Compressed Distributed Gradient Descent: Communication-Efficient Consensus over Networks

Xin Zhang[†]   Jia Liu[‡]   Zhengyuan Zhu[†]   Elizabeth S. Bentley[*]

[†]Department of Statistics, Iowa State University
[‡]Department of Computer Science, Iowa State University
[*]Air Force Research Laboratory, Information Directorate

Abstract—Network consensus optimization has received increasing attention in recent years and has found important applications in many scientific and engineering fields. To solve network consensus optimization problems, one of the most well-known approaches is the distributed gradient descent method (DGD). However, in networks with slow communication rates, DGD's performance is unsatisfactory for solving high-dimensional network consensus problems due to the communication bottleneck. This motivates us to design a communication-efficient DGD-type algorithm based on compressed information exchanges. Our contributions in this paper are three-fold: i) We develop a communication-efficient algorithm called amplified-differential compression DGD (ADC-DGD) and show that it converges under *any* unbiased compression operator; ii) We rigorously prove the convergence performances of ADC-DGD and show that they match with those of DGD without compression; iii) We reveal an interesting phase transition phenomenon in the convergence speed of ADC-DGD. Collectively, our findings advance the state-of-the-art of network consensus optimization theory.

I. INTRODUCTION

In recent years, network consensus optimization has received increasing attention thanks to its generality and wide applicability. To date, network consensus optimization has found important applications in many scientific and engineering fields, e.g., distributed sensing in wireless sensor networks [1]–[4], decentralized machine learning [5], [6], multi-agent robotic systems [7]–[9], smart grids [10], [11], to name just a few. Simply speaking, in a network consensus optimization problem, the global objective function is only partially known at each node. Through communications with local neighbors, all nodes in the network collaborate with each other to reach a consensus on an optimal solution, which minimizes the global objective function.

Among various algorithms for solving network consensus optimization problems, one of the most effective methods is the distributed gradient descent (DGD) algorithm, which was developed by Nedic and Ozdaglar approximately a decade ago [12]. The enduring popularity of DGD is primarily due to its implementation simplicity and elegant networking interpretation: In each iteration of DGD, each node performs an update by using a linear combination of a gradient step with respect to its local objective function and a weighted average from its neighbors. It has been shown that DGD enjoys the same $O(1/k)$ convergence speed as the classical gradient descent method, where k denotes the number of iterations [12]. The simplicity and salient features of DGD have further inspired a large number of extensions to various network settings (see Section II for more in-depth discussions).

However, despite its theoretical and engineering appeals, the performance of DGD is not satisfactory for solving a high-dimensional consensus problem over a network with low network communication speed: Due to the large amount of data sharing and the communication bottleneck, exchanging full high-dimensional information between neighboring nodes is time-consuming (or even infeasible), which significantly hinders the overall convergence of DGD. To improve the convergence speed, several second-order approaches using Hessian approximation (with respect to local objective function) have been proposed (see, e.g., [13], [14]). Although these second-order methods converge in a fewer number of iterations (hence less information exchanges), they require matrix inversion in each iteration, implying a $\Omega(d^2 \log d)$ per-iteration complexity for a d-dimensional problem. Hence, for high-dimensional problems (i.e., large d), low-complexity first-order methods remain more preferable in practice.

To address DGD's limitations in high-dimensional network consensus over low-speed networks, a natural idea is to *compress* the information exchanged between nodes. Specifically, by compressing the information in a high-dimensional state space to a smaller set of quantized states, each node can use a codebook to represent the quantized states with a small number of bits, which significantly reduces the communication burden. Moreover, from a cybersecurity standpoint, transmitting compressed information is also helpful because each node can encrypt its codebook and avoid revealing full information to potential eavesdroppers in the network.

However, with compressed information being adopted in DGD, several fundamental questions immediately arise: *i) Will DGD with compressed information exchanges still converge? ii) If the answer to i) is no, could we modify DGD to make it work with compressed information? iii) If the answer to ii) is yes, how fast does this modified DGD method converge?* The main contribution in this paper is that we provide concrete answers to all three fundamental questions. Our key results and their significance are summarized as follows:

- First, we show that DGD with straightforward compressed information exchange fails to converge because of a non-vanishing accumulated noise term resulted from compression over iterations. To address this problem, we propose a new algorithm called "amplified-differential compression DGD" (ADC-DGD), where, instead of directly exchanging compressed estimates of the optimization variable in DGD, we exchange an *amplified* version of the *state differential* between consecutive iterations, hence the name. We show that ADC-DGD effectively diminishes the accumulated noise from compression and induces convergence.

- We show that, under *any* unbiased compression operator, our ADC-DGD method converges at rate $O(1/k)$ to an $O(\alpha^2)$-neighborhood of an optimal solution with a constant step-size α. Under diminishing step-sizes, ADC-DGD converges asymptotically at rate $O(1/\sqrt{k})$ to an optimal solution. We note that these convergence rates are the *best possible* in the sense that they match with those of the original DGD without compression. This result is surprising since the information loss due to compression could be large.

- Based on the above convergence results of ADC-DGD, we further investigate the impacts of ADC-DGD's amplifying factor on convergence speed and communication load. Interestingly, we reveal a phase transition phenomenon of the convergence speed with respect to the amplification exponent γ in ADC-DGD. Specifically, when $\gamma \in (\frac{1}{2}, 1]$ (sublinear growth of amplification), convergence speed approaches that of DGD as γ increases. However, as soon as $\gamma > 1$, there is no further convergence speed improvement but network communication load continues to grow. This shows that $\gamma = 1$ is a critical point, under which we can trade communication overhead for convergence speed.

Collectively, our results contribute to a growing theoretical foundation of network consensus optimization. The rest of the paper is organized as follows. In Section II, we review related work. In Section III, we introduce the network consensus optimization problem and show that DGD with compressed information exchange fails to converge. In Section IV, we present our ADC-DGD algorithm and its convergence performance analysis. Numerical results are provided in Section V and Section VI concludes this paper.

II. RELATED WORK

In this section, we first provide a quick overview on the historical development of DGD-type algorithms. We then focus on the recent advances of communication-conscious network consensus optimization, including related work that utilize compression.

1) DGD-Based Algorithms for Network Consensus: Network consensus optimization can trace its roots to the seminal work by Tsitsiklis [15], where the system model and the analysis framework were first developed. As mentioned earlier, a well-known method for solving network consensus optimization is the distributed (sub)gradient descent (DGD) method, which was proposed by Nedic and Ozdaglar in [12].

DGD was recently reexamined in [16] by Yuan *et al.* using a new Lyapunov technique, which offers further mathematical understanding of its convergence performance. In their follow-up work [17], the convergence behavior of DGD was further analyzed for non-convex problems. Recently, several DGD variants have been proposed to enhance the convergence performance (e.g., achieving the same $O(1/k)$ convergence rate with constant step-size [18] or even under time-varying network graphs [19]).

2) Communication-Conscious Distributed Optimization: As mentioned earlier, studies have shown that communication costs of DGD could be a major concern in practice. To this end, Chow *et al.* [20] studied the tradeoff between communication requirements and prescribed accuracy. In [21], Berahas *et al.* developed an adaptive DGD framework called DGD^t to balance the costs between communication and computation. Here, the parameter t represents the number of consensus steps performed per gradient descent step ($t = 1$ corresponding to the original DGD). The larger the t-value, the cheaper the communication cost, and vice versa. The most related work to ours is by Tang *et al.* [22], which, to our knowledge, is also the only work in the literature that considers adopting compression in DGD. However, our algorithm differs from [22] in the following key aspects: i) The compression in [22] uses a quantized extrapolation between two successive iterates, which can be viewed as a diminishing step-size strategy. In contrast, our ADC-DGD algorithm uses an amplified differential of two successive iterates. As will be shown later, our algorithm can be interpreted as a *variance reduction* method; ii) Our convergence rate outperforms that of [22]. The fastest convergence rate of the algorithms in [22] is $O(\log(k)/\sqrt{k})$, while the convergence rate of our ADC-DGD algorithm is $o(1/\sqrt{k})$; iii) To reach the best convergence rate in [22], the extrapolation compression algorithm needs to solve a complex equation to obtain an optimal step-size. In contrast, our ADC-DGD algorithm uses the standard sublinearly diminishing step-sizes, which is of much *lower* complexity and can be easily implemented in practice.

III. NETWORK CONSENSUS OPTIMIZATION AND DISTRIBUTED GRADIENT DESCENT

In Section III-A, we first introduce the network consensus optimization problem, which is followed by the basic version of the DGD method. Then in Section III-B, we will illustrate an example where DGD with directly compressed information fails to converge, which motivates our subsequent ADC-DGD approach in Section IV.

A. Consensus Optimization over Networks: A Primer

Consider an undirected connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links, respectively, with $|\mathcal{N}| = N$ and $|\mathcal{L}| = E$. Let $x \in \mathbb{R}^P$ be some global decision variable to be optimized. Each node i has a local objective function $f_i(x)$, $i = 1, \cdots, N$ (only available to node i). The global objective function is the sum of all local objectives,

i.e., $f(x) \triangleq \sum_{i=1}^{N} f_i(x)$. Our goal is to solve the following network-wide optimization problem in a *distributed* fashion:

$$\min_{x \in \mathbb{R}^P} f(x) = \min_{x \in \mathbb{R}^P} \sum_{i=1}^{N} f_i(x). \qquad (1)$$

Problem (1) has a wide range of applications in practice. For example, consider a wireless sensor network, where each sensor node k distributively collects some local monitored temporal data $\mathbf{x}^k = (x_1^k, \cdots, x_T^k)$ and collaborates to detect the change-point in the global temporal data. This problem can be formulated as: $\min \sum_{i=1}^{N} f_i(\mathbf{x}^k)$, where $f_i(\mathbf{x}^k) \triangleq -|\sum_{i=1}^{t} x_i^k - \frac{t}{T} \sum_{i=1}^{T} x_i^k|^2$ is the CUSUM (cumulative sum control chart) statistics. Note that Problem (1) can be equivalently written in the following *consensus* form:

$$\text{Minimize} \quad \sum_{i=1}^{N} f_i(x_i) \qquad (2)$$
$$\text{subject to} \quad x_i = x_j, \qquad \forall (i,j) \in \mathcal{L}.$$

where $x_i \in \mathbb{R}^P$ is the local copy of x at node i. In Problem (2), the constraints enforce that the local copy at each node is equal to those of its neighbors, hence the name consensus. It is well-known [12] that Problem (2) can be reformulated as:

$$\text{Minimize} \quad \sum_{i=1}^{N} f_i(x_i) \qquad (3)$$
$$\text{subject to} \quad (\mathbf{W} \otimes \mathbf{I}_P)\mathbf{x} = \mathbf{x},$$

where $\mathbf{x} \triangleq [x_1^\top, \ldots, x_n^\top]^\top \in \mathbb{R}^{NP}$, $\mathbf{I}_P$ denotes the P-dimensional identity matrix, and the operator $\otimes$ denotes the Kronecker product. In (3), $\mathbf{W} \in \mathbb{R}^{N \times N}$ is referred to as the consensus matrix and satisfies the following properties:

1) $\mathbf{W}$ is doubly stochastic: $\sum_{i=1}^{N}[\mathbf{W}]_{ij} = \sum_{j=1}^{N}[\mathbf{W}]_{ij} = 1$.
2) The sparsity pattern of $\mathbf{W}$ follows the network topology: $[\mathbf{W}]_{ij} > 0$ for $\forall (i,j) \in \mathcal{L}$ and $[\mathbf{W}]_{ij} = 0$ otherwise.
3) $\mathbf{W}$ is symmetric and hence it has real eigenvalues.

The doubly stochastic property in 1) ensures that all eigenvalues of $\mathbf{W}$ are in $(-1, 1]$ and exactly one eigenvalue is equal to 1. Hence, it follows from Property 3) that one can sort eigenvalues as $1 = \lambda_1(\mathbf{W}) \geq \cdots \geq \lambda_N(\mathbf{W}) > -1$. Let $\beta \triangleq \max\{|\lambda_2(\mathbf{W})|, |\lambda_N(\mathbf{W})|\}$. Clearly, we have $\beta < 1$. It is shown in [12] that $(\mathbf{W} \otimes \mathbf{I}_P)\mathbf{x} = \mathbf{x}$ if and only if $x_i = x_j$, $(i,j) \in \mathcal{L}$. Therefore, Problems (2) and (3) are equivalent.

The equivalent network consensus formulation in Problem (3) motivates the design of the decentralized gradient descent (DGD) method as stated in Algorithm 1:

Algorithm 1: Decentralized Gradient Descent (DGD) [12].

Initialization:
1. Let $k = 1$. Choose initial values for $x_{i,1}$ and step-size α_1.
Main Loop:
2. In the k-th iteration, each node sends its local copy to its neighbors. Also, upon reception of all local copies from its neighbors, each node updates its local copy as follows:

$$x_{i,k+1} = \underbrace{\sum_{j \in \mathcal{N}_i}[\mathbf{W}]_{ij} x_{j,k}}_{\text{Consensus step}} - \underbrace{\alpha_k \nabla f_i(x_{i,k})}_{\text{Gradient step}}, \qquad (4)$$

---

where $[\mathbf{W}]_{ij}$ is the entry in the i-th row and j-th column in $\mathbf{W}$, $x_{i,k}$ and α_k represent x_i's value and step-size in the k-th iteration, respectively, and $\mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i,j) \in \mathcal{L}\}$.
3. Stop if a desired convergence criterion is met; otherwise, let $k \leftarrow k + 1$ and go to Step 2.

We can see that the DGD update in (4) consists of a consensus step and a local gradient step, which can be easily implemented in a network. Also, DGD achieves the same $O(1/k)$ convergence rate as in the classical gradient descent method. However, as mentioned in Section I, DGD may not work well for high-dimensional consensus problem in low-speed networks. Hence, we are interested in developing a DGD-type algorithm with compressed information exchanges in this paper. In what follows, we will first show that DGD fails to converge if compressed information is directly adopted in the consensus step.

B. DGD with Directly Compressed Information Exchange Does Not Converge: A Motivating Example

We first introduce the notion of unbiased stochastic compression operator, which has been widely used to represent compressions in the literature (see, e.g., [20], [21], [23]–[26]).

Definition 1 (Unbiased Stochastic Compression Operator). A stochastic compression operator $C(\cdot)$ is unbiased if it satisfies $C(z) = z + \epsilon_z$, with $\mathbb{E}[\epsilon_z] = 0$ and $\mathbb{E}[\epsilon_z^2] \leq \sigma^2$, $\forall z$.

Defintion 1 guarantees that the noise caused by the compression has no effect on the mean of the parameter and its variance is bounded. Many compressed operators satisfy the above definition. The followings are two examples:

Example 1 (The Ternary Compressed Operater [25]). *For* $\mathbf{z} = (z_1, \cdots, z_p)^\top \in \mathbb{R}^P$, $C(\mathbf{z}) = \|\mathbf{z}\|_\infty sign(\mathbf{z}) \circ b_{\mathbf{z}}$, *where* $\circ$ *is the Hadamard product and* $b_{\mathbf{z}}$ *is a random vector with the k-th element* $[b_{\mathbf{z}}]_k$ *following the Bernoulli distribution:*

$$\begin{cases} \Pr([b_{\mathbf{z}}]_k = 1) = |z_k|/\|\mathbf{z}\|_\infty, \\ \Pr([b_{\mathbf{z}}]_k = 0) = 1 - |z_k|/\|\mathbf{z}\|_\infty. \end{cases}$$
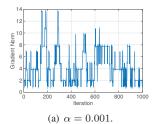
Example 2 (The Quantized Compressed Operater [24]). *For* $\mathbf{z} = (z_1, \cdots, z_p)^\top \in \mathbb{R}^P$, *the k-th element of* $[C(\mathbf{z})]_k$ *is:*

$$[C(\mathbf{z})]_k = \begin{cases} \lfloor z_k \rfloor + 1, \text{with probability } (1 - p_k), \\ \lfloor z_k \rfloor, \text{with probability } p_k. \end{cases}$$

where $\lfloor z \rfloor$ *represents the largest integer smaller than z and the probability* $p_k = z_k - \lfloor z_k \rfloor$.

Now, we consider the convergence of DGD with unbiased stochastic compressions. If local copies are compressed and then directly used in the consensus step in the DGD algorithm, then Eq. (4) in Algorithm 1 can be modified as:

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i}[\mathbf{W}]_{ij} C(x_{j,k}) - \alpha_k \nabla f_i(x_{i,k}) =$$
$$\underbrace{\sum_{j=\mathcal{N}_i}[\mathbf{W}]_{ij} x_{j,k} - \alpha_k \nabla f_i(x_{i,k})}_{\text{Exact DGD}} + \underbrace{\sum_{j \in \mathcal{N}_i}[\mathbf{W}]_{ij} \epsilon_{x_{j,k}}}_{\text{Accumulated noise term}}, (5)$$
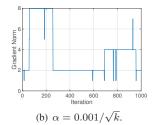
2433

Fig. 1. The simulation results for DGD with quantized compression operator for a 2-node network, for which DGD fails to converge after 1000 iterations.

which shows that there is a *non-vanishing* noise term accumulated over iterations, which prevents the DGD algorithm from converging. For example, consider a simple 2-node network with local objectives $f_1(x) = 4(x-2)^2$ and $f_2(x) = 2(x+3)^2$. The quantized compressed operator [25] is adopted in DGD. The simulation results are illustrated in Fig. 1, where we can see that DGD fails to converge after 1000 iterations even for such a small-size network consensus problem. This motivates us to pursue a new algorithmic design in Section IV.

IV. AMPLIFIED-DIFFERENTIAL DISTRIBUTED GRADIENT DESCENT METHOD (ADC-DGD)

In this Section, we will first introduce our ADC-DGD algorithm in Section IV-A. Then, we will present the main theoretical results and their intuitions in Section IV-B. The proofs for the main results are provided in Section IV-C.

A. The ADC-DGD Algorithm

Our ADC-DGD algorithm is stated in Algorithm 2:

Algorithm 2: Amplified-Differential Compression DGD.

Initialization:
1. Let $k=1$. Let $x_{i,0} = \tilde{x}_{i,0} = 0$, $\forall i$. Choose initial values for step-size α_1 and the amplification exponent γ. Let $x_{i,1} = y_{i,1} = -\alpha_1 \nabla f_i(x_{i,0})$, $\forall i$.

Main Loop:
2. In the k-th iteration, each node sends the compressed amplified-differential $d_{i,k} = C(k^\gamma y_{i,k})$ to its neighbors. Also, upon collecting all neighbors' information, each node estimates neighbors' (imprecise) values: $\tilde{x}_{j,k} = \tilde{x}_{j,k-1} + d_{j,k}/k^\gamma$. Then, each node updates its local value:

$$x_{i,k+1} = \underbrace{\sum_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij} \tilde{x}_{j,k}}_{\text{Compressed consensus}} - \underbrace{\alpha_k \nabla f_i(x_{i,k})}_{\text{gradient step}}. \quad (6)$$

Each node updates local differential: $y_{i,k+1} = x_{i,k+1} - \tilde{x}_{i,k}$.
3. Stop if a desired convergence criterion is met; otherwise, let $k \leftarrow k+1$ and go to Step 2.

Several important remarks on Algorithm 2 are in order: i) Compared to the original DGD, each node i under ADC-DGD requires additional memory to store the (imprecise) values of its neighbors in the previous iteration: $\{\tilde{x}_{j,k-1} : (i,j) \in \mathcal{L}\}$. This additional memory allows the neighbors to only transmit the difference between successive iterations

$y_{i,k} = x_{i,k} - \tilde{x}_{i,k-1}$ rather than $x_{i,k}$ directly. Note that this memory requirement is modest in practice since many computer networks are scale-free (i.e., node degree distribution follows a power law and hence most nodes have low degrees); ii) Each node sends out a compressed version of the amplified-differential $C(k^\gamma y_{i,k})$. This information will then be de-amplified at the receiving nodes as $d_{i,k}/k^\gamma$, which is a noisy version of $y_{i,k}$. Based on the memory of the previous version, each node obtains their neighbors' values estimation $\tilde{x}_{j,k}$, $j \in \mathcal{N}_i$. Clearly, ADC-DGD is more communication-efficient compared to the original DGD; iii) Once $\tilde{x}_{j,k}$, $j \in \mathcal{N}_i$, are available, the update in (6) follows the same structure as in DGD, which also contains a consensus step and a local gradient step. Therefore, the complexity of ADC-DGD are almost identical to the original DGD, which means that ADC-DGD enjoys the same low-complexity.

B. Main Convergence Results

Before presenting the convergence results of ADC-DGD, we first state several needed assumptions:

Assumption 1. *The local objective functions $f_i(\cdot)$ satisfy:*
- *(Lower boundedness) There exists an optimal x^* with $\|x^*\| < \infty$ such that $\forall x \neq x^*$, $\sum_{i=1}^N f_i(x) \geq \sum_{i=1}^N f_i(x^*)$;*
- *(Lipschitz continuous gradient) there exists a constant $L > 0$ such that $\forall x, y$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, $\forall i$.*

Assumption 2 (Growth rate at infinity). *If the domain for $\mathbf{x}$ is unbounded, then there exists a constant $M > 0$ such that*

$$\lim_{\|\mathbf{x}\| \to \infty} \frac{\|\mathbf{x}\|}{f(\mathbf{x})} = \lim_{\sum_{i=1}^n \|x_i\| \to \infty} \frac{\sum_{i=1}^N \|x_i\|}{\sum_{i=1}^N f_i(x_i)} \leq M,$$

where $f(\mathbf{x}) = \sum_{i=1}^N f_i(x_i)$ and $\mathbf{x} = (x_1^\top, \cdots, x_N^\top)^\top$.

Assumption 1 is standard in convergence analysis of gradient descent type algorithms: The first bullet ensures the existence of optimal solution and the second bullet guarantees the smoothness of the local objectives. Assumption 2 is a technical result coming out of our proofs and guarantees that, at infinity, the growth rate of the objective function is at least faster than linear. We note that Assumption 2 is a mild assumption, which is evidenced by the following lemma (proof details are relegated to Appendix A).

Lemma 1. *Any strictly convex function $f(\cdot)$ satisfying Assumption 1 also satisfies Assumption 2.*

In addition to convex objectives, many non-convex functions also satisfy Assumption 2, , as shown below and in Fig. 2:

Example 3. *(Non-convex functions satisfying Assumption 2):*
- *$f(x) = 5x^3 + x^4$, with $\lim_{x\to\infty} \|x\|/f(x) = 0$ but $\nabla^2 f(x) = 12x^2 + 30x$ is smaller than 0 when $x = -1$;*
- *$f(x) = 10\sin(x) + x^2$, with $\lim_{x\to\infty} \|x\|/f(x) = 0$ but $\nabla^2 f(x) = -10\cos(x) + 2$ is smaller than 0 when $x = 0$.*

Our first key result is on the convergence of local variables to the mean vector across nodes:
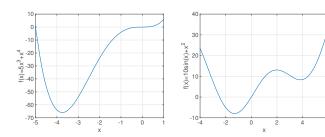
Fig. 2. Examples of non-convex functions satisfying Assumption 2.

Theorem 1. *Let the mean vector at the k-th iteration be defined as $\bar{\mathbf{x}}^k = \mathbf{1} \otimes \bar{x}^k \in \mathbb{R}^{NP}$, with $\bar{x}^k = \frac{1}{N}\sum_{i=1}^{N} x_{i,k}$. Under Assumptions 1, if $\mathbb{E}[\|\nabla \mathbf{f}(\mathbf{x}^i)\|]$ is bounded by D and the amplifying exponent is γ, then:*

- *For constant step-size $\alpha_i = \alpha$, $\forall i$, $\mathbb{E}[\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|] \leq \frac{\alpha D}{1-\beta} + O(\frac{\sqrt{NP}\sigma}{k^\gamma})$;*
- *For diminishing step-size $\alpha_i = O(\frac{1}{k^\eta})$ with some $\eta > 0$, $\mathbb{E}[\|\mathbf{x}^k - \bar{\mathbf{x}}^k\|] = O(\frac{1}{k^{\min(\eta,\gamma)}})$.*

Remark 1. Theorem 1 says that the local copies will converge to the mean vector asymptotically with a diminishing step-size, or stay within a bounded error ball of the mean vector if a constant step-size is adopted.

Our second key convergence result is on the convergence rate of ADC-DGD under constant step-sizes:

Theorem 2 (Constant Step-Size). *Let the step-size be constant, i.e., $\alpha_k = \alpha$, $\forall k$, with $\alpha < \frac{1+\lambda_N(\mathbf{W})}{L}$. Under Assumptions 1-2, if the amplified exponent $\gamma > \frac{1}{2}$, then it holds that:*

$$\min_k \left\{ \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\right\|^2\right]\right\} \leq$$
$$\frac{C_1 \alpha^2}{N(1-\beta)^2} + \frac{C_2}{\alpha k} + O\left(\frac{2L\sqrt{P}\sigma}{\sqrt{N}k^\gamma}\right), \quad (7)$$

where $C_1 \triangleq L^4(B + \|\mathbf{x}^*\|)^2$ and $C_2 \triangleq 2[\frac{1}{N}\sum_{i=1}^{N} f_i(0) - \frac{1}{N}\sum_{i=1}^{N} f_i(x^*)]$ are two constants.

Remark 2. Under the same conditions of Theorem 2, we immediately have that Algorithm 2 has an ergodic convergence rate $O\left(\frac{1}{k^{\min(1,\gamma)}}\right)$ until $\min_k\{\mathbb{E}[\|\frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2\}$ reaching the error ball $O((\frac{\alpha}{1-\beta})^2)$ and the fastest rate is $O(\frac{1}{k})$.

Our third key convergence result is concerned with the convergence rate of ADC-DGD under diminishing step-sizes:

Theorem 3 (Diminishing Step-Sizes). *Under Assumptions 1-2, if the local objectives have bounded graidents, i.e. there exists a positive constant D such that $\|\nabla f_i(x)\| \leq D$, $\forall x$, and $\gamma > \frac{1}{2}$, $\eta \geq \frac{1}{2}$, then with diminishing step-size $\alpha_k = O(\frac{1}{k^\eta})$, it holds that $\|\frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2 = o(\frac{1}{k^{1-\eta}})$ almost surely.*

Remark 3. In Theorem 3, the exponent for the diminishing rate of step-size is lower bounded ($\eta \geq \frac{1}{2}$). Thus, the best convergence rate for this algorithm is $o(1/\sqrt{k})$, which is faster than the rate $O(\log(k)/\sqrt{k})$ in [22]. We also note that our

convergence result is in "Small-O", which is stronger than conventional "Big-O" convergence results.

Remark 4 (Intuition and Design Rationale of ADC-DGD). To understand why ADC-DGD converges, a closer look at (6) in Algorithm 2 reveals that:

$$\tilde{x}_{j,k} = \tilde{x}_{j,k-1} + C(k^\gamma y_{j,k})/k^\gamma = \tilde{x}_{j,k-1} + (k^\gamma y_{j,k} + \epsilon_{k^\gamma y_{j,k}})/k^\gamma$$
$$= \tilde{x}_{j,k-1} + y_{j,k} + (\epsilon_{k^\gamma y_{j,k}})/k^\gamma = x_{j,k} + (\epsilon_{k^\gamma y_{j,k}})/k^\gamma. \quad (8)$$

Thanks to the properties of the unbiased stochastic operator (cf. Definition 1), the noise term in the last step of (8) has zero mean and a *vanishing* variance $\frac{\sigma^2}{k^{2\gamma}}$ as k gets large. This is in contrast to the accumulated non-vanishing noise term in DGD (cf. Eq. (5)). Eq. (8) also shows that our ADC-DGD algorithm can be interpreted as a *variance reduction* method. Indeed, our proofs in Section IV-C are based on these intuitions.

C. Proofs of the Main Theorems

Due to space limitation, in this subsection, we outline the key steps of the proofs of Theorems 1–3. We relegate proof details to appendices. Some appendices provide proof sketches due to the lengths of the proofs.

Step 1): Introducing a Lyapunov Function. Consider the following Lyapunov function, which is also used in [16], [21]:

$$L_{\alpha_k}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top(\mathbf{I} - \mathbf{Z})\mathbf{x} + \alpha_k \mathbf{1}^\top \mathbf{f}(\mathbf{x}), \quad (9)$$

where $\mathbf{x} = [x_1^\top, \cdots, x_N^\top]^\top$, $\mathbf{Z} \triangleq \mathbf{W} \otimes \mathbf{I}_P$, and $\mathbf{f}(\mathbf{x}) \triangleq [f_1(x_1), \cdots, f_N(x_N)]^\top$ so that $\mathbf{1}^\top \mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$. The following lemma is from [16], which says that the Lyapunov function $L_a(\mathbf{x})$ has Lipschitz-continuous gradient.

Lemma 2. *Under Assumption 1, the Lyapunov function $L_\alpha(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T(\mathbf{I} - \mathbf{Z})\mathbf{x} + \alpha\mathbf{1}^T\mathbf{f}(\mathbf{x})$ has $(1 - \lambda_N(\mathbf{W}) + \alpha L)$-Lipschitz gradient, i.e. $\|\nabla L_\alpha(\mathbf{x}) - \nabla L_\alpha(\mathbf{y})\| \leq (1 - \lambda_N(\mathbf{W}) + \alpha L)\|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^{np}$.*

Note that, using the notation $\mathbf{Z}$, we can compactly rewrite the updating step (6) in Algorithm 2 as follows:

$$\mathbf{x}^{k+1} = \mathbf{Z}(\tilde{\mathbf{x}}^k + \mathbf{d}^k/k^\gamma) - \alpha_k \nabla \mathbf{f}(\mathbf{x}^k)$$
$$= \mathbf{Z}\mathbf{x}^k + \mathbf{Z}\epsilon^k/k^\gamma - \alpha_k \nabla \mathbf{f}(\mathbf{x}^k)$$
$$= \mathbf{x}^k - [(\mathbf{I} - \mathbf{Z})\mathbf{x}^k + \alpha_k \nabla \mathbf{f}(\mathbf{x}^k)] + \mathbf{Z}\epsilon^k/k^\gamma$$
$$= \mathbf{x}^k - \nabla L_{\alpha_k}(\mathbf{x}^k) + \mathbf{Z}\epsilon^k/k^\gamma, \quad (10)$$

where $\nabla \mathbf{f}(\mathbf{x}^k) = [\nabla f_1(x_{1,k})^\top, \cdots, \nabla f_N(x_{N,k})^\top]^\top$, $\mathbf{x}^k = (x_{1,k}^\top, \cdots, x_{N,k}^\top)^\top$ is the parameter in the k-th iteration, $\tilde{\mathbf{x}}^k = [\tilde{x}_{1,k}^\top, \cdots, \tilde{x}_{N,k}^\top]^\top$ is the vector of imprecise parameters, and $\mathbf{d}^k \triangleq [d_{1,k}^\top, \cdots, d_{N,k}^\top]^\top$ and $\epsilon^k \triangleq [\epsilon_{k^\gamma y_{1,k}}^\top, \cdots, \epsilon_{k^\gamma y_{N,k}}^\top]^\top$. It can be seen that Eq. (10) is one-step stochastic gradient descent for $L_{\alpha_k}(\mathbf{x})$ and the noise term $\mathbf{Z}\epsilon^k/k^\gamma$ has zero mean and variance with diminishing bound $NP\sigma^2/k^{2\gamma}$, i.e.,

$$\mathbb{E}[\mathbf{Z}\epsilon^k/k^\gamma] = \mathbf{Z}\mathbb{E}[\epsilon^k]/k^\gamma = 0, \quad (11)$$

$$\mathbb{E}[\|\mathbf{Z}\epsilon^k/k^\gamma\|^2] \leq \|\mathbf{Z}\|^2 \mathbb{E}[\|\epsilon^k\|^2]/k^{2\gamma} \overset{(a)}{\leq} NP\sigma^2/k^{2\gamma}, \quad (12)$$

2435

where (a) follows from the fact that the eigenvalues of $\mathbf{Z}$ are in $(-1, 1]$ and $\epsilon^k \in \mathbb{R}^{NP}$.

Step 2) Convergence of the Objective Value. Note from (8) that the noise caused by compression is similar to the noise in the standard stochastic gradient descent method (SGD). Hence, we can apply similar analysis techniques from SGD on the iterations of ADC-DGD to obtain the following results:

Theorem 4 (Bounded Gradient). *Under Assumptions 1-2, if the step-size $\alpha < \frac{1+\lambda_N(\mathbf{W})}{L}$ and the amplified exponent $\gamma > \frac{1}{2}$ in Algorithm 2, then there exists a constant $B > 0$ such that $\mathbb{E}[\|\mathbf{x}^k\|] \leq B$ and $\mathbb{E}[\|\nabla \mathbf{f}(\mathbf{x}^k)\|] \leq L(B + \|\mathbf{x}^*\|)$, where $\mathbf{x}^* = \mathbf{1} \otimes x^* \in \mathbb{R}^{NP}$. Moreover, $\mathbb{E}[\|\nabla L_\alpha(\mathbf{x}^k)\|^2] = o(1/k)$.*

Theorem 4 shows that with an appropriate step-size and an amplifying exponent, Algorithm 2 converges. But due to the compression noise, the convergence rate is sublinear. To see this, note that $\nabla L_\alpha(\mathbf{x}) = (\mathbf{I} - \mathbf{Z})\mathbf{x} + \alpha \nabla \mathbf{f}(\mathbf{x})$, and $\mathbf{1}^\top (\mathbf{I} - \mathbf{Z}) = 0$. Thus, $\mathbf{1}^\top \nabla L_\alpha(\mathbf{x}) = \alpha \sum_{i=1}^N \nabla f_i(x_i)$, which implies $\|\alpha \sum_{i=1}^N \nabla f_i(x_i)\|^2 \leq \|\nabla L_\alpha(\mathbf{x})\|^2$. From Theorem 4, the convergence rate of $\mathbb{E}[\|\sum_{i=1}^N \nabla f_i(x_i)\|^2]$ is also $o(1/k)$.

Step 3) Proving Theorem 1. Note from Algorithm 2 and (10) that the following hold:

$$
\begin{cases}
\mathbf{x}^1 = \mathbf{Z}\mathbf{x}^0 - \alpha_0 \nabla \mathbf{f}(\mathbf{x}^0) = -\alpha_0 \nabla \mathbf{f}(\mathbf{x}^0), \\
\mathbf{x}^2 = \mathbf{Z}\mathbf{x}^1 - \alpha_1 \nabla \mathbf{f}(\mathbf{x}^1) + \mathbf{Z}\frac{\epsilon^1}{1^\gamma}, \\
\quad = -\alpha_0 \mathbf{Z}\nabla \mathbf{f}(\mathbf{x}^0) - \alpha_1 \nabla \mathbf{f}(\mathbf{x}^0) + \mathbf{Z}\epsilon^1, \\
\quad \vdots \\
\mathbf{x}^k = -\sum_{i=0}^{k-1} \alpha_i \mathbf{Z}^{k-i-1} \nabla \mathbf{f}(\mathbf{x}^i) + \sum_{i=1}^{k-1} \mathbf{Z}^{k-i} \frac{\epsilon^i}{i^\gamma}.
\end{cases}
\quad (13)
$$

Eq. (13) characterize the trajectory of the iterates. Each iterate consists of two parts, one from gradients and the other from noises. Note that in (13), the variance of accumulated noises are in the form of $h_k \triangleq \sum_{i=1}^k \frac{\beta^{k-i}}{i^\gamma}$. Next, we prove an interesting lemma for h_k, which is useful in proving Theorem 1.

Lemma 3. *Define $h_k \triangleq \sum_{i=1}^k \frac{\beta^{k-i}}{i^\gamma}$, where $\beta \in [0, 1)$ and $\gamma > 0$. It follows that $h_k = O(\frac{1}{k^\gamma})$.*

Lemma 3 implies that the negative effect of compression noises can be ignored asymptotically, which induces convergence. With (13), Theorem 4 and Lemma 3, we can finally prove Theorem 1 and the details are relegated to Appendix D.

Step 4) Proving Theorems 2 and 3. With some algebraic derivation, we can show the following fundamental result:

Lemma 4. *Let $\mathcal{F}_k = \sigma\langle \mathbf{x}^1, \cdots, \mathbf{x}^k \rangle$ be a filtration. Under Assumptions 1-2, the following inequality holds:*

$$
\mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^N f_i(\bar{x}^{k+1})\Big|\mathcal{F}_k\Big] + \frac{\alpha_k}{2}\Big\|\frac{1}{N}\sum_{i=1}^N \nabla f_i(\bar{x}^k)\Big\|^2
$$
$$
\leq \frac{1}{N}\sum_{i=1}^N f_i(\bar{x}^k) + \frac{\alpha_k L^2}{2N^2}\sum_{i=1}^N \Big\|\bar{x}^k - x_{i,k}\Big\|^2 + \frac{LP\sigma^2}{2Nk^{2\gamma}}, \quad (14)
$$

where α_k is the step-size at the k-the iteration.

Eq. (14) in Lemma 4 is similar to the contraction in stochastic gradient descent algorithm, which relates the objective

Fig. 3. A four-node network.

$$
\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 3/4 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 \\ 1/4 & 0 & 0 & 3/4 \end{bmatrix},
$$

Fig. 4. The consensus matrix for Fig. 3.

values and gradient norm. Then, by telescoping and the supermartingale convergence theorem, we can prove Theorems 2 and 3 (see Appendices F and G).

D. Understanding the Role of the Amplifying Exponent γ

In our algorithm, the amplifying exponent γ is a key component to adjust the communication rate. From Theorems 2 and 3, it can be seen that within $(1/2, 1]$, the larger γ means the faster convergence. However, since the transmitted value is $C(k^\gamma \mathbf{y}^k)$, we can see that a larger γ leads to a larger $k^\gamma \mathbf{y}^k$, which may lead to overflow error (for example, type 'int8' in Matlab could only present data within $(-128, 127)$). Hence, it is necessary to guarantee that $k^\gamma \mathbf{y}^k$ would not grow too fast. Recalling Eqs. (10) and (6) in Algorithm 2, we have

$$
\mathbf{y}^k = \mathbf{x}^k - \tilde{\mathbf{x}}^{k-1} = (\mathbf{I} + \mathbf{Z})\epsilon^k / k^\gamma - \nabla L_{\alpha_k}(\mathbf{x}^k).
$$

Under the expectation, the transmitted value is bounded by

$$
\mathbb{E}[\|k^\gamma \mathbf{y}^k\|] = \mathbb{E}[\|(\mathbf{I} + \mathbf{Z})\epsilon^k - k^\gamma \nabla L_{\alpha_k}(\mathbf{x}^k)\|]
$$
$$
\leq \mathbb{E}[\|(\mathbf{I} + \mathbf{Z})\epsilon^k\| + \|k^\gamma \nabla L_{\alpha_k}(\mathbf{x}^k)\|]
$$
$$
\leq 2\mathbb{E}[\|\epsilon^k\|] + k^\gamma \mathbb{E}[\|\nabla L_{\alpha_k}(\mathbf{x}^k)\|].
$$

From Definition 1, we have that each element of $\mathbb{E}[\|\epsilon^k\|]$ is bounded by σ. From Theorem 4, we have $\mathbb{E}[\|\nabla L_{\alpha_k}(\mathbf{x}^k)\|^2] = o(1/k)$. Thus, $\mathbb{E}[\|k^\gamma \mathbf{y}^k\|]$ is bounded by $o(k^{\gamma-\frac{1}{2}})$. We state this result in the following proposition:

Proposition 5. *Under Assumptions 1-2, with $\gamma > \frac{1}{2}$, the transmitted value $k^\gamma \mathbf{y}^k$ satisfies $\mathbb{E}[\|k^\gamma \mathbf{y}^k\|] = o(k^{\gamma-\frac{1}{2}})$.*
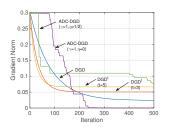
The insight from Proposition 5 is that with $\gamma \in (\frac{1}{2}, 1]$, the growth speed for the transmitted value is slower than $o(\sqrt{k})$, which is not very fast.

V. NUMERICAL RESULTS

In this section, we will present several numerical experiments to further validate the performance of ADC-DGD.

1) Effect of Compression: First, we compare ADC-DGD with some existing methods to show its convergence rate and communication-efficiency. Consider a four-node network as shown in Fig. 3 with the following global objective function: $\min_x f(x) = f_1(x) + f_2(x) + f_3(x) + f_4(x)$, where $f_1(x) = -4x^2$, $f_2(x) = 2(x-0.2)^2$, $f_3(x) = 2(x+0.3)^2$, and $f_4(x) = 5(x-0.1)^2$. It can be seen that $f_1(x)$ is non-convex, while the rest are convex. The communication consensus matrix used in this experiment is shown in Fig. 4.

In our simulation, we compare our ADC-DGD with the conventional DGD and DGD^t. For DGD^t, we consider two cases: $t = 3$ and $t = 5$. In ADC-DGD, the amplifying exponent γ is set to 1. We use two step-size strategies: 1)
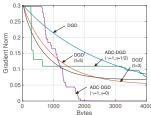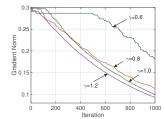
2436

Fig. 5. Convergence comparisons between ADC-DGD, DGD, and DGD^t.

Fig. 6. Amount of exchanged information (bytes) vs gradient norm.

Fig. 7. Convergence behaviors under different choices of γ.

Fig. 8. Growth of transmitted values vs. number of iterations.

constant step-size α (i.e. $\eta = 0$) and 2) diminishing step-size $\alpha/\sqrt{k}$ (i.e. $\eta = 1/2$). We adopt the quantized operator in [24] as the compression operator. After compression, the values are integer. Hence, they can be stored as type 'int16', which is 2 bytes. However, the uncompressed values are stored as type 'double', costing 8 bytes. The convergence results for one trial are illustrated in Fig. 5 and Fig. 6.

From the simulations, we can see that: 1) with a fixed step-size, all algorithms converge to an error ball, while the radiuses of the conventional DGD and ADC-DGD are relatively smaller. This is because with a larger t, β^t becomes smaller and hence the error ball $O(\alpha/(1 - \beta^t))$ for DGD^t becomes larger; 2) By using compression, the convergence process of ADC-DGD is relatively less smooth. But the compression noise does not affect convergence. With the same step-size, the conventional DGD and ADC-DGD have the almost the *same* convergence rate; 3) By using diminishing step-sizes, the convergence speed for ADC-DGD becomes slower. However, the objective value remains decreasing; 4) By comparing the amount of exchanged information, ADC-DGD with the fixed step-size converges the fastest, using only 2000 bytes. This shows that our algorithm is the *most* communication-efficient.

2) Effect of the Amplifying Exponent: Next, we show the effect of the amplifying exponent γ. As discussed in Section IV-D, with a small γ, the noise caused by compression could lead to a slow convergence. On the other hand, with a large γ, the transmitted value $k^\gamma y$ could be too large and cause overflow, especially for quantized compressed operator. Here, we change γ using $\{0.6, 0.8, 1.0, 1.2\}$ and keep the rest of the parameters the same. For each γ, we repeat the algorithm 100 times and compute the average objective values, as well as the maximum transmitted value from all the nodes in each iteration. The simulation results are shown in Figs. 7 and 8. We can see that, with a larger γ value, the algorithm converges faster and the curve is smoother, while the transmitted values are increasing a little bit faster. In this example, we can see that $\gamma = 0.8$ strikes a good balance between convergence and maximum transmitted value.

3) Effect of Network Size: The following simulations indicate that our algorithm could be scaled to large-size networks. In our simulation, we consider the 'circle' system: each node only connects with two neighboring nodes and forms a circle. For example, Fig. 9 shows a five-node circle. We set n to be 3, 5, 10, 20 in our experiment. The local objectives are in the
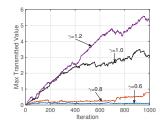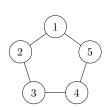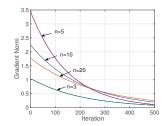

Fig. 9. The 5-node circle topology.

Fig. 10. The effect of network size.

form of $f_i(x) = a_i(x - b_i)^2$. In our simulation, $\{a_i, b_i\}_{i=1}^n$ are independently randomly generated: $a_i \sim \text{Uniform}[0, 10]$ and $b_i \sim \text{Uniform}[0, 1]$. For each value of n, we repeat 100 trials and compute the average gradient norm. The convergence results are shown in Fig. 10. It can be seen that our algorithm works well as the network size increases, demonstrating the scalability of ADC-DGD.

VI. CONCLUSION

In this paper, we considered designing communication-efficient network consensus optimization algorithms in networks with slow communication rates. We proposed a new algorithm called amplified-differential compression decentralized gradient descent (ADC-DGD), which is based on compression to reduce communication costs. We investigated the convergence behavior of ADC-DGD on smooth but possibly non-convex objectives in this work. We showed that: 1) by employing a fixed step-size α, ADC-DGD converges with the $O(1/k^{\min(1,\gamma)})$ ergodic rate until reaching an error ball of size $O(\alpha^2)$ with the amplified parameter γ; 2) ADC-DGD enjoys the best convergence rate $o(1/\sqrt{k})$ and converge to a stationary point almost surely with diminishing step-sizes. Consensus optimization with compressed information is an important and under-explored area. An interesting future topic is to generalize our ADC-DGD algorithmic framework to analyze cases with local stochastic gradients, which could further lower the implementation complexity of ADC-DGD.

REFERENCES

[1] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3816–3827, 2010.

[2] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *arXiv preprint cs/0503072*, 2005.

[3] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy linkspart i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[4] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2012.

[6] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 1543–1550.

[7] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[8] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.

[9] K. Zhou, S. I. Roumeliotis *et al.*, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[10] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, 2013.

[11] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617–1626, 2013.

[12] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[13] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *arXiv preprint arXiv:1605.07689*, 2016.

[14] J. Wang, M. Kolar, N. Srebro, and T. Zhang, "Efficient distributed learning with sparsity," *arXiv preprint arXiv:1605.07991*, 2016.

[15] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND DECISION SYSTEMS, Tech. Rep., 1984.

[16] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[17] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *arXiv preprint arXiv:1608.05766*, 2016.

[18] 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.

[19] 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.

[20] Y.-T. Chow, W. Shi, T. Wu, and W. Yin, "Expander graph and communication-efficient decentralized optimization," in *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE, 2016, pp. 1715–1720.

[21] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *arXiv preprint arXiv:1709.02999*, 2017.

[22] H. Tang, C. Zhang, S. Gan, T. Zhang, and J. Liu, "Decentralization meets quantization," *arXiv preprint arXiv:1803.06443*, 2018.

[23] Y. Zhang, M. J. Wainwright, and J. C. Duchi, "Communication-efficient algorithms for statistical optimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 1502–1510.

[24] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1707–1718.

[25] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 1508–1518.

[26] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *arXiv preprint arXiv:1710.09854*, 2017.

[27] R. Hannah and W. Yin, "On unbounded delays in asynchronous parallel fixed-point algorithms," *Journal of Scientific Computing*, pp. 1–28, 2016.

APPENDIX A
PROOF FOR LEMMA 1

Without loss of generality, we prove the case of one-dimensional objective. Firstly, we consider $f_i(\cdot)$ reach the minimal at $x_i^* = 0$ and $f_i(x_i^*) = 0$. With the convexity of $f_i(\cdot)$, $f_i(x) \leq (1 - \frac{1}{a})f_i(0) + \frac{1}{a}f_i(ax)$, $\forall x$, $a > 1$. Consider x_0 with $f_i(x_0) \neq 0$, and define $\max_i\{\frac{\|x_0\|}{f_i(x_0)}\} = M_0$. Consider $\forall y > x_0 > 0$, there exists a constant $a_y > 1$ such that $y = a_y x_0$ and thus $f_i(x_0) \leq \frac{1}{a_y}f_i(y)$. Hence, $\frac{\|y\|}{f_i(y)} \leq \frac{\|a_y x_0\|}{a_y f_i(x_0)} \leq M_0$. It is easy to obtain the same result for negative values y. Therefore, $\frac{\sum_{i=1}^N \|x_i\|}{\sum_{i=1}^N f_i(x_i)} \leq \sum_{i=1}^N \frac{\|x_i\|}{f_i(x_i)} \leq NM_0 := M$. Next, if $x_i^* \neq 0$ and $f_i(x_i^*) \neq 0$. Consider the transformation, $g_i(x) = f_i(x + x_i^*) - f_i(x_i^*)$, then $x = 0$ is the minimal solution for $g_i(x)$, $g_i(0) = 0$ and also $g_i(x)$ maintains the convexity. From the above, we know that $\frac{\|x\|}{g_i(x)} \leq M_0$. Therefore, $\frac{\|x\|}{f_i(x + x_i^*) - f_i(x_i^*)} \leq M_0$. Denote $y = x + x_i^*$, we have $\frac{\|y - x_i^*\|}{f_i(y) - f_i(x_i^*)} \leq M_0$. Consider the following limits:

$$\lim_{\|y\|\to\infty} \frac{\|y\|}{f_i(y)} \bigg/ \frac{\|y - x_i^*\|}{f(y) - f_i(x_i^*)} = \lim_{\|y\|\to\infty} \frac{\|y\|}{\|y - x_i^*\|} \bigg/ \frac{f_i(y)}{f_i(y) - f_i(x_i^*)}$$

$$= \lim_{\|y\|\to\infty} \frac{f_i(y) - f_i(x_i^*)}{f_i(y)} \leq 1 + \lim_{\|y\|\to\infty} |\frac{f_i(x_i^*)}{f_i(y)}| \leq 2,$$

which implies that $\lim_{\|y\|\to\infty} \frac{\|y\|}{f_i(y)} \leq 2M_0$. Similarly, we can show $\frac{\sum_{i=1}^N \|x_i\|}{\sum_{i=1}^N f_i(x_i)} \leq 2NM_0 := M$. This completes the proof.

APPENDIX B
PROOF SKETCH FOR THEOREM 4

First, we need the following useful lemma for series with convergent infinite sum.

Lemma 5. *Assume the infinite sum of the positive series* $\{a_k\}_{k=1}^\infty$, $\sum_{k=1}^\infty a_k < \infty$, *then* $a_k = o(1/k)$.

Define filtration $\{\mathcal{F}_k = \sigma\langle \mathbf{x}^1, \cdots, \mathbf{x}^k\rangle\}$, $\mathcal{F}_k \subset \mathcal{F}_{k+1}$. From Lemma 2, given α, the Lyapunov function $L_\alpha(\mathbf{x})$ has $(1 - \lambda_N(\mathbf{W}) + \alpha L)$-Lipschitz gradient. Hence, we can show that:

$$L_\alpha(\mathbf{x}^{k+1}) \leq L_\alpha(\mathbf{x}^k) - \|\nabla L_\alpha(\mathbf{x}^k)\|^2 + \langle \nabla L_\alpha(\mathbf{x}^k), \frac{\mathbf{Z}\epsilon^k}{k^\gamma}\rangle +$$

$$\frac{(1 - \lambda_N(\mathbf{W}) + \alpha L)}{2}[\|\nabla L_\alpha(\mathbf{x}^k)\|^2 + \|\frac{\mathbf{Z}\epsilon^k}{k^\gamma}\|^2 - 2\langle \nabla L_\alpha(\mathbf{x}^k), \frac{\mathbf{Z}\epsilon^k}{k^\gamma}\rangle],$$

Taking conditional expectation on the above inequality, using $\alpha < \frac{1 + \lambda_N(\mathbf{W})}{L}$, and then taking full expectation, we have:

$$\mathbb{E}[L_\alpha(\mathbf{x}^{k+1})] + \frac{(1 + \lambda_N(\mathbf{W}) - \alpha L)}{2}\mathbb{E}[\|\nabla L_\alpha(\mathbf{x}^k)\|^2]$$

$$\leq \mathbb{E}[L_\alpha(\mathbf{x}^k)] + \frac{(1 - \lambda_N(\mathbf{W}) + \alpha L)NP\sigma^2}{2k^{2\gamma}},$$

Telescoping and rearranging terms, we have:

$$\sum_{i=0}^k \frac{(1 + \lambda_N(\mathbf{W}) - \alpha L)}{2}\mathbb{E}[\|\nabla L_\alpha(\mathbf{x}^i)\|^2] \leq L_\alpha(\mathbf{x}^0)$$

$$- \alpha \sum_{i=1}^n f_i(x^*) + \sum_{i=0}^k \frac{(1 - \lambda_N(\mathbf{W}) + \alpha L)NP\sigma^2}{2i^{2\gamma}}. \qquad (15)$$

With the condition $\gamma > \frac{1}{2}$, Equation (15) indicates that $\sum_{i=0}^{\infty} \mathbb{E}[\|\nabla L_\alpha(\mathbf{x}^i)\|^2]$ is bounded and $\mathbb{E}[\|\nabla L_\alpha(\mathbf{x}^i)\|^2] = o(1/k)$, due to Lemma 5. Also, note that $\mathbb{E}[L_\alpha(\mathbf{x}^k)] < +\infty$ holds, $\forall k$. Due to the fact that $\mathbb{E}[\sum_{i=1}^{N} f_i(x_{i,k})] \leq \mathbb{E}[\frac{1}{\alpha} L_\alpha(\mathbf{x}^k)] < +\infty$, with Assumption 2, we can show:

$$\mathbb{E}[\|\mathbf{x}^k\|] \leq A + (M + \delta_A)\mathbb{E}[f(\mathbf{x}^k)] \leq +\infty,$$

where $p(\mathbf{x}^k)$ is the density of $\mathbf{x}^k$ and δ_A is from the fact that $\forall A > 0$, there exist $\delta_A < \infty$, such that $|\frac{\|\mathbf{x}^k\|}{f(\mathbf{x}^k)}| < M + \delta_A$. Hence, $\mathbb{E}[\|\mathbf{x}^k\|]$ is bounded, $\forall k$, by some constant B. Then, $\mathbb{E}[\|\nabla f(\mathbf{x}^k)\|] = \mathbb{E}[\|\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*)\|] \leq L\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|] \leq L(B + \|\mathbf{x}^*\|)$. This completes the proof.

APPENDIX C
PROOF FOR LEMMA 3

First of all, h_k could be bounded by a integral $h_k = \beta^k \sum_{i=1}^{k} \frac{\beta^{-i}}{i^\gamma} \leq \beta^k (\int_{x=1}^{k} \frac{\beta^{-x}}{x} dx + k)$. Thus, we focus on the increasing rate of $\int_{x=1}^{k} \frac{\beta^{-x}}{x} dx$. Denote $z := -\log(\beta) > 0$ and $s := 1 - \gamma \in (0, 1)$, we have the following complex integral:

$$\int_{x=1}^{k} \frac{\beta^{-x}}{x^\gamma} = \int_{x=1}^{k} \frac{e^{zx}}{x^{1-s}} dx \stackrel{(a)}{=} \int_{y=z}^{zk} \frac{e^y}{(y/z)^{1-s} z} dy$$

$$= z^{-s} \int_{y=z}^{zk} \frac{e^y}{y^{1-s}} dy \stackrel{(b)}{=} z^{-s} \int_{t=e^{-i\pi}z}^{e^{-i\pi}zk} e^{-t} t^{s-1} e^{i\pi s} dt$$

$$= z^{-s} e^{i\pi s} [\int_{t=e^{-i\pi}z}^{\infty} e^{-t} t^{s-1} dt - \int_{t=e^{-i\pi}kz}^{\infty} e^{-t} t^{s-1} dt]$$

$$\stackrel{(c)}{=} z^{-s} e^{i\pi s} [\Gamma(s, e^{-i\pi}z) - \Gamma(s, e^{-i\pi}kz)],$$

where $y = zx$ in (a), $t = e^{-i\pi}y$ in (b) with $e^{-i\pi} + 1 = 0$ and $\Gamma(\cdot, \cdot)$ is the upper incomplete gamma function in (c). With the property of $\Gamma(\cdot, \cdot)$, it follows that as $k \to +\infty$, $\Gamma(s, e^{-i\pi}kz) = (e^{-i\pi}kz)^{s-1} e^{kz} [1 + O(\frac{1}{k})]$, which implies the second term on the above equation is $z^{-s} e^{i\pi s} \Gamma(s, e^{-i\pi}kz) = -z^{-1} k^{-\gamma} \beta^{-k} [1 + O(\frac{1}{k})]$. Hence, the integral is $\int_{x=1}^{k} \frac{\beta^{-x}}{x^\gamma} = \beta^k \{z^{-s} e^{i\pi s} \Gamma(s, e^{-i\pi}z) + z^{-1} k^{-\gamma} \beta^{-k} [1 + O(\frac{1}{k})]\} = \frac{1}{k^{-\gamma}z} [1 + O(\frac{1}{k})] + O(\beta^k)$, and $h_k = O(\frac{1}{k^\gamma})$. This completes the proof.

APPENDIX D
PROOF SKETCH FOR THEOREM 1

Using $\bar{\mathbf{x}}^k = \frac{1}{N} \mathbf{1}\mathbf{1}^\top \mathbf{x}^k$, we can show that $\|\mathbf{x}^k - \bar{\mathbf{x}}^k\| = \|\mathbf{x}^k - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \mathbf{x}^k\| \leq \sum_{i=1}^{k-1} \alpha_i \beta^{k-i-1} \|\nabla f(\mathbf{x}^i)\| + \sum_{i=1}^{k-1} \beta^{k-i} \|\frac{\epsilon^i}{i^\gamma}\|$. Taking full expectation on the both sides yields:

$$\mathbb{E}[\|\mathbf{x}^k - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \mathbf{x}^k\|] \leq$$
$$\sum_{i=1}^{k-1} \alpha_i \beta^{k-i-1} \mathbb{E}[\|\nabla \mathbf{f}(\mathbf{x}^i)\|] + \sum_{i=1}^{k-1} \beta^{k-i} \frac{\sqrt{NP}\sigma}{i^\gamma}. \quad (16)$$

Hence, with Lemma 3, if $\mathbb{E}[\|\nabla \mathbf{f}(\mathbf{x}^i)\|]$ is bounded by D and step-size $\alpha_i = \alpha$, $\forall i$, then it follows that $\mathbb{E}[\|\mathbf{x}^k - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \mathbf{x}^k\|] \leq \frac{\alpha D}{1-\beta} + O(\frac{\sqrt{NP}\sigma}{k^\gamma})$; if the step-size $\alpha_i = O(\frac{1}{k^\eta})$ with $\eta > 0$, then $\mathbb{E}[\|\mathbf{x}^k - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \mathbf{x}^k\|] \leq \sum_{i=1}^{k-1} \beta^{k-i} (\frac{D}{\beta} \alpha_i + \sqrt{NP}\sigma \frac{1}{k^\gamma}) = O(\frac{1}{k^{\min(\eta,\gamma)}})$. This completes the proof.

APPENDIX E
PROOF SKETCH FOR LEMMA 4

With L-Lipschitz condition for local objectives, we can show the following: $\frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^{k+1}) \leq \frac{1}{N} \sum_{i=1}^{N} \langle \frac{\epsilon_k^\gamma y_{i,k}}{k^\gamma} \rangle + \frac{L}{2} \| - \frac{\alpha_k}{N} \sum_{i=1}^{N} \nabla f_i(x_{i,k}) + \frac{1}{N} \sum_{i=1}^{N} \frac{\epsilon_k^\gamma y_{i,k}}{k^\gamma} \|^2$. Then, taking conditional expectation and after some algebraic derivations yields the stated result.

APPENDIX F
PROOF SKETCH FOR THEOREM 2

From Theorem 4, we know that $\mathbb{E}\|\nabla f(\mathbf{x}^k)\| \leq L(B + \|\mathbf{x}^*\|)$. With Lemma 4 and Theorem 1, we can show that

$$\frac{\alpha}{2} \| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2 \leq= \frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^k) -$$

$$\mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^{k+1})|\mathcal{F}_k] + \frac{\alpha^3 L^4 (B + \|\mathbf{x}^*\|)^2}{2N(1-\beta)^2} + O(\frac{\alpha L \sqrt{NP}\sigma}{Nk^\gamma}).$$

Taking full expectation and telescoping yields:

$$\sum_{i=0}^{k} \frac{\alpha}{2} \mathbb{E}[\| \frac{1}{N} \sum_{i=1}^{N}] \nabla f_i(\bar{x}^k)\|^2 \leq \frac{1}{N} \sum_{i=1}^{N} f_i(0) - \frac{1}{N} \sum_{i=1}^{N} f_i(x^*)$$
$$+ \frac{k\alpha^3 L^4 (B + \|\mathbf{x}^*\|)^2}{2N(1-\beta)^2} + kO(\frac{\alpha L \sqrt{P}\sigma}{\sqrt{N}k^\gamma})$$

Then, by rearranging terms and some algebraic derivations, it can be further shown that $\min_k \{\mathbb{E}[\| \frac{1}{N} \sum_{i=1}^{N}] \nabla f_i(\bar{x}^k)\|^2\} \leq \frac{2}{\alpha k} [\frac{1}{N} \sum_{i=1}^{N} f_i(0) - \frac{1}{N} \sum_{i=1}^{N} f_i(x^*)] + \frac{\alpha^2 L^4 (B + \|\mathbf{x}^*\|)^2}{N(1-\beta)^2} + O(\frac{2L\sqrt{P}\sigma}{\sqrt{N}k^\gamma})$. This completes the proof.

APPENDIX G
PROOF SKETCH FOR THEOREM 3

From Lemma 4 and Theorem 1, we have:

$$\mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^{k+1})|\mathcal{F}_k] + \frac{\alpha_k}{2} \| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2$$

$$\leq \frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^k) + O(\frac{\alpha_k L^2}{2N^2 k^{\min(\eta,\gamma)}}) + \frac{LP\sigma^2}{2Nk^{2\gamma}}.$$

Consider the supermartingale convergence theorem [27]:

Theorem 6. *(Supermartingale Convergence Theorem) Let a^k, θ^k and ρ^k be positive sequences adapted to $\mathcal{F}_k$, and let ρ^k be summable with probably 1. If $\mathbb{E}[a^{k+1}|\mathcal{F}_k] + \theta^k \leq a^k + \rho^k$, then with probability 1, a^k converages to a $[0, \infty)$-valued random variable, and $\sum_{k=1}^{\infty} \theta^k < +\infty$.*

In our case, denote $a^k = \frac{1}{N} \sum_{i=1}^{N} f_i(\bar{x}^k) - \frac{1}{N} \sum_{i=1}^{N} f_i(x^*)$, $\theta^k = \frac{\alpha_k}{2} \| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2$, and $\rho^k = O(\frac{\alpha_k L^2}{2N^2 k^{\min(\eta,\gamma)}}) + \frac{LP\sigma^2}{2Nk^{2\gamma}} = O(\frac{1}{k^{\eta+\min(\eta,\gamma)}} + \frac{1}{k^{2\gamma}})$. With $\gamma > \frac{1}{2}$ and $\eta \geq \frac{1}{2}$, there exists $\epsilon > 0$ and $\rho^k = O(\frac{1}{k^{(1+\epsilon)}})$. With Lemma 5, ρ^k is summable with probably 1. Then θ^k is summable almost surely, i.e. $\sum_{i=1}^{\infty} \frac{\alpha_k}{2} \| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2 < +\infty$. Again, with Lemma 5, $\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{x}^k)\|^2 = o(\frac{1}{k^{1-\eta}})$ almost surely. This completes the proof.

2439