Communication-Adaptive Stochastic Gradient Methods for Distributed Learning

Tianyi Chen, *Member, IEEE*, Yuejiao Sun, and Wotao Yin, *Member, IEEE*

***Abstract*—This paper targets developing algorithms for solving distributed learning problems in a communication-efficient fashion, by generalizing the recent method of lazily aggregated gradient (LAG) to deal with stochastic gradient — justifying the name of the new method LASG. While LAG is effective at reducing communication without sacrificing the rate of convergence, we show it only works with deterministic gradients. We introduce new rules and analysis for LASG that are tailored for stochastic gradients, so it effectively saves downloads, uploads, or both for distributed stochastic gradient descent. LASG achieves impressive empirical performance — it typically saves total communication by an order of magnitude. LASG can be used together with gradient quantization to bring more savings.**

***Index Terms*—Distributed optimization, machine learning, federated learning, communication-efficient.**

I. Introduction

STOCHASTIC gradient descent (SGD) method [1] is prevalent in solving large-scale machine learning problems during the last decades. Although simple to use, the plain-vanilla SGD often becomes less efficient when it is applied to the distributed setting, especially in terms of the communication efficiency.

In this paper, we aim to solve the distributed learning problem in a communication-efficient fashion while maintaining the learning accuracy. Consider a setting consisting of a cloud server and a set of M devices (workers) collected in $\mathcal{M} := \{1, \ldots, M\}$. Each device m has its local dataset $\{\xi_n,\, n \in \mathcal{N}_m\}$, which defines the loss function of device m as

$$\mathcal{L}_m(\theta) := \sum_{n \in \mathcal{N}_m} \ell(\theta; \xi_n), \quad m \in \mathcal{M} \tag{1}$$

where $\theta \in \mathbb{R}^p$ is the sought vector (e.g., parameters of a prediction model) and ξ_n is a data sample. For example, in linear regression, $\ell(\theta; \xi_n)$ is the square loss; and, in deep learning, $\ell(\theta; \xi_n)$ is the loss function of a neural network, and θ concatenates the weights. The goal is to solve

$$\min_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta) \text{ with } \mathcal{L}(\theta) := \frac{1}{M} \sum_{m \in \mathcal{M}} \mathcal{L}_m(\theta). \tag{2}$$

Problem (2) also arises in a number of areas, such as multi-agent optimization [2], distributed signal processing [3], and distributed machine learning [4]. While our algorithms can be applied to other settings, we focus on the setting that for bandwidth and privacy concerns, local data $\{\xi_n, n \in \mathcal{N}_m\}$ at each worker m are not uploaded to the server. This setting naturally arises in e.g., federated learning, in which collaboration is needed through communication between the server and multiple workers (e.g., mobile devices).

To solve (2), we can in principle apply the distributed version of SGD. In this case, at iteration k, the server broadcasts the current model θ^k to *all* the workers; each worker m computes $\nabla \ell(\theta^k; \xi_m^k)$ using a randomly selected sample or a minibatch of samples $\{\xi_m^k\} \subseteq \{\xi_n, n \in \mathcal{N}_m\}$, and then uploads it to the server; and once receiving stochastic gradients from all workers, the server updates the model parameters via

$$\textbf{SGD} \qquad \theta^{k+1} = \theta^k - \frac{\eta_k}{M} \sum_{m \in \mathcal{M}} \nabla \ell(\theta^k; \xi_m^k) \tag{3}$$

where $\eta_k > 0$ is the (possibly) time-varying stepsize used at iteration k. When $\nabla \ell(\theta^k; \xi_m^k)$ is an unbiased gradient estimator of $\mathcal{L}_m(\theta)$, the convergence of SGD update (3) is guaranteed [5]. To implement (3), however, the server has to communicate with *all* workers to obtain fresh $\{\nabla \ell(\theta^k; \xi_m^k)\}$. This prevents the efficient implementation of SGD in scenarios where communication between the server and the workers is costly [6]. For example, consider using SGD to iteratively train an image classification model over a group of wireless devices. The start-of-the-art deep neural network models (e.g., ResNet, LSTM) for computer vision, speech and natural language processing tasks involve millions of parameters (e.g., 500 MB). This training process is costly because one SGD update generates around 500 MB data on each device's up- and down-link transmission, and SGD takes thousands of iterations to converge. Therefore, *our goal* is to find the parameter θ that minimizes (2) with minimal communication overhead.

A. Related Work

Communication-efficient distributed learning methods have gained popularity recently [7], [8]. Most methods belong to two

Manuscript received June 30, 2020; revised November 10, 2020 and April 4, 2021; accepted July 12, 2021. Date of publication July 27, 2021; date of current version August 20, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yao Xie. The work of Tianyi Chen was supported in part by National Science Foundation under the project NSF 2047177, and in part by RPI-IBM Artificial Intelligence Research Collaboration. The work of Yuejiao Sun was supported in part by ONR under Grant N000141712162, and in part by AFOSR MURI under Grant FA9550-18-1-0502. *(Corresponding author: Tianyi Chen.)*

Tianyi Chen is with the Department of Electrical, Computer and Systems Engineering, and the Institute for Data Exploration and Applications, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: chentianyi19@gmail.com).

Yuejiao Sun and Wotao Yin are with the Department of Mathematics, University of California, Los Angeles, CA 90095 USA (e-mail: sunyj@math.ucla.edu; wotaoyin@math.ucla.edu).

Digital Object Identifier 10.1109/TSP.2021.3099977

categories: c1) reducing the bits per communication round; and, c2) reducing the communication rounds.

Reducing communication bits. For c1), methods are centered around the ideas of *quantization* and *sparsification*.

Quantization has been successfully applied to wireless sensor networks [9], [10]. In the context of distributed machine learning, a 1-bit and multi-bits quantization methods have been developed in [11]–[13]. Other variants of quantized gradient schemes include error compensation [14], variance-reduced quantization [15], and quantization to a ternary vector [16].

Sparsification amounts to transmitting only gradient coordinates with large enough magnitudes exceeding a certain threshold [17]. To avoid losing information of skipping communication, small gradient components will be accumulated and then transmitted when they are large enough [18]–[22].

Quantization and sparsification address c1) but not address c2), so they are still affected by latencies due to initiating communication, queuing, and propagating messages [23].

Reducing communication rounds. Methods using periodic averaging include elastic averaging SGD [24], local SGD (FedAvg) [6], [25]–[29] and momentum SGD [30]. Except [26], [29], [31], local SGD methods follow a fixed communication schedule. They work well in the *homogeneous* setting where data are i.i.d. over all workers, but often sacrifice the learning accuracy in the non-i.i.d. case. Work tailored for the heterogeneous setting includes FedProx [32]. Other methods that reduce the number of iterations include the gradient tracking [33], [34], primal-dual update [35], [36], opportunistic communication [37], and higher-order methods [38], [39]. Roughly speaking, algorithms in [32], [38]–[40] reduce communication by increasing local gradient computation.

This paper is based on the method of lazily aggregated gradient (LAG) [41], [42]. LAG is adaptive and works well for the *heterogeneous* setting. Parameters in LAG are updated at the server, and workers only upload information that is informative enough. LAG has great performance with full gradient, but its performance degrades significantly with stochastic gradients, which make its rule of communication highly unreliable.

B. Our Approach

This paper proposes **L**azily **A**ggregated **S**tochastic **G**radient (**LASG**), which includes a set of SGD-based methods that considerably reduce the communication of distributed SGD. Compared with popular communication-efficient algorithms such as local SGD [6], [25]–[27], our LASG does not sacrifice learning accuracy in the non-i.i.d. settings. Observing that not all communications between the server and the workers are equally important, LASG uses conditions to decide communication adaptively. When a worker skips a round of communication, the server uses its stale gradient to perform parameter updates.

Define $\mathcal{M}^k$ as the set of uploading workers at iteration k, and define τ_m^k as the staleness of the gradient from worker m used

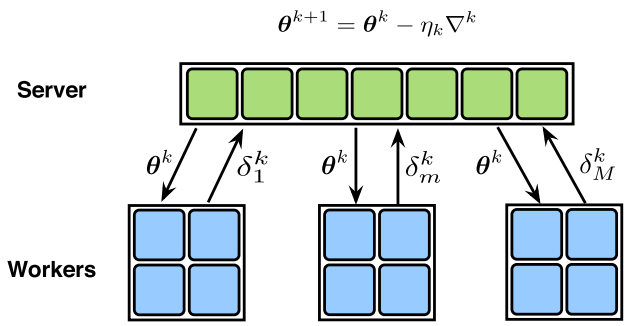


Fig. 1. Generic LASG implementation.

at iteration k. LASG has the following update

$$\theta^{k+1} = \theta^k - \frac{\eta_k}{M} \sum_{m \in \mathcal{M} \setminus \mathcal{M}^k} \nabla \ell(\theta^{k-\tau_m^k}; \xi_m^{k-\tau_m^k}) - \frac{\eta_k}{M} \sum_{m \in \mathcal{M}^k} \nabla \ell(\theta^k; \xi_m^k) \quad (4)$$

or equivalently (see also Fig. 1)

$$\textbf{GenericLASG} \quad \theta^{k+1} = \theta^k - \eta_k \nabla^k \quad \text{with} \quad \nabla^k = \nabla^{k-1} + \frac{1}{M} \sum_{m \in \mathcal{M}^k} \delta_m^k \quad (5)$$

where the stochastic gradient innovation is defined as

$$\delta_m^k := \nabla \ell(\theta^k; \xi_m^k) - \nabla \ell(\theta^{k-\tau_m^k}; \xi_m^{k-\tau_m^k}). \quad (6)$$

The staleness $\{\tau_m^k\}$ depend on $\mathcal{M}^k$: at iteration k, if worker $m \notin \mathcal{M}^k$, the server increases staleness $\tau_m^{k+1} = \tau_m^k + 1$; otherwise, worker m uploads its stochastic gradient, and the server resets $\tau_m^{k+1} = 1$.

Clearly, selection of subset $\mathcal{M}^k$ is critical. The challenges are 1) the importance of each communication round is dynamic, thus a fixed condition is ineffective; and 2) checking the condition must be numerically efficiently. To address these challenges, we develop two types of adaptive condition based on different *communication*, *computation* and *memory* requirements. The first type is computed by each worker (**WK**), and the second by the server (**PS**).

LASG-WK: At iteration k, the server broadcasts θ^k to all workers; each worker m computes $\nabla \ell(\theta^k; \xi_m^k)$, and checks whether $m \in \mathcal{M}^k$; only those in $\mathcal{M}^k$ upload δ_m^k to the server, which executes (5).

LASG-PS: At iteration k, the server determines $\mathcal{M}^k$ and sends θ^k to those workers $m \in \mathcal{M}^k$; each worker $m \in \mathcal{M}^k$ computes $\nabla \ell(\theta^k; \xi_m^k)$ and uploads δ_m^k; those workers $m \notin \mathcal{M}^k$ do nothing; the server executes (5);

How $\mathcal{M}^k$ are computed are deferred to Section II. We summarize the contributions of this paper as follows.

1) We introduce LASG, a set of communication-skipping methods for distributed SGD. It reuses stale stochastic gradients to reduce redundant communication.

2) We establish convergence of our proposed methods. The convergence rates match those of SGD.

3) We tested LASG on logistic regression and neural network training and confirm its performance gains.

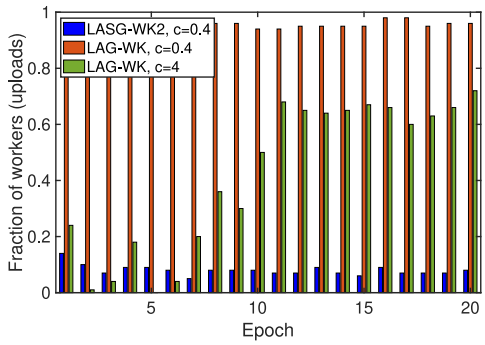


Fig. 2. Comparison of upload numbers (10 iterations per epoch). Applying LAG-WK with stochastic gradients is ineffective. Even using an aggressive parameter $c = 4$, it is significantly less effective than LASG-WK2 (proposed).

C. Why LAG Does Not Work Well With SGD?

Let us revisit the LAG method [41] and provide why it works poorly with stochastic gradients.

Similar to what is described above, LAG has both WK and PS types of conditions to decide $\mathcal{M}^k$. Since they are equally ineffective with stochastic gradients, we limit our discussion to LAG-WK. Applying LAG-WK to stochastic gradients amounts to, in the condition of [41], replacing worker m's gradient by its stochastic gradient, that is, *exclude* m from $\mathcal{M}^k$ if

$$\left\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2 \le \frac{c}{d_{\max}}\sum_{d=1}^{d_{\max}}\left\|\theta^{k+1-d}-\theta^{k-d}\right\|^2, \quad (7)$$

where $c \ge 0$ is a pre-defined constant, and $d_{\max}$ is the number of consecutive past iterates. This condition compares the new stochastic gradient to the stale copy at the server; if the difference is small compared to the recent changes in θ, then the server will reuse the stale copy.

When used with (standard) gradients, LAG [41] proves the condition leads to *"larger descent per upload"*. Unfortunately, the two stochastic gradients in (7) are evaluated with two different samples, ξ_m^k and $\xi_m^{k-\tau_m^k}$. The left-hand side (LHS) is almost never small. So, (7) becomes ineffective at judging the contribution of $\nabla\ell(\theta^k;\xi_m^k)$ to the *stochastic* descent.

Fig. 2 compares the stochastic LAG and one of our new algorithms LASG-WK2 (introduced later) on a synthetically generated logistic regression task, which demonstrates that the stochastic LAG is ineffective in saving communication — when c is small (e.g., 0.4), (7) is almost never satisfied due to the inherent variance of the computed stochastic gradients; and when c is large (e.g., 4), (7) is satisfied only initially. Mathematically, this can be explained by expanding the LHS of (7) by (see the supplemental material for the deduction)

$$\mathbb{E}\left[\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\|^2\right] \quad (8a)$$

$$\ge \frac{1}{2}\mathbb{E}\left[\left\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\mathcal{L}_m(\theta^k)\right\|^2\right] \quad (8b)$$

$$+\frac{1}{2}\mathbb{E}\left[\left[\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\|^2\right]\right] \quad (8c)$$

$$-\mathbb{E}[\|\nabla\mathcal{L}_m(\theta^k)-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\|^2]. \quad (8d)$$

When θ^k converges, e.g., $\theta^k \to \theta^*$, the right-hand side (RHS) of (7) $\left\|\theta^{k+1-d}-\theta^{k-d}\right\|^2 \to 0$. But the LHS of (7) does not since the gradient variances in (8b) and (8c) do not vanish.

Therefore, the key issue is the variance of stochastic gradients is not diminishing and fails the LAG rule (7) eventually.

II. LASG: Lazily Aggregated Stochastic Gradients

In this section, we formally develop our LASG method. To overcome the limitations of LAG in stochastic settings, the key of the LASG design is to **reduce the variance of the innovation measure** appeared in the adaptive condition. As discussed, LASG-WK uses a condition checked by each worker; LASG-PS uses one checked by the parameter server.

A. Worker LASG: Save Communication Uploads

We first introduce two LASG-WK variants. The first one, which we term **LASG-WK1**, calculates two stochastic gradient innovations with one at sample ξ_m^k as

$$\tilde{\delta}_m^k := \nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\tilde{\theta};\xi_m^k)$$

and one at sample $\xi_m^{k-\tau_m^k}$ as

$$\tilde{\delta}_m^{k-\tau_m^k} := \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k}).$$

where $\tilde{\theta}$ is a snapshot of the previous iterate θ that will be updated every D ($\ge d_{\max}$) iterations. As we will show in (10), $\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}$ can be viewed as the difference of two variance-reduced gradients calculated at θ^k and $\theta^{k-\tau_m^k}$. Using $\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}$ as the error induced by using stale information, LASG-WK1 excludes m from $\mathcal{M}^k$ if

$$\boxed{\left\|\tilde{\delta}_m^k-\tilde{\delta}_m^{k-\tau_m^k}\right\|^2 \le \frac{c}{d_{\max}}\sum_{d=1}^{d_{\max}}\left\|\theta^{k+1-d}-\theta^{k-d}\right\|^2.} \quad (9)$$

Recall if (9) is satisfied, we increment staleness $\tau_m^{k+1} = \tau_m^k + 1$; otherwise, worker m uploads the fresh stochastic gradient and resets staleness as $\tau_m^{k+1} = 1$.

Behind (9) is the reduction of its inherent variance. To see this, decompose the LHS of (9) as the difference of two *variance reduced* stochastic gradients at iteration k and $k - \tau_m^k$:

$$\tilde{\delta}_m^k-\tilde{\delta}_m^{k-\tau_m^k} = \left(\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\tilde{\theta};\xi_m^k)+\nabla\mathcal{L}_m(\tilde{\theta})\right) - \left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k})+\nabla\mathcal{L}_m(\tilde{\theta})\right). \quad (10)$$

To provide some intuition, we define the minimizer of (2) as $\theta^\star$ and assume that $\nabla\ell(\theta;\xi_m)$ is $\bar{L}$-Lipschitz continuous[1] for any ξ_m. The LHS of (9) is *upper-bounded* in expectation by

$$\mathbb{E}\left[\left\|\tilde{\delta}_m^k-\tilde{\delta}_m^{k-\tau_m^k}\right\|^2\right] \le 8\bar{L}(\mathbb{E}\mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star)) + 8\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k})-\mathcal{L}(\theta^\star))+16\bar{L}(\mathbb{E}\mathcal{L}(\tilde{\theta})-\mathcal{L}(\theta^\star)). \quad (11)$$

When the iterate θ^k converges, e.g., $\theta^k, \theta^{k-\tau_m^k}, \tilde{\theta} \to \theta^*$, the RHS of (11) diminishes, and thus the LHS of (9) diminishes. This is

[1]This Lipschitz continuous assumption is needed only when we provide some intuitions of our design, but in our subsequent analysis.

TABLE I
A COMPARISON OF COMMUNICATION, COMPUTATION AND MEMORY REQUIREMENTS. **PS** DENOTES THE PARAMETER SERVER, **WK** DENOTES THE WORKER, **PS** $\rightarrow$ **WK** m IS THE DOWNLOAD FROM THE SERVER TO WORKER m, AND **WK** $m \rightarrow$ **PS** IS THE UPLOAD FROM WORKER m TO THE SERVER

Metric	Communication		Computation		Memory	
Algorithm	PS$\rightarrow$WK m	WK $m \rightarrow$PS	PS	WK m	PS	WK m
Sync SGD	always	always	(3)	(3)	$\mathcal{O}(p)$	/
LASG-WK1	always	only if $m \in \mathcal{M}^k$	(5)	(9)	$\mathcal{O}(p)$	$\mathcal{O}(p)$
LASG-WK2	always	only if $m \in \mathcal{M}^k$	(5)	(12)	$\mathcal{O}(p)$	$\mathcal{O}(p)$
LASG-PS	only if $m \in \mathcal{M}^k$	only if $m \in \mathcal{M}^k$	(5), (14)	only if $m \in \mathcal{M}^k$	$\mathcal{O}(Mp)$	$\mathcal{O}(p)$
LASG-PSE	only if $m \in \mathcal{M}^k$	only if $m \in \mathcal{M}^k$	(5), (16)	only if $m \in \mathcal{M}^k$	$\mathcal{O}(Mp)$	$\mathcal{O}(p)$

in contrast to the stochastic LAG-WK rule in (8) that is *lower-bounded* by a non-diminishing value.

The second rule **LASG-WK2** excludes m from $\mathcal{M}^k$ if

$$\left\|\nabla \ell(\theta^k;\xi_m^k) - \nabla \ell(\theta_m^{k-\tau_m^k};\xi_m^k)\right\|^2 \leq \frac{c}{d_{\max}} \sum_{d=1}^{d_{\max}} \left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2. \quad (12)$$

Note that different from (7), condition (12) is evaluated at two different iterates but on the same sample ξ_m^k.

LASG-WK2 (12) also reduces its inherent variance since the LHS of (12) can be written as the difference between a *variance reduced* stochastic gradient and a *deterministic* gradient, that is

$$\nabla \ell(\theta^k;\xi_m^k) - \nabla \ell(\theta^{k-\tau_m^k};\xi_m^k) = \big(\nabla \ell(\theta^k;\xi_m^k) - \nabla \ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla \mathcal{L}_m(\theta^{k-\tau_m^k})\big) - \nabla \mathcal{L}_m(\theta^{k-\tau_m^k}). \quad (13)$$

With derivations deferred to the supplementary, we conclude that $\mathbb{E}[\|\nabla \ell(\theta^k;\xi_m^k) - \nabla \ell(\theta^{k-\tau_m^k};\xi_m^k)\|^2] \to 0$ as $\theta^k \to \theta^\star$.

B. Server LASG: Save Up/Downloads and Calculations

We next introduce two LASG-PS variants. The rationale is that if the model difference is small, the gradient difference used in Section II-A is likely to be small.

The first variant **LASG-PS** excludes m from $\mathcal{M}^k$ if

$$L_m^2 \left\|\theta^k - \theta^{k-\tau_m^k}\right\|^2 \leq \frac{c}{d_{\max}} \sum_{d=1}^{d_{\max}} \left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2 \quad (14)$$

where L_m is the smoothness constant of $\mathcal{L}_m(\theta)$. Condition (14) can be checked at the server side without computing new gradients if the server stores $\{\theta^{k-\tau_m^k}\}$ for each worker m.

The LHS of (14) can be *upper-bounded* in expectation by

$$\mathbb{E}\left[\left\|\theta^k - \theta^{k-\tau_m^k}\right\|^2\right] \leq 2D \sum_{d=1}^{D} \mathbb{E}\left[\left\|\theta^{k-d} - \theta^{k-d-\tau_m^{k-d}}\right\|^2\right] \eta_{k-D}^2 + 2D \sum_{d=1}^{D} \mathbb{E}\left\|\nabla \mathcal{L}(\theta^{k-d})\right\|^2 \eta_{k-D}^2 + D^2 \left(\sum_{m\in\mathcal{M}} \sigma_m^2\right) \eta_{k-D}^2. \quad (15)$$

Assume $\|\nabla \mathcal{L}(\theta^k)\|^2$ is bounded; then the diminishing stepsizes $\{\eta_k\}$ ensure that the 2nd and 3 rd terms in the RHS of (15) vanish. Using mathematical induction, the LHS of (14) also diminishes. Therefore, this condition remains effective asymptotically.

When an estimate L_m is not available, one can use LASG-PSE, a variation of LASG-PS that estimates L_m "on-the-fly." With $\hat{L}_m^k$ denoting the estimate of L_m, **LASG-PSE** excludes m from $\mathcal{M}^k$ if

$$(\hat{L}_m^k)^2 \|\theta^k - \theta^{k-\tau_m^k}\|^2 \leq \frac{c}{d_{\max}} \sum_{d=1}^{d_{\max}} \|\theta^{k+1-d} - \theta^{k-d}\|^2 \quad (16)$$

where the estimated constant $\hat{L}_m^k$ is updated iteratively via

$$\hat{L}_m^{k+1} = \max\left\{\hat{L}_m^k, \frac{\|\nabla \ell(\theta^k;\xi_m^k) - \nabla \ell(\theta^{k-\tau_m^k};\xi_m^k)\|}{\|\theta^k - \theta^{k-\tau_m^k}\|}\right\}. \quad (17)$$

We give LASG-PS and LASG-PSE in Algorithms 3 and 4, respectively, and compare all LASG methods in Table I .

Comparison of all LASG methods. All the LASG rules can be computed efficiently without storing all previous θ^k. LASG-PS and LASG-PSE need *extra memory* at the server but save both *local computation and download communication* while LASG-WK1 and LASG-WK2 save only upload communication. LASG-WK1 is more conservative as LASG-WK1 measures the change of gradients at two model states for both new and old data samples, but LASG-WK2 measures only the change of gradient at the new sample.

C. Quantized LASG: Save Also Communication Bits

We further reduce communication bits per round by applying quantization. We define the gradient under a quantization operator $\mathcal{Q}$ as

$$Q(\theta;\xi) := \mathcal{Q}\left(\nabla \ell(\theta;\xi)\right). \quad (18)$$

We adopt the stochastic quantization scheme in [13] and develop *quantized LASG* (LAQSG) as

$$\theta^{k+1} = \theta^k - \eta_k \sum_{m\in\mathcal{M}\setminus\mathcal{M}^k} Q(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \eta_k \sum_{m\in\mathcal{M}^k} Q(\theta^k;\xi_m^k)$$

where $\mathcal{M}^k$ is determined by one of four described rules.

III. MAIN RESULTS

In this section we present the convergence results of LASG-WK1, LASG-WK2 and LASG-PS under both the nonconvex condition and the Polyak-Łojasiewicz condition, and the convergence results of LAQSG under the nonconvex condition only. We leave the analysis of LASG-PSE for future work, but it empirically has very impressive performance.

First, we make some basic assumptions.

Assumption 1: The loss function $\mathcal{L}(\theta)$ is L-smooth, i.e. for any $\theta_1, \theta_2 \in \mathbb{R}^p$, it follows that

$$\mathcal{L}(\theta_2) \leq \mathcal{L}(\theta_1) + \left\langle \nabla \mathcal{L}(\theta_1), \theta_2 - \theta_1 \right\rangle + \frac{L}{2}\|\theta_2 - \theta_1\|^2. \quad (19)$$

TABLE II
A COMPARISON OF LASG-WK1 AND LASG-WK2

Algorithm 1 LASG-WK1

Input: Delay counter $\{\tau_m^0\}$, stepsizes $\{\eta_k\}$, max delay D.
for $k = 0, 1, \ldots, K-1$ **do**
  Server broadcasts θ^k to all workers.
  All workers save $\tilde{\theta} = \theta^k$ if $k \mod D = 0$.
  for Worker $m = 1, 2, \ldots, M$ **do in parallel**
    Compute $\nabla \ell(\theta^k; \xi_m^k)$ and $\nabla \ell(\tilde{\theta}; \xi_m^k)$.
    Check condition (9) with stored $\tilde{\delta}_m^{k-\tau_m^k}$.
    if (9) is violated, or, $k \mod D = 0$ **then**
      Upload δ_m^k. ▷ Save $\tilde{\delta}_m^k$ and set $\tau_m^{k+1} = 1$
    else
      Upload nothing. ▷ Set $\tau_m^{k+1} = \tau_m^k + 1$
    end if
  end for
  Server updates via (4).
end for

Algorithm 2 LASG-WK2

Input: Delay counter $\{\tau_m^0\}$, stepsizes $\{\eta_k\}$, max delay D.
for $k = 0, 1, \ldots, K-1$ **do**
  Server broadcasts θ^k to all workers.
  for Worker $m = 1, 2, \ldots, M$ **do in parallel**
    Compute $\nabla \ell(\theta^k; \xi_m^k)$ and $\nabla \ell(\theta_m^{k-\tau_m^k}; \xi_m^k)$.
    Check condition (12).
    if (12) is violated, or, $\tau_m^k \geq D$ **then**
      Upload δ_m^k. ▷ Save θ^k and set $\tau_m^{k+1} = 1$
    else
      Upload nothing. ▷ Set $\tau_m^{k+1} = \tau_m^k + 1$
    end if
  end for
  Server updates via (4).
end for

TABLE III
A COMPARISON OF LASG-PS AND LASG-PSE

Algorithm 3 LASG-PS

Input: θ^0, delay counter $\{\tau_m^0\}$, smoothness contants $\{L_m\}$, stepsizes $\{\eta_k\}$, maximum delay D.
for $k = 0, 1, \ldots, K-1$ **do**
  for Worker $m = 1, 2, \ldots, M$ **do in parallel**
    Server checks condition (14).
    if (14) is violated or $\tau_m^k \geq D$ **then**
      Server sends θ^k to worker m
      Worker m computes $\nabla \ell(\theta^k; \xi_m^k)$.
      Worker m uploads δ_m^k. ▷ Save θ^k and $\tau_m^{k+1} = 1$
    else
      No action. ▷ $\tau_m^{k+1} = \tau_m^k + 1$
    end if
  end for
  Server updates via (4).
end for

Algorithm 4 LASG-PSE

Input: θ^0, delay counter $\{\tau_m^0\}$, smoothness estimates $\{\hat{L}_m^0\}$, stepsizes $\{\eta_k\}$, maximum delay D.
for $k = 0, 1, \ldots, K-1$ **do**
  for Worker $m = 1, 2, \ldots, M$ **do in parallel**
    Server checks condition (16).
    if (16) is violated or $\tau_m^k \geq D$ **then**
      Server sends θ^k to worker m.
      Worker m computes $\nabla \ell(\theta^k; \xi_m^k)$.
      Worker m uploads δ_m^k. ▷ Save θ^k and $\tau_m^{k+1} = 1$
      Worker m uploads $\hat{L}_m^{k+1}$ in (17).
    else
      No action. ▷ $\tau_m^{k+1} = \tau_m^k + 1$
    end if
  end for
  Server updates via (4).
end for

Assumption 2: The samples $\xi_m^1, \xi_m^2, \ldots$ are independent, and the stochastic gradient $\nabla \ell(\theta; \xi_m^k)$ satisfies

$$\mathbb{E}_{\xi_m^k}\left[\nabla \ell(\theta; \xi_m^k)\right] = \nabla \mathcal{L}_m(\theta), \tag{20a}$$

$$\mathbb{E}_{\xi_m^k}\left[\|\nabla \ell(\theta; \xi_m^k) - \nabla \mathcal{L}_m(\theta)\|^2\right] \leq \sigma_m^2. \tag{20b}$$

For LASG-PS, we require an extra smoothness assumption.

Assumption 3: The local gradient $\nabla \mathcal{L}_m$ is L_m-Lipschitz continuous, i.e. for any $\theta_1, \theta_2 \in \mathbb{R}^p$, we have

$$\|\nabla \mathcal{L}_m(\theta_1) - \nabla \mathcal{L}_m(\theta_2)\| \leq L_m \|\theta_1 - \theta_2\|. \tag{21}$$

Assumption 1 implies that the loss function $\mathcal{L}$ can be upper bounded by a quadratic function at any point. Assumption 2 ensures that the stochastic gradient is unbiased, and has bounded variance. Assumption 3 bounds the change of local gradients when they are evaluated at two points. Assumptions 1-3 are common in analyzing SGD [13], [19]–[22], [42]–[44].

A. Convergence in the Nonconvex Case

We first present the convergence in the nonconvex case.

Theorem 1 (nonconvex): Under Assumptions 1, 2 (for Algorithm 3 also Assumption 3), if the stepsize is chosen as $\eta_k = \eta = \mathcal{O}(\frac{1}{\sqrt{K}})$, and the threshold is $c \leq \min\{\frac{1}{12D\eta^2}, \frac{\sqrt{M}L^2}{18}\}$, then $\{\theta^k\}$ generated by Algorithms 1-3 satisfy

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\|\nabla \mathcal{L}(\theta^k)\|^2\right] = \mathcal{O}\left(\frac{\sqrt{M}}{K} + \frac{\sqrt{\sum_{m=1}^M \sigma_m^2}}{M^{\frac{3}{4}}\sqrt{K}}\right). \tag{22}$$

From Theorem 1, the convergence rate of LASG in terms of the average gradient norms is still $\mathcal{O}(1/\sqrt{K})$, matching standard SGD [43]. When $K \gg M$, the second term is dominant. If we simplify $\sigma_m = \sigma$, $\forall m$, then the bound becomes $\mathcal{O}(1/(M^{\frac{1}{4}}K^{\frac{1}{2}}))$, and the convergence rate will be improved as the number of workers M increases. Note that async method [45] has better speedup as it artifically assumes that uploading workers are independent of the past.

The assumption below bounds the variance of the quantized stochastic gradient.

Assumption 4: For any $\theta \in \mathbb{R}^p$ and any $m \in \mathcal{M}$, we have $\mathbb{E}_{\xi_m}[\|\nabla \ell(\theta; \xi_m)\|^2] \leq B$.

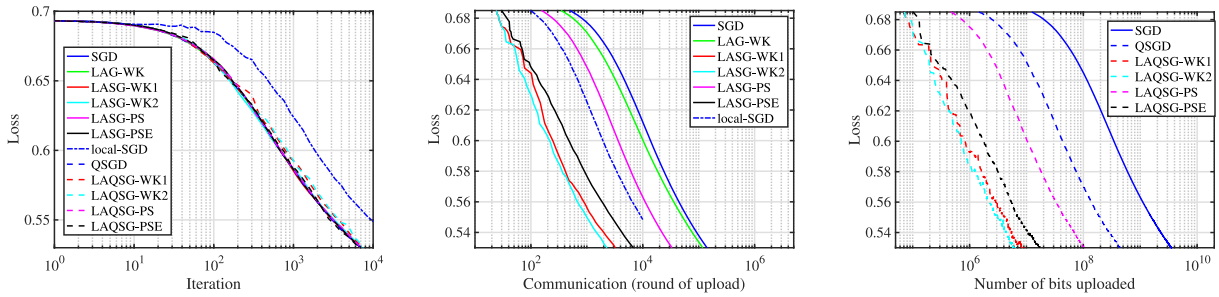


Fig. 3. Logistic regression on *covtype* dataset.

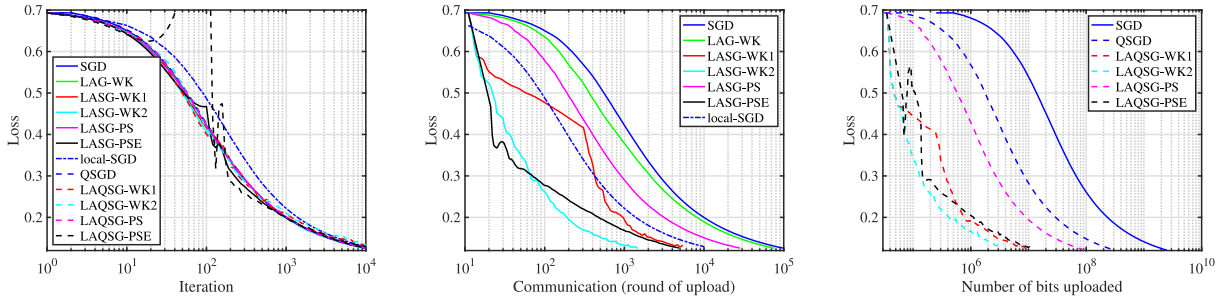


Fig. 4. Logistic regression on *mnist* digits 3 and 5.

Based on this assumption, we have the following result.

Theorem 2 (LAQSG): Under Assumptions 1, 2, 4 (also Assumption 3 for Algorithm 3), if $\eta_k = \eta = \mathcal{O}(\frac{1}{\sqrt{K}})$, $c \leq \min\{\frac{d_{\max}}{16D\eta^2}, \frac{d_{\max}\sqrt{M}L^2}{24}\}$ where $c_\eta > 0$ is a constant, then $\{\theta^k\}$ generated by quantized Algorithms 1 - 3 satisfy

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right] = \mathcal{O}\left(1/\sqrt{K}\right). \tag{23}$$

The rate $\mathcal{O}(1/\sqrt{K})$ matches the standard QSGD [13].

B. Convergence Under the Polyak-Łojasiewicz Condition

Assumption 5: The loss function $\mathcal{L}$ satisfies the Polyak-Łojasiewicz (PL) condition with constant $\mu > 0$, that is

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^*) \leq \frac{1}{2\mu}\|\mathcal{L}(\theta)\|^2. \tag{24}$$

The PL condition is weaker than strong convexity and may hold with convexity [46]. It is met by underdetermined least squares and logistic regression.

Theorem 3 (PL-condition): Under Assumption 1,2,5 (for Algorithm 3 also Assumption 3), if $\eta_k = \frac{2}{\mu(k+K_0)} \leq \eta_0$ for a given constant K_0, and $c \leq \min\{\frac{d_{\max}}{24D\eta_0^2}, \frac{d_{\max}\sqrt{M}L^2}{18}\}$, then θ^K generated by Algorithms 1, 2 and 3 satisfies

$$\mathbb{E}\left[\mathcal{L}(\theta^K)\right] - \mathcal{L}(\theta^\star) = \mathcal{O}\left(1/K\right). \tag{25}$$

The rate $\mathcal{O}(1/K)$ matches that of SGD [47]. Under the same (or even slightly stronger) assumptions of Theorem 3, it has been shown that $\mathcal{O}(1/K)$ is the best rate by any stochastic gradient-based algorithm; see [48, Theorems 5.3.1 and 7.2.6].

IV. Numerical Tests

We conducted numerical tests on both logistic regression and neural network models. We benchmarked LA(Q)SG with SGD, LAG-WK, local SGD (with varying intervals H) and QSGD. We did a grid search for best learning rates.

A. Logistic Regression

The data are distributed across $M = 10$ workers for ijcnn1, MNIST (with digits 3, 5) and $M = 20$ for Covtype. For each worker, the batch size is selected to be 0.01 of the local data size for ijcnn1, MNIST and 0.001 for Covtype. The ℓ_2-regularization parameter is set to be 10^{-5}. We choose stepsize $\eta = 0.1$. For all LASG algorithms, $D = 100$, $d_{\max} = 10$ and $c = 1/\eta^2$. For local-SGD, the communication period is $H = 50, 10, 20$ iterations for ijcnn1, MNIST, Covtype respectively. This is optimized to save communication as much as possible without largely affecting the convergence speed. For quantization methods, we perform 4-bit stochastic quantization [13]. Numerical results are reported in Figs. 3–5.

B. Training Neural Networks

We train a convolutional neural network with two convolution-ELU-maxpooling layers (ELU is a smoothed ReLU) followed by two fully-connected layers for 10 classes classification on MNIST. The data are distributed on $M = 10$ workers. We choose stepsize $\eta = 0.05$. Since the objective function is nonsmooth (L_m is unavailable), LASG-PS is not tested. For other LASG algorithms, we set $D = 50$, $d_{\max} = 10$, and $c = 1/\eta^2$. For local-SGD, we set $H = 4$. For all quantization methods, we set 8 bits. We first report the the total number of uploads needed to achieve the training loss 0.1 and the test

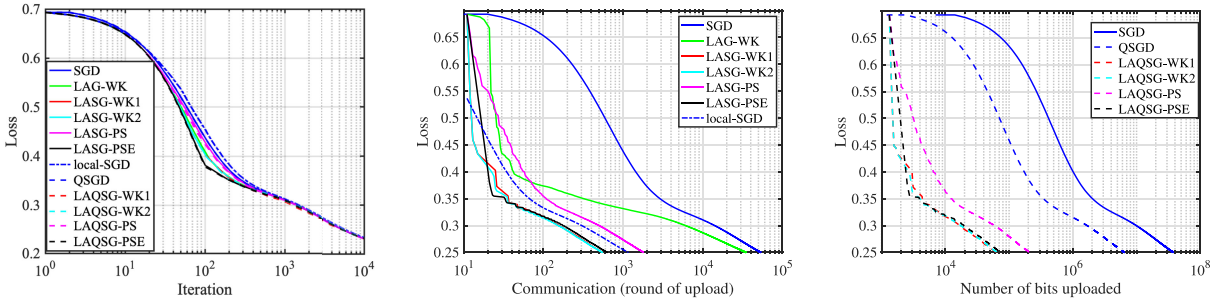


Fig. 5. Logistic regression on *ijcnn1* dataset.

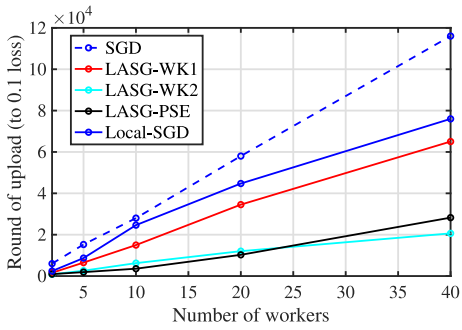


Fig. 6. Training loss on *mnist* dataset under different number of workers.

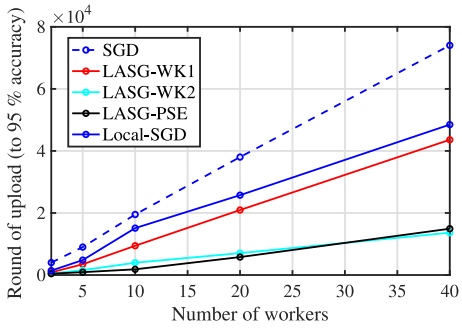


Fig. 7. Test accuracy on *mnist* dataset under different number of workers.

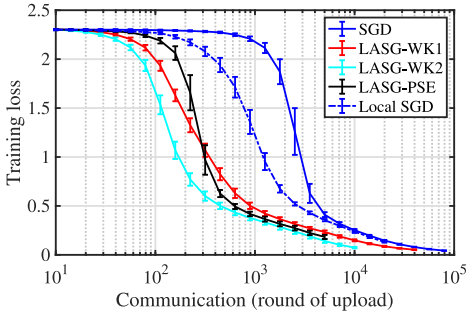


Fig. 8. Training loss on *mnist* dataset averaged over 30 trials.

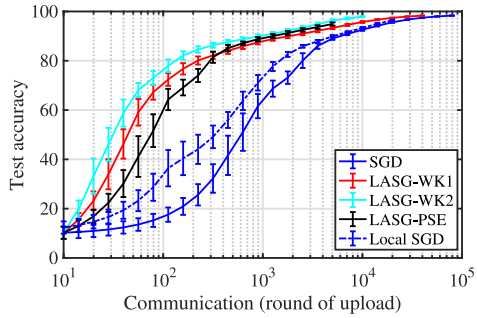


Fig. 9. Test accuracy on *mnist* dataset averaged over 30 trials.

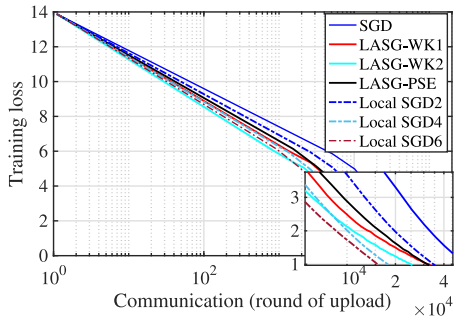


Fig. 10. Training loss on *tiny imagenet* dataset.

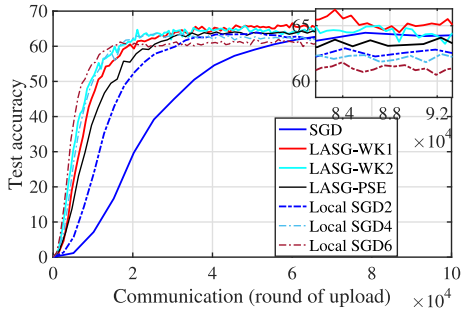


Fig. 11. Test accuracy on *tiny imagenet* dataset.

accuracy 95% under different number of workers M in Figs. 6 and 7, respectively. We also report the numerical results averaged over 30 Monte Carlo runs in Figs. 8 and 9.

All algorithms have been tested on the popular *tiny imagenet* dataset, which contains 200 classes and 500 images per class for training and 10 000 images for testing. All images in *tiny imagenet* are 64x64 colored ones. We use the Resnet18 model initialized by weights pretrained on ImageNet1000; see the accuracy versus the number of communication uploads in Figs. 10 and 11. For training loss, LASG-WK1 and -WK2 require much less total time than SGD and local SGD with $H = 2$, but slightly more than local SGD with $H = 4$ and 6. However, as shown in Fig. 11, local SGD with larger communication period sacrifices the testing accuracy by 3-4%. This reduced test accuracy is

common among local SGD methods, which has been studied theoretically; see e.g., [27].

All LASG algorithms has the same iteration complexity as SGD and outperform local-SGD in most cases. Compared with SGD, LASG-WK2 and LASG-PSE reduce communication rounds by about one order of magnitude for neural network training and even more for logistic regression. LASG-WK1 also reduce communication by more than one order of magnitude for logistic regression. Based on the results of LAG-WK, it is evident that the selection rules (9), (12) and (16) achieve more significant improvement in terms of saving communication than the selection rule (7) of LAG-WK. Moreover, the performance of LAQSG validates that LASG can be easily equipped with stochastic quantization with extra benefits from quantization.

V. Concluding Remarks

In this paper, we developed a class of communication-efficient variants of SGD that we term LASG. The LASG methods leverage a set of adaptive communication rules to detect and then skip less informative or redundant communication rounds between the server and workers during distributed learning. To further reduce communication bandwidth, the quantized version of LASG is also presented. Both LASG and their quantized version are simple to implement, and have convergence rate comparable to the original SGD.

Future research includes studying the proximal extension of LASG for certain nonsmooth problems, and developing the primal-dual counterpart of LASG for distributed training of generative adversarial networks (GANs). While the setting in this paper considers a central server and a set of workers in a communication error-free system, developing the decentralized version of LASG for learning without a central server is in our future agenda. Studying the effect of errors induced by noisy wireless channels on distributed learning is also of great practical importance, which can be typically tackled by controlling the transmit power to maintain a desired SNR [49].

VI. Derivations of Missing Steps in Section II

We will provide the detailed derivations of some missing steps in Section II We define an auxiliary function as

$$\psi_m(\theta) := \mathcal{L}_m(\theta) - \mathcal{L}_m(\theta^\star) - \left\langle \nabla\mathcal{L}_m(\theta^\star), \theta - \theta^\star \right\rangle$$

where $\theta^\star$ is a minimizer of $\mathcal{L}(\theta)$. Assume that $\nabla\ell(\theta;\xi_m)$ is $\bar{L}$-Lipschitz continuous for all ξ_m, we have $\|\nabla\ell(\theta;\xi_m) - \nabla\ell(\theta^\star;\xi_m)\|^2 \le 2\bar{L}(\ell(\theta;\xi_m) - \ell(\theta^\star;\xi_m) - \langle\nabla\ell(\theta^\star;\xi_m), \theta-\theta^\star\rangle)$. Taking expectation with respect to ξ_m, we can obtain

$$\mathbb{E}_{\xi_m}[\|\nabla\ell(\theta;\xi_m) - \nabla\ell(\theta^\star;\xi_m)\|^2] \le 2\bar{L}\left(\mathcal{L}_m(\theta) - \mathcal{L}_m(\theta^\star) - \left\langle \nabla\mathcal{L}_m(\theta^\star), \theta-\theta^\star\right\rangle\right) = 2\bar{L}\psi_m(\theta). \tag{26}$$

Note that $\nabla\mathcal{L}_m$ is also $\bar{L}$-Lipschitz continuous and thus

$$\|\nabla\mathcal{L}_m(\theta) - \nabla\mathcal{L}_m(\theta^\star)\|^2 \le 2\bar{L}\psi_m(\theta).$$

1) Derivations of (8): By (38), we can derive that $\|\theta_1\|^2 \ge \frac{1}{2}\|\theta_1+\theta_2\|^2 - \|\theta_2\|^2$. As a consequence, we can obtain

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2\right]\\
\ge\ &\frac{1}{2}\mathbb{E}\left[\left\|\left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\mathcal{L}_m(\theta^k)\right)\right.\right.\\
&\left.\left.+\left(\nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right)\right\|^2\right]\\
&-\mathbb{E}\left[\left\|\nabla\mathcal{L}_m(\theta^k) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\|^2\right]\\
=\ &\frac{1}{2}\mathbb{E}\left[\left\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\mathcal{L}_m(\theta^k)\right\|^2\right]\\
&+\frac{1}{2}\mathbb{E}\left[\left[\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\|^2\right]\right]\\
&+\mathbb{E}\left[\left\langle\nabla\ell(\theta^k;\xi_m^k) - \nabla\mathcal{L}_m(\theta^k), \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right.\right.\\
&\left.\left.-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\
&-\mathbb{E}\left[\left\|\nabla\mathcal{L}_m(\theta^k) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\|^2\right].
\end{aligned}$$

To obtain (8), we use that

$$\left\langle \underbrace{\mathbb{E}\left[\nabla\ell(\theta^k;\xi_m^k)\middle|\Theta^k\right]}_{=\nabla\mathcal{L}_m(\theta^k)} - \nabla\mathcal{L}_m(\theta^k), \nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle = 0.$$

2) Derivations of (11): Recall that

$$\begin{aligned}
&\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\\
=\ &\left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\tilde{\theta};\xi_m^k) + \nabla\mathcal{L}_m(\tilde{\theta})\right)\\
&-\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k}) + \nabla\mathcal{L}_m(\tilde{\theta})\right)\\
=\ &\underbrace{\left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\tilde{\theta};\xi_m^k) + \nabla\psi_m(\tilde{\theta})\right)}_{:=g_m^k}\\
&-\underbrace{\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k}) + \nabla\psi_m(\tilde{\theta})\right)}_{:=g_m^{k-\tau_m^k}}.
\end{aligned}$$

And by (38), we have $\|\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\|^2 \le 2\|g_m^k\|^2 + 2\|g_m^{k-\tau_m^k}\|^2$. We decompose the first term as

$$\begin{aligned}
\mathbb{E}[\|g_m^k\|^2] &\le 2\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k)\|^2]\\
&\quad+2\mathbb{E}[\|\nabla\ell(\tilde{\theta};\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k) - \nabla\psi_m(\tilde{\theta})\|^2]\\
&=2\mathbb{E}[\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k)\|^2|\Theta^k]]\\
&\quad+2\mathbb{E}[\|\nabla\ell(\tilde{\theta};\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k)\\
&\quad-\mathbb{E}[\nabla\ell(\tilde{\theta};\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k)|\Theta^k]\|^2]\\
&\le 4\bar{L}\mathbb{E}[\psi_m(\theta^k)] + 2\mathbb{E}[\|\nabla\ell(\tilde{\theta};\xi_m^k) - \nabla\ell(\theta^\star;\xi_m^k)\|^2]\\
&\overset{(a)}{\le} 4\bar{L}\mathbb{E}[\psi_m(\theta^k)] + 4\bar{L}\mathbb{E}\psi_m(\tilde{\theta}).
\end{aligned}$$

where (a) follows from (26).

By nonnegativity of ψ_m, we have

$$\mathbb{E}[\|g_m^k\|^2] \le 4\bar{L}\sum_{m\in\mathcal{M}}\mathbb{E}\psi_m(\theta^k) + 4\bar{L}\sum_{m\in\mathcal{M}}\mathbb{E}\psi_m(\tilde{\theta})$$
$$= 4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 4M\bar{L}(\mathbb{E}\mathcal{L}(\tilde{\theta}) - \mathcal{L}(\theta^\star)). \quad (27)$$

Similarly, we can prove

$$\mathbb{E}[\|g_m^{k-\tau_m^k}\|^2] \le$$
$$4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)) + 4M\bar{L}(\mathbb{E}\mathcal{L}(\tilde{\theta}) - \mathcal{L}(\theta^\star)). \quad (28)$$

Therefore, it follows that

$$\mathbb{E}[\|\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\|^2] \le 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star))$$
$$+ 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)) + 16M\bar{L}(\mathbb{E}\mathcal{L}(\tilde{\theta}) - \mathcal{L}(\theta^\star)).$$

3) Derivations of (13): The LHS of (12) can be written as

$$\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)$$
$$= \left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right)$$
$$- \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})$$
$$= \left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\psi_m(\theta^{k-\tau_m^k})\right)$$
$$- \nabla\psi_m(\theta^{k-\tau_m^k}).$$

Similar to (27), we can obtain

$$\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\psi_m(\theta^{k-\tau_m^k})\|^2]$$
$$\le 4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)).$$

Combined with the fact

$$\mathbb{E}[\|\nabla\psi_m(\theta^{k-\tau_m^k})\|^2] = \mathbb{E}[\|\nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) - \nabla\mathcal{L}_m(\theta^\star)\|^2]$$
$$\le 2\bar{L}\mathbb{E}\psi_m(\theta^{k-\tau_m^k})$$
$$\le 2M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star))$$

we have

$$\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)\|^2]$$
$$\le 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 12M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)).$$

4) Derivations of (15): Expanding LASG update, we have

$$\mathbb{E}\left[\|\theta^k - \theta^{k-\tau_m^k}\|^2\right]$$
$$= \frac{1}{M^2}\mathbb{E}\left[\left\|\sum_{d=1}^{\tau_m^k}\sum_{m\in\mathcal{M}}\eta_{k-d}\nabla\ell(\theta^{k-d-\tau_m^{k-d}};\xi_m^{k-d-\tau_m^{k-d}})\right\|^2\right]$$
$$\le \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\eta_{k-d}^2\,\mathbb{E}\left[\left\|\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-d-\tau_m^{k-d}};\xi_m^{k-d-\tau_m^{k-d}})\right\|^2\right]$$

where we used the Cauchy-Schwartz inequality.

Using $\mathbb{E}[\|A - \mathbb{E}[A]\|^2] + \|\mathbb{E}[A]\|^2 = \mathbb{E}[\|A\|^2]$, we have

$$\mathbb{E}\left[\|\theta^k - \theta^{k-\tau_m^k}\|^2\right] = \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\eta_{k-d}^2$$
$$\times\mathbb{E}\left[\left\|\sum_{m\in\mathcal{M}}\left(\nabla\ell(\theta^{k-d-\tau_m^{k-d}};\xi_m^{k-d-\tau_m^{k-d}}) - \nabla\mathcal{L}_m(\theta^{k-d-\tau_m^{k-d}})\right)\right\|^2\right]$$
$$+ \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\eta_{k-d}^2\,\mathbb{E}\left[\left\|\sum_{m\in\mathcal{M}}\nabla\mathcal{L}_m(\theta^{k-d-\tau_m^{k-d}})\right\|^2\right]$$
$$\le \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\eta_{k-d}^2\sum_{m\in\mathcal{M}}\sigma_m^2 + \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\eta_{k-d}^2\,\mathbb{E}\left[\left\|\sum_{m\in\mathcal{M}}\nabla\mathcal{L}_m(\theta^{k-d-\tau_m^{k-d}})\right\|^2\right]$$
$$\le \frac{\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\sum_{m\in\mathcal{M}}\sigma_m^2\eta_{k-d}^2 + \frac{2\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^{k-d})\right\|^2\right]\eta_{k-d}^2$$
$$+ \frac{2\tau_m^k}{M^2}\sum_{d=1}^{\tau_m^k}\sum_{m\in\mathcal{M}}L_m^2\,\mathbb{E}\left[\left\|\theta^{k-d} - \theta^{k-d-\tau_m^{k-d}}\right\|^2\right]\eta_{k-d}^2.$$

We arrive at our statement since $\tau_m^k \le D$ and $\eta_{k-d} \le \eta_{k-D}$.

VII. Proofs of Main Theorems

We first highlight the key steps, present some supporting lemmas that will be used frequently in the subsequent analysis, which is followed by the proofs of the results in Section III.

A. Key Steps of Lyapunov Analysis

With these assumptions, LASG will yield descent of $\mathcal{L}(\theta^k)$.

Lemma 1: Under Assumptions 1, 2 and 3, $\{\theta^k\}$ generated by Algorithms 1, 2 and 3 satisfy

$$\mathbb{E}[\mathcal{L}(\theta^{k+1})] - \mathbb{E}[\mathcal{L}(\theta^k)] \le -\left(\eta_k - \frac{L\eta_k^2}{2}\right)\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]$$
$$+ \frac{L\eta_k^2}{2}\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\mathcal{L}(\theta^k)\right\|^2\right]$$
$$+ \frac{(\eta_k - L\eta_k^2)}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k), \delta_m^k\right\rangle\right] \quad (29)$$

where $\delta_m^k := \nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})$.

Among three terms in the RHS of (29): the first term resembles the standard unbiased stochastic descent; the second term captures the variance of the *stale* aggregated stochastic gradient; and, the last term quantifies the correlations between the gradient direction $\nabla\mathcal{L}(\theta^k)$ and the error induced by the *stale* stochastic gradient ∇^k.

Analyzing the progress of $\mathcal{L}(\theta^k)$ under LASG is challenging. Below we characterize the regularity of the stale stochastic gradients ∇^k, which lays the foundation for incorporating the properly controlled staleness into the SGD update.

Lemma 2: Under Assumptions 1 and 2, if the stepsizes satisfy $\eta_{k+1} \le \eta_k \le 1/L$, then we have

$$\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k), \delta_m^k\right\rangle\right] \le \frac{L\eta_k}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right] + \frac{6DL\eta_k}{2M\sqrt{M}}\sum_{m\in\mathcal{M}}\sigma_m^2$$
$$+ \sum_{d=1}^{D}\left(\frac{c}{2L\eta_k d_{\max}} + \frac{\sqrt{M}L}{12\eta_k}\right)\mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right]. \quad (30)$$

Lemma 2 justifies the relevance of the stale yet properly selected stochastic gradients. Intuitively, the first term in the RHS of (30) will reduces the magnitude of descent in (29), and

the second and third terms will diminish if the stepsizes are diminishing since $\mathbb{E}[\|\theta^k-\theta^{k-1}\|^2]=\mathcal{O}(\eta_k^2)$.

The next lemma implies that the variance of the *stale* aggregated stochastic gradient reduces to that of standard SGD if the stepsizes are diminishing since $\mathbb{E}[\|\theta^k-\theta^{k-1}\|^2]=\mathcal{O}(\eta_k^2)$.

Lemma 3: Under Assumptions 1 and 2, if the stepsizes satisfy $\eta_{k+1}\le\eta_k\le 1/L$, then we have

$$\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}(\theta^k)\right\|^2\right]$$
$$\le\frac{3c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\|\theta^{k+1-d}-\theta^{k-d}\|^2+\frac{9}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2. \quad (31)$$

In view of Lemmas 1-3, we introduce the following **Lyapunov function** to capture the progress of LASG:

$$V^k:=\mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star)+\sum_{d=1}^{D}\gamma_d\|\theta^{k+1-d}-\theta^{k-d}\|^2 \quad (32)$$

where $\{\gamma_d\}_{d=1}^D$ are constants to be determined later. The following lemma is a direct application of Lemmas 1–3.

Lemma 4: Under Assumptions 1 and 2, there exist nonnegative constants $\{A_d^k\}_{d=1}^D$, B_0^k and B_1^k such that

$$\mathbb{E}[V^{k+1}]-\mathbb{E}[V^k]\le -B_0^k\,\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+B_1^k\sum_{m=1}^{M}\sigma_m^2$$
$$-\sum_{d=1}^{D}A_d^k\,\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]. \quad (33)$$

The constants $\{A_d^k\}_{d=1}^D$, B_0^k and B_1^k depend on the stepsize η_k, the threshold c and the parameters $\{\gamma_d\}_{d=1}^D$. Their expressions are specified in the proof. By choosing proper η_k and c, we are able to ensure the convergence of LASG.

B. Supporting Lemmas

Define the σ-algebra $\Theta^k=\{\theta^l,1\le l\le k\}$. For convenience, we initialize parameters as $\theta^{-D}=\cdots=\theta^{-1}=\theta^0$, and define the difference between θ^{k+1-d} and θ^{k-d} as

$$\Delta^{k-d}:=\theta^{k+1-d}-\theta^{k-d} \quad (34)$$

which implies that $\Delta^k:=\theta^{k+1}-\theta^k$.

Some basic facts used in the proof are reviewed as follows.

Fact 1. Assume that $X_1,X_2,\ldots,X_n\in\mathbb{R}^p$ are independent random variables, and $EX_1=\cdots=EX_n=0$. Then

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n}X_i\right\|^2\right]=\sum_{i=1}^{n}\mathbb{E}\left[\|X_i\|^2\right]. \quad (35)$$

Fact 2. (Young's inequality) For any $\theta_1,\theta_2\in\mathbb{R}^p,\varepsilon>0$,

$$\left\langle\theta_1,\theta_2\right\rangle\le\frac{\|\theta_1\|^2}{2\varepsilon}+\frac{\varepsilon\|\theta_2\|^2}{2}. \quad (36)$$

As a consequence, we have

$$\|\theta_1+\theta_2\|^2\le\left(1+\frac{1}{\varepsilon}\right)\|\theta_1\|^2+(1+\varepsilon)\|\theta_2\|^2. \quad (37)$$

Fact 3. (Cauchy-Schwartz) For $\theta_1,\ldots,\theta_n\in\mathbb{R}^p$, we have

$$\left\|\sum_{i=1}^{n}\theta_i\right\|^2\le n\sum_{i=1}^{n}\|\theta_i\|^2. \quad (38)$$

Lemma 5: For $k-D\le l\le k-\tau_m^k$, if $\{\theta^k\}$ are the iterates generated by LASG, we have

$$\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\rangle\right]$$
$$\le\frac{\sqrt{M}L}{12\eta_k}\sum_{d=1}^{D}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{6DL\eta_k}{\sqrt{M}}\sigma_m^2 \quad (39)$$

and similarly, we have

$$\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\mathcal{L}_m(\theta^l)-\nabla\ell(\theta^l;\theta^{k-\tau_m^k})\right\rangle\right]$$
$$\le\frac{\sqrt{M}L}{12\eta_k}\sum_{d=1}^{D}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{3DL\eta_k}{\sqrt{M}}\sigma_m^2. \quad (40)$$

Proof: We first prove (39) by decomposing it as

$$\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\rangle\right]$$
$$\overset{(a}{=}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k)-\nabla\mathcal{L}(\theta^l),\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\rangle\right]$$
$$\le L\mathbb{E}\left[\|\theta^k-\theta^l\|\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\|\right]$$
$$\overset{(b)}{\le}\frac{\sqrt{M}L}{12D\eta_k}\underbrace{\mathbb{E}\left[\|\theta^k-\theta^l\|^2\right]}_{:=T_1}$$
$$+\frac{6DL\eta_k}{2\sqrt{M}}\underbrace{\mathbb{E}\left[\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\|^2\right]}_{:=T_2} \quad (41)$$

where (a) holds due to

$$\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^l),\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\rangle\right]$$
$$=\mathbb{E}\left[\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^l),\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\rangle\middle|\Theta^l\right]\right]$$
$$=\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^l),\mathbb{E}\left[\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})|\Theta^l\right]\right\rangle\right]$$
$$=\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^l),\nabla\mathcal{L}_m(\theta^l)-\nabla\mathcal{L}_m(\theta^l)\right\rangle\right]=0$$

and (b) is a direct application of Fact 2.

Applying Fact 3 to T_1, we have

$$T_1=\mathbb{E}\left[\left\|\sum_{d=1}^{k-l}\Delta^{k-d}\right\|^2\right]\le(k-l)\sum_{d=1}^{k-l}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]$$
$$\le D\sum_{d=1}^{D}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right] \quad (42)$$

and applying Fact 1 to T_2, we have

$$T_2=\mathbb{E}\left[\left\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\|^2\right]$$
$$=\mathbb{E}\Big[\Big\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\mathcal{L}_m(\theta^l)+\nabla\mathcal{L}_m(\theta^l)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\Big\|^2\Big]$$
$$=\mathbb{E}\left[\left\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\mathcal{L}_m(\theta^l)\right\|^2\right]$$
$$+\mathbb{E}\left[\left\|\nabla\mathcal{L}_m(\theta^l)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right\|^2\right]\le 2\sigma_m^2 \quad (43)$$

where the last inequality uses Assumption 2. Plugging (42) and (43) into (41), it leads to (39).

Likewise, following the steps to (41), it can be verified that (40) also holds true.

C. Proof of Lemma 1

Due to the smoothness of $\mathcal{L}(\theta)$ in Assumption 1, we have

$$\begin{aligned}&\mathbb{E}\left[\mathcal{L}(\theta^{k+1})\right]-\mathbb{E}\left[\mathcal{L}(\theta^k)\right]\\ &\le \eta_k\underbrace{\mathbb{E}\left[-\left\langle\nabla\mathcal{L}(\theta^k),\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]}_{:=I_1}\\ &\quad+\frac{L\eta_k^2}{2}\underbrace{\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2\right]}_{:=I_2}. \qquad (44)\end{aligned}$$

With $\delta_m^k:=\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})$ denoting the stochastic gradient innovation, we decompose I_1 as

$$\begin{aligned}I_1=&-\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k;\xi_m^k)\right\rangle\right]\\ &+\underbrace{\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\delta_m^k\right\rangle\right]}_{:=H_1}\\ =&-\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\nabla\ell(\theta^k;\xi_m^k)\big|\,\Theta^k\right]\right\rangle\right]+H_1\\ =&-\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+H_1 \qquad (45)\end{aligned}$$

and likewise decompose I_2 as

$$\begin{aligned}I_2=&\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}(\theta^k)+\nabla\mathcal{L}(\theta^k)\right\|^2\right]\\ =&\underbrace{\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}(\theta^k)\right\|^2\right]}_{:=H_2}\\ &+\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]-2\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k;\xi_m^k)\right\rangle\right]\\ &+2\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\ =&H_2+\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]-2H_1. \qquad (46)\end{aligned}$$

We obtain Lemma 1 by plugging (45) and (46) into (44).

D. Proof of Lemma 2

We next bound H_1 defined in (45) separately for different LASG rules. First for LASG-WK1's rule (9), we have

$$\begin{aligned}H_1\overset{(a)}{=}&\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\tilde{\delta}_m^k-\tilde{\delta}_m^{k-\tau_m^k}\right\rangle\right]\\ &+\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\tilde{\theta};\xi^k)-\nabla\ell(\tilde{\theta},\xi_m^{k-\tau_m^k})\right\rangle\right]\\ \overset{(b)}{\le}&\frac{L\eta_k}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right]+\frac{6DL\eta_k}{M\sqrt{M}}\sum_{m\in\mathcal{M}}\sigma_m^2\\ &+\sum_{d=1}^{D}\left(\frac{c}{2L\eta_k d_{\max}}+\frac{\sqrt{M}L}{12\eta_k}\right)\mathbb{E}\left[\left\|\Delta^{k-d}\right\|^2\right]\end{aligned}$$

where (a) is due to the definition of δ_m^k, and (b) is obtained by (9), (36) with $\varepsilon=\frac{1}{L\eta_k}$, and (39) with $\theta^l=\tilde{\theta}$. Note that the definition of $\tilde{\theta}$ in Algorithm 1 implies $l=\lfloor\frac{k}{D}\rfloor\le k-\tau_m^k$.

For LASG-WK2's rule (12), we apply (36) with $\varepsilon=\frac{1}{L\eta_k}$ and (39) with $l=k-\tau_m^k$, which leads to

$$\begin{aligned}H_1=&\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)\right\rangle\right]\\ &+\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\ \le&\frac{L\eta_k}{2}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+\frac{6DL\eta_k}{M\sqrt{M}}\sum_{m\in\mathcal{M}}\sigma_m^2\\ &+\sum_{d=1}^{D}\left(\frac{c}{2L\eta_k d_{\max}}+\frac{\sqrt{M}L}{12\eta_k}\right)\mathbb{E}\left[\|\Delta^{k-d}\|^2\right].\end{aligned}$$

For LASG-PS's rule (14), applying $\mathbb{E}[\nabla\ell(\theta^k;\xi_m^k)|\Theta^k]=\nabla\mathcal{L}_m(\theta^k)$, we get

$$\begin{aligned}H_1=&\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\mathcal{L}_m(\theta^k)-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\rangle\right]\\ &+\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\ \overset{(c)}{\le}&\frac{L\eta_k}{2}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+\frac{6DL\eta_k}{2M\sqrt{M}}\sum_{m\in\mathcal{M}}\sigma_m^2\\ &+\sum_{d=1}^{D}\left(\frac{c}{2L\eta_k d_{\max}}+\frac{\sqrt{M}L}{12\eta_k}\right)\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]\end{aligned}$$

where (c) uses (36) with $\varepsilon=\frac{1}{L\eta_k}$ and (40) with $l=k-\tau_m^k$.

E. Proof of Lemma 3

We next bound H_2 defined in (46) separately for different LASG rules. For LASG-WK1, using (38), we first have

$$\begin{aligned}H_2\le&3\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\tilde{\delta}_m^k-\tilde{\delta}_m^{k-\tau_m^k}\right\|^2\right]\\ &+3\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k,\xi_m^k)-\nabla\mathcal{L}(\theta^k))\right\|^2\right]\\ &+3\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}(\nabla\ell(\tilde{\theta};\xi_m^k)-\nabla\mathcal{L}_m(\tilde{\theta}))\right.\right.\\ &\left.\left.+\frac{1}{M}\sum_{m\in\mathcal{M}}(\nabla\mathcal{L}_m(\tilde{\theta})-\nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k}))\right\|^2\right]\\ \overset{(a)}{\le}&\frac{3c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{9}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2\end{aligned}$$

where (a) follows from (9), (20b), and (35).

For LASG-WK2, using (38), we have

$$H_2 \le 2\mathbb{E}\left[\Big\|\frac{1}{M}\sum_{m\in\mathcal{M}}\Big(\nabla\ell(\theta^{k-\tau_m^k},\xi_m^{k-\tau_m^k})-\nabla\ell(\theta^k;\xi_m^k)\Big)\Big\|^2\right] + 2\mathbb{E}\left[\Big\|\frac{1}{M}\sum_{m\in\mathcal{M}}\big(\nabla\ell(\theta^k;\xi_m^k)-\nabla\mathcal{L}_m(\theta^k)\big)\Big\|^2\right]$$
$$\overset{(b)}{\le} \frac{2c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{2}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2$$

where (b) uses (12), (20b) and (35).

For LASG-PS, using (38), we have

$$H_2 \le 2\mathbb{E}\left[\Big\|\sum_{m\in\mathcal{M}}\Big(\nabla\ell(\theta^{k-\tau_m^k},\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\Big)\Big\|^2\right] + 2\mathbb{E}\left[\Big\|\sum_{m\in\mathcal{M}}\Big(\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})-\nabla\mathcal{L}_m(\theta^k)\Big)\Big\|^2\right]$$
$$\overset{(c)}{\le} \frac{2c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{2}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2$$
$$\le \frac{3c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\|\Delta^{k-d}\|^2+\frac{9}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2$$

where (c) holds due to (14), (20b), and (35).

F. Proof of Lemma 4

Plugging Lemmas 2 and 3 into Lemma 1 leads to

$$\begin{aligned}&\mathbb{E}\left[\mathcal{L}(\theta^{k+1})\right]-\mathbb{E}\left[\mathcal{L}(\theta^k)\right]\\ \le&-\left(\eta_k-L\eta_k^2+\frac{L^2\eta_k^3}{2}\right)\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]\\ &+\sum_{d=1}^{D}\left(\left(\frac{3\eta_k}{2d_{\max}}+\frac{1-L\eta_k}{2Ld_{\max}}\right)c+\frac{\sqrt{M}L}{12}\right)\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]\\ &+L\eta_k^2\left(\frac{9}{2}+6\sqrt{M}D\right)\frac{1}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2 \qquad (47)\end{aligned}$$

where we use the fact that $L\eta_k \le 1$.

By definition of $\mathbb{E}[V^k]$, it follows that (with $\gamma_{D+1}=0$)

$$\begin{aligned}&\mathbb{E}[V^{k+1}]-\mathbb{E}[V^k]=\mathbb{E}\left[\mathcal{L}(\theta^{k+1})\right]-\mathbb{E}\left[\mathcal{L}(\theta^k)\right]\\ &+\gamma_1\mathbb{E}\left[\|\Delta^k\|^2\right]+\sum_{d=1}^{D}(\gamma_{d+1}-\gamma_d)\mathbb{E}\left[\|\Delta^{k-d}\|^2\right].\end{aligned}$$

First we decompose $\mathbb{E}[\|\Delta^k\|^2]$ as

$$\begin{aligned}&\frac{1}{\eta_k^2}\mathbb{E}\left[\|\Delta^k\|^2\right]\\ &=\mathbb{E}\left[\Big\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}(\theta^k)+\nabla\mathcal{L}(\theta^k)\Big\|^2\right]\\ &\le 2\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]\\ &\quad+2\mathbb{E}\left[\Big\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}(\theta^k)\Big\|^2\right]\\ &\overset{(a)}{\le} 2\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+\frac{6c}{d_{\max}}\sum_{d=1}^{D}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{18}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2\end{aligned}$$

where (a) uses Lemma 3.

Together with (47), it follows that

$$\begin{aligned}&\mathbb{E}[V^{k+1}]-\mathbb{E}[V^k]\le-\underbrace{\left(\eta_k-(L+2\gamma_1)\eta_k^2\right)}_{:=B_0^k}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]\\ &+\sum_{d=1}^{D}\underbrace{\left(\left(\eta_k+\frac{1}{2L}\right)\frac{c}{d_{\max}}+\frac{\sqrt{M}L}{12}+\frac{6c\gamma_1\eta_k^2}{d_{\max}}+\gamma_{d+1}-\gamma_d\right)}_{:=A_d^k}\\ &\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]\\ &+\underbrace{\left(\left(\frac{9}{2}+6\sqrt{M}D\right)L+18\gamma_1\right)}_{:=B_1^k}\frac{\eta_k^2}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2 \qquad (48)\end{aligned}$$

from which the proof is complete.

G. Proof of Theorem 1

To ensure $A_d^k \le 0$ in (48) of Lemma 4, it is sufficient to choose $\{\gamma_d\}$ satisfying (with $\gamma_{D+1}=0$)

$$\left(\eta_k+\frac{1}{2L}\right)\frac{c}{d_{\max}}+\frac{\sqrt{M}L}{12}+\frac{6c\gamma_1\eta_k^2}{d_{\max}}+\gamma_{d+1}-\gamma_d\le 0, 0\le d\le D$$

where the stepsize is chosen as $\eta_k=\eta$, $k=1,\ldots,K$.

Solve the linear equations above and get

$$\gamma_1=\frac{(\eta+\frac{1}{2L})cD/d_{\max}+\frac{\sqrt{M}DL}{12}}{1-6cD\eta^2/d_{\max}}. \qquad (49)$$

Select $c\le\min\{\frac{d_{\max}}{12D\eta^2},\frac{d_{\max}\sqrt{M}L^2}{18}\}$ such that $\gamma_1\le\frac{\sqrt{M}DL}{3}$. If we further select $\eta\le\frac{1}{2L+\frac{4}{3}\sqrt{M}DL}\le\frac{1}{2L+4\gamma_1}$ and then

$$B_0^k=\eta_k-(L+2\gamma_1)\eta_k^2\ge\frac{\eta}{2}. \qquad (50)$$

Summation up (33) over $k=0,\ldots,K-1$, it follows that

$$\begin{aligned}\sum_{k=0}^{K-1}\frac{\eta_k}{2}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]&\le\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*)\\ &+\sum_{k=0}^{K-1}\left(\frac{9}{2}+12\sqrt{M}D\right)\frac{L\eta_k^2}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2. \qquad (51)\end{aligned}$$

Specifically, if we choose a constant stepsize

$$\eta_k=\eta:=\min\left\{\frac{1}{2L+\frac{4}{3}\sqrt{M}DL},\frac{c_\eta}{\sqrt{K}}\right\} \qquad (52)$$

where $c_\eta>0$ is a constant, then

$$\begin{aligned}&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]\\ &\le\frac{2}{K\eta}\left(\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*)+K\left(\frac{9}{2}+12\sqrt{M}D\right)\frac{L\eta^2}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2\right)\end{aligned}$$

$$\leq \left(\frac{4L+\frac{8}{3}\sqrt{M}DL}{K}+\frac{2}{c_\eta\sqrt{K}}\right)(\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*)) + \frac{c_\eta}{\sqrt{K}}\left(9+24\sqrt{M}D\right)\frac{L}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2. \quad (53)$$

Choosing $c_\eta=\mathcal{O}(M^{\frac{3}{4}}(\sum_{m\in\mathcal{M}}\sigma_m^2)^{-\frac{1}{2}})$ leads to the theorem.

H. Proof of Theorem 2

Let $\mathbb{E}_Q$ and $\mathbb{E}_{Q,\xi_m}$ denote the expectation with respect to the stochastic quantization Q and both the stochastic quantization Q and the datum ξ_m, respectively.

As a result of [13, Lemma 3.1] and Assumption 4, b-bit quantized gradients have the following unbiasedness property

$$\mathbb{E}_Q\left[Q(\theta;\xi_m)\right]=\nabla\ell(\theta;\xi_m) \quad (54)$$

and the bounded variance (with B defined in Assumption 4)

$$\mathbb{E}_{Q,\xi_m}\left[\|Q(\theta;\xi_m)-\nabla\ell(\theta;\xi_m)\|^2\right] \leq \min\left\{\frac{d}{(2^{b-1}-1)^2},\frac{\sqrt{d}}{2^{b-1}-1}\right\}B=:\sigma_Q^2. \quad (55)$$

Analogous to the proof of Lemma 1, we can get

$$\mathbb{E}\left[\mathcal{L}(\theta^{k+1})\right]-\mathbb{E}\left[\mathcal{L}(\theta^k)\right]\leq -\left(\eta_k-\frac{L\eta_k^2}{2}\right)\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right] +\left(\eta_k-L\eta_k^2\right)H_3+\frac{L\eta_k^2}{2}H_4$$

where H_3 and H_4 are defined similar to H_1 and H_2 in (44).

We first bound H_3 as

$$\begin{aligned}H_3&:=\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\\&\times\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^k;\xi_m^k)-Q(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\&=H_1+\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\\&\times\left[\left\langle\nabla\mathcal{L}(\theta^k),\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-Q(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\rangle\right]\\&\overset{(a)}{\leq}H_1+\frac{\sqrt{M}L}{12\eta_k}\sum_{d=1}^{D}\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{6DL\eta_k}{2M\sqrt{M}}\sigma_Q^2 \quad (56)\end{aligned}$$

where (a) is obtained by steps similar to those of (39).

Plugging the bound on H_1 in Lemma 2 into (56), we have

$$H_3\leq\frac{L\eta_k}{2}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]+\sum_{d=1}^{D}\left(\frac{c/d_{\max}}{2L\eta_k}+\frac{\sqrt{M}L}{6\eta_k}\right)\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]+\frac{6DL\eta_k}{\sqrt{M}}\sum_{m\in\mathcal{M}}\left(\sigma_m^2+\frac{\sigma_Q^2}{2}\right).$$

Likewise, H_4 can be bounded as

$$\begin{aligned}H_4&=\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)-\frac{1}{M}\sum_{m\in\mathcal{M}}Q(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2\right]\\&\overset{(b)}{\leq}4\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-Q(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2\right]+\frac{4}{3}H_2\\&\overset{(c)}{\leq}\frac{4c}{d_{\max}}\sum_{d=1}^{d_{\max}}\mathbb{E}\|\Delta^{k-d}\|^2+\frac{12}{M^2}\sum_{m\in\mathcal{M}}\left(\sigma_m^2+\frac{\sigma_Q^2}{2}\right)\end{aligned}$$

where (b) uses (37) with $\varepsilon=3$, and (c) uses Lemma 3.

The remaining steps follow those of Theorem 1 with σ_m^2 replaced with $\sigma_m^2+\frac{\sigma_Q^2}{2}$.

I. Proof of Theorem 3

Using the PL-condition of $\mathcal{L}(\theta)$, (33) can be rewritten as

$$\mathbb{E}[V^{k+1}]-\mathbb{E}[V^k]\leq -2\mu B_0^k\mathbb{E}[\mathcal{L}(\theta^k)-\mathcal{L}(\theta^*)]+B_1^k\sum_{m\in\mathcal{M}}\sigma_m^2+\sum_{d=1}^{D}A_d^k\mathbb{E}\left[\|\Delta^{k-d}\|^2\right]. \quad (57)$$

If we choose γ_d such that $A_d^k\leq-2\mu B_0^k\gamma_d$ for $d=1,2\ldots,D$, then we have

$$\begin{aligned}\mathbb{E}[V^{k+1}]&\leq(1-2\mu B_0^k)\mathbb{E}[V^k]+B_1^k\frac{1}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2\\&\leq\prod_{j=0}^{k}(1-2\mu B_0^j)V^0+\sum_{j=0}^{k}B_1^j\prod_{i=j+1}^{k}\frac{1-2\mu B_0^i}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2.\end{aligned} \quad (58)$$

To ensure $A_d^k\leq-2\mu B_0^k\gamma_d$, note that if $\eta_k\leq\eta\leq\frac{1}{L+2\gamma_1}$, then

$$B_0^k=\eta_k-(L+2\gamma_1)\eta_k^2\in[0,\eta_k]. \quad (59)$$

Hence, it is sufficient to choose γ_d satisfying ($\gamma_{D+1}=0$)

$$\left(\eta_k+\frac{1}{2L}\right)\frac{c}{d_{\max}}+\frac{\sqrt{M}L}{12}+\frac{6c\gamma_1\eta_k^2}{d_{\max}}+\gamma_{d+1}-\gamma_d\leq-2\mu\eta\gamma_1,\ \forall d.$$

Solve the linear equations above and get

$$\gamma_1=\frac{(\eta+\frac{1}{2L})cD/d_{\max}+\sqrt{M}DL/12}{1-6cD\eta^2/d_{\max}-2\mu D\eta}. \quad (60)$$

Let $\eta_k=\frac{2}{\mu(k+K_0)}$ with $K_0=\max\{\frac{2(L+\frac{2}{3}\sqrt{M}DL)}{\mu},16D\}$, which ensures that

$$\eta_k\leq\eta:=\min\left\{\frac{1}{L+2\gamma_1},\frac{1}{8\mu D}\right\}. \quad (61)$$

Together with the selection $c\leq\min\{\frac{d_{\max}}{24D\eta^2},\frac{d_{\max}\sqrt{M}L^2}{18}\}$, this ensures that $\gamma_1\leq\frac{\sqrt{M}DL}{3}$.

Plugging into (58) leads to

$$\mathbb{E}[V^{k+1}]\leq(1-\mu\eta_k)\mathbb{E}[V^k]+\underbrace{\left(\frac{9}{2}+12\sqrt{M}D\right)\frac{L}{M^2}\sum_{m\in\mathcal{M}}\sigma_m^2\eta_k^2}_{:=R}.$$

Multiplying over $k = 0, \ldots, K-1$, it follows that

$$\begin{aligned}\mathbb{E}[V^K] &\leq \prod_{k=0}^{K-1}(1-\mu\eta_k)V^0 + R\sum_{k=0}^{K-1}\eta_k^2\prod_{j=k+1}^{K-1}(1-\mu\eta_j)\\ &\leq \frac{(K_0-2)(K_0-1)}{(K+K_0-2)(K+K_0-1)}V^0\\ &\quad + \frac{R}{\mu^2}\sum_{k=0}^{K-1}\frac{4}{(k+K_0)^2}\frac{(k+K_0-1)(k+K_0)}{(K+K_0-2)(K+K_0-1)}\\ &\leq \frac{(K_0-1)^2}{(K+K_0-1)^2}V^0 + \frac{4RK}{\mu^2(K+K_0-1)^2}. \qquad (62)\end{aligned}$$

Using the definition of V^0 and the initialization $\theta^{-D} = \cdots = \theta^{-1} = \theta^0$, we complete the proof.

REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automat. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[3] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication and Imaging, Science and Engineering*, New York, NY, USA: Springer, 2016.

[4] J. Dean *et al.* "Large scale distributed deep networks," in *Proc. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1223–1231.

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

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

[7] A. Nedić, A. Olshevsky, and M. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.

[8] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. Amer. Stat. Assoc.*, vol. 114, no. 526, 2019.

[9] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[10] E. J. Msechu and G. B. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 400–414, Jan. 2011.

[11] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proc. Conf. Int. Speech Commun. Assoc.*, Singapore, 2014.

[12] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 559–568.

[13] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1709–1720.

[14] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 5325–5333.

[15] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang, "Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning," in *Proc. Int. Conf. Mach. Learn.*, Sydney, Australia, 2017, pp. 4035–4043.

[16] W. Wen *et al.*, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, 2017, pp. 1509–1519.

[17] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 440–445.

[18] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Intl. Conf. Learn. Representations*, Vancouver, Canada, 2018.

[19] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018, pp. 4447–4458.

[20] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018, pp. 5973–5983.

[21] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018, pp. 1299–1309.

[22] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018, pp. 9850–9861.

[23] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. Burlington, MA, USA: Morgan Kaufman, 2007.

[24] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2015, pp. 685–693.

[25] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Intl. Conf. Learn. Representations*, New Orleans, LA, 2019.

[26] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," in *Proc. Int. Conf. Mach. Learn. Workshop Coding Theory Large-Scale ML*, Long Beach, CA, 2019.

[27] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 5693–5700.

[28] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," in *Proc. Intl. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[29] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe, "Local sgd with periodic averaging: Tighter analysis and adaptive synchronization," in *Proc. Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 11080–11092.

[30] J. Wang, V. Tantia, N. Ballas, and M. Rabbat, "SlowMo: Improving communication-efficient distributed SGD with slow momentum," in *Proc. Intl. Conf. Learn. Representations*, 2020.

[31] M. Kamp *et al.*, "Efficient decentralized deep learning by dynamic model averaging," in *Proc. Eur. Conf. Mach. Learn. Knowl. Disc. Data.*,, Dublin, Ireland, 2018, pp. 393–409.

[32] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*.

[33] H. Sun, S. Lu, and M. Hong, "Improving the sample and communication complexity for decentralized non-convex optimization: A joint gradient estimation and tracking approach," 2019, *arXiv :1910.05857*.

[34] B. Li, S. Cen, Y. Chen, and Y. Chi, "Communication-efficient distributed optimization in networks with gradient tracking and variance reduction," in *Proc. Int. Conf. Artif. Intell. Stat.*, Palermo, Italy, 2020, pp. 1662–1672.

[35] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored ADMM for decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2565–2579, Mar. 2019.

[36] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, "Walkman: A communication-efficient random-walk algorithm for decentralized optimization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2513–2528, Mar. 2020.

[37] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Communication-efficient distributed strongly convex stochastic optimization: non-asymptotic rates," Sep. 2018, *arXiv :1809.02920*.

[38] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1000–1008.

[39] Y. Zhang and X. Lin, "DiSCO: Distributed optimization for self-concordant empirical loss," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 362–370.

[40] M. Jaggi *et al.*, "Communication-efficient distributed dual coordinate ascent," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, Canada, 2014, pp. 3068–3076.

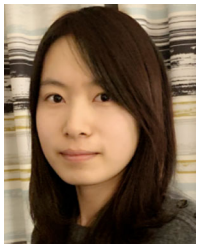
[41] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018, pp. 5050–5060.

[42] J. Sun, T. Chen, G. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," in *Proc. Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 3370–3380.

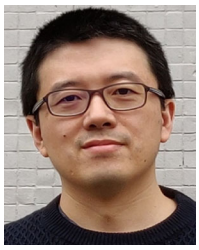
[43] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.
[44] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Conf. Neural Inf. Process. Syst.*, 2013, pp. 315–323.
[45] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Proc. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 3054–3062.
[46] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in *Proc. Eur. Conf. Mach. Learn.*, Riva del Garda, Italy, 2016, pp. 795–811.
[47] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," 2011, *arXiv:1109.5647*.
[48] A. S. Nemirovski and D. B. Yudin, *Problem Complexity and Efficiency in Optimization*. Wiley, 1983.
[49] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

Tianyi Chen (Member, IEEE) received the B.Eng. degree in communication science and engineering from Fudan University, Shanghai, China, in 2014, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Minnesota (UMN), Minneapolis, MN, USA, in 2016 and 2019, respectively.

Since August 2019, he has been with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA, as an Assistant Professor. His research focuses on the theory and application of optimization and statistical signal processing to problems emerging in machine learning and wireless networks. He was the recipient of the Doctoral Dissertation Fellowship at UMN, the IEEE Signal Processing Society Best Ph.D. Dissertation Award in 2020, and the NSF CAREER Award in 2021. He was also the recipient of the Best Student Paper awards, including those from Asilomar and ICASSP (as the coauthor).



Yuejiao Sun received the bachelor's degree in applied mathematics from Peking University, Beijing, China, in 2016 and the Ph.D. degree in applied mathematics from the University of California, Los Angeles, CA, USA, in 2021. Her research focuses on developing efficient stochastic optimization methods for machine learning applications. She developed efficient algorithms for large-scale distributed optimization problems and hierarchically coupled problems. She was the recipient of the Dissertation Year Fellowship and the Balbes Award at UCLA in 2020, and the Outstanding Student Paper Award for ICASSP 2021.



Wotao Yin (Member, IEEE) received the Ph.D. degree in operations research from Columbia University, New York, NY, USA, in 2006, respectively. During 2006–2013, he was with Rice University, Houston, TX, USA. Between 2013 and 2021, he was a Professor with the Department of Mathematics, University of California, Los Angeles, CA, USA. In 2019, he joined Alibaba US Damo Academy as a Researcher. His research interests include computational optimization and its applications in signal processing, machine learning, and other data science problems. He invented fast algorithms for sparse optimization and large-scale distributed optimization problems. He was the recipient of the NSF CAREER Award in 2008, the Alfred P. Sloan Research Fellowship in 2009, and the Morningside Gold Medal in 2016, and his coauthored six papers received the Best or Outstanding Paper-Type awards. Since 2018, he has been among the top 1% cited researchers by Clarivate Analytics.