, and thus requires m_i component gradient evaluations. Finally, we compare the algorithms' convergence. We notice that all algorithms, except for [14], provide (sub-linear) convergence guarantees for convex and nonconvex problems. Additionally, some works show linear convergence for strongly convex problems. We further distinguish between algorithms which achieve exact convergence due to the use of variance reduction, or inexact convergence with an error proportional to the stochastic gradient variance ($\propto \sigma^2$).

Outline: The outline of the paper is as follows. Section II formulates the problem at hand, and presents the proposed algorithms design. Section III analyzes their convergence, and discusses the results. Section IV reports and discusses numerical results comparing the proposed algorithms with the state of the art. Section V presents some concluding remarks.

Notation: ∇f denotes the gradient of a differentiable function f. Given a matrix $A \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denotes the smallest and largest eigenvalue of A, respectively. $A > 0$ represents that matrix A is positive definite. With $n \in \mathbb{N}$, we let $\mathbf{1}_n \in \mathbb{R}^n$ be the vector with all elements equal to 1, $\mathbf{I} \in \mathbb{R}^{n\times n}$ the identity matrix and $\mathbf{0} \in \mathbb{R}^{n\times n}$ the zero matrix. $\langle x, y\rangle = \sum_{h=1}^{n} x_h y_h$ represents the standard inner product of two vectors $x, y \in \mathbb{R}^n$. $\|\cdot\|$ denotes the Euclidean norm of a vector and the matrix-induced 2-norm of a matrix. The proximal of a cost f, with penalty $\rho > 0$, is defined as $\operatorname{prox}_f^\rho(x) = \arg\min_{y\in\mathbb{R}^n} \left\{ f(y) + \frac{1}{2\rho}\|y - x\|^2 \right\}$.

II. Problem Formulation and Algorithm Design

In this section, we formulate the problem at hand and present our proposed algorithms.

A. Problem formulation

We target the solution of (2) over a (undirected) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., N\}$ is the set of N agents, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges (i, j), $i, j \in \mathcal{V}$. In particular, we assume that the local costs $f_i : \mathbb{R}^n \to \mathbb{R}$ are in the *empirical risk minimization* form (1). We make the following assumptions for (2), which are commonly used to support the convergence analysis of distributed learning algorithms (see *e.g.* [9], [10], [11], [12], [15], [16]).

Assumption 1: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a connected, undirected graph.

Assumption 2: The cost function f_i of each agent $i \in \mathcal{V}$ is L-smooth. That is, there exists $l > 0$ such that $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathbb{R}^n$. Moreover, f_i is proper: $f_i(x) > -\infty$, $\forall x \in \mathbb{R}^n$.

When, in the following, we specialize our results to convex scenarios, we resort to the additional assumption below.

Assumption 3: Each function f_i, $i \in \mathcal{V}$, is convex.

B. Algorithm design

We start our design from the distributed ADMM, characterized by the updates[1] [21]:

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

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

where $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ denotes the neighbors of agent i, $\rho > 0$ is a penalty parameter, and $z_{ij} \in \mathbb{R}^n$ are auxiliary variables, one for each neighbor of agent i. This algorithm converges in a wide range of scenarios, and, differently from most gradient tracking approaches, shows robustness to many challenges (asynchrony, limited communications, *etc*) [21], [22]. However, the drawback of (3) is that the agents need to solve an optimization problem to update x_i, which in general does not have a closed-form solution. Therefore, in practice, the agents need to compute an approximate update (3a), which can lead to inexact convergence [22].

In this paper, we modify (3) to use approximate local updates, *while ensuring that this choice does not compromise*

[1] We remark that, more precisely, (3) corresponds to the algorithm in [21] with relaxation parameter $\alpha = 1/2$.

exact convergence. In particular, we allow the agents to use $\tau \in \mathbb{N}$ iterations of a gradient-based solver to approximate (3a), which yields the update:

$$\begin{aligned}
&\phi_{i,k}^0 = x_{i,k}\,,\\
&\phi_{i,k}^{t+1} = \phi_i^t - \left(\gamma g_i(\phi_{i,k}^t) + \beta\Big(\rho\,|\mathcal{N}_i|\,x_{i,k} - \sum_{j\in\mathcal{N}_i} z_{ij,k}\Big)\right),\\
&\qquad t = 0,\dots,\tau-1\,,\\
&x_{i,k+1} = \phi_{i,k}^\tau\,,
\end{aligned} \tag{4}$$

where γ, β are the positive step-sizes, and $g_i(\phi_{i,k}^\ell)$ is an estimate of the gradient ∇f_i. Notice that for efficiency's sake we "freeze" the penalty term $\rho\,|\mathcal{N}_i|\,x_{i,k} - \sum_{j\in\mathcal{N}_i} z_{ij,k}$, and for flexibility we multiply gradient estimate and penalty term by two different step-sizes. The resulting algorithm is a distributed gradient method, with the difference that each communication round (3b) is preceded by $\tau > 1$ local gradient evaluations. This is an application of the *local training* paradigm [10]. We remark that the convergence of the proposed algorithm rests on the initialization $\phi_{i,k}^0 = x_{i,k}$, which enacts a feedback loop on the local training. In general, without this initialization, exact convergence cannot be achieved [22].

The local training (4) requires a local gradient evaluation or at least its estimate. In the following, we introduce two different estimator options. Notice that the gradient of the penalty term, $\rho\,|\mathcal{N}_i|\,x_{i,k} - \sum_{j\in\mathcal{N}_i} z_{ij,k}$, is exactly known (and frozen) and does not need an estimator. The most straightforward idea is to simply employ a local gradient $g_i(\phi) = \nabla f_i(\phi)$. However, in learning applications, the agents may store large datasets ($m_i \gg 1$). Therefore, computing $\nabla f_i(\phi)$ becomes computationally expensive. To remedy this, the agents can instead use *stochastic gradients*, choosing

$$g_i(\phi) = \frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i} \nabla f_{i,h}(\phi)\,, \tag{5}$$

where $\mathcal{B}_i$ are randomly drawn indices from $\{1,\dots,m_i\}$, with $|\mathcal{B}_i| < m_i$. While reducing the computational complexity of the local training iterations, the use of stochastic gradients results in inexact convergence. The idea, therefore, is to employ a gradient estimator based on a *variance reduction* scheme. In particular, we adopt the scheme proposed in [25], characterized by the following procedure. Each agent maintains a table of component gradients $\{\nabla f_{i,h}(r_{i,h,k}^t)\}, h = 1,\dots,m_i$, where $r_{i,h,k}^t$ is the most recent iterate at which the component gradient was evaluated. This table is reset at the beginning of every new local training (that is, for any $k \in \mathbb{N}$ when $t = 0$). Using the table, the agents then estimate their local gradients as

$$\begin{aligned}
g_i\left(\phi_{i,k}^t\right) = &\frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\left(\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \nabla f_{i,h}\left(r_{i,h,k}^t\right)\right)\\
&+ \frac{1}{m_i}\sum_{h=1}^{m_i}\nabla f_{i,h}(r_{i,h,k}^t).
\end{aligned} \tag{6}$$

The gradient estimate is then used to update $\phi_{i,k}^{t+1}$ according to (4); afterwards, the agents update 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. Notice that this update requires a full gradient evaluation at the beginning of each local training, to populate the memory with $\{\nabla f_{i,h}(r_{i,h,k}^0) = \nabla f_{i,h}(\phi_{i,k}^0)\}, h = 1,\dots,m_i$. In the following steps ($t > 0$), each agent only computes $|\mathcal{B}_i|$ component gradients.

Selecting the stochastic gradient estimator (5) yields the proposed algorithm LT-ADMM, while selecting the variance reduction scheme (6) yields the proposed algorithm LT-ADMM-VR. The two methods are reported, with color-coding, in Algorithm 1.

Algorithm 1 LT-ADMM and LT-ADMM-VR

Input: For each node i, initialize $x_{i,0} = z_{ij,0}$, $j \in \mathcal{N}_i$. Set the penalty ρ, the number of local training steps τ, the number of iterations K, and the local step-size γ, β.

for $k = 0, 1, \dots, K-1$ every agent i **do**
  // local training
  $\phi_{i,k}^0 = x_{i,k}$, $r_{i,h,k}^0 = x_{i,k}$, for all $h \in \{1,\dots.m_i\}$
  for $t = 0, 1, \dots, \tau-1$ **do**
    draw the batch $\mathcal{B}_i$ uniformly at random
    update the gradient estimator according to (5)
    update the gradient estimator according to (6)
    update $\phi_{i,k}$ according to (4)
    if $h \in \mathcal{B}_i$ update $r_{i,h,k}^{t+1} = \phi_{i,k}^{t+1}$, else $r_{i,h,k}^{t+1} = r_{i,h,k}^t$
  end for
  $x_{i,k+1} = \phi_{i,k}^\tau$
  // communication
  transmit $z_{ij,k} - 2\rho x_{i,k+1}$ to each neighbor $j \in \mathcal{N}_i$, and receive the corresponding transmissions
  // auxiliary update
  update $z_{ij,k+1}$ according to (3b)
end for

III. Convergence Analysis and Discussion

In this section, we analyze the convergence rate of Algorithm 1 in both nonconvex and convex scenarios. Throughout, we will employ the following metric of convergence

$$\mathcal{D}_k = \mathbb{E}\left[\left\|\nabla F\left(\bar{x}_k\right)\right\|^2 + \frac{1}{\tau}\sum_{t=0}^{\tau-1}\left\|\frac{1}{N}\sum_{i=1}^N \nabla f_i\left(\phi_{i,k}^t\right)\right\|^2\right], \tag{7}$$

where $F(x) = \frac{1}{N}\sum_{i=1}^N f_i(x)$ and $\bar{x}_k = \frac{1}{N}\sum_{i=1}^N x_{i,k}$. If the agents converge to a stationary point of (2), then $\mathcal{D}_k \to 0$. We note that this performance measure is standard in the literature on stochastic gradient methods and distributed optimization [10], [11]. Although it does not, in general, imply almost sure convergence of the sequence $\{\mathcal{D}_k\}_{k=1}^K$, it provides meaningful performance guarantees in expectation. Specifically, if the index K' is selected uniformly at random from $1,\dots,K$, then $\mathbb{E}[\mathcal{D}_{K'}] = \frac{1}{K}\sum_{k=0}^{K-1}\mathcal{D}_k$.

A. Convergence with SGD

We start by characterizing the convergence of Algorithm 1 when the agents use SGD during local training (LT-ADMM).

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2026.3675513

To this end, we make the following standard assumption on the variance of the gradient estimators, see *e.g.* [9], [10].

Assumption 4: For all $\phi \in \mathbb{R}^n$ the gradient estimators $g_i(\phi)$, $i \in \mathcal{V}$, in (5) are unbiased and their variance is bounded by some $\sigma^2 > 0$:

$$\mathbb{E}\left[g_i(\phi) - \nabla f_i(\phi)\right] = 0$$
$$\mathbb{E}\left[\left\|g_i(\phi) - \nabla f_i(\phi)\right\|^2\right] \leq \sigma^2.$$

We are now ready to state our convergence results. All the proofs are deferred to the Appendix, where Appendix I provides a sketch of the proofs, followed by the full proofs. We remark that prior analyses of distributed ADMM based on operator-theoretic approaches [21], [22] are not directly applicable to LT-ADMM, and the convergence proofs must therefore be specifically tailored to this algorithm.

Theorem 1 (Nonconvex case): Let Assumptions 1, 2, and 4 hold. If the local step-sizes satisfy $\gamma \leq \mathcal{O}(\frac{\lambda_l}{L\tau^2}) < \gamma_{\text{sgd}} := \min_{i=1,2,\ldots,6} \bar{\gamma}_i$ (see (10) in Appendix II-A for the precise bound), and $1/(\tau\lambda_u\rho) \leq \beta < 2/(\tau\lambda_u\rho)$, then the output of LT-ADMM satisfies:

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathcal{D}_k \leq \mathcal{O}\left(\frac{F(\bar{x}_0) - F(x^*)}{K\gamma\tau}\right) + \mathcal{O}\left(\gamma\tau\sigma^2\right) + \mathcal{O}\left(\frac{\|\widehat{\mathbf{d}}_0\|^2}{\rho^2 KN}\right) \tag{8}$$

where x^* is a stationary point of (2), λ_u is the largest eigenvalue of the graph $\mathcal{G}$'s Laplacian matrix, λ_l is the smallest nonzero eigenvalue of graph $\mathcal{G}$' Laplacian matrix, and $\|\widehat{\mathbf{d}}_0\|$ is related to the initial conditions (see (26)).

Theorem 1 shows that LT-ADMM converges to a neighborhood of a stationary point x^* as $K \to \infty$. The radius of this neighborhood is proportional to the step-size γ, to the number of local training epochs τ, and to the stochastic gradient variance σ^2. The result can then be particularized to the convex case as follows.

Corollary 1 (Convex case): In the setting of Theorem 1, with the additional Assumption 3, then the output of LT-ADMM converges to a neighborhood of an optimal solution x^* characterized by (8).

Remark 1 (Exact convergence): Clearly, if we employ full gradients (and thus $\sigma = 0$), then these results prove exact convergence to a stationary/optimal point. This verifies that our algorithm design achieves convergence despite the use of approximate local updates.

B. Convergence with variance reduction

The results of the previous section shows that only inexact convergence can be achieved when employing SGD. The following results show how Algorithm 1 achieves exact convergence when using variance reduction (LT-ADMM-VR).

Theorem 2 (Nonconvex case): Let Assumptions 1, 2 hold. If the local step-sizes satisfy $\gamma \leq \mathcal{O}(\frac{\lambda_l}{L\tau^3}) < \gamma_{\text{vr}} := \min_{i=1,7,8,\ldots,15} \bar{\gamma}_i$ (see (11) in Appendix II-A for the precise bound), and $1/(\tau\lambda_u\rho) \leq \beta < 2/(\tau\lambda_u\rho)$, then the output of LT-ADMM-VR converges to a stationary point x^* of (2), and in particular it holds:

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathcal{D}_k \leq \mathcal{O}\left(\frac{F(\bar{x}_0) - F(x^*)}{K\gamma\tau}\right) + \mathcal{O}\left(\frac{\|\widehat{\mathbf{d}}_0\|^2}{\rho^2 K}\right), \tag{9}$$

where x^* is a stationary point of (2), λ_u is the largest eigenvalue of the graph $\mathcal{G}$'s Laplacian matrix, λ_l is the smallest nonzero eigenvalue of graph $\mathcal{G}$' Laplacian matrix, and $\|\widehat{\mathbf{d}}_0\|$ is related to the initial conditions (see (26)).

Corollary 2 (Convex case): In the setting of Theorem 2, with the additional Assumption 3, then the output of LT-ADMM-VR converges to an optimal solution x^*, with rate characterized by (9).

C. Discussion

1) Choice of step-size: The upper bounds to the step-sizes of LT-ADMM and LT-ADMM-VR ((10) and (11) in Appendix II-A), highlight a dependence on several *features of the problem*. In particular, the step-size bounds decrease as the smoothness constant L increases, as is usually the case for gradient-based algorithms. Moreover, the bounds are proportional to the network connectivity, represented by the smallest nonzero eigenvalue of $\mathcal{G}$'s Laplacian (the algebraic connectivity λ_l). Thus, less connected graphs (smaller λ_l) result in smaller bounds. Finally, we remark that the step-size bound for LT-ADMM-VR is proportional to $\frac{m_l}{m_u} = \frac{\min_{i\in\mathcal{V}} m_i}{\max_{i\in\mathcal{V}} m_i}$, where m_i is the number of data points available to agent i (see (1)). This ratio can be viewed as a measure of heterogeneity between the agents. Smaller values of $\frac{m_l}{m_u}$ highlight higher imbalance in the amount of data available to the agents. The step-size bound thus is smaller for less balanced scenarios.

The step-size bounds also depend on the *tunable parameters* τ, the number of local updates, and ρ, the penalty parameter. Therefore, these two parameters can be tuned in order to increase the step-size bounds, which translates in faster convergence.

2) Convergence rates: As discussed in section I-A, various distributed algorithms with variance reduction have been recently proposed, for example, [12], [14] for strongly convex problems, and [16], [15], [13] for nonconvex problems. Focusing on [16], [15], [13], we notice that their convergence rate is $\mathcal{O}(\frac{1}{K})$, while Theorem 2 shows that LT-ADMM-VR has rate of $\mathcal{O}(\frac{1}{\tau K})$. This shows that *employing local training accelerates convergence*.

Similarly to LT-ADMM-VR, [16], [12] also use batch gradient computations, *i.e.*, they update a subset of components to estimate the gradient (see (6)). Interestingly, the step-size upper bound and, hence, the convergence rate in [16], [12] depend on the batch size. On the other hand, our theoretical results are not affected by the batch size, since we use a different variance reduction technique.

We also remark that, as shown in (48) and (70) in the Appendix, better network connectivity (corresponding to larger λ_l) leads to smaller upper bounds on the right-hand side of the convergence results. This indicates that stronger network connectivity accelerates the convergence rate.

Finally, the bound in Theorem 1 also highlights a trade-off: a larger γ accelerates convergence through the term $\mathcal{O}(\frac{1}{K\gamma\tau})$, but it also enlarges the steady-state neighborhood via the term $\mathcal{O}\left(\gamma\tau\sigma^2\right)$. Thus, γ must be tuned to balance convergence speed and steady-state precision – and a similar discussion holds for τ. This trade-off is also explored in the numerical results of section IV-B.

3) Choice of variance reduction mechanism: In variance reduction, we distinguish two main classes of algorithms: those that need to periodically (or randomly) perform a full gradient evaluation (SARAH-type [36]), and those that do not (SAGA-type [25]). In distributed learning, SARAH-type algorithms were proposed in *e.g.*, [16], [14], while SAGA-type algorithms in *e.g.* [14]. Also the proposed LT-ADMM-VR requires a periodic full gradient evaluation, as the agents re-initialize their local gradient memory at the start of each local training (since they set $r^0_{i,h,k} = x_{i,k}$). Clearly, periodically computing a full gradient significantly increases the computational complexity of the algorithm. Thus, one can design a SAGA-type variant of LT-ADMM-VR by removing the gradient memory re-initialization at the start of local training (choosing now $r^0_{i,h,k} = r^\tau_{i,h,k-1}$). This variant is computationally cheaper and shows promising empirical performance, see the results for LT-ADMM-VR v2 in section IV. However, using the outdated gradient memory leads to a more complex theoretical analysis, which we leave for future work.

4) Uncoordinated parameters: In principle, the agents could employ *uncoordinated* parameters, depending on their available resources (*e.g.*, heterogeneous computational capabilities). For instance, different agents could adopt distinct local solvers, numbers of updates (see results in Section IV-B), and batch sizes. Alternatively, they could use the same solver but with step-sizes tailored to the smoothness of their local cost functions.

IV. Numerical Results

In this section we compare the proposed algorithms with the state of the art, applying them to a classification problem with nonconvex regularization, characterized by [10]:

$$f_i(x) = \frac{1}{m_i}\sum_{h=1}^{m_i} \log\left(1+\exp\left(-b_{i,h}a_{i,h}^\top x\right)\right) + \epsilon\sum_{\ell=1}^{n}\frac{[x]_\ell^2}{1+[x]_\ell^2}$$

where $[x]_\ell$ is the ℓ-th component of $x \in \mathbb{R}^n$, and $a_{i,h} \in \mathbb{R}^n$ and $b_{i,h} \in \{-1, 1\}$ are the pairs of feature vector and label. As data we use 8×8 gray-scale images of handwritten digits[2], with pixels normalized to $[0, 1]$; we divide the images in the two classes 'even' and 'odd'. We choose a ring graph with $N = 10$, and have $n = 64$, $m_i = 180$, $\epsilon = 0.01$; the initial conditions are randomly chosen as $x_{i,0} \sim \mathcal{N}(0, 100I_n)$. We use stochastic gradients with a batch of $|\mathcal{B}| = 1$. All results are averaged over 10 Monte Carlo iterations. For the algorithms with local training we select $\tau = 2$. We also tune the step-sizes of all algorithms to ensure best performance. Finally, as performance metric we employ $\|\nabla F(\bar{x}_k)\|^2$, which is zero if the agents have reached a solution of (2)[3]. The simulations are implemented in Python and run on a Windows laptop with Intel i7-1265U and 16GB of RAM.

A. Comparison with the state of the art

We start by comparing LT-ADMM and LT-ADMM-VR with local training algorithms LED [10] and K-GT [9], as well as variance reduction algorithms GT-SARAH [16], and GT-SAGA [14]. We also compare with the alternative version LT-ADMM-VR v2 discussed in section III-C.3. When evaluating the performance, we account for the computation time of each algorithm, rather than the more commonly used iteration count. In particular, letting $t_{\rm G}$ be the time for a component gradient evaluation ($\nabla f_{i,h}$), and $t_{\rm C}$ the time for a round of communications, Table II reports the computation time incurred by each algorithm over the course of τ iterations.

TABLE II
COMPUTATION TIME OF THE ALGORITHMS OVER τ ITERATIONS.

Algorithm [Ref.]	Time
LED [10] & K-GT [9]	$\tau t_{\rm G} + 2t_{\rm C}$
GT-SARAH [16]	$(m_i+\tau-1)t_{\rm G} + 2\tau t_{\rm C}$
GT-SAGA [14]	$\tau\,(t_{\rm G} + 2t_{\rm C})$
LT-ADMM & LT-ADMM-VR v2	$\tau t_{\rm G} + t_{\rm C}$
LT-ADMM-VR	$(m_i+\tau-1)t_{\rm G} + t_{\rm C}$

We start by comparing in Table III the algorithms with variance reduction, in terms of the computation time they require to reach $\|\nabla F(\bar{x}_k)\|^2 < 10^{-7}$, that is, to reach a stationary point up to numerical precision. We see that, depending on the ratio $t_{\rm G}/t_{\rm C}$, their relative speed of convergence changes. When gradient computations are cheaper than communications ($t_{\rm G}/t_{\rm C} = 0.1$), the proposed algorithm LT-ADMM-VR (and LT-ADMM-VR v2) outperform both GT-SARAH and GT-SAGA, since the latter do not employ local training. *This testifies to the benefit of employing local training in scenarios where communications are expensive*. As the ratio $t_{\rm G}/t_{\rm C}$ increases to 1 and then 10, we see how LT-ADMM-VR and GT-SARAH, on the one hand, and LT-ADMM-VR v2 and GT-SAGA, on the other hand, tend to align in terms of performance, as the bulk of the computation time is now due to gradient evaluations, of which the two pairs of algorithms have a similar number (see Table II). Nonetheless, local training still gives an edge to the proposed algorithms.

TABLE III
COMPARISON OF COMPUTATION TIME FOR VARIANCE-REDUCED ALGORITHMS TO REACH $\|\nabla F(\bar{x}_k)\|^2 < 10^{-7}$.

Algorithm [Ref.]	$t_{\rm G}/t_{\rm C} = 0.1$	$t_{\rm G}/t_{\rm C} = 1$	$t_{\rm G}/t_{\rm C} = 10$
GT-SARAH [16]	7.57×10^5	6.33×10^6	6.20×10^7
GT-SAGA [14]	1.55×10^5	2.21×10^5	8.85×10^5
LT-ADMM-VR	6.04×10^5	5.76×10^6	5.73×10^7
LT-ADMM-VR v2	3.81×10^4	9.52×10^4	6.66×10^5

The remaining algorithms (LT-ADMM, LED, K-GT) do not guarantee exact convergence as they do not employ variance reduction. Thus in Table IV we report the asymptotic value of $\|\nabla F(\bar{x}_k)\|^2$ achieved by the different methods. The algorithms have close performance, with the proposed LT-ADMM

TABLE IV
COMPARISON OF ALGORITHMS WITHOUT VARIANCE REDUCTION.

Algorithm [Ref.]	$\|\nabla F(\bar{x}_K)\|^2$
LED [10]	1.29×10^{-3}
K-GT [9]	2.01×10^{-3}
LT-ADMM	1.07×10^{-3}

[2]From `https://doi.org/10.24432/C50P49`.

[3]We choose this metric as it can be defined for all algorithms considered in the comparison, whereas $\mathcal{D}_k$ is defined specifically for Algorithm 1.

outperforming the state of the art slightly, that is, converging closer to a stationary point.

B. Tuning the parameters

In this section we focus on evaluating the impact of the proposed algorithms' tunable parameters. As discussed in section III-C.2, the step-size of LT-ADMM regulates both the speed of convergence and how close it converges to a stationary point. In Table V then we apply different step-sizes and evaluate both the asymptotic value of $\|\nabla F(\bar{x}_k)\|^2$ and the computation time needed for LT-ADMM to reach such value. As expected, a smaller step-size leads to a smaller

TABLE V
PERFORMANCE OF LT-ADMM FOR DIFFERENT γ.

γ	$\|\nabla F(\bar{x}_K)\|^2$	Computation time
0.1	6.01×10^{-5}	4.80×10^4
1	5.79×10^{-4}	3.22×10^4
2	1.07×10^{-3}	2.27×10^2
3	1.72×10^{-3}	2.38×10^2
4	2.68×10^{-3}	7.9×10^1
5	4.43×10^{-3}	4.10×10^1

asymptotic distance from the stationary point, while a larger step-size improves the speed of convergence.

We turn now to LT-ADMM-VR and evaluate its speed of convergence for different numbers of local training epochs τ. Figure 1 reports the computation time to reach $\|\nabla F(\bar{x}_k)\|^2 < 10^{-7}$. Interestingly, it appears that there is a finite optimal

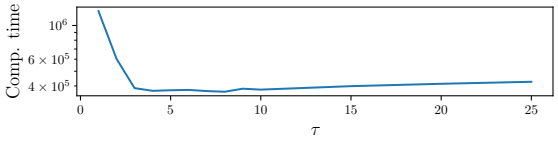

Fig. 1. Computation time for LT-ADMM-VR to reach $\|\nabla F(\bar{x}_k)\|^2 < 10^{-7}$ for different numbers of local training epochs τ.

value ($\tau = 8$), while smaller and larger values lead to slower convergence.

Finally, as discussed in section III-C.4, we can actually choose uncoordinated parameters in LT-ADMM-VR. In Table VI then we test the use of different τ_i, $i \in \{1, \dots, N\}$, for different agents. In particular, we compare the computation

TABLE VI
COMPUTATION TIME OF LT-ADMM-VR WITH UNCOORDINATED NUMBERS OF LOCAL TRAINING EPOCHS.

τ_i	Computation time
$2\ \forall i$	6.04×10^5
$\begin{cases} 2 & i < N/2 \\ 5 & i \geq N/2 \end{cases}$	5.80×10^5
$5\ \forall i$	3.75×10^5

time required to reach $\|\nabla F(\bar{x}_k)\|^2 < 10^{-7}$ in two coordinated scenarios and an uncoordinated scenario where half of the agents are "slow" ($\tau_i = 2$) and half are "fast" ($\tau_i = 5$). Interestingly, the algorithm still converges even with uncoordinated parameters, with the presence of "fast" agents improving the performance.

V. CONCLUDING REMARKS

In this paper, we considered (non)convex distributed learning problems. In particular, to address the challenge of expensive communication, we proposed two communication-efficient algorithms, LT-ADMM and LT-ADMM-VR, that use local training. The algorithms employ SGD and SGD with variance reduction, respectively. We have shown that LT-ADMM converges to a neighborhood of a stationary point, while LT-ADMM-VR converges exactly. We have thoroughly compared our algorithms with the state of the art, both theoretically and in simulations. Future research will focus on analyzing convergence for strongly convex problems and on extending our algorithmic framework to asynchronous scenarios and to the broader class of composite problems, as in [37].

APPENDIX I
PROOF SKETCH OF THE MAIN THEOREMS

A. Proof sketch of Theorem 1

Step 1 (Lemma 1): Reformulate the algorithm into a compact linear dynamical system, in which $\mathbf{h}_k$ contains all nonlinearities. Decompose the system into average and deviation components via a projection matrix $\widehat{\mathbf{Q}}$. Use graph connectivity to show that the linear part of the deviation system

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

is stable ($\|\mathbf{\Delta}\| = 1 - \lambda_l \rho \tau \beta/2 < 1$) when $\beta < \frac{2}{\tau \lambda_u \rho}$ is satisfied.

Step 2 (Lemmas 2, 3): Bound the error from local training steps: the deviation of local states $\mathbf{\Phi}_k^t$ from the global average $\bar{\mathbf{X}}_k$ as in (29).

$$\mathbb{E}\left[\|\widehat{\mathbf{\Phi}}_k\|^2\right] \leq \left(\frac{72\beta\tau^2}{\lambda_l \rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + 4N\tau^2\gamma^2\sigma^2 + 16\tau^3\gamma^2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2].$$

Incorporate this bound into the perturbation term $\widehat{\mathbf{h}}_k$ to derive a recursive inequality for the deviation system as in (37),

$$\begin{aligned}&\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] \\ &\leq (\delta + \frac{c_0}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{c_1}{1-\delta}\mathbb{E}[\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2] \\ &+ \frac{c_2}{1-\delta}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + \frac{c_3}{1-\delta}\sigma^2,\end{aligned}$$

where sufficiently small γ ensuring stability.

Step 3 (Theorem 1): Apply the smoothness inequality to the averaged iterate

$$\bar{x}_{k+1} - x^* = \bar{x}_k - x^* - \frac{\gamma}{N}\sum_{t=0}^{\tau-1}\sum_{i=1}^{N} g_i(\phi_{i,k}^t),$$

show a descent in the global objective $F(\bar{x}_k)$.

$$\begin{aligned}\mathbb{E}[F(\bar{x}_{k+1})] &\leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{2}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\ &- \frac{\gamma}{2}(1 - 2\gamma\tau L)\sum_t \mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] \\ &+ \frac{\gamma L^2}{2N}\mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] + \gamma^2\tau^2 L\sigma^2.\end{aligned}$$

Sum over iterations, pick an appropriate step-size, and use the bounds obtained in Lemmas 2, 3, leading to the convergence result.

B. Proof sketch of Theorem 2

The proof of Theorem 2 follows a structure similar to that of Theorem 1. The key distinction lies in the treatment of the gradient variance. We bound the gradient variance with t_k,

$$\mathbb{E}[\sum_i \sum_t \|g_i\left(\phi_{i,k}^t\right) - \nabla f_i(\phi_{i,k}^t)\|^2] \leq 2L^2\|\widehat{\mathbf{\Phi}}_k\|^2 + 2L^2\mathbb{E}[t_k],$$

using $\mathbb{E}[\|a - \mathbb{E}[a]\|^2] \leq \mathbb{E}[\|a\|^2]$ with $a = \nabla f_{i,h}(\phi_{i,k}^t) - \nabla f_{i,h}(r_{i,h,k}^t)$. And show that t_k can be bound by the deviation bound $\|\widehat{\mathbf{d}}_k\|$ and the global gradient at the average state $\nabla F(\bar{x}_k)$ as in Lemma 5. And we further bound the deviation bound $\|\widehat{\mathbf{d}}_k\|$ with t_k as in Lemma 6. As a result, the final convergence expression no longer depends on a constant σ, and we can have exact convergence.

Appendix II
Preliminary Analysis

In this section we summarize the step-size bounds, and present preliminary results underpinning Theorems 1 and 2.

A. Step-size bounds

The step-size upper bounds for LT-ADMM and LT-ADMM-VR are, respectively:

$$\bar{\gamma}_{\text{sgd}} \coloneqq \min_{i=1,2,\dots,6} \bar{\gamma}_i, \tag{10}$$

$$\bar{\gamma}_{\text{vr}} \coloneqq \min_{i=1,7,8,\dots,15} \bar{\gamma}_i, \tag{11}$$

where:

$$\bar{\gamma}_1 \coloneqq \min\left\{1, \frac{1}{L\tau 2\sqrt{2}}\right\},\ \bar{\gamma}_2 \coloneqq \frac{\sqrt{3}}{8L\tau},$$

$$\bar{\gamma}_3 \coloneqq \frac{3}{8L\tau},\quad \bar{\gamma}_4 \coloneqq \frac{\lambda_l}{\lambda_u L\sqrt{16(1+2\rho^2\|\tilde{\mathbf{L}}\|^2)\tau\|\widehat{\mathbf{V}}^{-1}\|^2\beta_0}},$$

$$\bar{\gamma}_5 \coloneqq \frac{\sqrt{\lambda_l}}{4\tau\sqrt{\lambda_u L}\sqrt[4]{c_4(1+2\rho^2\|\tilde{\mathbf{L}}\|^2)2\|\widehat{\mathbf{V}}^{-1}\|^2}},$$

$$\bar{\gamma}_6 \coloneqq \frac{\sqrt{\lambda_l\beta}}{2\sqrt{\tau\lambda_u L}\sqrt[4]{c_4 6N\|\widehat{\mathbf{V}}^{-1}\|^2}},$$

$$\bar{\gamma}_7 \coloneqq \frac{1}{4\tau L\sqrt{3}}, \bar{\gamma}_8 \coloneqq \sqrt{\frac{m_l}{32L^2m_u}}, \bar{\gamma}_9 \coloneqq \sqrt{\frac{1}{12L^2}},$$

$$\bar{\gamma}_{10} \coloneqq \sqrt{\frac{m_l}{512m_u\tau^3L^2}}, \bar{\gamma}_{11} \coloneqq \sqrt{\frac{3\tau}{8\kappa_3}}, \bar{\gamma}_{12} \coloneqq \frac{3}{8\tau L},$$

$$\bar{\gamma}_{13} \coloneqq \frac{\lambda_l}{\lambda_u\sqrt{8(\kappa_0\tilde{\beta}_0 + 2(\tilde{s}_0+\tilde{s}_1)(\kappa_1+32\tau^2L^2\kappa_0))}},$$

$$\bar{\gamma}_{14} \coloneqq \sqrt[3]{\frac{\lambda_l^2\beta^2}{768L^2N\lambda_u^2\kappa_4}},\quad \bar{\gamma}_{15} \coloneqq \sqrt[3]{\frac{\lambda_l^2\tau}{128\lambda_u^2\kappa_4\kappa_2}}$$

These bounds depend on the following quantities. $d_u = \max\{|\mathcal{N}_i|\}_{i\in\mathcal{V}}$ denotes the maximum agents' degree. $\widehat{\mathbf{V}}$ is defined in (25). λ_u is the largest eigenvalue of the graph $\mathcal{G}$'s Laplacian matrix, λ_l is the smallest nonzero eigenvalue of graph $\mathcal{G}$'s Laplacian matrix. We denote $m_u = \max_{i=1,\dots,N} m_i$ and $m_l = \min_{i=1,\dots,N} m_i$, where m_i is the number of local data points of agent i. Additionally, we have the following definitions, used both in the upper bound above and throughout the convergence analysis:

$$\beta_0 \coloneqq \frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2,$$

$$\tilde{\beta}_0 \coloneqq \frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2,$$

$$c_4 \coloneqq \frac{4L^2}{N}\left(\frac{72\beta\tau}{\lambda_l\rho} + 144\tau^2\beta^2\right)$$

$$\kappa_0 \coloneqq (1+2\rho^2\|\tilde{\mathbf{L}}\|^2)6\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2 + \frac{6L^2}{\beta^2}2\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2,$$

$$\kappa_1 \coloneqq (1+2\rho^2\|\tilde{\mathbf{L}}\|^2)4\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2 + \frac{6L^2}{\beta^2}2\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2,$$

$$\kappa_2 \coloneqq 16\tau^3N\kappa_0 + 2\tilde{s}_2(\kappa_1 + 32\tau^2L^2\kappa_0),$$

$$\tilde{s}_0 \coloneqq \frac{36\beta\tau^2m_u}{\lambda_l\rho} + \frac{144\tau^2m_u}{m_l}\beta^2,$$

$$\tilde{s}_1 \coloneqq \left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\frac{8m_u\tau}{m_l},$$

$$\tilde{s}_2 \coloneqq \frac{16Nm_u\tau^2}{m_l} + \frac{8m_u\tau}{m_l}16\tau^3N,$$

$$\kappa_3 \coloneqq 16\tau^3N(\frac{L^2}{2N} + 2\tau L^3)$$
$$+ 2\tilde{s}_2(2\tau L^3 + 32\tau^2L^2(\frac{L^2}{2N} + 2\tau L^3)),$$

$$\kappa_4 \coloneqq (\frac{L^2}{2N} + 2\tau L^3)\tilde{\beta}_0$$
$$+ 2(\tilde{s}_0+\tilde{s}_1)(2\tau L^3 + 32\tau^2L^2(\frac{L^2}{2N} + 2\tau L^3)),$$

where $\frac{1}{\tau\lambda_u\rho} \leq \beta < \frac{2}{\tau\lambda_u\rho}$.

B. Preliminary transformation

We start by rewriting the algorithm in a compact form. To this end, we introduce the following auxiliary variables: $\mathbf{Z} = \text{col}\{z_{ij}\}_{i,j\in\mathcal{E}}$, $\mathbf{\Phi}_k^t = \text{col}\{\phi_{1,k}^t, \phi_{2,k}^t, ..., \phi_{N,k}^t\}$, $G(\Phi_k^t) = \text{col}\{g_1(\phi_{1,k}^t), g_2(\phi_{2,k}^t), ..., g_N(\phi_{N,k}^t)\}$, $\text{F}(\mathbf{X}) = \text{col}\{f_1(x_1), f_2(x_2), ..., f_N(x_N)\}$, $F(x) = \frac{1}{N}\sum_{i=1}^N f_i(x)$. Define $\mathbf{A} = \text{blk diag}\{\mathbf{1}_{d_i}\}_{i\in\mathcal{V}} \in \mathbb{R}^{M\times N}$, where $d_i = |\mathcal{N}_i|$ is the degree of node i, and $M = \sum_i |\mathcal{N}_i|$. $\mathbf{P} \in \mathbb{R}^{M\times M}$ is a permutation matrix that swaps e_{ij} with e_{ji}. If there is an edge between nodes i, j, then $A^T[i,:]PA[:,j] = 1$, otherwise $A^T[i,:]PA[:,j] = 0$. Therefore $\mathbf{A}^T\mathbf{P}\mathbf{A} = \tilde{\mathbf{A}}$ is the adjacency matrix.

The compact form of LT-ADMM and LT-ADMM-VR then is:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \sum_{t=0}^{\tau-1}(\gamma G(\mathbf{\Phi}_k^t) + \beta(\rho\mathbf{A}^T\mathbf{A}\otimes\mathbf{I}_n\mathbf{X}_k - \mathbf{A}^T\otimes\mathbf{I}_n\mathbf{Z}_k)) \tag{12a}$$

$$\mathbf{Z}_{k+1} = \frac{1}{2}\mathbf{Z}_k - \frac{1}{2}\mathbf{P}\otimes\mathbf{I}_n\mathbf{Z}_k + \rho\mathbf{P}\mathbf{A}\otimes\mathbf{I}_n\mathbf{X}_{k+1}. \tag{12b}$$

Moreover, we introduce the following useful variables

$$\begin{aligned}\mathbf{Y}_k &= \mathbf{A}^T \otimes \mathbf{I}_n \mathbf{Z}_k - \frac{\gamma}{\beta}\nabla F(\bar{\mathbf{X}}_k) - \rho \mathbf{D} \otimes \mathbf{I}_n \mathbf{X}_k \\ \tilde{\mathbf{Y}}_k &= \mathbf{A}^T \mathbf{P} \otimes \mathbf{I}_n \mathbf{Z}_k + \frac{\gamma}{\beta}\nabla \mathrm{F}(\bar{\mathbf{X}}_k) - \rho \mathbf{D} \otimes \mathbf{I}_n \mathbf{X}_k,\end{aligned} \tag{13}$$

where $\bar{\mathbf{X}}_k = \mathbf{1}_N \otimes \bar{x}_k$, with $\bar{x}_k = \frac{1}{N}\mathbf{1}^T\mathbf{X}_k$, and $\mathbf{D} = \mathbf{A}^T\mathbf{A} = \mathrm{diag}\{d_i\}_{i\in\mathcal{V}}$ is the degree matrix.

Multiplying both sides of (12b) by $\mathbf{1}^T$, and using the initial condition, we obtain $\mathbf{1}^T\mathbf{A}^T\mathbf{Z}_{k+1} = \rho\mathbf{1}^T\mathbf{D}\mathbf{X}_{k+1}$ for all $k \in \mathbb{N}$. As a consequence $\bar{\mathbf{Y}}_k = \frac{\gamma}{\beta}\mathbf{1} \otimes \frac{1}{N}\mathbf{1}^T\nabla \mathrm{F}(\bar{\mathbf{X}}_k) = \frac{\gamma}{\beta}\mathbf{1} \otimes \frac{1}{N}\sum_i \nabla f_i(\bar{x}_k)$, and (12) can be further rewritten as

$$\begin{bmatrix}\mathbf{X}_{k+1}\\ \mathbf{Y}_{k+1}\\ \tilde{\mathbf{Y}}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \beta\tau\mathbf{I} & \mathbf{0}\\ \rho\tilde{\mathbf{L}} & \rho\tilde{\mathbf{L}}\beta\tau + \frac{1}{2}\mathbf{I} & -\frac{1}{2}\mathbf{I}\\ \mathbf{0} & -\frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I}\end{bmatrix} \otimes \mathbf{I}_n \begin{bmatrix}\mathbf{X}_k\\ \mathbf{Y}_k\\ \tilde{\mathbf{Y}}_k\end{bmatrix} - \mathbf{h}_k, \tag{14}$$

where

$$\tilde{\mathbf{L}} = \tilde{\mathbf{A}} - \mathbf{D} \tag{15}$$

and

$$\begin{aligned}\mathbf{h}_k = \big[&\gamma\sum_{t=0}^{\tau-1}(\nabla G(\mathbf{\Phi}_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k));\\ &\gamma\rho\tilde{\mathbf{L}} \otimes \mathbf{I}_n \sum_{t=0}^{\tau-1}(\nabla G(\mathbf{\Phi}_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k))\\ &+ \frac{\gamma}{\beta}(\nabla F(\bar{\mathbf{X}}_{k+1}) - \nabla F(\bar{\mathbf{X}}_k));\\ &\frac{\gamma}{\beta}(-\nabla F(\bar{\mathbf{X}}_{k+1}) + \nabla F(\bar{\mathbf{X}}_k))\big].\end{aligned}$$

We remark that (14) can be interpreted as a linear dynamical system, with the non-linearity of the gradients as input in $\mathbf{h}_k$.

C. Deviation from the average

The following lemma illustrates how far the states deviate from the average and will be used later in the proofs of Lemmas 2 and 6.

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

$$\|\bar{\mathbf{X}}_k - \mathbf{X}_k\|^2 \le \frac{18\beta\tau}{\lambda_l\rho}\|\widehat{\mathbf{d}}_k\|^2, \quad \|\bar{\mathbf{Y}}_k - \mathbf{Y}_k\|^2 \le 9\|\widehat{\mathbf{d}}_k\|^2, \tag{16}$$

and

$$\|\widehat{\mathbf{d}}_{k+1}\|^2 \le \delta\|\widehat{\mathbf{d}}_k\|^2 + \frac{1}{1-\delta}\|\widehat{\mathbf{h}}_k\|^2 \tag{17}$$

where $\delta = 1 - \lambda_l\rho\tau\beta/2 < 1$, $\widehat{\mathbf{d}}_k = \widehat{\mathbf{V}}^{-1}\left[\widehat{\mathbf{Q}}^T \otimes \mathbf{I}_n\mathbf{X}_k; \widehat{\mathbf{Q}}^T \otimes \mathbf{I}_n\mathbf{Y}_k; \widehat{\mathbf{Q}}^T \otimes \mathbf{I}_n\tilde{\mathbf{Y}}_k\right]$, $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{V}}^{-1}$ are matrices used to define the deviation term $\widehat{\mathbf{d}}_k$.

Proof: By Assumption 1, graph $\mathcal{G}$ is undirected and connected, hence its Laplacian $-\tilde{\mathbf{L}}$ is symmetric; moreover, it has one zero eigenvalue with eigenvector $\mathbf{1}$, with all eigenvalues being positive. Denote by $\widehat{\mathbf{Q}} \in \mathbf{R}^{N\times(N-1)}$ the matrix satisfying $\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T = \mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\widehat{\mathbf{Q}}^T\widehat{\mathbf{Q}} = \mathbf{I}_{N-1}$ and $\mathbf{1}^T\widehat{\mathbf{Q}} = 0$, $\widehat{\mathbf{Q}}^T\mathbf{1} = 0$. We have that

$$\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}} = \widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}(\mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T) = \widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T. \tag{18}$$

Without loss of generality, in the rest of the proof, we consider $n = 1$, then it holds that $\|\widehat{\mathbf{Q}}^T\mathbf{X}_k\|^2 = \mathbf{X}_k^T\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T\mathbf{X}_k = \left\|\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T\mathbf{X}_k\right\|^2 = \|\mathbf{X}_k - \bar{\mathbf{X}}_k\|^2$, and $\|\widehat{\mathbf{Q}}\| = 1$. Multiplying both sides of (14) by $\widehat{\mathbf{Q}}^T$ and using (18) yields:

$$\begin{bmatrix}\widehat{\mathbf{Q}}^T\mathbf{X}_{k+1}\\ \widehat{\mathbf{Q}}^T\mathbf{Y}_{k+1}\\ \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_{k+1}\end{bmatrix} = (\mathbf{\Theta} \otimes \mathbf{I}_n)\begin{bmatrix}\widehat{\mathbf{Q}}^T\mathbf{X}_k\\ \widehat{\mathbf{Q}}^T\mathbf{Y}_k\\ \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k\end{bmatrix} - \widehat{\mathbf{Q}}^T\mathbf{h}_k \tag{19}$$

where $\mathbf{\Theta} = \begin{bmatrix}\mathbf{I} & \beta\tau\mathbf{I} & \mathbf{0}\\ \rho\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}} & \rho\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}\beta\tau\mathbf{I} + \frac{1}{2}\mathbf{I} & -\frac{1}{2}\mathbf{I}\\ \mathbf{0} & \frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I}\end{bmatrix}$.

The next step is to show that $\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}$ is negative definite by contradiction. Let $x \in R^{N-1}$ be an arbitrary vector, since $-\tilde{\mathbf{L}}$ is the positive semi-definite Laplacian matrix, the quadratic form $x^T\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}x = (\widehat{\mathbf{Q}}x)^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}x \le 0$. Moreover, if $(\widehat{\mathbf{Q}}x)^T(\tilde{\mathbf{A}} - \mathbf{D})\widehat{\mathbf{Q}}x = 0$, we have $\widehat{\mathbf{Q}}x = \mathbf{1}$. Now, the properties of $\widehat{\mathbf{Q}}$ imply that $\widehat{\mathbf{Q}}^T\widehat{\mathbf{Q}}x = x = \widehat{\mathbf{Q}}^T\mathbf{1} = 0$. Therefore, for all non-zero vectors x, the quadratic form $x^T\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}x < 0$, thus $\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}$ is a symmetric negative-definite matrix.

We proceed now to diagonalize each block of $\mathbf{\Theta}$ with $\boldsymbol{\phi} \in \mathbb{R}^{(N-1)\times(N-1)}$:

$$\begin{aligned}\tilde{\mathbf{\Theta}} = \boldsymbol{\phi}\mathbf{\Theta}\boldsymbol{\phi}^T &= \begin{bmatrix}\phi & 0 & 0\\ 0 & \phi & 0\\ 0 & 0 & \phi\end{bmatrix}\mathbf{\Theta}\begin{bmatrix}\phi^T & 0 & 0\\ 0 & \phi^T & 0\\ 0 & 0 & \phi^T\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{I} & \beta\tau & 0\\ \rho\phi\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}\phi^T & \rho\phi\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}\phi^T\beta\tau + \frac{1}{2}\mathbf{I} & -\frac{1}{2}\mathbf{I}\\ 0 & -\frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I}\end{bmatrix}.\end{aligned}$$

We denote $\phi\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}\phi^T = \mathrm{diag}\{\tilde{\lambda}_i\}_{i=2,\dots,N}$, where $\tilde{\lambda}_i < 0$ is the eigenvalue of $\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}}$, $\tilde{\lambda}_{\min} = \lambda_{\min}(\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}})$, and $\tilde{\lambda}_{\max} = \lambda_{\max}(\widehat{\mathbf{Q}}^T\tilde{\mathbf{L}}\widehat{\mathbf{Q}})$, note that $|\tilde{\lambda}_{\max}|$ and $|\tilde{\lambda}_{\min}|$ are the smallest nonzero eigenvalue and the largest eigenvalue of the Laplacian matrix of the graph $\mathcal{G}$, respectively. In the following, we denote $\lambda_l = |\tilde{\lambda}_{\max}|$ and $\lambda_u = |\tilde{\lambda}_{\min}|$. Since each block of $\tilde{\mathbf{\Theta}}$ is a diagonal matrix, there exists a permutation matrix $\mathbf{P}_0$ such that $\mathbf{P}_0\tilde{\mathbf{\Theta}}\mathbf{P}_0^T = \mathbf{P}_0\boldsymbol{\phi}\mathbf{\Theta}\boldsymbol{\phi}^T\mathbf{P}_0^T = \mathrm{blkdiag}\{\mathbf{D}_i\}_{i=2}^N$, where

$$\mathbf{D}_i = \begin{bmatrix}1 & \beta\tau & 0\\ \rho\tilde{\lambda}_i & \rho\tilde{\lambda}_i\beta\tau + 0.5 & -0.5\\ 0 & -0.5 & 0.5\end{bmatrix}. \tag{20}$$

We diagonalize $\mathbf{D}_i = \mathbf{V}_i\mathbf{\Delta}_i\mathbf{V}_i^{-1}$, where $\mathbf{\Delta}_i$ is the diagonal matrix of $\mathbf{D}_i$'s eigenvalues, and

$$\mathbf{V}_i = \begin{bmatrix}-\beta\tau & d_{12} & d_{13}\\ 1 & d_{22} & d_{23}\\ 1 & 1 & 1\end{bmatrix} \tag{21}$$

with $d_{12} = -\beta\tau + ((\beta\tilde{\lambda}_i\rho\tau(\beta\tilde{\lambda}_i\rho\tau + 2))^{0.5})/(\tilde{\lambda}_i\rho)$, $d_{13} = -\beta\tau - ((\beta\tilde{\lambda}_i\rho\tau(\beta\tilde{\lambda}_i\rho\tau + 2))^{0.5})/(\tilde{\lambda}_i\rho)$, $d_{22} = \tilde{\lambda}_i\rho d_{12} - 1$, $d_{23} = \tilde{\lambda}_i\rho d_{13} - 1$. The nonzero eigenvalues λ of $\mathbf{D}_i$, $i = 2,\dots,N$, satisfy $2\lambda^2 + (-2\tilde{\lambda}_i\rho\tau\beta - 4)\lambda + \tilde{\lambda}_i\rho\tau\beta + 2 = 0$, which can be written in the form:

$$2\lambda^2 - 2t\lambda + t = 0 \tag{22}$$

where $t = \tilde{\lambda}_i\rho\tau\beta + 2$. The modulus of the roots of (22) is $1 - \frac{|\tilde{\lambda}|\rho\tau\beta}{2}$ when $-2 < \tilde{\lambda}\rho\tau\beta < 0$. We conclude that we can write

$\mathbf{\Theta} = (\mathbf{P}_0\boldsymbol{\phi})^T\mathbf{V}\mathbf{\Delta}\mathbf{V}^{-1}(\mathbf{P}_0\boldsymbol{\phi})$ where $\mathbf{V} = \text{blkdiag}\{V_i\}_{i=2}^N$ and

$$\mathbf{\Delta} = \text{blkdiag}\{\mathbf{\Delta}_i\}_{i=2}^N. \tag{23}$$

Moreover, $\|\mathbf{\Delta}\| = 1 - \lambda_l\rho\tau\beta/2$ when

$$\lambda_u\rho\tau\beta < 2. \tag{24}$$

Then, left multiplying both sides of (19) by the inverse of

$$\widehat{\mathbf{V}} = (\mathbf{P}_0\boldsymbol{\phi})^T\mathbf{V}, \tag{25}$$

which is given by $\widehat{\mathbf{V}}^{-1} = \mathbf{V}^{-1}(\mathbf{P}_0\boldsymbol{\phi})$, yields

$$\widehat{\mathbf{d}}_{k+1} = \mathbf{\Delta}\widehat{\mathbf{d}}_k - \widehat{\mathbf{h}}_k, \tag{26}$$

where $\widehat{\mathbf{d}}_k = \widehat{\mathbf{V}}^{-1}\left[\widehat{\mathbf{Q}}^T\mathbf{X}_k; \widehat{\mathbf{Q}}^T\mathbf{Y}_k; \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k\right]$, $\widehat{\mathbf{h}}_k = \widehat{\mathbf{V}}^{-1}\,\text{blkdiag}\{\widehat{\mathbf{Q}}^T, \widehat{\mathbf{Q}}^T, \widehat{\mathbf{Q}}^T\} \otimes \mathbf{I}_n\mathbf{h}_k$, and $\left[\widehat{\mathbf{Q}}^T\mathbf{X}_k; \widehat{\mathbf{Q}}^T\mathbf{Y}_k; \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k\right] = \widehat{\mathbf{V}}\widehat{\mathbf{d}}_k = \boldsymbol{\phi}^T\mathbf{P}_0^T\mathbf{V}\widehat{\mathbf{d}}_k = \boldsymbol{\phi}^T\mathbf{P}_0^T\mathbf{V}\mathbf{P}_0\mathbf{P}_0^T\widehat{\mathbf{d}}_k$. As a consequence, from (19) it holds that

$$\begin{aligned}
\begin{bmatrix}\widehat{\mathbf{Q}}^T\mathbf{X}_k\\ \widehat{\mathbf{Q}}^T\mathbf{Y}_k\\ \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k.\end{bmatrix} &= \boldsymbol{\phi}^T\begin{bmatrix}-\beta\tau\mathbf{I} & d_{12}\mathbf{I} & d_{13}\mathbf{I}\\ \mathbf{I} & d_{22}\mathbf{I} & d_{23}\mathbf{I}\\ \mathbf{I} & \mathbf{I} & \mathbf{I}\end{bmatrix}\mathbf{P}_0^T\widehat{\mathbf{d}}_k\\
&= \boldsymbol{\phi}^T\begin{bmatrix}-\beta\tau\mathbf{I}\mathbf{P}_0^T[1] + d_{12}\mathbf{P}_0^T[2] + d_{13}\mathbf{P}_0^T[3]\\ \mathbf{P}_0^T[1] + d_{22}\mathbf{P}_0^T[2] + d_{23}\mathbf{P}_0^T[3]\\ \mathbf{P}_0^T[1] + \mathbf{P}_0^T[2] + \mathbf{P}_0^T[3]\end{bmatrix}\widehat{\mathbf{d}}_k,
\end{aligned}$$

where $\mathbf{P}_0^T[1], \mathbf{P}_0^T[2], \mathbf{P}_0^T[3]$ are the top, middle and bottom blocks of $\mathbf{P}_0^T$ respectively. Moreover, we have $|d_{12}|^2 = |d_{13}|^2 \le \frac{2\beta\tau}{\lambda_l\rho}$, $|d_{22}| = |d_{23}| = 1$. Now, if we let $\beta\tau \le \frac{2}{\lambda_l\rho}$, and using $\|\boldsymbol{\phi}\| = 1$, $\|\mathbf{P}_0^T[i]\| = 1$, $i = 1, 2, 3$, we derive that

$$\begin{aligned}
\|\bar{\mathbf{X}}_k - \mathbf{X}_k\|^2 &= \|\widehat{\mathbf{Q}}^T\mathbf{X}_k\|^2\\
&= \|\boldsymbol{\phi}^T(-\beta\tau\mathbf{I}\mathbf{P}_0^T[1] + d_{12}\mathbf{P}_0^T[2] + d_{13}\mathbf{P}_0^T[3])\widehat{\mathbf{d}}_k\|^2\\
&\le 3(\beta^2\tau^2 + \frac{4\beta\tau}{\lambda_l\rho})\|\widehat{\mathbf{d}}_k\|^2 \le \frac{18\beta\tau}{\lambda_l\rho}\|\widehat{\mathbf{d}}_k\|^2.
\end{aligned}$$

Applying the same manipulations to $\|\bar{\mathbf{Y}}_k - \mathbf{Y}_k\|^2$, we obtain (16) holds. Denote now $\|\widehat{\mathbf{\Phi}}_k\|^2 = \sum_{i=1}^N\sum_{t=0}^{\tau-1}\left\|\phi_{i,k}^t - \bar{x}_k\right\|^2 = \sum_{t=0}^{\tau-1}\left\|\mathbf{\Phi}_k^t - \bar{X}_k\right\|^2$. Using Assumption 2 we derive that

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

Denote $\overline{G}(\mathbf{\Phi}_k^t) = \frac{1}{N}\sum_{i=1}^N g_i(\phi_{i,k}^t)$ and $\overline{\nabla\mathrm{F}}(\mathbf{\Phi}_k^t) = \frac{1}{N}\sum_{i=1}^N\nabla f_i(\phi_{i,k}^t)$, we have

$$\begin{aligned}
&\|\sum_{t=0}^{\tau-1}\overline{G}(\mathbf{\Phi}_k^t)\|^2\\
&= \|\frac{1}{N}\sum_i\sum_t(\nabla f_i\left(\phi_{i,k}^t\right) + g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right))\|^2\\
&\le 2\|\sum_{t=0}^{\tau-1}\overline{\nabla\mathrm{F}}(\mathbf{\Phi}_k^t)\|^2 + \frac{2\tau}{N}\sum_i\sum_t\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right)\|^2
\end{aligned} \tag{27}$$

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

$$\begin{aligned}
\|\mathbf{h}_k\|^2 &\le \gamma^2(1 + 2\rho^2\|\tilde{\mathbf{L}}\|^2)(2\tau L^2\|\widehat{\mathbf{\Phi}}_k\|^2\\
&+ 2\tau\sum_{t=0}^{\tau-1}\|G(\mathbf{\Phi}_k^t) - \nabla F(\mathbf{\Phi}_k^t)\|^2)\\
&+ 6L^2\frac{\gamma^4}{\beta^2}\Bigg(\tau\sum_i\sum_t\left\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i\left(\phi_{i,k}^t\right)\right\|^2\\
&+ N\|\sum_{t=0}^{\tau-1}\overline{\nabla\mathrm{F}}(\mathbf{\Phi}_k^t)\|^2\Bigg)
\end{aligned} \tag{28}$$

Recalling (26), using Jensen's inequality $\|\widehat{\mathbf{d}}_{k+1}\|^2 \le \frac{1}{\|\mathbf{\Delta}\|}\|\mathbf{\Delta}\|^2\|\widehat{\mathbf{d}}_k\|^2 + \frac{1}{1-\|\mathbf{\Delta}\|}\|\widehat{\mathbf{h}}_k\|^2$ yields (17). ■

APPENDIX III
CONVERGENCE ANALYSIS FOR LT-ADMM

A. Key bounds

Lemma 2: Let Assumptions 1, 2, and 4 hold; when $\beta < \frac{2}{\tau\lambda_u\rho}$ and $\gamma \le \bar{\gamma}_1$, we have

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\mathbf{\Phi}}_k\|^2\right] \le \left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + 4N\tau^2\gamma^2\sigma^2\\
+ 16\tau^3N\gamma^2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2].
\end{aligned} \tag{29}$$

Proof: From (12) we can derive that

$$\bar{x}_{k+1} - x^* = \bar{x}_k - x^* - \frac{\gamma}{N}\sum_{t=0}^{\tau-1}\sum_{i=1}^N g_i(\phi_{i,k}^t) \tag{30}$$

and

$$\mathbf{\Phi}_k^{t+1} = \mathbf{\Phi}_k^t + \beta\mathbf{Y}_k - \gamma(G(\mathbf{\Phi}_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k)) \tag{31}$$

Recall that by Assumption 4, $\|G(\mathbf{\Phi}_k^t) - \nabla F(\mathbf{\Phi}_k^t)\|^2 \le N\sigma^2$. Now, suppose that $\tau \ge 2$, using Jensen's inequality we obtain

$$\begin{aligned}
&\mathbb{E}[\left\|\mathbf{\Phi}_k^{t+1} - \bar{\mathbf{X}}_k\right\|^2]\\
&= \mathbb{E}[\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k + \beta\mathbf{Y}_k - \gamma(\nabla\mathrm{F}(\mathbf{\Phi}_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k))\|^2] + N\gamma^2\sigma^2\\
&\le \left(1 + \frac{1}{\tau-1}\right)\mathbb{E}[\left\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\right\|^2] + N\gamma^2\sigma^2 \qquad (32)\\
&+ \tau\mathbb{E}[\|\beta Y_k - \gamma(\nabla\mathrm{F}(\mathbf{\Phi}_k^t) - \nabla\mathrm{F}(\bar{\mathbf{X}}_k))\|^2]\\
&\le \left(1 + \frac{1}{\tau-1} + 2\gamma^2\tau L^2\right)\mathbb{E}[\left\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\right\|^2] \qquad (33)\\
&+ 2\tau\beta^2\mathbb{E}[\|\mathbf{Y}_k\|^2] + \gamma^2N\sigma^2\\
&\le \left(1 + \frac{5/4}{\tau-1}\right)\mathbb{E}[\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\|^2] + \gamma^2N\sigma^2 + 2\tau\beta^2\mathbb{E}[\|\mathbf{Y}_k\|^2], \qquad (34)
\end{aligned}$$

where the last inequality holds when

$$2\gamma^2\tau L^2 \le \frac{1/4}{\tau-1}, \tag{35}$$

which is satisfied by $\gamma \le \bar{\gamma}_1$. Iterating the above inequality for $t = 0, ..., \tau-1$

$$\mathbb{E}[\left\|\mathbf{\Phi}_k^{t+1} - \bar{\mathbf{X}}_k\right\|^2] \le \left(1 + \frac{5/4}{\tau-1}\right)^t\mathbb{E}[\left\|\mathbf{X}_k - \bar{\mathbf{X}}_k\right\|^2] +$$

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2026.3675513

$$+ 2\tau\beta^2 \sum_{l=0}^{t} \left(1 + \frac{5/4}{\tau - 1}\right)^l \mathbb{E}[\left\|\mathbf{Y}_k - \bar{\mathbf{Y}}_k + \bar{\mathbf{Y}}_k\right\|^2]$$
$$+ N\gamma^2\sigma^2 \sum_{l=0}^{t} \left(1 + \frac{5/4}{\tau - 1}\right)^l$$
$$\leq 4\mathbb{E}[\left\|\mathbf{X}_k - \bar{\mathbf{X}}_k\right\|^2] + 4\tau N\gamma^2\sigma^2 + 8\tau^2\beta^2\mathbb{E}[\left\|\mathbf{Y}_k - \bar{\mathbf{Y}}_k + \bar{\mathbf{Y}}_k\right\|^2],$$

where the last inequality holds by $(1 + \frac{a}{\tau-1})^t \leq \exp(\frac{at}{\tau-1}) \leq \exp(a)$ for $t \leq \tau - 1$ and $a = 5/4$.

Summing over t, it follows that

$$\mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] \leq 4\tau\mathbb{E}[\|\mathbf{X}_k - \bar{\mathbf{X}}_k\|^2] + 4N\tau^2\gamma^2\sigma^2 + 16\tau^3\beta^2\mathbb{E}[\|\mathbf{Y}_k - \bar{\mathbf{Y}}_k\|^2] + 16\tau^3 N\gamma^2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2]; \tag{36}$$

moreover, it is easy to verify that (36) also holds for $\tau = 1$. Using (16) concludes the proof. ∎

Lemma 3: Let Assumptions 1, 2, and 4 hold. When $\beta < \frac{2}{\tau\lambda_u\rho}$ and $\gamma \leq \bar{\gamma}_1$,

$$\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] \leq (\delta + \frac{c_0}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{c_1}{1-\delta}\mathbb{E}[\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2] + \frac{c_2}{1-\delta}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + \frac{c_3}{1-\delta}\sigma^2 \tag{37}$$

where

$$\delta \coloneqq 1 - \frac{\lambda_l\rho\tau\beta}{2},$$
$$\beta_0 \coloneqq \frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2$$
$$c_0 \coloneqq \gamma^2(1 + 2\rho^2\|\tilde{\mathbf{L}}\|^2)2\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2\beta_0,$$
$$c_1 \coloneqq \gamma^4\frac{6L^2}{\beta^2}N\|\widehat{\mathbf{V}}^{-1}\|^2,$$
$$c_2 \coloneqq \gamma^4(1 + 2\rho^2\|\tilde{\mathbf{L}}\|^2)L^2 32\tau^4\|\widehat{\mathbf{V}}^{-1}\|^2,$$
$$c_3 \coloneqq \gamma^4(1 + 2\rho^2\|\tilde{\mathbf{L}}\|^2)8L^2N\tau^3\|\widehat{\mathbf{V}}^{-1}\|^2 + \gamma^2(1 + 2\rho^2\|\tilde{\mathbf{L}}\|^2)2\tau^2N\|\widehat{\mathbf{V}}^{-1}\|^2 + 6L^2\frac{\gamma^4}{\beta^2}N\tau^2\|\widehat{\mathbf{V}}^{-1}\|^2.$$

Proof: When $\beta < \frac{2}{\tau\lambda_u\rho}$ and $\gamma \leq \bar{\gamma}_1$, using (28), (29) and Assumption 4, we have

$$\|\widehat{\mathbf{h}}_k\|^2 \leq c_0\|\widehat{\mathbf{d}}_k\|^2 + c_1\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2 + c_2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + c_3\sigma^2,$$

together with (17) we can then derive that (37) holds. ∎

B. Theorem 1

We start our proof by recalling that the following inequality holds for all L-smooth function f, $\forall y, z \in \mathbb{R}^n$ [38]:

$$f(y) \leq f(z) + \langle\nabla f(z), y - z\rangle + (L/2)\|y - z\|^2 \tag{38}$$

Based on (30), substituting $y = \bar{x}_{k+1}$ and $z = \bar{x}_k$ into (38), using Assumption 4, we get

$$\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \gamma\mathbb{E}[\langle\nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t)\rangle] + \frac{\gamma^2 L}{2}\mathbb{E}[\|\frac{1}{N}\sum_t\sum_i g_i(\phi_{i,k}^t)\|^2]$$
$$\leq \mathbb{E}[F(\bar{x}_k)] - \gamma[\langle\nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t)\rangle)] + \gamma^2\tau L\mathbb{E}[\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + \gamma^2\tau^2 L\sigma^2.$$

Using now $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$, we have

$$-\langle\nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t)\rangle = -\frac{\tau}{2}\|\nabla F(\bar{x}_k)\|^2 - \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2 + \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t) - \nabla F(\bar{x}_k)\|^2$$
$$\leq -\frac{\tau}{2}\|\nabla F(\bar{x}_k)\|^2 - \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2 + \frac{L^2}{2N}\|\widehat{\mathbf{\Phi}}_k\|^2.$$

Now, combining the two equations above and using (16), yields

$$\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{2}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] - \frac{\gamma}{2}(1 - 2\gamma\tau L)\sum_t\mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + \frac{\gamma L^2}{2N}\mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] + \gamma^2\tau^2 L\sigma^2.$$

Substituting (29) into the above inequality yields

$$\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] + - \frac{\gamma\tau}{2}\left(1 - 16L^2\tau^2\gamma^2\right)\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] - \frac{\gamma}{2}(1 - 2\gamma L\tau)\sum_t\mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + \frac{\gamma L^2}{2N}\left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \gamma^2\tau^2 L\sigma^2 + 2\tau^2\gamma^3\sigma^2L^2.$$

When $\gamma \leq \min\{\bar{\gamma}_2, \bar{\gamma}_3\}$, then

$$16L^2\tau^2\gamma^2 \leq \frac{3}{4}, \quad 2\gamma L\tau \leq \frac{3}{4}, \tag{39}$$

and we can upper bound the previous inequality by

$$\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{8}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] - \frac{\gamma}{8}\sum_t\mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + \frac{\gamma L^2}{2N}\left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \gamma^2\tau^2 L\sigma^2 + 2\tau^2\gamma^3\sigma^2L^2.$$

Rearranging the above relation, we get

$$\mathcal{D}_k \leq \frac{8}{\gamma\tau}\mathbb{E}\left[\left(\tilde{F}(\bar{x}_k) - \tilde{F}(\bar{x}_{k+1})\right)\right] + \frac{8}{\gamma\tau}\frac{\gamma L^2}{2N}\left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + 8\gamma\tau L\sigma^2 + 16\tau\gamma^2\sigma^2L^2,$$

where $\mathcal{D}_k$ is defined in (7), and $\tilde{F}(\bar{x}_k) = F(\bar{x}_k) - F(x^*)$.

Summing over $k = 0, 1, \ldots, K-1$, using $-\tilde{F}(\bar{x}_k) \leq 0$, it holds that

$$\sum_{k=0}^{K-1} \mathcal{D}_k \leq \frac{8\tilde{F}(\bar{x}^0)}{\gamma\tau} + c_4 \sum_{k=0}^{K-1} \mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + Kc_5\sigma^2 \tag{40}$$

where

$$\begin{aligned} c_4 &:= \frac{4L^2}{N}\left(\frac{72\beta\tau}{\lambda_l\rho} + 144\tau^2\beta^2\right) \\ c_5 &:= 8\gamma\tau L + 16\tau\gamma^2L^2 \end{aligned} \tag{41}$$

We now bound the term $\sum_{k=0}^{K-1}\|\widehat{\mathbf{d}}_k\|^2$. From (37), we have

$$\begin{aligned} &\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] \\ &\leq (\delta + \frac{c_0}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{c_1\tau}{1-\delta}\mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i\left(\phi_{i,k}^t\right)\|^2] \\ &+ \frac{c_2}{1-\delta}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + \frac{c_3}{1-\delta}\sigma^2 \\ &\leq \bar{\delta}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{c_3}{1-\delta}\sigma^2 + R\mathcal{D}_k \end{aligned} \tag{42}$$

where

$$R := \max\{\frac{c_2}{1-\delta}, \frac{c_1\tau^2}{1-\delta}\}. \tag{43}$$

Moreover, letting $\gamma \leq \bar{\gamma}_4$ and $\frac{1}{\tau\lambda_u\rho} \leq \beta < \frac{2}{\tau\lambda_u\rho}$, we have

$$\bar{\delta} = \delta + \frac{c_0}{1-\delta} < 1 - \frac{\lambda_l}{4\lambda_u}. \tag{44}$$

Iterating (42) now gives

$$\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \leq \bar{\delta}^k\mathbb{E}[\|\widehat{\mathbf{d}}_0\|^2] + R\sum_{\ell=0}^{k-1}\bar{\delta}^{k-1-\ell}\mathcal{D}_\ell + \frac{c_3\sigma^2}{1-\bar{\delta}}$$

and summing this inequality over $k = 0, \ldots, K-1$, it follows that

$$\sum_{k=0}^{K-1}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \leq \frac{\|\widehat{\mathbf{d}}_0\|^2}{1-\bar{\delta}} + \frac{R}{1-\bar{\delta}}\sum_{k=0}^{K-1}\mathcal{D}_k + \frac{c_3\sigma^2K}{1-\bar{\delta}}. \tag{45}$$

Substituting (45) into (40) and rearranging, we obtain

$$(1-q_0)\sum_{k=0}^{K-1}\mathcal{D}_k \leq \frac{8\tilde{F}(\bar{x}^0)}{\gamma\tau} + q_1\|\hat{\mathbf{d}}_0\|^2 + Kq_2\sigma^2,$$

where

$$q_0 := \frac{c_4R}{1-\bar{\delta}}, \quad q_1 := \frac{c_4}{1-\bar{\delta}} \quad q_2 := \frac{c_4c_3}{1-\bar{\delta}} + c_5. \tag{46}$$

Since $1-\bar{\delta} \geq \frac{\lambda_l}{4\lambda_u}$ and $1-\delta \geq \frac{\lambda_l}{2\lambda_u}$, when $\gamma \leq \min\{1, \bar{\gamma}_5, \bar{\gamma}_6\}$, we have

$$q_0 \leq \frac{1}{2}, \tag{47}$$

and it follows that

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathcal{D}_k \leq \frac{16\tilde{F}(\bar{x}^0)}{\gamma\tau K} + \frac{2q_1}{K}\|\hat{\mathbf{d}}_0\|^2 + 2q_2\sigma^2. \tag{48}$$

By collecting all step-size conditions, if the step-size $\gamma < \bar{\gamma}_{\text{sgd}} := \min_{i=1,2,\ldots,6}\bar{\gamma}_i$, then (48) holds, the states $\{\mathbf{X}_k\}$ generated by LT-ADMM converge to the neighborhood of the stationary point, concluding the proof.

APPENDIX IV
CONVERGENCE ANALYSIS FOR LT-ADMM-VR

A. Key bounds

We start by deriving an upper bound for the variance of the gradient estimator $\mathbb{E}[\|g_i(\phi_{i,k}^t) - \nabla f_i(\phi_{i,k}^t)\|^2]$. Define t_i^k as the averaged consensus gap of the auxiliary variables of $\{r_{i,h,k}^k\}_{h=1}^{m_i}$ at node i:

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

By the updates of $g_i(\phi_{i,k}^t)$ in LT-ADMM-VR,

$$\begin{aligned} &\mathbb{E}\left[\|g_i(\phi_{i,k}^t) - \nabla f_i\left(\phi_{i,k}^t\right)\|^2\right] \\ &= \mathbb{E}[\|\frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\nabla f_{i,h}\left(r_{i,h,k}^t\right) \\ &- (\nabla f_i\left(\phi_{i,k}^t\right) - \frac{1}{m_i}\sum_{h=1}^{m_j}\nabla f_{i,h}(r_{i,h,k}^k))\|^2] \\ &\leq \mathbb{E}\left[\|\frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\nabla f_{i,h}\left(r_{i,h,k}^t\right)\|^2\right] \\ &\leq \mathbb{E}\left[\frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\left\|\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \nabla f_{i,h}\left(r_{i,h,k}^t\right)\right\|^2\right] \\ &= \frac{1}{|\mathcal{B}_i|}\sum_{h\in\mathcal{B}_i}\mathbb{E}\left[\left\|\nabla f_{i,h}\left(\phi_{i,k}^t\right) - \nabla f_{i,h}\left(r_{i,h,k}^t\right)\right\|^2\right] \\ &\leq 2L^2\left\|\phi_{i,k}^t - \bar{x}_k\right\|^2 + 2L^2\mathbb{E}[t_{i,k}^t], \end{aligned}$$

where in the first inequality we use $\mathbb{E}[\|a-\mathbb{E}[a]\|^2] \leq \mathbb{E}[\|a\|^2]$ with $a = \nabla f_{i,h}(\phi_{i,k}^t) - \nabla f_{i,h}(r_{i,h,k}^t)$; and in the second inequality we use the smoothness of the costs. As a consequence, we have

$$\mathbb{E}[\sum_i\sum_t\|g_i\left(\phi_{i,k}^t\right) - \nabla f_i(\phi_{i,k}^t)\|^2] \leq 2L^2\|\widehat{\mathbf{\Phi}}_k\|^2 + 2L^2\mathbb{E}[t_k]. \tag{49}$$

Lemma 4: Let Assumptions 1 and 2 hold; when $\beta\tau \leq \frac{2}{\lambda_u\rho}$ and $\gamma \leq \bar{\gamma}_7$, we have

$$\begin{aligned} \mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] &\leq \left(\frac{72\beta\tau^2}{\lambda_l\rho} + 144\tau^3\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\ &+ 16\tau^3\gamma^2N\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] + 32\tau^2\gamma^2L^2\mathbb{E}[t_k]. \end{aligned} \tag{50}$$

Proof: Suppose that $\tau \geq 2$, using (31) and (49) we have

$$\begin{aligned}
&\mathbb{E}[\left\|\Phi_k^{t+1} - \bar{\mathbf{X}}_k\right\|^2] \\
&= \mathbb{E}[\|\Phi_k^t - \bar{\mathbf{X}}_k + \beta\mathbf{Y}_k - \gamma(G(\Phi_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k))\|^2] \\
&\leq \left(1 + \frac{1}{\tau - 1}\right) \mathbb{E}[\left\|\Phi_k^t - \bar{\mathbf{X}}_k\right\|^2] \\
&+ \tau \mathbb{E}[\|\beta\mathbf{Y}_k - \gamma(G(\Phi_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k))\|^2] \\
&\leq \left(1 + \frac{1}{\tau - 1} + 4\gamma^2\tau(2L^2 + L^2)\right) \mathbb{E}[\left\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\right\|^2] \\
&+ 2\tau\beta^2 \mathbb{E}[\|\mathbf{Y}_k\|^2] + 4\tau\gamma^2(2L^2 \mathbb{E}[t_k^t]) \\
&\leq \left(1 + \frac{5/4}{\tau - 1}\right) \mathbb{E}[\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\|^2] \\
&+ 8\tau\gamma^2 L^2 \mathbb{E}[t_k^t] + 2\tau\beta^2 \mathbb{E}[\|\mathbf{Y}_k\|^2],
\end{aligned} \tag{51}$$

where the last inequality holds when

$$4\gamma^2\tau(2L^2 + L^2) \leq \frac{1/4}{\tau - 1}, \tag{52}$$

which can be satisfied when $\gamma \leq \bar{\gamma}_7$. Similar to Lemma 2, we can derive that (50) holds, which concludes the proof. ∎

The following lemma provides the bound on t_k.

Lemma 5: Let $\{t_k\}$ be the iterates generated by LT-ADMM-VR. If $\beta\tau \leq \frac{2}{\lambda_l \rho}$ and $\gamma \leq \min\{\bar{\gamma}_8, \bar{\gamma}_9, \bar{\gamma}_{10}\}$, we have for all $k \in \mathbb{N}$:

$$\mathbb{E}[t_k] \leq 2(s_0 + s_1)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + 2s_2 \mathbb{E}[\|\nabla F(\bar{x}_k)\|^2], \tag{53}$$

where

$$\begin{aligned}
s_0 &= \frac{36\beta\tau^2 m_u}{\lambda_l \rho} + \frac{144\tau^2 m_u}{m_l}\beta^2 \\
s_1 &= \left(\frac{72\beta\tau^2}{\lambda_l \rho} + 144\tau^3\beta^2\right)\frac{8m_u\tau}{m_l} \\
s_2 &= \frac{16N\gamma^2 m_u \tau^2}{m_l} + \frac{8m_u\tau}{m_l} 16\tau^3\gamma^2 N.
\end{aligned} \tag{54}$$

Proof: From Algorithm 1, $\forall k, r_{i,h,k}^{t+1} = r_{i,h,k}^k$ with probability $1 - \frac{1}{m_i}$ and $r_{i,h,k}^{t+1} = \phi_{i,k}^{t+1}$ with probability $\frac{1}{m_i}$, therefore,

$$\begin{aligned}
&\mathbb{E}[t_k^{t+1}] = \frac{1}{m_i}\sum_{h=1}^{m_i} \mathbb{E}[\|\mathbf{r}_{h,k}^{t+1} - \bar{\mathbf{X}}_k\|^2] \\
&= \frac{1}{m_i}\sum_{h=1}^{m_i} \mathbb{E}[(1 - \frac{1}{m_i})\|\mathbf{r}_{h,k}^t - \bar{\mathbf{X}}_k\|^2 + \frac{1}{m_i}\|\Phi_k^{t+1} - \bar{\mathbf{X}}_k\|^2] \\
&= \left(1 - \frac{1}{m_i}\right)\frac{1}{m_i}\sum_{h=1}^{m_i} \mathbb{E}\left[\|\mathbf{r}_{h,k}^t - \bar{\mathbf{X}}_k\|^2\right] \\
&+ \frac{1}{m_i}\mathbb{E}[\|\Phi_k^{t+1} - \bar{\mathbf{X}}_k\|^2].
\end{aligned}$$

Denote $q_k^t = \beta\mathbf{Y}_k - \gamma(G(\Phi_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k))$, we have $\left\|\Phi_k^{t+1} - \bar{\mathbf{X}}_k\right\|^2 = \left\|\Phi_k^{t+1} - \Phi_k^t + \Phi_k^t - \bar{\mathbf{X}}_k\right\|^2 \leq 2\|\Phi_k^t - \bar{\mathbf{X}}_k\|^2 + 2\|q_k^t\|^2$, and

$$\begin{aligned}
&\mathbb{E}[\|q_k^t\|^2] \\
&\leq 2\gamma^2 \mathbb{E}[\|G(\Phi_k^t) - \nabla \mathrm{F}(\bar{\mathbf{X}}_k)\|^2] + 2\beta^2 \mathbb{E}[\|\mathbf{Y}_k\|^2] \\
&\leq 4\gamma^2(2L^2 + L^2)\mathbb{E}[\left\|\Phi_k^t - \bar{\mathbf{X}}_k\right\|^2] + 2\beta^2 \mathbb{E}[\|\mathbf{Y}_k\|^2] \\
&+ 4\gamma^2(2L^2 \mathbb{E}[t_k^t]) \\
&\leq 12\gamma^2 L^2 \left\|\Phi_k^t - \bar{\mathbf{X}}_k\right\|^2 + 4\gamma^2 N\|\nabla F(\bar{x}_k)\|^2 \\
&+ 4\beta^2\|\mathbf{Y}_k - \bar{\mathbf{Y}}_k\|^2 + 8\gamma^2 L^2 \mathbb{E}[t_k^t],
\end{aligned}$$

it follows that

$$\begin{aligned}
\mathbb{E}[t_k^{t+1}] &= (1 - \frac{1}{m_i})\frac{1}{m_i}\sum_{h=1}^{m_i} \|\mathbf{r}_{h,k}^t - \bar{\mathbf{X}}_k\|^2 \\
&+ \frac{1}{m_i}\|\Phi_k^{t+1} - \bar{\mathbf{X}}_k\|^2 \\
&\leq (1 - \frac{1}{m_i})t_k^t + \frac{1}{m_i}\left(2\|\Phi_k^t - \bar{\mathbf{X}}_k\|^2 + 2\|q_k^t\|^2\right) \\
&\leq \left(1 - \frac{1}{m_u} + \frac{16\gamma^2 L^2}{m_l}\right)\mathbb{E}[t_k^t] \qquad (56) \\
&+ \left(\frac{2}{m_l} + \frac{24\gamma^2 L^2}{m_l}\right)\mathbb{E}[\|\Phi_k^t - \bar{\mathbf{X}}_k\|^2 \\
&+ \frac{72}{m_l}\beta^2\|\widehat{\mathbf{d}}_k\|^2 + \frac{8N}{m_l}\gamma^2\|\nabla F(\bar{\mathbf{X}}_k)\|^2 \\
&\leq \left(1 - \frac{1}{2m_u}\right)\mathbb{E}[t_k^t] + \frac{4}{m_l}\mathbb{E}[\|\Phi_k^t - \bar{\mathbf{X}}_k\|^2 \qquad (57) \\
&+ \frac{72}{m_l}\beta^2\|\widehat{\mathbf{d}}_k\|^2 + \frac{8N}{m_l}\gamma^2\|\nabla F(\bar{\mathbf{X}}_k)\|^2 \qquad (58)
\end{aligned}$$

where the last inequality holds when

$$\frac{16\gamma^2 L^2}{m_l} < \frac{1}{2m_u}, \quad 24\gamma^2 L^2 < 2. \tag{59}$$

Iterating (58) for $t = 0, ..., \tau - 1$ then yields:

$$\begin{aligned}
&\mathbb{E}[t_k^t] \leq (1 - \frac{1}{2m_u})^t \mathbb{E}[\|\mathbf{X}_k - \bar{\mathbf{X}}_k\|^2] \\
&+ \frac{72}{m_l}\beta^2 \sum_{l=0}^{t-1}(1 - \frac{1}{2m_u})^{t-1-l}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\
&+ \frac{8N\gamma^2}{m_l}\sum_{l=0}^{t-1}(1 - \frac{1}{2m_u})^l\|\nabla F(\bar{x}_k)\|^2 \\
&+ \frac{4}{m_l}\sum_{l=0}^{t-1}(1 - \frac{1}{2m_u})^{t-1-l}\mathbb{E}[\|\Phi_k^l - \bar{\mathbf{X}}_k\|^2 \\
&\leq \frac{36\beta\tau m_u}{\lambda_l \rho}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{16N\gamma^2 m_u\tau}{m_l}\|\nabla F(\bar{x}_k)\|^2 \\
&+ \frac{8m_u\tau}{m_l}\sum_{l=0}^{t-1}\mathbb{E}[\|\Phi_k^l - \bar{\mathbf{X}}_k\|^2 + \frac{144m_u\tau}{m_l}\beta^2\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2].
\end{aligned}$$

Summing the above relation over $t = 0, 1, ..., \tau - 1$ we get:

$$\begin{aligned}
&\mathbb{E}[t_k] \\
&\leq \left(\frac{36\beta\tau^2 m_u}{\lambda_l \rho} + \frac{144\tau^2 m_u}{m_l}\beta^2\right)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \frac{8m_u\tau}{m_l}\|\widehat{\mathbf{\Phi}}_k\|^2 \\
&+ \frac{16N\gamma^2 m_u\tau^2}{m_l}\|\nabla F(\bar{x}_k)\|^2,
\end{aligned}$$

and using (50) then yields

$$\begin{aligned}
\mathbb{E}[t_k] &\leq (s_0 + s_1)\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + s_2 \mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\
&+ \frac{8m_u\tau}{m_l} 32\tau^2\gamma^2 L^2 \mathbb{E}[t_k],
\end{aligned}$$

where s_0, s_1 and s_2 are defined in (54). Letting

$$\frac{8m_u\tau}{m_l} 32\tau^2\gamma^2 L^2 < \frac{1}{2}, \tag{60}$$

and thus (53) holds. The conditions (59) and (60) hold when $\gamma \leq \min\{\bar{\gamma}_8, \bar{\gamma}_9, \bar{\gamma}_{10}\}$. ■

Lemma 6: Let Assumptions 1 and 2 hold; when $\beta\tau \leq \frac{2}{\lambda_u \rho}$ and $\gamma < \min\{\bar{\gamma}_1, \bar{\gamma}_7\}$, it holds that $\forall k \geq 0$

$$\begin{aligned}&\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] \leq (\delta + \frac{\tilde{q}_0}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\ &+ \frac{\tilde{q}_1}{1-\delta}\mathbb{E}[\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2] + \frac{\tilde{q}_2}{1-\delta}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2],\end{aligned} \tag{61}$$

where

$$\tilde{\beta}_0 \coloneqq \frac{72\beta\tau^2}{\lambda_l \rho} + 144\tau^3\beta^2,$$

$$\tilde{c}_0 \coloneqq \gamma^2(1+2\rho^2\|\tilde{\mathbf{L}}\|^2)6\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2 + 6L^2\frac{\gamma^4}{\beta^2}2\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2,$$

$$\tilde{c}_1 \coloneqq \gamma^2(1+2\rho^2\|\tilde{\mathbf{L}}\|^2)4\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2 + 6L^2\frac{\gamma^4}{\beta^2}2\tau L^2\|\widehat{\mathbf{V}}^{-1}\|^2,$$

$$\tilde{c}_2 \coloneqq 6L^2\frac{\gamma^4}{\beta^2}N\|\widehat{\mathbf{V}}^{-1}\|^2,$$

$$\tilde{q}_0 \coloneqq \tilde{c}_0\tilde{\beta}_0 + 2(s_0+s_1)(\tilde{c}_1 + 32\tau^2\gamma^2L^2\tilde{c}_0),$$

$$\tilde{q}_1 \coloneqq 6L^2\frac{\gamma^4}{\beta^2}N,$$

$$\tilde{q}_2 \coloneqq 16\tau^3\gamma^2N\tilde{c}_0 + 2s_2(\tilde{c}_1 + 32\tau^2\gamma^2L^2\tilde{c}_0).$$

Proof: When $\beta\tau \leq \frac{2}{\lambda_u\rho}$ and $\gamma < \min\{\bar{\gamma}_1, \bar{\gamma}_7\}$, substituting (49) and (50) into (28) then yields

$$\begin{aligned}&\|\mathbf{h}_k\|^2 \leq \gamma^2(1+2\rho^2\|\tilde{\mathbf{L}}\|^2)(6\tau L^2\|\widehat{\mathbf{\Phi}}_k\|^2 + 4\tau L^2\mathbb{E}[t_k]) \\ &+ 6L^2\frac{\gamma^4}{\beta^2}\left(4\tau L^2\|\widehat{\mathbf{\Phi}}_k\|^2 + 4\tau L^2\mathbb{E}[t_k] + N\|\sum_{t=0}^{\tau-1}\overline{\nabla \mathrm{F}}(\Phi_k^t)\|^2\right)\end{aligned}$$

and

$$\begin{aligned}&\|\widehat{\mathbf{h}}_k\|^2 \leq \tilde{c}_0\|\widehat{\mathbf{\Phi}}_k\|^2 + \tilde{c}_1 t_k + \tilde{c}_2\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2 \\ &\leq \tilde{q}_0\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \tilde{q}_1\mathbb{E}[\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2] + \tilde{q}_2\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2],\end{aligned}$$

together with (17), it proves that (61) holds. ■

B. Theorem 2

Based on (30), substituting $y = \bar{x}_{k+1}$ and $z = \bar{x}_k$ into (38) and using (49), we get

$$\begin{aligned}&\mathbb{E}[F(\bar{x}_{k+1})] \\ &\leq \mathbb{E}[F(\bar{x}_k)] - \gamma\mathbb{E}[\langle \nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t)\rangle] \\ &+ \frac{\gamma^2 L}{2}\mathbb{E}[\|\frac{1}{N}\sum_t\sum_i g_i(\phi_{i,k}^t)\|^2] \\ &\leq \mathbb{E}[F(\bar{x}_k)] - \gamma[\langle \nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t))\rangle] \\ &+ \gamma^2\tau L\mathbb{E}[\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + 2\gamma^2\tau L^3(\|\widehat{\mathbf{\Phi}}_k\|^2 + \mathbb{E}[t_k]).\end{aligned}$$

Using now $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$, we have

$$\begin{aligned}&-\langle \nabla F(\bar{x}_k), \frac{1}{N}\sum_t\sum_i \nabla f_i(\phi_{i,k}^t)\rangle \\ &= -\frac{\tau}{2}\|\nabla F(\bar{x}_k)\|^2 - \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2 \\ &+ \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t) - \nabla F(\bar{x}_k)\|^2 \\ &\leq -\frac{\tau}{2}\|\nabla F(\bar{x}_k)\|^2 - \frac{1}{2}\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2 + \frac{L^2}{2N}\|\widehat{\mathbf{\Phi}}_k\|^2.\end{aligned}$$

Now, combining the two equations above and using (16), yields

$$\begin{aligned}&\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{2}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\ &- \frac{\gamma}{2}(1-2\gamma\tau L)\sum_t\mathbb{E}[\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] \\ &+ \frac{\gamma L^2}{2N}\mathbb{E}[\|\widehat{\mathbf{\Phi}}_k\|^2] + 2\gamma^2\tau L^3(\|\widehat{\mathbf{\Phi}}_k\|^2 + \mathbb{E}[t_k]).\end{aligned}$$

Using (50) and (53) we have

$$\begin{aligned}&\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{2}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\ &- \frac{\gamma}{2}(1-2\gamma\tau L)\mathbb{E}[\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] \\ &+ \tilde{q}_3\mathbb{E}[\|\nabla F(x)\|^2] + \tilde{q}_4\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2],\end{aligned}$$

where

$$\begin{aligned}\tilde{q}_3 &\coloneqq 16\tau^3\gamma^2N(\frac{\gamma L^2}{2N} + 2\gamma^2\tau L^3) \\ &+ 2s_2(2\gamma^2\tau L^3 + 32\tau^2\gamma^2L^2(\frac{\gamma L^2}{2N} + 2\gamma^2\tau L^3)), \\ \tilde{q}_4 &\coloneqq (\frac{\gamma L^2}{2N} + 2\gamma^2\tau L^3)\tilde{\beta}_0 \\ &+ 2(s_0+s_1)(2\gamma^2\tau L^3 + 32\tau^2\gamma^2L^2(\frac{\gamma L^2}{2N} + 2\gamma^2\tau L^3)).\end{aligned}$$

Letting $\gamma \leq \min\{1, \bar{\gamma}_{11}, \bar{\gamma}_{12}\}$, then

$$\tilde{q}_3 \leq \frac{3\gamma\tau}{8}, \quad 2\gamma\tau L \leq \frac{3}{4}, \tag{62}$$

and we can upper bound the previous inequality by

$$\begin{aligned}&\mathbb{E}[F(\bar{x}_{k+1})] \leq \mathbb{E}[F(\bar{x}_k)] - \frac{\gamma\tau}{8}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\ &- \frac{\gamma}{8}\mathbb{E}[\sum_t\|\frac{1}{N}\sum_i \nabla f_i(\phi_{i,k}^t)\|^2] + \tilde{q}_4\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2].\end{aligned}$$

Rearranging we get

$$\mathcal{D}_k \leq \frac{8}{\gamma\tau}(\mathbb{E}[\tilde{F}(\bar{x}_k)] - \mathbb{E}[\tilde{F}(\bar{x}_{k+1})]) + \frac{8}{\gamma\tau}\tilde{q}_4\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2], \tag{63}$$

where $\mathcal{D}_k$ is defined in (7), and $\tilde{F}(\bar{x}_k) = F(\bar{x}_k) - F(x^*)$. Summing (63) over $k = 0, 1, \ldots, K-1$ and using $-\tilde{F}(\bar{x}_k) \leq 0$, it holds that

$$\sum_{r=0}^{K-1}\mathcal{D}_k \leq \frac{8\tilde{F}(\bar{x}_0)}{\gamma\tau} + \frac{8\tilde{q}_4}{\gamma\tau}\sum_{k=0}^{K-1}\|\mathbb{E}[\hat{\mathbf{d}}_k]\|^2]. \tag{64}$$

According to (61), we derive that $\forall k \geq 0$,

$$\begin{aligned}&\mathbb{E}[\|\widehat{\mathbf{d}}_{k+1}\|^2] \leq (\delta + \frac{\tilde{q}_0}{1-\delta})\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \\ &+ \frac{\tilde{q}_1}{1-\delta}\mathbb{E}[\|\sum_t \overline{\nabla F}(\mathbf{\Phi}_k^t)\|^2] + \frac{\tilde{q}_2}{1-\delta}\mathbb{E}[\|\nabla F(\bar{x}_k)\|^2] \\ &\leq \tilde{\delta}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] + \tilde{R}\mathcal{D}_k,\end{aligned} \tag{65}$$

where

$$\tilde{R} = \max\left\{\frac{\tilde{q}_1\tau^2}{1-\delta}, \frac{\tilde{q}_2}{1-\delta}\right\}.$$

Letting $\gamma \leq \min\{\bar{\gamma}_1, \bar{\gamma}_{13}\}$ and $\frac{1}{\tau\lambda_u\rho} \leq \beta < \frac{2}{\tau\lambda_u\rho}$, then

$$\tilde{\delta} = \delta + \frac{\tilde{q}_0}{1-\delta} \leq 1 - \frac{\lambda_l}{4\lambda_u}. \tag{66}$$

Iterating (65) yields $\forall k \geq 1$, $\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \leq \tilde{\delta}^k\mathbb{E}[\|\widehat{\mathbf{d}}_0\|^2] + \tilde{R}\sum_{\ell=0}^{k-1}\tilde{\delta}^{k-1-\ell}\mathcal{D}_\ell$, and summing over $k = 0, \ldots, K-1$ it holds that

$$\sum_{k=0}^{K-1}\mathbb{E}[\|\widehat{\mathbf{d}}_k\|^2] \leq \frac{1}{1-\tilde{\delta}}\|\widehat{\mathbf{d}}_0\|^2 + \sum_{k=0}^{K-1}\frac{\tilde{R}}{1-\tilde{\delta}}\mathcal{D}_k. \tag{67}$$

Substituting (67) into (64), and rearranging, we obtain

$$\sum_{r=0}^{K-1}\mathcal{D}_k \leq \frac{8\tilde{F}(\bar{x}_0)}{\gamma\tau} + \frac{8\tilde{q}_4}{\gamma\tau}\sum_{k=0}^{K-1}\|\mathbb{E}[\hat{\mathbf{d}}_k\|^2]. \tag{68}$$

$$(1 - \frac{\tilde{R}}{1-\tilde{\delta}}\frac{8\tilde{q}_4}{\gamma\tau})\sum_{k=0}^{K-1}\mathcal{D}_k \leq \frac{8\tilde{F}(\bar{x}_0)}{\gamma\tau} + \frac{8\tilde{q}_4}{\gamma\tau(1-\tilde{\delta})}\|\widehat{\mathbf{d}}_0\|^2.$$

Since $1 - \tilde{\delta} \geq \frac{\lambda_l}{4\lambda_u}$, let $\gamma \leq \min\{\bar{\gamma}_{14}, \bar{\gamma}_{15}\}$, then

$$\frac{\tilde{R}}{1-\tilde{\delta}}\frac{8\tilde{q}_4}{\gamma\tau} \leq \frac{1}{2}, \tag{69}$$

and therefore it follows that

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathcal{D}_k \leq \frac{16\tilde{F}(\bar{x}_0)}{K\gamma\tau} + \frac{16\tilde{q}_4}{K\gamma\tau(1-\tilde{\delta})}\|\widehat{\mathbf{d}}_0\|^2. \tag{70}$$

By collecting all step-size conditions, if the step-size γ satisfies $\gamma \leq \min\bar{\gamma}_{i=1,7,8,\ldots,15}$, then the states $\{\mathbf{X}_k\}$ generated by LT-ADMM-VR converge to the stationary point, concluding the proof.

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] O. Shorinwa, T. Halsted, J. Yu, and M. Schwager, "Distributed optimization methods for multi-robot systems: Part 1—a tutorial," *IEEE Robotics & Automation Magazine*, 2024.

[3] ——, "Distributed optimization methods for multi-robot systems: Part 2—a survey," *IEEE Robotics & Automation Magazine*, 2024.

[4] R. Mohebifard and A. Hajbabaie, "Distributed optimization and coordination algorithms for dynamic traffic metering in urban street networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1930–1941, 2018.

[5] 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.

[6] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated Learning: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 14–41, May 2022.

[7] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.

[8] S. A. Alghunaim and K. Yuan, "A unified and refined convergence analysis for non-convex decentralized learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3264–3279, 2022.

[9] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," *Optimization Methods and Software*, pp. 1–28, 2024.

[10] S. A. Alghunaim, "Local exact-diffusion for decentralized optimization and learning," *IEEE Transactions on Automatic Control*, vol. 69, no. 11, pp. 7371–7386, 2024.

[11] L. Guo, S. A. Alghunaim, K. Yuan, L. Condat, and J. Cao, "Randcom: Random communication skipping method for decentralized stochastic optimization," *CoRR*, 2023.

[12] 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.

[13] 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.

[14] R. Xin, U. A. Khan, and S. Kar, "Variance-Reduced Decentralized Stochastic Optimization With Accelerated Convergence," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6255–6271, 2020.

[15] ——, "A Fast Randomized Incremental Gradient Method for Decentralized Nonconvex Optimization," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5150–5165, 2022.

[16] ——, "Fast Decentralized Nonconvex Finite-Sum Optimization with Recursive Variance Reduction," *SIAM Journal on Optimization*, vol. 32, no. 1, pp. 1–28, Mar. 2022.

[17] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[18] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[19] F. Saadatniaki, R. Xin, and U. A. Khan, "Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4769–4780, 2020.

[20] X. Ren, D. Li, Y. Xi, and H. Shao, "An accelerated distributed gradient method with local memory," *Automatica*, vol. 146, p. 110260, 2022.

[21] 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.

[22] N. Bastianello, D. Deplano, M. Franceschelli, and K. H. Johansson, "Robust online learning over networks," *IEEE Transactions on Automatic Control*, vol. 70, no. 2, p. 933–946, 2025.

[23] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed admm over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[24] V. Khatana and M. V. Salapaka, "Dc-distadmm: Admm algorithm for constrained optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 68, no. 9, pp. 5365–5380, 2022.

[25] 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.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, Apr. 2017, pp. 1273–1282.

[27] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021.

[28] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtarik, "ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!" in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, Jul. 2022, pp. 15 750–15 769.

[29] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear Convergence in Federated Learning: Tackling Client Heterogeneity and Sparse Gradients," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 14 606–14 619.

[30] L. Condat, I. Agarský, G. Malinovsky, and P. Richtárik, "TAMUNA: Doubly Accelerated Federated Learning with Local Training, Compression, and Partial Participation," May 2023.

[31] 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.

[32] A. S. Berahas, R. Bollapragada, and E. Wei, "On the convergence of nested decentralized gradient methods with multiple consensus and gradient steps," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4192–4203, 2021.

[33] C. Iakovidou and E. Wei, "S-NEAR-DGD: A Flexible Distributed Stochastic Gradient Method for Inexact Communication," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 1281–1287, Feb. 2023.

[34] Y. Hou, W. Hu, J. Li, and T. Huang, "Prescribed performance control for double-integrator multi-agent systems: A unified event-triggered consensus framework," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 9, pp. 4222–4232, 2024.

[35] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in neural information processing systems*, vol. 26, 2013.

[36] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "Sarah: A novel method for machine learning problems using stochastic recursive gradient," in *International conference on machine learning*. PMLR, 2017, pp. 2613–2621.

[37] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

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

Xiaoxing Ren (M'24) received her B.S. degree in Automation from Dalian University of Technology, Dalian, China, in 2016, and her Ph.D. degree from the Department of Automation at Shanghai Jiao Tong University, Shanghai, China, in 2022. She is currently a Postdoctoral Researcher at the School of Civil and Environmental Engineering, Cornell University, Ithaca, NY, USA. Prior to joining Cornell, she was a Research Associate at the Department of Electrical and Electronic Engineering at Imperial College London, UK, and previously a Senior Engineer at Huawei Technologies, Shanghai, China. Her research interests include optimization and learning in multi-agent systems, as well as data-driven methods.

Nicola Bastianello (M'18) is a post-doc at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden. From 2021 to 2022 he was a post-doc at the Department of Information Engineering (DEI), University of Padova, Italy. He received the Ph.D. in Information Engineering at the University of Padova, Italy in 2021. During the Ph.D. he was a visiting student at the Department of Electrical, Computer, and Energy Engineering (ECEE), University of Colorado Boulder, Colorado, USA. He received the master degree in Automation Engineering (2018) and the bachelor degree in Information Engineering (2015) from the University of Padova, Italy. He currently serves in the IEEE CSS and EUCA Conference Editorial Boards. His research lies at the intersection of optimization and learning, with a focus on multi-agent systems.

Karl H. Johansson is Swedish Research Council Distinguished Professor in Electrical Engineering and Computer Science at KTH Royal Institute of Technology in Sweden and Founding Director of Digital Futures. He earned his MSc degree in Electrical Engineering and PhD in Automatic Control from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU and other prestigious institutions. His research interests focus on networked control systems and cyber-physical systems with applications in transportation, energy, and automation networks. For his scientific contributions, he has received numerous best paper awards and various distinctions from IEEE, IFAC, and other organizations. He has been awarded Distinguished Professor by the Swedish Research Council, Wallenberg Scholar by the Knut and Alice Wallenberg Foundation, Future Research Leader by the Swedish Foundation for Strategic Research. He has also received the triennial IFAC Young Author Prize and IEEE CSS Distinguished Lecturer. He is the recipient of the 2024 IEEE CSS Hendrik W. Bode Lecture Prize. His extensive service to the academic community includes being President of the European Control Association, IEEE CSS Vice President Diversity, Outreach & Development, and Member of IEEE CSS Board of Governors and IFAC Council. He has served on the editorial boards of Automatica, IEEE TAC, IEEE TCNS and many other journals. He has also been a member of the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He is Fellow of both the IEEE and the Royal Swedish Academy of Engineering Sciences.

Thomas Parisini (Fellow, IEEE) received the Ph.D. degree in electronic engineering and computer science from the University of Genoa, Italy, in 1993. He was an Associate Professor with Politecnico di Milano, Italy. He currently holds the Chair of industrial control and is the Head of the Control and Power Research Group, Imperial College London, U.K. He also holds a Distinguished Professorship at Aalborg University, Denmark. Since 2001, he has been the Danieli Endowed Chair of automation engineering with the University of Trieste, Italy, where from 2009 to 2012, he was the Deputy Rector. In 2023, he held a "Scholar-in-Residence" visiting position with Digital Futures-KTH, Sweden. He has coauthored a research monograph in the Communication and Control Series (Springer Nature) and over 400 publications, including journal articles, book chapters, and conference papers. In 2023 he was the recipient of the Knighthood of the Order of Merit of the Italian Republic for scientific achievements abroad awarded by the Italian President of the Republic. In 2018 he received the Honorary Doctorate from the University of Aalborg, Denmark and in 2024, the IEEE CSS Transition to Practice Award. Moreover, he was awarded the 2007 IEEE Distinguished Member Award, and was co-recipient of the IFAC Best Application Paper Prize of the Journal of Process Control for the period 2011-2013 and of the 2004 Outstanding Paper Award of IEEE Transactions on Neural Networks. In 2016, he was awarded as Principal Investigator with Imperial of the H2020 European Union flagship Teaming Project KIOS Research and Innovation Centre of Excellence led by the University of Cyprus with an overall budget of over 40 million Euros. He was the 2021-2022 President of the IEEE Control Systems Society, the Editor-in-Chief of IEEE Transactions on Control Systems Technology (2009-2016). He is currently an Editor of Automatica and the Editor-in-Chief of the European Journal of Control. He is a Fellow of IFAC, a Member of IEEE TAB Periodicals Review and Advisory Committee and chairs the IEEE CSS Awards Committee.