

# A Compressed Gradient Tracking Method for Decentralized Optimization With Linear Convergence

Yiwei Liao , Zhuorui Li , Kun Huang , and Shi Pu 

**Abstract**—Communication compression techniques are of growing interests for solving the decentralized optimization problem under limited communication, where the global objective is to minimize the average of local cost functions over a multiagent network using only local computation and peer-to-peer communication. In this article, we propose a novel compressed gradient tracking algorithm (C-GT) that combines gradient tracking technique with communication compression. In particular, C-GT is compatible with a general class of compression operators that unifies both unbiased and biased compressors. We show that C-GT inherits the advantages of gradient tracking-based algorithms and achieves linear convergence rate for strongly convex and smooth objective functions. Numerical examples complement the theoretical findings and demonstrate the efficiency and flexibility of the proposed algorithm.

**Index Terms**—Communication compression, decentralized optimization, gradient tracking, linear convergence.

## I. INTRODUCTION

In this article, we study the problem of decentralized optimization over a multiagent network that consists of  $n$  agents. The goal is to collaboratively solve the following optimization problem:

$$\min_{x \in \mathbb{R}^p} f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (1)$$

where  $x$  is the global decision variable, and each agent has a local objective function  $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ . The agents are connected through a communication network and can only exchange information with their immediate neighbors in the network. Through local computation and

local information exchange, they seek a consensual and optimal solution that minimizes the average of all the local cost functions. Decentralized optimization is widely applicable when central controllers or servers are not available or preferable, when centralized communication that involves a large amount of data exchange is prohibitively expensive due to limited communication resources, and when privacy preservation is desirable.

Problem (1) has attracted much attention in recent years and has found a variety of applications in wireless networks, distributed control of robotic systems, and machine learning, etc., [1]–[3]. To solve (1) over a multiagent network, early work considered the distributed subgradient descent (DGD) method with a diminishing step-size policy [4]. Under a constant step-size, EXTRA [5] first achieved linear convergence rate for strongly convex and smooth cost functions by introducing an extra correction term to DGD. Distributed gradient tracking-based methods were later developed in [6]–[9], where the local gradient descent direction in DGD was replaced by an auxiliary variable that is able to track the average gradient of the local objective functions. As a result, each agent's local iteration is moving in the global descent direction and converges exponentially to the optimal solution for strongly convex and smooth objective functions [8], [9]. Compared with EXTRA, gradient tracking-based methods are also suitable for uncoordinated step-sizes [6], [10], and possibly asymmetric weight matrices while preserving linear convergence rates. Some variants were also proposed to deal with stochastic gradient information and time-varying or directed network topologies, etc. For example, in [11], a distributed stochastic gradient tracking method was considered which exhibits comparable performance to a centralized stochastic gradient algorithm. Combining an approximate Newton-type method and gradient tracking leads to Network-DANE, which enables further computational savings by performing variance-reduced techniques [12]. Time-varying networks were considered in [8], [13]–[15], and more recent development on directed graphs can be found in [13], [16]–[20], and the references therein.

In many application scenarios, it is vital to design communication-efficient protocols for distributed computation due to limited communication bandwidth and power constraints. Recently, in order to improve system scalability and communication efficiency, researchers have considered a variety of communication compression methods, such as sparsification and quantization [21]–[30], under the master-worker centralized architecture. Several techniques were introduced to alleviate compression errors, including compression error compensation and gradient difference compression [21], [24], [26], [27].

In the decentralized setting, the difference compression and extrapolation compression techniques were introduced to reduce model compression error in [31]. A novel algorithm with communication compression (CHOCO-SGD), which combines with DGD and preserves the model average, was presented in [32] and [33]. But the method converges sublinearly even when the objective functions are strongly convex. In [34], a linearly convergent decentralized optimization algorithm with compression (LEAD) was introduced for

Manuscript received 14 February 2022; accepted 21 May 2022. Date of publication 7 June 2022; date of current version 27 September 2022. This work was supported in part by the Shenzhen Research Institute of Big Data (SRI BD) under Grant J00120190011, in part by the National Natural Science Foundation of China (NSFC) under Grant 62003287, in part by the Shenzhen Science and Technology Program under Grant RCYX20210609103229031 and Grant GXWD20201231105722002-20200901175001001, and in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS) under Grant AC01202101108. Recommended by Associate Editor Z. Kan. (Yiwei Liao and Zhuorui Li contributed as co-first authors). (Corresponding author: Shi Pu.)

Yiwei Liao is with the School of Data Science, Chinese University of Hong Kong, Shenzhen 518172, China, and also with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: lyw@stu.scu.edu.cn).

Zhuorui Li is with the Shenzhen Research Institute of Big Data, Shenzhen 518116, China (e-mail: lizhuorui27@gmail.com).

Kun Huang is with the School of Data Science, Shenzhen Research Institute of Big Data, Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: kunhuang@link.cuhk.edu.cn).

Shi Pu is with the School of Data Science, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: pushi@cuhk.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3180695>.

Digital Object Identifier 10.1109/TAC.2022.3180695

strongly convex and smooth objective functions. The method is based on NIDS [35], a sibling of EXTRA. In light of an incremental primal-dual method, a linearly convergent quantized decentralized optimization algorithm was developed for unbiased randomized compressors in [36]. In [37], a black-box model was provided for distributed algorithms based on finite-bit quantizers.

In light of the advantages of gradient tracking-based methods for decentralized optimization, it is natural to consider the marriage between gradient tracking and communication compression. The first such effort was made in [38] which considered a quantized gradient tracking method based on a special quantizer. It was shown to achieve linear convergence rate for strongly convex and smooth objective functions. However, the algorithm design is rather complicated and relies on a specific quantizer. In addition, the convergence conditions are not easy to verify.

In this article, we consider a novel gradient tracking-based method (C-GT) for decentralized optimization with communication compression. The algorithm compresses both the decision variables and the gradient trackers to provide a communication-efficient implementation. Unlike the existing methods which are mostly based on unbiased compressors or biased but contractive compressors, C-GT is provably efficient for a general class of compressors, including those which are neither unbiased nor biased but contractive, e.g., the composition of quantization and sparsification and the norm-sign compression operators. We show that C-GT achieves linear convergence for strongly convex and smooth objective functions under such a general class of communication compression techniques, where the agents may choose different, uncoordinated step-sizes.

The main contributions of the article are summarized as follows.

- 1) We propose a novel compressed gradient tracking algorithm (C-GT) for decentralized optimization, which inherits the advantages of gradient tracking-based methods and saves communication costs at the same time.
- 2) The proposed C-GT algorithm is applicable to a general class of compression operators and works under arbitrary compression precision. In particular, the general condition on the compression operators unifies the commonly considered unbiased and biased but contractive compressors and also includes other compression methods such as the composition of quantization and sparsification and the norm-sign compressors.
- 3) C-GT provably achieves linear convergence for minimizing strongly convex and smooth objective functions under the general condition on the compression operators, where the agents may choose different, uncoordinated step-sizes.
- 4) Simulation examples show that C-GT is efficient compared to the state-of-the-art methods and widely applies to various compressors.

The rest of this article is organized as follows. We present the general condition on the compressors and the C-GT algorithm in Section III. In Section IV, we perform the convergence analysis for C-GT. Numerical examples are provided in Section V. Finally, Section VI concludes this article.

## A. Notation

Vectors are columns if not otherwise specified in this article. Let each agent  $i$  hold a local copy  $\mathbf{x}_i \in \mathbb{R}^p$  of the decision variable and a gradient tracker (auxiliary variable)  $\mathbf{y}_i \in \mathbb{R}^p$ . At the  $k$ th iteration, their values are denoted by  $\mathbf{x}_i^k$  and  $\mathbf{y}_i^k$ , respectively. For notational convenience, define  $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times p}$ ,  $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times p}$ , and  $\bar{\mathbf{X}} := \frac{1}{n} \mathbf{1}^\top \mathbf{X} \in \mathbb{R}^{1 \times p}$ ,  $\bar{\mathbf{Y}} := \frac{1}{n} \mathbf{1}^\top \mathbf{Y} \in \mathbb{R}^{1 \times p}$ , where  $\mathbf{1}$  is the column vector with each entry given by 1. At the  $k$ th iteration, their

values are denoted by  $\mathbf{X}^k$ ,  $\mathbf{Y}^k$ ,  $\bar{\mathbf{X}}^k$  and  $\bar{\mathbf{Y}}^k$ , respectively. Auxiliary variables of the agents (in an aggregative matrix form)  $\mathbf{H}_x$ ,  $\mathbf{H}_y$ ,  $\mathbf{Q}_x$ ,  $\mathbf{Q}_y$ ,  $\hat{\mathbf{X}}$ , and  $\hat{\mathbf{Y}}$  are defined similarly. Denote the aggregative gradient  $\nabla \mathbf{F}(\mathbf{X}) := [\nabla f_1(\mathbf{x}_1), \nabla f_2(\mathbf{x}_2), \dots, \nabla f_n(\mathbf{x}_n)]^\top \in \mathbb{R}^{n \times p}$ , and  $\nabla \mathbf{F}(\mathbf{X}) := \frac{1}{n} \mathbf{1}^\top \nabla \mathbf{F}(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i)$ .

We use  $\|\cdot\|$  to denote the Frobenius norm of vectors and matrices by default. Specially, for square matrices,  $\|\cdot\|$  represents the spectral norm. The spectral radius of a square matrix  $\mathbf{M}$  is denoted by  $\rho(\mathbf{M})$ .

## II. PROBLEM FORMULATION

In this section, we provide the assumptions on the communication graphs and the objective functions. Then, we discuss different kinds of compression methods and provide a general description for compression operators.

### A. Preliminaries

We start with introducing the conditions on the communication network/graph and the objective functions. Assume the agents are connected over an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{1, 2, \dots, n\}$  is the set of vertices (nodes) and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges. For an arbitrary agent  $i \in \mathcal{V}$ , we define the set of its neighbors as  $\mathcal{N}_i$ . Regarding the network structure, we make the following standing assumption.

*Assumption 1:* The undirected graph  $\mathcal{G}$  is strongly connected and permits a nonnegative doubly stochastic weight matrix  $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$ . That is, agent  $i$  can receive information from agent  $j$  if and only if  $w_{ij} > 0$ , and  $\mathbf{W}\mathbf{1} = \mathbf{1}$  and  $\mathbf{1}^\top \mathbf{W} = \mathbf{1}^\top$ .

*Remark 1:* Although we assume an undirected graph  $\mathcal{G}$ , note that the considered C-GT method also works with any balanced directed graph, where it is convenient to construct a doubly stochastic weight matrix.

The assumption on the objective functions is given below.

*Assumption 2:* The local cost function  $f_i$  is  $\mu_i$ -strongly convex, and its gradient is  $L_i$ -Lipschitz continuous, i.e., for any  $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$

$$\langle \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}'), \mathbf{x} - \mathbf{x}' \rangle \geq \mu_i \|\mathbf{x} - \mathbf{x}'\|^2 \quad (2)$$

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}')\| \leq L_i \|\mathbf{x} - \mathbf{x}'\|. \quad (3)$$

From Assumption 2, the objective function  $f$  is  $\mu$ -strongly convex and the gradient of  $f$  is  $L$ -Lipschitz continuous, where  $\mu = \frac{1}{n} \sum_{i=1}^n \mu_i$  and  $L = \max \{L_i\}$ .<sup>1</sup> Moreover, there exists a unique solution denoted by  $\mathbf{x}^* \in \mathbb{R}^{1 \times p}$  to problem (1) under Assumption 2.

### B. Compression Methods

In this section, we introduce some common assumptions on the compression operators and then present a more general and unified assumption.

#### 1) Unbiased Compression Operators:

*Assumption 3:* The compression operator  $\mathcal{Q} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  satisfies  $\mathbb{E} \mathcal{Q}(\mathbf{x}) = \mathbf{x}$  and there exists a constant  $C \geq 0$  such that  $\mathbb{E} \|\mathcal{Q}(\mathbf{x}) - \mathbf{x}\|^2 \leq C \|\mathbf{x}\|^2 \forall \mathbf{x} \in \mathbb{R}^d$ .

*Remark 2:* The expectation is taken with respect to the random vector corresponding to the internal compression randomness of  $\mathcal{Q}$ . Some instances of feasible stochastic compression operators satisfying Assumption 3, such as the unbiased  $b$ -bits  $q$ -norm quantization compression method, can be found in [32]–[34], and the references therein.

<sup>1</sup>We denote  $\kappa = L/\mu$  as the condition number.

## 2) Biased Compression Operators:

**Assumption 4:** The compression operator  $\mathcal{C}_\delta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  satisfies  $\mathbb{E}_{\mathcal{C}_\delta} \|\mathcal{C}_\delta(\mathbf{x}) - \mathbf{x}\|^2 \leq (1 - \delta) \|\mathbf{x}\|^2, \forall \mathbf{x} \in \mathbb{R}^d$ , where  $\delta \in (0, 1]$ .

**Remark 3:** If  $\delta = 1$ , there is no compression error, i.e.,  $\mathcal{C}_\delta(\mathbf{x}) = \mathbf{x}$ . For instance, the Top- $k$  and Random- $k$  methods (see e.g., [29], [33]) satisfy Assumption 4, where  $\delta$  is given by  $\delta = \frac{k}{p}$ .

**3) General Compression Operators:** We now present a general assumption on the compression operators, which contains Assumptions 3 and 4 as special cases.

**Assumption 5:** The compression operator  $\mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  satisfies

$$\mathbb{E}_{\mathcal{C}} \|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|^2 \leq C \|\mathbf{x}\|^2 \quad \forall \mathbf{x} \in \mathbb{R}^d \quad (4)$$

and the  $r$ -scaling of  $\mathcal{C}$  satisfies

$$\mathbb{E}_{\mathcal{C}} \left\| \frac{\mathcal{C}(\mathbf{x})}{r} - \mathbf{x} \right\|^2 \leq (1 - \delta) \|\mathbf{x}\|^2 \quad \forall \mathbf{x} \in \mathbb{R}^d \quad (5)$$

for some constants  $\delta \in (0, 1]$  and  $r > 0$ .

**Remark 4:** On one hand, if  $C < 1$ , Assumption 5 degenerates to Assumption 4 by setting  $r = 1$  and  $\delta = 1 - C$ . On the other hand, if  $\mathcal{C}$  is unbiased, i.e.,  $\mathbb{E}\mathcal{C}(\mathbf{x}) = \mathbf{x}$ , then Assumption 5 degenerates to Assumption 3 by setting  $r = C + 1$  and  $\delta = \frac{1}{C+1}$ . In short, Assumption 5 gives a unified description of unbiased and biased compression operators and thus Assumptions 3 and 4 can be regarded as its special cases.

However, there also exist compression operators where  $\mathcal{C}$  is biased and  $C \geq 1$  in Assumption 5, that is, they do not satisfy Assumptions 3 and 4. Examples include the norm-sign compressor  $\mathcal{C}(\mathbf{x}) = \|\mathbf{x}\|_q \text{sign}(\mathbf{x})$  and the composition of quantization and sparsification [23], [25], [39].

**Remark 5:** Although some compression operators (e.g., composition of quantization and sparsification) can be rescaled so that the new compression operator satisfies the contractive condition in Assumption 4, applying the rescaled operator may hurt the performance of the algorithm when compared with directly using the original compression operator  $\mathcal{C}$ . Considering Assumption 5 provides us with more flexibility in choosing the most suitable compression method.

## III. A COMPRESSED GRADIENT TRACKING ALGORITHM

In this section, we introduce the communication-efficient compressed gradient tracking algorithm (C-GT). We also give some interpretations as well as how C-GT connects to existing works.

Denote  $\mathbf{D} = \text{diag}([\eta_1, \eta_2, \dots, \eta_n])$ , where  $\eta_i$  is the step-size of agent  $i$ . Let **Compress** be the compression function, and the compression operators are associated with the function **Compress**. The proposed compressed gradient tracking algorithm (C-GT) is presented in Algorithm 1.

The compression and communication steps are included in the procedure **COMMZ, H, H<sub>w</sub>**. The function **Compress** is the compression operator that independently compresses the variables for each agent per iteration. In Line 10, the difference between  $\mathbf{Z}$  and the auxiliary variable  $\mathbf{H}$  is compressed and then added back to  $\mathbf{H}$  in Line 11 for obtaining  $\hat{\mathbf{Z}}$ . Here,  $\mathbf{H}$  acts as a reference point, and when it gradually approaches  $\mathbf{Z}$  such that the difference vanishes to 0, the compression error on the difference will also decrease to 0 under Assumption 5. The low-bit compressed value  $\mathbf{Q}$  is transmitted in Line 12.

To control the compression error, particularly for a relatively large constant  $C$  in Assumption 5, we introduce a momentum update  $\mathbf{H} = (1 - \alpha_z)\mathbf{H} + \alpha_z \hat{\mathbf{Z}}$  motivated by the centralized distributed method DIANA [26] and the decentralized algorithm LEAD [34]. If  $\alpha_z = 1$ , the update degenerates to that in the decentralized stochastic algorithm CHOCO-SGD [32].

### Algorithm 1: A Compressed Gradient Tracking (C-GT) Algorithm.

**Input:** stopping time  $K$ , step-size  $\{\eta_i\}$ , consensus step-size  $\gamma$ , scaling parameters  $\alpha_x, \alpha_y$ , and initial values  $\mathbf{X}^0, \mathbf{H}_x^0, \mathbf{H}_y^0, \mathbf{Y}^0 = \nabla \mathbf{F}(\mathbf{X}^0)$

**Output:**  $\mathbf{X}^K, \mathbf{Y}^K$

- 1:  $\mathbf{H}_{x,w}^0 = \mathbf{W}\mathbf{H}_x^0$
- 2:  $\mathbf{H}_{y,w}^0 = \mathbf{W}\mathbf{H}_y^0$
- 3: **for**  $k = 0, 1, 2, \dots, K - 1$  **do**
- 4:  $\hat{\mathbf{X}}^k, \hat{\mathbf{X}}_w^k, \mathbf{H}_{x,w}^{k+1}, \mathbf{H}_{x,w}^{k+1} = \text{COMM}\mathbf{X}^k, \mathbf{H}_x^k, \mathbf{H}_{x,w}^k$
- 5:  $\hat{\mathbf{Y}}^k, \hat{\mathbf{Y}}_w^k, \mathbf{H}_{y,w}^{k+1}, \mathbf{H}_{y,w}^{k+1} = \text{COMM}\mathbf{Y}^k, \mathbf{H}_y^k, \mathbf{H}_{y,w}^k$
- 6:  $\mathbf{X}^{k+1} = \mathbf{X}^k - \gamma(\hat{\mathbf{X}}^k - \hat{\mathbf{X}}_w^k) - \mathbf{D}\mathbf{Y}^k$
- 7:  $\mathbf{Y}^{k+1} = \mathbf{Y}^k - \gamma(\hat{\mathbf{Y}}^k - \hat{\mathbf{Y}}_w^k) + \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \nabla \mathbf{F}(\mathbf{X}^k)$
- 8: **end for**
- 9: **procedure** **COMMZ, H, H<sub>w</sub>**
- 10:  $\mathbf{Q} = \text{Compress}(\mathbf{Z} - \mathbf{H}) \triangleright$  Compression
- 11:  $\hat{\mathbf{Z}} = \mathbf{H} + \mathbf{Q}$
- 12:  $\hat{\mathbf{Z}}_w = \mathbf{H}_w + \mathbf{W}\mathbf{Q} \triangleright$  Communication
- 13:  $\mathbf{H} \leftarrow (1 - \alpha_z)\mathbf{H} + \alpha_z \hat{\mathbf{Z}}$
- 14:  $\mathbf{H}_w \leftarrow (1 - \alpha_z)\mathbf{H}_w + \alpha_z \hat{\mathbf{Z}}_w$
- 15: **Return:**  $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}_w, \mathbf{H}, \mathbf{H}_w$
- 16: **end procedure**

<sup>a</sup>In Lines 4 and 5,  $\alpha_z$  in the compression function is replaced by  $\alpha_x$  for decision difference compression and  $\alpha_y$  for gradient tracker difference compression, respectively.

In Line 14,  $\mathbf{H}_w$  is used as a backup copy for the neighboring information. By introducing such an auxiliary variable, there is no need to store all the neighbors' reference points  $\mathbf{H}$  [32]. Noticing that  $\mathbf{H}_w^0 = \mathbf{W}\mathbf{H}^0$  from the initialization in Lines 1 and 2, we have  $\hat{\mathbf{Z}}_w = \mathbf{W}\hat{\mathbf{Z}}$  and then  $\mathbf{H}_w = \mathbf{W}\mathbf{H}$  by induction. It follows that  $\hat{\mathbf{X}}_w^k = \mathbf{W}\hat{\mathbf{X}}^k$  and  $\hat{\mathbf{Y}}_w^k = \mathbf{W}\hat{\mathbf{Y}}^k$  from Lines 4 and 5. Therefore, the decision variable update in Line 6 becomes

$$\begin{aligned} \mathbf{X}^{k+1} &= \mathbf{X}^k - \gamma(\hat{\mathbf{X}}^k - \mathbf{W}\hat{\mathbf{X}}^k) - \mathbf{D}\mathbf{Y}^k \\ &= \mathbf{X}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{X}}^k - \mathbf{D}\mathbf{Y}^k \end{aligned} \quad (6)$$

and the gradient tracker update in Line 7 is given by

$$\begin{aligned} \mathbf{Y}^{k+1} &= \mathbf{Y}^k - \gamma(\hat{\mathbf{Y}}^k - \mathbf{W}\hat{\mathbf{Y}}^k) + \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \nabla \mathbf{F}(\mathbf{X}^k) \\ &= \mathbf{Y}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{Y}}^k + \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \nabla \mathbf{F}(\mathbf{X}^k). \end{aligned} \quad (7)$$

One key property of C-GT is that gradient tracking is efficient regardless of the compression errors, i.e., for  $k \geq 0$

$$\begin{aligned} \mathbf{1}^\top \mathbf{Y}^{k+1} &= \mathbf{1}^\top (\mathbf{Y}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{Y}}^k + \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \nabla \mathbf{F}(\mathbf{X}^k)) \\ &= \mathbf{1}^\top \mathbf{Y}^k + \mathbf{1}^\top \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \mathbf{1}^\top \nabla \mathbf{F}(\mathbf{X}^k) \\ &= \mathbf{1}^\top \nabla \mathbf{F}(\mathbf{X}^{k+1}). \end{aligned} \quad (8)$$

The second equality holds because  $\mathbf{1}^\top (\mathbf{I} - \mathbf{W}) = 0$ , and the last equality is obtained by induction under the initial condition  $\mathbf{Y}^0 = \nabla \mathbf{F}(\mathbf{X}^0)$ . Therefore, as long as  $\mathbf{y}_i^k$  reaches (approximate) consensus among all the agents, each  $\mathbf{y}_i^k$  is able to track the average gradient  $\mathbf{1}^\top \nabla \mathbf{F}(\mathbf{X}^k)/n$ . Moreover, by multiplying  $\mathbf{1}^\top$  and dividing  $n$  on both sides of Line 8, we obtain

$$\bar{\mathbf{X}}^{k+1} = \bar{\mathbf{X}}^k - \frac{1}{n} \mathbf{1}^\top \mathbf{D}\mathbf{Y}^k. \quad (9)$$

Hence, the update of  $\bar{\mathbf{X}}^k$  does not involve any compression error from the current step.

*Remark 6:* If no communication compression is performed in the algorithm, i.e.,  $\hat{\mathbf{X}}^k = \mathbf{X}^k$  and  $\hat{\mathbf{Y}}^k = \mathbf{Y}^k$ , then C-GT recovers the typical distributed gradient tracking algorithm in [8] and [9] for  $\eta_i = \eta$ . To see such a connection, note that C-GT reads

$$\begin{aligned}\mathbf{X}^{k+1} &= \mathbf{X}^k - \gamma(\mathbf{I} - \mathbf{W})\mathbf{X}^k - \eta\mathbf{Y}^k \\ &= [(1 - \gamma)\mathbf{I} + \gamma\mathbf{W}]\mathbf{X}^k - \eta\mathbf{Y}^k\end{aligned}\quad (10)$$

and

$$\begin{aligned}\mathbf{Y}^{k+1} &= \mathbf{Y}^k - \gamma(\mathbf{I} - \mathbf{W})\mathbf{Y}^k + \nabla\mathbf{F}(\mathbf{X}^{k+1}) - \nabla\mathbf{F}(\mathbf{X}^k) \\ &= [(1 - \gamma)\mathbf{I} + \gamma\mathbf{W}]\mathbf{Y}^k + \nabla\mathbf{F}(\mathbf{X}^{k+1}) - \nabla\mathbf{F}(\mathbf{X}^k)\end{aligned}\quad (11)$$

where we substitute  $\hat{\mathbf{X}}^k = \mathbf{X}^k$  and  $\hat{\mathbf{Y}}^k = \mathbf{Y}^k$  in (6) and (7), respectively. By denoting  $\tilde{\mathbf{W}} := (1 - \gamma)\mathbf{I} + \gamma\mathbf{W}$ , C-GT takes the same form as the typical gradient tracking method.

On the other hand, C-GT performs an implicit error compensation operation that mitigates the impact of the compression error, as can be seen from the following argument. The decision variable is updated as

$$\begin{aligned}\mathbf{X}^{k+1} &= \mathbf{X}^k - \gamma(\mathbf{I} - \mathbf{W})(\mathbf{X}^k - \mathbf{E}^k) - \mathbf{D}\mathbf{Y}^k \\ &= [(1 - \gamma)\mathbf{I} + \gamma\mathbf{W}]\mathbf{X}^k - \mathbf{D}\mathbf{Y}^k + \gamma(\mathbf{I} - \mathbf{W})\mathbf{E}^k\end{aligned}\quad (12)$$

where  $\mathbf{E}^k := \mathbf{X}^k - \hat{\mathbf{X}}^k$  measures the compression error for the decision variable. The additional term  $(\mathbf{I} - \mathbf{W})\mathbf{E}^k$  implies that each agent  $i$  transmits its total compression error  $-\sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ji}\mathbf{e}_i^k = -\mathbf{e}_i^k$  to its neighboring agents and compensates this error locally by adding  $\mathbf{e}_i^k$ , where  $\mathbf{e}_i^k \in \mathbb{R}^{1 \times p}$  is the  $i$ th row of  $\mathbf{E}^k$ . Similarly, the compression errors for the gradient trackers are also mitigated.

#### IV. CONVERGENCE ANALYSIS

In this section, we study the convergence properties of the proposed compressed gradient tracking algorithm for minimizing strongly convex and smooth cost functions. Our analysis relies on constructing a linear system of inequalities that is related to the optimization error  $\Omega_o^k := \mathbb{E}[\|\bar{\mathbf{X}}^k - \mathbf{x}^*\|^2]$ , consensus error  $\Omega_c^k := \mathbb{E}[\|\mathbf{X}^k - \mathbf{1}\bar{\mathbf{X}}^k\|^2]$ , gradient tracking error  $\Omega_g^k := \mathbb{E}[\|\mathbf{Y}^k - \mathbf{1}\bar{\mathbf{Y}}^k\|^2]$ , and compression errors  $\Omega_{cx}^k := \mathbb{E}[\|\mathbf{X}^k - \mathbf{H}_x^k\|^2]$ , and  $\Omega_{cy}^k := \mathbb{E}[\|\mathbf{Y}^k - \mathbf{H}_y^k\|^2]$ .

In order to derive the main results, we introduce some useful lemmas first.

*Lemma 1:* Under Assumption 2, for all  $k \geq 0$ , there holds

$$\|\nabla f((\bar{\mathbf{X}}^k)^\top) - (\bar{\mathbf{Y}}^k)^\top\| \leq \frac{L}{\sqrt{n}}\|\mathbf{X}^k - \mathbf{1}\bar{\mathbf{X}}^k\|. \quad (13)$$

In addition, if  $\eta < 2/(\mu + L)$ , then we have

$$\|\mathbf{x} - \eta\nabla f(\mathbf{x}) - (\mathbf{x}^*)^\top\| \leq (1 - \eta\mu)\|\mathbf{x} - (\mathbf{x}^*)^\top\| \quad \forall \mathbf{x} \in \mathbb{R}^p. \quad (14)$$

*Lemma 2:* Suppose Assumption 1 holds. For any  $\omega \in \mathbb{R}^{n \times p}$ , we have  $\|\mathbf{W}\omega - \mathbf{1}\bar{\omega}\| \leq \rho_w\|\omega - \mathbf{1}\bar{\omega}\|$ , where  $\bar{\omega} = \frac{1}{n}\mathbf{1}^\top\omega$  and  $\rho_w < 1$  is the spectral norm of the matrix  $\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ .

The proofs of Lemma 1 and Lemma 2 can be found in [9, Lemma 10].

*Remark 7:* For  $\gamma \in (0, 1]$ , if we define  $\tilde{\mathbf{W}} = (1 - \gamma)\mathbf{I} + \gamma\mathbf{W}$ , then  $\|\tilde{\mathbf{W}}\omega - \mathbf{1}\bar{\omega}\| \leq \tilde{\rho}\|\omega - \mathbf{1}\bar{\omega}\|$ , where  $\tilde{\rho} = 1 - \gamma s$ , and  $s = 1 - \rho_w$ .

Denote  $\bar{\eta} = \frac{1}{n} \sum_{i=1}^n \eta_i$ ,  $\hat{\eta} = \max_i \eta_i$ . We introduce below the key lemma for establishing the linear convergence of the C-GT algorithm under Assumptions 1, 2, and 5.

*Lemma 3:* Suppose Assumptions 1, 2, and 5 hold and  $\hat{\eta} < \min\{\frac{2}{\mu+L}, \frac{1}{3\mu}\}$ . Then we have the following linear system of inequalities:

$$\mathbf{w}^{k+1} \leq \mathbf{A}\mathbf{w}^k$$

where  $\mathbf{w}^k := [\Omega_o^k, \Omega_c^k, \Omega_g^k, \Omega_{cx}^k, \Omega_{cy}^k]^\top$ . The inequality is to be taken component-wisely, where  $\mathbf{A} \in \mathbb{R}^{5 \times 5}$  is nonnegative.<sup>2</sup>

*Proof:* See Appendix B in [40]. ■

Based on Lemma 3, we present the preliminary convergence result for the C-GT algorithm below.

*Lemma 4:* Suppose Assumptions 1, 2 and 5 hold, the scaling parameters  $\alpha_x, \alpha_y \in (0, \frac{1}{r}]$ ,  $\bar{\eta} \geq M\hat{\eta}$  for some  $M > 0$ , and

$$\begin{aligned}n\epsilon_1 &\geq \frac{12\kappa^2}{M^2}\epsilon_3, \quad \epsilon_2 \geq \frac{12C\lambda}{s^2}\epsilon_4, \quad \epsilon_3 \geq M\epsilon_2 \\ \epsilon_3 &\geq \frac{12\lambda(3\epsilon_2 + C(3\epsilon_4 + \epsilon_5))}{s^2}, \quad \epsilon_4 > 0, \quad \epsilon_5 > 0\end{aligned}\quad (15)$$

$$\gamma \leq \min\left\{1, \frac{1 - c_x}{m_x}\epsilon_4, \frac{1 - c_y}{m_y}\epsilon_5\right\} \quad (16)$$

$$\hat{\eta} \leq \min\left\{\sqrt{\frac{\epsilon_2}{12}}, \sqrt{\frac{\epsilon_3}{36}}\right\} \frac{s\gamma}{\sqrt{2n\epsilon_1 + 2\epsilon_2 + \epsilon_3}L} \quad (17)$$

where  $s = 1 - \rho_w$ ,  $\lambda := \|\mathbf{I} - \mathbf{W}\|^2$ ,  $m_x := t_x(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + t_x\lambda(\epsilon_2 + C\epsilon_4) + \frac{M\epsilon_4}{2\kappa}$ ,  $m_y := 3t_y(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + t_y\lambda(3\epsilon_2 + \epsilon_3 + 3C\epsilon_4 + C\epsilon_5) + \frac{M\epsilon_5}{2\kappa}$ ,  $c_x = \tau_x(1 - \alpha_x r\delta) < 1$ ,  $c_y = \tau_y(1 - \alpha_y r\delta) < 1$ ,  $t_x = \frac{3\tau_x}{\tau_x - 1}$ ,  $t_y = \frac{3\tau_y}{\tau_y - 1}$ , constants  $\tau_x, \tau_y > 1$ , and  $\epsilon_1 - \epsilon_5$  are some positive constants. Then, the spectral radius of  $\mathbf{A}$  satisfies  $\rho(\mathbf{A}) \leq 1 - \frac{1}{2}M\hat{\eta}\mu$ , and the optimization error  $\Omega_o^k$  and the consensus error  $\Omega_c^k$  both converge to 0 at the linear rate  $\mathcal{O}((1 - \frac{1}{2}M\hat{\eta}\mu)^k)$ .

*Proof:* See Appendix A2. ■

#### A. Main Results

By taking some concrete values for the constants in Lemma 4, we derive the main convergence result for the C-GT algorithm under Assumptions 1, 2, and 5 in the following theorem, which demonstrates the linearly convergent property of C-GT for the general compression operators.

*Theorem 1:* Suppose Assumptions 1, 2, 5 hold,  $\bar{\eta} \geq M\hat{\eta}$  for some  $M > 0$ ,  $\alpha_x, \alpha_y = 1/r$ , the consensus step-size  $\gamma$  satisfies

$$\gamma \leq \min\left\{1, \frac{\delta}{m(2 - \delta)}\right\} \quad (18)$$

and the maximum step-size  $\hat{\eta}$  satisfies

$$\hat{\eta} \leq \frac{s^2}{\sqrt{12\left[\left(\frac{24\kappa^2}{M^2} + 1\right)(4s^2 + 36\lambda) + 2s^2\right]}} \frac{\gamma}{L} \quad (19)$$

where  $\kappa = L/\mu$  and  $m = \frac{6}{\delta}\left[\left(\frac{72\kappa^2}{M^2} + \lambda + 3\right)\left(\frac{432C\lambda^2}{s^4} + \frac{48C\lambda}{s^2}\right) + (3\lambda + 6)\frac{12C\lambda}{s^2} + 4C\lambda\right] + \frac{M}{2\kappa}$  with  $\lambda = \|\mathbf{I} - \mathbf{W}\|^2$ . Then, the optimization error  $\Omega_o^k$  and the consensus error  $\Omega_c^k$  both converge to 0 at the linear rate  $\mathcal{O}\left((1 - \frac{1}{2}M\hat{\eta}\mu)^k\right)$ .

*Proof:* See Appendix A3. ■

<sup>2</sup>The elements of the transition matrix  $\mathbf{A}$  correspond to the parameters of the inequalities in the proof.



*Remark 8:* If the compression error is sufficiently small, i.e.,  $C \rightarrow 0$  and  $\delta \rightarrow 1$ , we have  $m \approx \frac{M}{2\kappa} < 1$ . Then, we obtain  $\gamma \leq 1$  and

$$\hat{\eta} \leq \frac{s^2}{\sqrt{12 \left[ \left( \frac{24\kappa^2}{M^2} + 1 \right) (4s^2 + 36\lambda) + 2s^2 \right]}}$$

$$\frac{1}{L} \sim \mathcal{O} \left( \frac{(1 - \rho_w)^2}{\kappa L} \right).$$

The convergence rate of C-GT is then comparable to those of the typical gradient tracking methods; see, e.g., [9].

*Remark 9:* In practice, the restrictions on  $\alpha_x$  and  $\alpha_y$  can be relaxed to  $\alpha_x, \alpha_y \in (0, \frac{1}{r})$  as in Lemma 4. The condition  $\bar{\eta} \geq M\hat{\eta}$  is always satisfied for some fixed  $M$ , e.g.,  $M = \frac{1}{n}$ . If in addition that all  $\eta_i$  are equal, then we can take  $M = 1$ .

*Remark 10:* Comparing the performance of C-GT with the existing linearly convergent algorithm LEAD [35], C-GT enjoys more flexibility in the mixing matrix, compression methods, and the stepsize policy, while LEAD achieves faster convergence in theory under more restricted conditions.

## V. NUMERICAL EXAMPLES

In this part, we provide some numerical examples to confirm our theoretical results. Consider the ridge regression problem

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \left( (u_i^\top x - v_i)^2 + \rho \|x\|^2 \right) \quad (20)$$

where  $\rho > 0$  is a penalty parameter. The pair  $(u_i, v_i)$  is a sample that belongs to the  $i$ th agent, where  $u_i \in \mathbb{R}^p$  represents the features and  $v_i \in \mathbb{R}$  represents the observations or outputs. In the simulations, pairs  $(u_i, v_i)$  are pregenerated: input  $u_i \in [-1, 1]^p$  is uniformly distributed, and the output  $v_i$  satisfies  $v_i = u_i^\top \tilde{x}_i + \varepsilon_i$ , where  $\varepsilon_i$  are independent Gaussian noises with mean 0 and variance 25, and  $\tilde{x}_i$  are predefined parameters evenly located in  $[0, 1]^p$ . Then, the  $i$ th agent can calculate the gradient of its local objective function  $f_i(x)$  with  $g_i(x, u_i, v_i) = 2(u_i^\top x - v_i)u_i + 2\rho x$ . The unique optimal solution of the problem is  $x^* = (\sum_{i=1}^n u_i u_i^\top + n\rho \mathbf{I})^{-1} \sum_{i=1}^n u_i v_i$ .

In our experimental settings, we consider penalty parameter  $\rho = 0.1$ . The number of nodes is  $n = 100$ , and the dimension of variables is  $p = 500$ . Meanwhile,  $\mathbf{x}_i^0$  is randomly generated in  $[0, 1]^p$  and other initial values satisfy  $\mathbf{H}_x^0 = \mathbf{0}$ ,  $\mathbf{H}_y^0 = \mathbf{0}$ , and  $\mathbf{Y}^0 = \nabla \mathbf{F}(\mathbf{X}^0)$ .

We compare C-GT with CHOCO-SGD [32], LB [36], LEAD [34] and the uncompressed linearly convergent methods, NIDS [35] and GT [9], for decentralized optimization over a randomly generated undirected graph. In order to guarantee the fairness, all algorithms use their equivalent matrix forms. The considered compression methods are 2-b  $\infty$ -norm quantization (Q), Top-10 sparsification, composition of quantization and sparsification (Q-T) and its rescaled version (Q-T-R). Note that Q-T only satisfies Assumption 5 and does not satisfy Assumptions 3 and 4. The communication bits of these compression methods are given in [32]–[34], [39].<sup>3</sup> The parameter settings of the algorithms are given in Table I, which are hand-tuned to achieve the best performance for each algorithm.

In Fig. 1, we compare the communication efficiency of C-GT, LEAD, LB, and CHOCO-SGD with the uncompressed methods GT and NIDS. For C-GT, LEAD, LB, and CHOCO-SGD, we apply the compressors that work the best for them,

<sup>3</sup>Note that C-GT requires two times the communication bits of the other algorithms per iteration since it compressed both the decision variable and the gradient tracker.

TABLE I  
PARAMETER SETTING FOR DIFFERENT ALGORITHMS AND COMPRESSION METHODS

Algorithm	Compressor	$\alpha_x$	$\alpha_y$	$\gamma$	$\eta$
LEAD	Top-10	1	1	$7 \times 10^{-5}$	0.005
C-GT	Top-10	1	1	0.05	0.005
LEAD	Q	1	1	0.09	0.005
C-GT	Q	1	1	0.09	0.005
CHOCO-SGD	Q	1	1	0.7	0.005
LB	Q	1	1	0.1	0.005
LEAD	Q-T	1	1	$7 \times 10^{-5}$	0.005
C-GT	Q-T	1	1	0.06	0.005
LEAD	Q-T-R	1	1	$3 \times 10^{-5}$	0.005
C-GT	Q-T-R	1	1	0.009	0.005
GT	/	/	/	/	0.009
NIDS	/	/	/	/	0.006

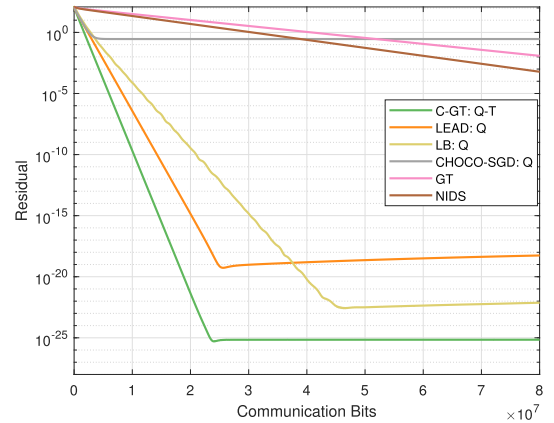


Fig. 1. Residuals  $\mathbb{E}[\|\bar{\mathbf{X}}^h - \mathbf{x}^*\|^2]$  against the communication bits for C-GT, LEAD, LB, CHOCO-SGD, and the uncompressed methods GT and NIDS.

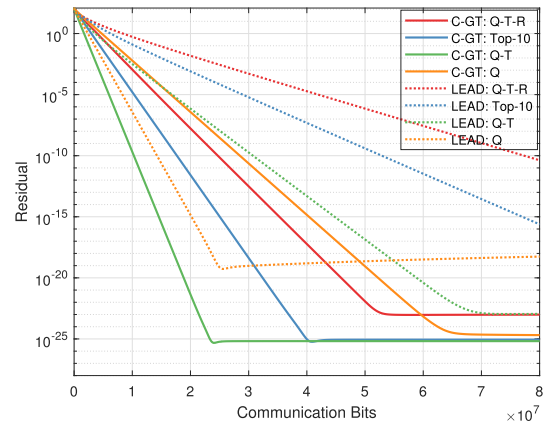


Fig. 2. Residuals  $\mathbb{E}[\|\bar{\mathbf{X}}^h - \mathbf{x}^*\|^2]$  against the communication bits for C-GT and LEAD under different compression methods.

respectively. Apparently, C-GT and LEAD outperform the other methods, while C-GT achieves the best communication efficiency.

In Fig. 2, we further present a detailed comparison between C-GT and LEAD under different types of compressors. Note that LEAD works the best under the unbiased compressor Q, while C-GT is more efficient under the biased compression operator Top-10 and the

composition of quantization and sparsification Q-T. In particular, the performance of C-GT under Q-T is the most favorite among all the combinations.

It can also be seen from Fig. 2 that using the rescaled compressors leads to slower convergence, which suggests that rescaling the compression operators to satisfy the typical contractive requirement (i.e., Assumption 4) may harm the algorithmic performance. Therefore, we can conclude that considering Assumption 5 provides users with more freedom in choosing the best compression method. These experimental findings demonstrate the effectiveness of C-GT.

## VI. CONCLUSION

In this article, we consider the problem of decentralized optimization with communication compression over a multiagent network. Specifically, we propose a compressed gradient tracking algorithm, termed C-GT, and show the algorithm converges linearly for strongly convex and smooth objective functions. C-GT not only inherits the advantages of gradient tracking-based methods, but also works with a wide class of compression operators. Simulation examples demonstrate the effectiveness and flexibility of C-GT for undirected networks. Future work will consider equipping C-GT with accelerated techniques such as Nesterov's acceleration and momentum methods. Nonconvex objective functions are also of future concern.

## APPENDIX A PROOFS FOR THE C-GT ALGORITHM

### A. Supplementary Lemmas

The following vector and matrix inequalities are often invoked.

**Lemma 5:** For  $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times p}$  and any constant  $\tau > 0$ , we have the following inequality:

$$\|\mathbf{U} + \mathbf{V}\|^2 \leq (1 + \tau)\|\mathbf{U}\|^2 + \left(1 + \frac{1}{\tau}\right)\|\mathbf{V}\|^2. \quad (21)$$

In particular, taking  $\tau = \tau' - 1, \tau' > 1$ , we have

$$\|\mathbf{U} + \mathbf{V}\|^2 \leq \tau'\|\mathbf{U}\|^2 + \frac{\tau'}{\tau' - 1}\|\mathbf{V}\|^2. \quad (22)$$

In addition, for any  $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3 \in \mathbb{R}^{n \times p}$ , we have  $\|\mathbf{U}_1 + \mathbf{U}_2 + \mathbf{U}_3\|^2 \leq \tau'\|\mathbf{U}_1\|^2 + \frac{2\tau'}{\tau' - 1}(\|\mathbf{U}_2\|^2 + \|\mathbf{U}_3\|^2)$  and  $\|\mathbf{U}_1 + \mathbf{U}_2 + \mathbf{U}_3\|^2 \leq 3\|\mathbf{U}_1\|^2 + 3\|\mathbf{U}_2\|^2 + 3\|\mathbf{U}_3\|^2$ .

**Lemma 6:** (Corollary 8.1.29 in [41]) Let  $\mathbf{M} \in \mathbb{R}^{l \times l}$  and  $\mathbf{v} \in \mathbb{R}^l$  be a nonnegative matrix and an element-wise positive vector, respectively. If  $\mathbf{M}\mathbf{v} \leq \theta\mathbf{v}$ , then  $\rho(\mathbf{M}) \leq \theta$ .

### B. Proof of Lemma 4

In light of Lemma 3, we consider the following linear system of inequalities:

$$\mathbf{A}\epsilon \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon \quad (23)$$

where  $\epsilon := [\epsilon_1, \epsilon_2, L^2\epsilon_3, \epsilon_4, L^2\epsilon_5]^\top$ , and the elements of  $\mathbf{A}$  correspond to the coefficients in Lemma 3. From Lemma 6, if there exists an element-wise positive  $\epsilon$ , then we obtain  $\rho(\mathbf{A}) \leq (1 - \frac{1}{2}M\hat{\eta}\mu)$ .

#### 1) First Inequality in (23):

$$\left(1 - \frac{3}{2}M\hat{\eta}\mu\right)\epsilon_1 + \frac{6\hat{\eta}L^2}{\mu n}\epsilon_2 + \frac{6\hat{\eta}L^2}{\mu n M}\epsilon_3 \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon_1. \quad (24)$$

Inequality (24) holds if  $\frac{6\hat{\eta}L^2}{\mu n}\epsilon_2 \leq \frac{6\hat{\eta}L^2}{\mu n M}\epsilon_3$  and  $2\frac{6\hat{\eta}L^2}{\mu n M}\epsilon_3 \leq M\hat{\eta}\mu\epsilon_1$ . That is,  $\epsilon_3 \geq M\epsilon_2$  and  $n\epsilon_1 \geq \frac{12\kappa^2}{M^2}\epsilon_3$ , where  $\kappa = L/\mu$  is the condition number.

#### 2) Second Inequality in (23):

$$\begin{aligned} & \left(\frac{1 + \tilde{\rho}^2}{2} + \frac{4L^2\hat{\eta}^2}{s\gamma}\right)\epsilon_2 + \frac{2\hat{\eta}^2}{s\gamma}L^2\epsilon_3 + \frac{2C\lambda\gamma}{s}\epsilon_4 + \frac{4nL^2\hat{\eta}^2}{s\gamma}\epsilon_1 \\ & \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon_2. \end{aligned} \quad (25)$$

Recalling  $\frac{1 - \tilde{\rho}^2}{2} = (1 + \tilde{\rho})\frac{1 - \tilde{\rho}}{2} > \frac{1 - \tilde{\rho}}{2} = \frac{\gamma s}{2}$ , relation (25) holds if

$$\frac{2(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}{s}L^2\frac{\hat{\eta}^2}{\gamma} + \frac{2C\lambda}{s}\gamma\epsilon_4 \leq \left(\frac{s\gamma}{2} - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon_2. \quad (26)$$

Dividing  $\gamma$  on both sides of (26), we get

$$\frac{M\mu\epsilon_2}{2}\frac{\hat{\eta}}{\gamma} + \frac{2(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}{s}L^2\frac{\hat{\eta}^2}{\gamma^2} + \frac{2C\lambda}{s}\epsilon_4 \leq \frac{s\epsilon_2}{2}. \quad (27)$$

It is sufficient that  $3\frac{M\mu\epsilon_2}{2}\frac{\hat{\eta}}{\gamma} \leq \frac{s\epsilon_2}{2}$ ,  $3\frac{2(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}{s}L^2\frac{\hat{\eta}^2}{\gamma^2} \leq \frac{s\epsilon_2}{2}$ , and  $3\frac{2C\lambda}{s}\epsilon_4 \leq \frac{s\epsilon_2}{2}$ . Therefore, if we have  $\hat{\eta} \leq \min\{\frac{s}{3M}\frac{\gamma}{\mu}, \sqrt{\frac{\epsilon_2}{12(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}}s\frac{\gamma}{L}\}$  and  $\epsilon_2 \geq \frac{12C\lambda}{s^2}\epsilon_4$ , then (25) can be demonstrated.

#### 3) Third Inequality in (23):

$$\begin{aligned} & 12nL^4\frac{\hat{\eta}^2}{s\gamma}\epsilon_1 + \left(12L^4\frac{\hat{\eta}^2}{s\gamma} + \frac{6\lambda L^2\gamma}{s}\right)\epsilon_2 + \left(\frac{1 + \tilde{\rho}^2}{2} + \frac{6L^2\hat{\eta}^2}{s\gamma}\right)L^2\epsilon_3 \\ & + \frac{6C\lambda L^2\gamma}{s}\epsilon_4 + \frac{2C\lambda\gamma}{s}L^2\epsilon_5 \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)L^2\epsilon_3. \end{aligned} \quad (28)$$

Dividing  $L^2$  on the both side of (28), we have

$$\begin{aligned} & 12nL^2\frac{\hat{\eta}^2}{s\gamma}\epsilon_1 + \left(12L^2\frac{\hat{\eta}^2}{s\gamma} + \frac{6\lambda\gamma}{s}\right)\epsilon_2 + \left(\frac{1 + \tilde{\rho}^2}{2} + \frac{6L^2\hat{\eta}^2}{s\gamma}\right)\epsilon_3 \\ & + \frac{6C\lambda\gamma}{s}\epsilon_4 + \frac{2C\lambda\gamma}{s}\epsilon_5 \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon_3. \end{aligned} \quad (29)$$

Based on arguments similar to those for deriving the second inequality, the third inequality holds if

$$\begin{aligned} & \frac{M\mu\epsilon_3}{2}\frac{\hat{\eta}}{\gamma} + \frac{6(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}{s}L^2\frac{\hat{\eta}^2}{\gamma^2} \\ & + \frac{2\lambda(3\epsilon_2 + C(3\epsilon_4 + \epsilon_5))}{s} \leq \frac{s\epsilon_3}{2}. \end{aligned} \quad (30)$$

It is sufficient that  $3M\mu\epsilon_3\frac{\hat{\eta}}{\gamma} \leq \frac{s\epsilon_3}{2}$ ,  $3\frac{6(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}{s}L^2\frac{\hat{\eta}^2}{\gamma^2} \leq \frac{s\epsilon_3}{2}$ , and  $3\frac{2\lambda(3\epsilon_2 + C(3\epsilon_4 + \epsilon_5))}{s} \leq \frac{s\epsilon_3}{2}$ . Thus, if there holds  $\epsilon_3 \geq \frac{12\lambda(3\epsilon_2 + C(3\epsilon_4 + \epsilon_5))}{s^2}$  and  $\hat{\eta} \leq \min\{\frac{s}{3M}\frac{\gamma}{\mu}, \sqrt{\frac{\epsilon_3}{36(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}}s\frac{\gamma}{L}\}$ , we can demonstrate (28).

#### 4) Fourth Inequality in (23):

$$\begin{aligned} & 2nt_xL^2\hat{\eta}^2\epsilon_1 + (2t_xL^2\hat{\eta}^2 + t_x\lambda\gamma^2)\epsilon_2 \\ & + t_x\hat{\eta}^2L^2\epsilon_3 + (c_x + t_xC\lambda\gamma^2)\epsilon_4 \leq \left(1 - \frac{1}{2}M\hat{\eta}\mu\right)\epsilon_4. \end{aligned} \quad (31)$$

It is equivalent to

$$\begin{aligned} & t_x(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)L^2\hat{\eta}^2 \\ & + (t_x\lambda\epsilon_2 + t_xC\lambda\epsilon_4)\gamma^2 + \frac{1}{2}M\hat{\eta}\mu\epsilon_4 \leq (1 - c_x)\epsilon_4. \end{aligned} \quad (32)$$

Inequality (32) holds if  $\hat{\eta} \leq \frac{\gamma}{L}$  and  $\gamma \leq \min\{1, \frac{1 - c_x}{m_x}\epsilon_4\}$ , where  $m_x := t_x(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + t_x\lambda(\epsilon_2 + C\epsilon_4) + \frac{M\epsilon_4}{2\kappa}$ .

### 5) Fifth Inequality in (23):

$$6nt_y L^4 \hat{\eta}^2 \epsilon_1 + (6t_y L^4 \hat{\eta}^2 + 3t_y \lambda L^2 \gamma^2) \epsilon_2 + (t_y \lambda \gamma^2 + 3t_y L^2 \hat{\eta}^2) L^2 \epsilon_3 + 3t_y C \lambda L^2 \gamma^2 \epsilon_4 + (c_y + t_y C \lambda \gamma^2) L^2 \epsilon_5 \leq \left(1 - \frac{1}{2} M \hat{\eta} \mu\right) L^2 \epsilon_5. \quad (33)$$

Dividing  $L^2$  on both sides of (33), we have

$$3t_y(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) L^2 \hat{\eta}^2 + t_y \lambda (3\epsilon_2 + \epsilon_3 + 3C\epsilon_4 + C\epsilon_5) \gamma^2 + \frac{M\mu\epsilon_5}{2} \hat{\eta} \leq (1 - c_y) \epsilon_5. \quad (34)$$

Inequality (34) holds if  $\hat{\eta} \leq \frac{\gamma}{L}$  and  $\gamma \leq \min\{1, \frac{1-c_y}{m_y} \epsilon_5\}$ , where  $m_y := 3t_y(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + t_y \lambda (3\epsilon_2 + \epsilon_3 + 3C\epsilon_4 + C\epsilon_5) + \frac{M\epsilon_5}{2\kappa}$ . In short, if the positive constants  $\epsilon_1$ - $\epsilon_5$ , consensus step-size  $\gamma$  and step-size  $\hat{\eta}$  satisfy the following conditions:

$$n\epsilon_1 \geq \frac{12\kappa^2}{M^2} \epsilon_3, \epsilon_2 \geq \frac{12C\lambda}{s^2} \epsilon_4, \epsilon_3 \geq M\epsilon_2$$

$$\epsilon_3 \geq \frac{12\lambda(3\epsilon_2 + C(3\epsilon_4 + \epsilon_5))}{s^2}, \epsilon_4 > 0, \epsilon_5 > 0 \quad (35)$$

$$\gamma \leq \min\left\{1, \frac{1-c_x}{m_x} \epsilon_4, \frac{1-c_y}{m_y} \epsilon_5\right\} \quad (36)$$

$$\hat{\eta} \leq \min\left\{\frac{s}{3M} \frac{\gamma}{\mu}, \frac{\gamma}{L}, \sqrt{\frac{\epsilon_2}{12(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}} \frac{s\gamma}{L}, \sqrt{\frac{\epsilon_3}{36(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}} \frac{s\gamma}{L}\right\}. \quad (37)$$

We can establish the linear system of inequalities in (23). For (35), it is easy to verify that there exist solutions to  $\epsilon_1$ - $\epsilon_5$ . Noticing that  $\sqrt{\frac{\epsilon_2}{12(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}} \frac{s\gamma}{L} = \sqrt{\frac{1}{12(2n\epsilon_1/\epsilon_2 + 2 + \epsilon_3/\epsilon_2)}} \frac{s\gamma}{L}$  and  $\sqrt{\frac{\epsilon_3}{36(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}} \frac{s\gamma}{L} = \sqrt{\frac{1}{36(2n\epsilon_1/\epsilon_3 + 2\epsilon_2/\epsilon_3 + 1)}} \frac{s\gamma}{L}$  are both less than  $\frac{s}{3M} \frac{\gamma}{\mu}$  and  $\frac{\gamma}{L}$ , we obtain the upper bound on the maximum step-size.

### C. Proof of Theorem 1

Taking  $\tau_x = \tau_y = \frac{1}{1-0.5\delta}$  and noticing that  $\alpha_x = \alpha_y = 1/r$ , we have  $t_x = t_y = \frac{6}{\delta}$  and  $c_x = c_y = \frac{1-\delta}{1-0.5\delta}$ . Recalling the relations of  $\epsilon_1$ - $\epsilon_5$  in Lemma 4, we can take  $\epsilon_4 = \epsilon_5 = 1$ ,  $\epsilon_2 = \frac{12C\lambda}{s^2}$ ,  $\epsilon_3 = \frac{432C\lambda^2}{s^4} + \frac{48C\lambda}{s^2} \geq 4\epsilon_2$  and  $n\epsilon_1 = \frac{12\kappa^2}{M^2} \left(\frac{432C\lambda^2}{s^4} + \frac{48C\lambda}{s^2}\right)$ . Then, we know  $\frac{\epsilon_2}{12(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)} \leq \frac{\epsilon_3}{36(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3)}$ . Meanwhile, noticing that  $m_y := 3\frac{6}{\delta}[(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + \lambda(\epsilon_2 + C)] + \frac{M}{2\kappa} + \frac{6}{\delta}\lambda(\epsilon_3 + C)$  and  $m_x = \frac{6}{\delta}[(2n\epsilon_1 + 2\epsilon_2 + \epsilon_3) + \lambda(\epsilon_2 + C)] + \frac{M}{2\kappa}$ , we get  $m_y \geq m_x$ . For simplicity, denote  $m = m_y$  where the specific values for the constants  $\epsilon_1$ - $\epsilon_5$  are given above. Then from Lemma 4, we obtain the upper bounds on the consensus step-size and the maximum step-size, which completes the proof.

### ACKNOWLEDGMENT

The authors would like to thank B. Wang from The Chinese University of Hong Kong, Shenzhen for providing helpful feedback.

### REFERENCES

- [1] K. Cohen, A. Nedić, and R. Srikant, "Distributed learning algorithms for spectrum sharing in spatial random access wireless networks," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2854–2869, Jun. 2017.
- [2] A. Nedić and J. Liu, "Distributed optimization for control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, no. 1, pp. 77–103, 2018.
- [3] A. Nedić, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 92–101, May 2020.
- [4] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
- [5] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
- [6] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant step-sizes," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 2055–2060.
- [7] P. D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.
- [8] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [9] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.
- [10] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," in *Proc. Amer. Control Conf.*, 2017, pp. 3950–3955.
- [11] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Program.*, vol. 187, pp. 409–457, 2020.
- [12] 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. Statist.*, 2020, pp. 1662–1672.
- [13] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.
- [14] P. Xie, K. You, R. Tempo, S. Song, and C. Wu, "Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4331–4337, Dec. 2018.
- [15] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM J. Optim.*, vol. 32, no. 2, pp. 354–385, 2022.
- [16] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. 51st IEEE Conf. Decis. Control*, 2012, pp. 5453–5458.
- [17] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 315–320, Jul. 2018.
- [18] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Jan. 2021.
- [19] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proc. IEEE*, vol. 108, no. 11, pp. 1869–1889, Nov. 2020.
- [20] S. Pu, "A robust gradient tracking method for distributed optimization over directed networks," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 2335–2341.
- [21] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and application to data-parallel distributed training of speech DNNs," in *Proc. Interspeech*, 2014, pp. 1058–1062.
- [22] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1709–1720.
- [23] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SIGNSGD: Compressed optimisation for non-convex problems," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 559–568.
- [24] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4452–4463.
- [25] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signSGD and other gradient compression schemes," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3252–3261.
- [26] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, "Distributed learning with compressed gradient differences," 2019, *arXiv:1901.09269*.
- [27] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6155–6165.

- [28] S. U. Stich, "On communication compression for distributed optimization on heterogeneous data," 2020, *arXiv:2009.02388*.
- [29] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan, "On biased compression for distributed learning," 2020, *arXiv:2002.12410*.
- [30] H. Xu *et al.*, "GRACE: A compressed communication framework for distributed machine learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 561–572.
- [31] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7652–7662.
- [32] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3479–3487.
- [33] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *Proc. Int. Conf. Learn. Representations*, 2020.
- [34] X. Liu, Y. Li, R. Wang, J. Tang, and M. Yan, "Linear convergent decentralized optimization with compression," in *Proc. Int. Conf. Learn. Representations*, 2020.
- [35] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4494–4506, Sep. 2019.
- [36] D. Kovalev, A. Koloskova, M. Jaggi, P. Richtárik, and S. Stich, "A linearly convergent algorithm for decentralized optimization: Sending less bits for free!," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 4087–4095.
- [37] N. Michelusi, G. Scutari, and C.-S. Lee, "Finite-bit quantization for distributed algorithms with linear convergence," in *IEEE Trans. Inf. Theory*, 2021, *arXiv:2107.11304*.
- [38] Y. Kajiyama, N. Hayashi, and S. Takai, "Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1254–1261, Mar. 2021.
- [39] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 217–226, May 2020.
- [40] Y. Liao, Z. Li, K. Huang, and S. Pu, "A compressed gradient tracking method for decentralized optimization with linear convergence," 2022, *arXiv:2205.12623*.
- [41] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2012.