DIFFERENTIAL ERROR FEEDBACK FOR COMMUNICATION-EFFICIENT DECENTRALIZED OPTIMIZATION

Roula Nassif [1], *Stefan Vlaski* [2], *Marco Carpentiero* [3], *Vincenzo Matta* [3], *Ali H. Sayed* [4]

[1] Université Côte d'Azur, I3S Laboratory, CNRS, France

[2] Imperial College London, UK

[3] University of Salerno, Italy

[4] Ecole Polytechnique Fédérale de Lausanne, Switzerland

ABSTRACT

Communication-constrained algorithms for decentralized learning and optimization rely on the exchange of quantized signals coupled with local updates. In this context, *differential quantization* is an effective technique to mitigate the negative impact of quantization by leveraging correlations between subsequent iterates. In addition, the use of *error feedback*, which consists of incorporating the quantization error into subsequent steps, is a powerful mechanism to compensate for the bias caused by the quantization. Under error feedback, performance guarantees in the literature have so far focused on algorithms employing a fusion center or a special class of contractive quantizers that cannot be implemented with a finite number of bits. In this work, we propose and study a new *decentralized* communication-efficient learning approach that blends differential quantization with error feedback. The results show that, under some general conditions on the quantization noise, and for sufficiently small step-sizes μ, it is possible to keep the estimation errors small (on the order of μ) in steady state. The results also suggest that, in the *small step-size regime*, it is possible to attain the performance achievable in the absence of compression.

***Index Terms*—Error feedback, differential quantization, communication-efficient learning, decentralized learning and adaptation.**

I. INTRODUCTION

Data is increasingly being collected in a distributed and streaming manner, in an environment where communication efficiency and data privacy are becoming major concerns. In this context, centralized learning schemes with fusion centers tend to be replaced by new paradigms, such as federated and decentralized learning [1]–[6]. In these approaches, each participating device (which is referred to as *agent* or *node*) has a local training dataset, which is never uploaded to the server. The training data is kept locally on the users' devices, and the devices act as agents performing local computations to learn global models of interest. In applications where communication with a server becomes a bottleneck, *decentralized* topologies (where agents only communicate with their neighbors) become attractive alternatives to federated topologies (where a server connects with all remote devices). These decentralized implementations reduce the communication burden since model updates are exchanged locally between agents without relying on a central coordinator [4]–[8].

In traditional decentralized implementations, agents need to exchange (possibly *high-dimentional* and *dense*) parameter vectors at every iteration of the learning algorithm, leading to high communication costs. In practice, if not addressed adequately, the scarcity of the communication resources may limit the application of decentralized learning. A variety of methods have been proposed to reduce the communication overhead of decentralized learning. These methods can be divided into two main categories. In the first one, communication is reduced by skipping communication rounds while performing a certain number of local updates in between [2], [9]. In the second one, information is compressed by employing either quantization (e.g., employing dithered quantization [10]), sparsification (e.g., employing rand-k sparsifiers [7]), or both (e.g., employing top-c combined with dithering [11]), before being exchanged. Compression operators and learning algorithms are then jointly designed to prevent the compression error from accumulating during the learning process and from significantly deteriorating the performance of the decentralized approach [7], [8], [12]–[16]. Other works propose to combine the aforementioned two categories to further reduce the communication overhead [17].

In this work, we introduce a new communication-efficient approach for decentralized learning. The approach exploits *differential quantization* and *error feedback* to mitigate the negative impact of compressed communications on the learning performance. Differential quantization is a common technique for mitigating the impact of compression by leveraging correlations between subsequent iterates. In this case, instead of communicating compressed versions of the iterates, agents communicate compressed versions of the differences between current estimates and their predictions based on previous iterations. Several recent works have focused on studying the benefits of differential quantization in the context of decentralized learning [8], [13]–[15], [17]. In the same token, error feedback consists of locally storing the compression error (i.e., the difference between the input and output of the compression operator), and incorporating it back into the next iteration. This technique has been previously employed for stochastic gradient descent (SGD) algorithms. Specifically, it has been applied to the signSGD algorithm in the single-agent context and under 1-bit quantization [18], and to the distributed SGD to handle biased compression operators [11]. In the context of decentralized learning, the DeepSqueeze approach in [19] uses error feedback without differential quantization.

In the current paper, we show how to blend differential quantization and error feedback in order to obtain a communication-efficient decentralized learning algorithm. First, we describe in Sec. II the decentralized learning framework and the class of compression operators considered in the study. Then, we present and analyze the proposed learning strategy in Secs. III and IV, respectively. While there exist several theoretical works investigating communication-efficient decentralized learning, these works (with some exceptions [14], [15]) assume that some quantities (such as the norm or some components of the vector to be quantized) are represented perfectly or with machine precision. The analysis in the current work removes this limitation. We also remove the assumption of *unbiased* quantizers adopted in [15]. Moreover, unlike the studies in [8], [9], [12], [14], [16], [17], [19], we do not require the combination matrices (which, as explained later,

The work of R. Nassif was supported by ANR JCJC grant ANR-22-CE23-0015-01 (CEDRO project). The work of S. Vlaski was supported in part by EPSRC Grants EP/X04047X/1 and EP/Y037243/1.

control the exchange of information between neighboring agents) to be symmetric. Finally, we do not assume bounded gradients as in [8], [17]. We establish in Sec. IV the mean-square-error stability of the proposed decentralized communication-efficient approach. The analysis shows that, under some general conditions on the quantization noise, and for sufficiently small step-sizes μ, it is possible to keep the estimation errors small (on the order of μ) in steady state. Our theoretical and experimental results reveal that, in the *small step-size regime*, the proposed strategy attains the performance achievable in the absence of compression.

II. PROBLEM SETUP

In this section, we formally state the decentralized optimization problem and introduce important quantities and assumptions that will be used in subsequent sections.

Decentralized optimization: We consider single-task or consensus-based optimization problems of the form:

$$w^o = \operatorname*{argmin}_{w\in\mathbb{R}^M} J^{\text{glob}}(w), \quad \text{where } J^{\text{glob}}(w) \triangleq \frac{1}{K}\sum_{k=1}^{K} J_k(w) \quad (1)$$

where K is the number of agents in the network, $w \in \mathbb{R}^M$ is the parameter of interest, and $J_k(w) : \mathbb{R}^M \to \mathbb{R}$ is a differentiable convex cost associated with agent k. It is expressed as the expectation of some loss function $L_k(\cdot)$ and written as $J_k(w) = \mathbb{E}L_k(w; \boldsymbol{y}_k)$, where $\boldsymbol{y}_k$ denotes the random data (throughout the paper, random quantities are denoted in boldface). The expectation is computed over the data distribution. In the stochastic setting, when the data distribution is unknown, the risks $J_k(\cdot)$ and their gradients $\nabla_w J_k(\cdot)$ are unknown. In this case, instead of using the true gradient, it is common to use approximate gradient vectors of the form $\widehat{\nabla_w J_k}(w) = \nabla_w L_k(w; \boldsymbol{y}_{k,i})$ where $\boldsymbol{y}_{k,i}$ represents the data observed at iteration i [5], [20].

In order to solve problem (1), agents are only allowed to perform local computations, and to exchange information with their neighbors over a prescribed network topology specified by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertices (agents) and the set of possible communication links or edges, respectively. Let $\mathcal{N}_k$ denote the set of nodes connected to agent k by a communication link (including node k itself). For optimization problems of the form (1), information sharing across agents is in general implemented by means of a $K \times K$ combination matrix $A = [a_{k\ell}]$ that has a zero element (k, ℓ) if nodes k and ℓ are not neighbors, i.e., $a_{k\ell} = 0$ if $\ell \notin \mathcal{N}_k$, and satisfies the following conditions [21]:

$$A\mathbb{1}_K = \mathbb{1}_K, \ \mathbb{1}_K^\top A = \mathbb{1}_K^\top, \ \rho\left(A - \frac{1}{K}\mathbb{1}_K\mathbb{1}_K^\top\right) < 1, \quad (2)$$

where $\mathbb{1}_K$ represents the $K \times 1$ vector of all 1's, and $\rho(\cdot)$ denotes the spectral radius of its matrix argument.

Compression operators: Compression is performed through the application of a mapping $\mathcal{C} : \mathbb{R}^M \to \mathbb{R}^M$, where $\mathcal{C}(x)$ represents a compressed version (e.g., a finite-bit representation or a sparsified version) of the original message x. In this study, we assume that each agent k employs randomized compression operators satisfying the following general property.

Property 1. ***(Bounded variance).*** *The randomized compression operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ at agent k satisfies the following condition:*

$$\mathbb{E}\|x - \boldsymbol{\mathcal{C}}_k(x)\|^2 \le \beta_{q,k}^2\|x\|^2 + \sigma_{q,k}^2, \quad (3)$$

for some $0 \le \beta_{q,k}^2 < 1$ and $\sigma_{q,k}^2 \ge 0$, and where the expectation is evaluated w.r.t. the randomness of $\boldsymbol{\mathcal{C}}_k(\cdot)$. Note that the considered class of randomized compression operators includes deterministic operators as a particular case for which we now have:

$$\|x - \mathcal{C}_k(x)\|^2 \le \beta_{q,k}^2\|x\|^2 + \sigma_{q,k}^2. \quad (4)$$

□

Property 1 is satisfied by many compression operators of interest in decentralized learning such as rand-c, top-c, gradient sparsifier, QSGD, probabilistic ANQ, and probabilistic uniform quantizer – see Table 3 in [11] and Table 1 in [15] for details. While many existing works focus on studying decentralized learning in the presence of compression operators that satisfy the variance bound conditions in Property 1 with a zero *absolute noise* term $\sigma_{q,k}^2 = 0$ [7], [8], [11]–[13], [16], [17], [19], the analysis in the current work is general and does not require $\sigma_{q,k}^2$ to be 0. As explained in [14], neglecting the effect of $\sigma_{q,k}^2$ requires that some quantities (e.g., the norm of the vector to be quantized) be represented with no quantization error, or at machine precision in practice. The advantage of employing compression operators with non-zero $\sigma_{q,k}^2$ will be further illustrated in the simulation section V.

III. DECENTRALIZED ALGORITHMIC FRAMEWORK: COMPRESSED COMMUNICATIONS

In this work, we propose the DEF-ATC (differential error feedback - adapt then combine) diffusion strategy listed in Algorithm 1 for solving problem (1) in a decentralized and communication-efficient manner. At each iteration i, each agent k in the network performs three steps. In the first step, which corresponds to the *adaptation* (or *self-learning*) step, agent k updates its estimate $\boldsymbol{w}_{k,i-1}$ to an intermediate estimate $\boldsymbol{\psi}_{k,i}$ using its approximation for its own gradient ($\mu > 0$ is a small step-size parameter). Note that replacing the step-size μ by $\frac{\mu}{\zeta}$ is necessary to compensate for the impact of the damping coefficient $\zeta \in (0, 1]$ on the algorithm's learning rate. The coefficient is used in the compression step (7) to control the network stability in scenarios where compression leads to network instability. The second step is the *compression* step. To update $\{\boldsymbol{\phi}_{\ell,i}\}_{\ell\in\mathcal{N}_k}$, each agent k first computes a compressed message $\boldsymbol{\delta}_{k,i}$ that encodes the error compensated difference $\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}$ (using a compression operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ satisfying Property 1), and broadcasts it to its neighbors. Then, agent k updates the compression error vector $\boldsymbol{z}_{k,i}$ according to (6) and performs the reconstruction on each received vector $\boldsymbol{\delta}_{\ell,i}$ according to (7). Observe that implementing the compression step in Algorithm 1 requires storing the previous compression error $\boldsymbol{z}_{k,i-1}$ and the previous estimates $\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell\in\mathcal{N}_k}$ by agent k. The compression step is followed by the *combination* step (8) where agent k combines the reconstructed vectors $\{\boldsymbol{\phi}_{\ell,i}\}$ using the combination coefficients $\{a_{k\ell}\}$ and a *mixing* parameter $\gamma \in (0, 1]$. The resulting vector $\boldsymbol{w}_{k,i}$ is the estimate of w^o in (1) at agent k and iteration i. As for ζ, the parameter γ in (8) can also be used to control the network stability.

To proceed, and for the sake of convenience, we rewrite Algorithm 1 in the following compact form:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \dfrac{\mu}{\zeta}\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1}) & (9a) \\ \boldsymbol{\phi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \zeta\boldsymbol{\mathcal{C}}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) & (9b) \\ \boldsymbol{w}_{k,i} = (1-\gamma)\boldsymbol{\phi}_{k,i} + \gamma\displaystyle\sum_{\ell\in\mathcal{N}_k} a_{k\ell}\boldsymbol{\phi}_{\ell,i} & (9c) \end{cases}$$

where the compression error $\boldsymbol{z}_{k,i}$ is updated according to:

$$\boldsymbol{z}_{k,i} = (\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) - \boldsymbol{\mathcal{C}}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}). \quad (10)$$

Observe that, in the absence of compression (i.e., when the operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ in (9b) and (10) is replaced by the identity operator and the parameters ζ and γ are set to 1), we obtain the diffusion ATC-type approach for solving the optimization problem (1) [4], [5]. Therefore, Algorithm 1 can be seen as a communication-efficient variant of the Adapt-Then-Combine (ATC) approach. To mitigate the negative impact of compression, the DEF-ATC approach uses *differential quantization* and *error-feedback* in step (9b).

Algorithm 1: DEF-ATC (differential error feedback - adapt then combine)

Input: initializations $\boldsymbol{w}_{k,0}=0$, $\boldsymbol{\phi}_{k,0}=0$, and $\boldsymbol{z}_{k,0}=0$, step-size μ, mixing parameter γ, combination matrix A.

for $i=1,2,\ldots,$ *on the k-th node* **do**

Adapt: update $\boldsymbol{w}_{k,i-1}$ according to:

$$\boldsymbol{\psi}_{k,i}=\boldsymbol{w}_{k,i-1}-\frac{\mu}{\zeta}\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1}) \tag{5}$$

Compress and broadcast:

- generate $\boldsymbol{\delta}_{k,i}=\boldsymbol{\mathcal{C}}_k(\boldsymbol{\psi}_{k,i}-\boldsymbol{\phi}_{k,i-1}+\boldsymbol{z}_{k,i-1})$ and broadcast it to neighbors $\mathcal{N}_k$
- update the compression error:

$$\boldsymbol{z}_{k,i}=(\boldsymbol{\psi}_{k,i}-\boldsymbol{\phi}_{k,i-1}+\boldsymbol{z}_{k,i-1})-\boldsymbol{\delta}_{k,i} \tag{6}$$

- upon receiving the compressed vectors $\boldsymbol{\delta}_{\ell,i}$ from neighbors $\ell\in\mathcal{N}_k$, reconstruct according to:

$$\boldsymbol{\phi}_{\ell,i}=\boldsymbol{\phi}_{\ell,i-1}+\zeta\boldsymbol{\delta}_{\ell,i},\qquad \ell\in\mathcal{N}_k \tag{7}$$

Combine: Update local model according to:

$$\boldsymbol{w}_{k,i}=(1-\gamma)\boldsymbol{\phi}_{k,i}+\gamma\sum_{\ell\in\mathcal{N}_k}a_{k\ell}\boldsymbol{\phi}_{\ell,i} \tag{8}$$

IV. MEAN-SQUARE-ERROR ANALYSIS

We analyze strategy (9) with a combination matrix A satisfying (2) and compression operators $\{\boldsymbol{\mathcal{C}}_k(\cdot)\}$ satisfying Property 1 by examining the average squared distance between w^o and $\boldsymbol{w}_{k,i}$, namely, $\mathbb{E}\|w^o-\boldsymbol{w}_{k,i}\|^2$, under the following assumptions on the risks $\{J_k(\cdot)\}$ and on the gradient noise processes $\{\boldsymbol{s}_{k,i}(\cdot)\}$ defined as [5]:

$$\boldsymbol{s}_{k,i}(w)\triangleq\nabla_w J_k(w)-\widehat{\nabla_w J_k}(w). \tag{11}$$

Assumption 1. *The individual costs $J_k(w)$ are assumed to be twice differentiable and convex with at least one of them being strongly convex. It follows that $J^{glob}(w)$ is twice-differentiable and strongly convex. It is further assumed to satisfy:*

$$0<\nu I_M\leq\nabla_w^2 J^{glob}(w)\leq\delta I_M, \tag{12}$$

for some positive parameters $\nu\leq\delta$. For two matrices X and Y, the notation $X\geq Y$ means that $X-Y$ is positive semi-definite.

Assumption 2. *The gradient noise process defined in* (11) *satisfies for $k=1,\ldots,K$:*

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^K]=0, \tag{13}$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^K]\leq\beta_{s,k}^2\|\widetilde{\boldsymbol{w}}_{k,i-1}\|^2+\sigma_{s,k}^2, \tag{14}$$

for some $\beta_{s,k}^2\geq 0$ and $\sigma_{s,k}^2\geq 0$. □

In the following, we derive a useful recursion that allows to examine the time-evolution across the network of the error dynamics relative to the reference vector w^o. Then, we present the mean-square-error stability theorem and highlight the main conclusions.

Let $\widetilde{\boldsymbol{w}}_{k,i}=w^o-\boldsymbol{w}_{k,i}$, $\widetilde{\boldsymbol{\psi}}_{k,i}=w^o-\boldsymbol{\psi}_{k,i}$, and $\widetilde{\boldsymbol{\phi}}_{k,i}=w^o-\boldsymbol{\phi}_{k,i}$. Using similar arguments as in [5], we can show that the vector $\widetilde{\boldsymbol{\psi}}_i=\text{col}\{\widetilde{\boldsymbol{\psi}}_{k,i}\}_{k=1}^K$ evolves according to:

$$\widetilde{\boldsymbol{\psi}}_i=\left(I_{MK}-\frac{\mu}{\zeta}\boldsymbol{\mathcal{H}}_{i-1}\right)\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}-\frac{\mu}{\zeta}\boldsymbol{s}_i+\frac{\mu}{\zeta}b, \tag{15}$$

where $\widetilde{\boldsymbol{\mathcal{W}}}_i=\text{col}\{\widetilde{\boldsymbol{w}}_{k,i}\}_{k=1}^K$ and:

$$b\triangleq\text{col}\{\nabla_w J_k(w^o)\}_{k=1}^K, \tag{16}$$

$$\boldsymbol{s}_i\triangleq\text{col}\{\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\}_{k=1}^K, \tag{17}$$

$$\boldsymbol{\mathcal{H}}_{i-1}\triangleq\text{diag}\{\boldsymbol{H}_{k,i-1}\}_{k=1}^K, \tag{18}$$

with $\boldsymbol{H}_{k,i-1}\triangleq\int_0^1\nabla_w^2 J_k(w^o-t\widetilde{\boldsymbol{w}}_{k,i-1})dt$. By subtracting w^o from both sides of (9c), replacing w^o by $(1-\gamma)w^o+\gamma w^o$, and using $w^o=\sum_{\ell\in\mathcal{N}_k}a_{k\ell}w^o$, we obtain:

$$\widetilde{\boldsymbol{w}}_{k,i}=(1-\gamma)\widetilde{\boldsymbol{\phi}}_{k,i}+\gamma\sum_{\ell\in\mathcal{N}_k}a_{k\ell}\widetilde{\boldsymbol{\phi}}_{\ell,i}, \tag{19}$$

from which we conclude that $\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}$ in (15) evolves according to:

$$\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}=\mathcal{A}'\widetilde{\boldsymbol{\phi}}_{i-1}, \tag{20}$$

where $\widetilde{\boldsymbol{\phi}}_i=\text{col}\{\widetilde{\boldsymbol{\phi}}_{k,i}\}_{k=1}^K$ and

$$\mathcal{A}'=\big((1-\gamma)I_K+\gamma A\big)\otimes I_M, \tag{21}$$

where the symbol $\otimes$ denotes the Kronecker product. Now, by subtracting w^o from both sides of (9b), and by adding and subtracting w^o to the difference $\boldsymbol{\psi}_{k,i}-\boldsymbol{\phi}_{k,i-1}$, we can write:

$$\begin{aligned}\widetilde{\boldsymbol{\phi}}_{k,i}&=\widetilde{\boldsymbol{\phi}}_{k,i-1}-\zeta\boldsymbol{\mathcal{C}}_k(\widetilde{\boldsymbol{\phi}}_{k,i-1}-\widetilde{\boldsymbol{\psi}}_{k,i}+\boldsymbol{z}_{k,i-1})\\&\overset{(10)}{=}(1-\zeta)\widetilde{\boldsymbol{\phi}}_{k,i-1}+\zeta\widetilde{\boldsymbol{\psi}}_{k,i}+\zeta(\boldsymbol{z}_{k,i}-\boldsymbol{z}_{k,i-1}).\end{aligned} \tag{22}$$

Combining (15), (20), and (22), we obtain the following recursion that describes the evolution of the error dynamics relative to w^o:

$$\widetilde{\boldsymbol{\phi}}_i=\boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{\phi}}_{i-1}-\mu\boldsymbol{s}_i+\mu b-\boldsymbol{z}_{i-1}+\boldsymbol{z}_i, \tag{23}$$

where

$$\boldsymbol{\mathcal{B}}_{i-1}\triangleq(1-\zeta)I_{MK}+\zeta\left(I_{MK}-\frac{\mu}{\zeta}\boldsymbol{\mathcal{H}}_{i-1}\right)\mathcal{A}', \tag{24}$$

$$\boldsymbol{z}_i\triangleq\zeta\text{col}\{\boldsymbol{z}_{k,i}\}_{k=1}^K. \tag{25}$$

The mean-square-error analysis exploits the eigenstructure of the matrix $\mathcal{A}'$ in (21). It can be shown that the matrix A satisfying (2) has a Jordan decomposition of the form $A=V_\epsilon J V_\epsilon^{-1}$, where [5]:

$$V_\epsilon=[\alpha\mathbb{1}_K|V_R],\ J=\left[\begin{array}{c|c}1&0\\\hline 0&J_\epsilon\end{array}\right],\ V_\epsilon^{-1}=\left[\begin{array}{c}\alpha\mathbb{1}_K^\top\\\hline V_L^\dagger\end{array}\right], \tag{26}$$

with $\alpha=1/\sqrt{K}$ and J_ϵ a Jordan matrix with eigenvalues (which may be complex and have magnitude less than one) on the diagonal and $\epsilon>0$ on the super-diagonal [5]. The parameter ϵ is chosen small enough to ensure $\rho(J_\epsilon)+\epsilon\in(0,1)$. Consequently, the matrix $\mathcal{A}'$ in (21) has a Jordan decomposition of the form $\mathcal{A}'=\mathcal{V}_\epsilon\mathcal{J}'\mathcal{V}_\epsilon^{-1}$ [15] where $\mathcal{V}_\epsilon=V_\epsilon\otimes I_M$, $\mathcal{V}_\epsilon^{-1}=V_\epsilon^{-1}\otimes I_M$, and:

$$\mathcal{J}'=\left[\begin{array}{c|c}I_M&0\\\hline 0&\mathcal{J}_\epsilon'\end{array}\right],\quad\text{with }\mathcal{J}_\epsilon'=[(1-\gamma)I_{K-1}+\gamma J_\epsilon]\otimes I_M. \tag{27}$$

Theorem 1. (Network mean-square-error stability). *Consider a network of K agents running the quantized decentralized Algorithm 1 under Assumptions 1 and 2 with a matrix A satisfying* (2) *and compression operators $\{\boldsymbol{\mathcal{C}}_k(\cdot)\}$ satisfying Property 1. In the absence of the relative quantization noise term (i.e., $\beta_{q,k}^2=0$, $\forall k$), let $\gamma=\zeta=1$. In the presence of the relative quantization noise, let $\zeta\in(0,1]$ and $\gamma\in(0,1]$ be such that the two following conditions are satisfied:*

$$\|\mathcal{J}_\epsilon''\|+4v_1^2v_2^2\beta_{q,\max}^2\zeta^2\|I-\mathcal{J}_\epsilon'\|^2<1, \tag{28}$$

$$\frac{2\zeta^2\|I-\mathcal{J}_\epsilon'\|^2}{1-\|\mathcal{J}_\epsilon''\|}+2\beta_{q,\max}^2\zeta^2v_1^2v_2^2\left((1+\mu\sigma_{11})^2+\|2I-\mathcal{J}_\epsilon'\|^2\right)<1, \tag{29}$$

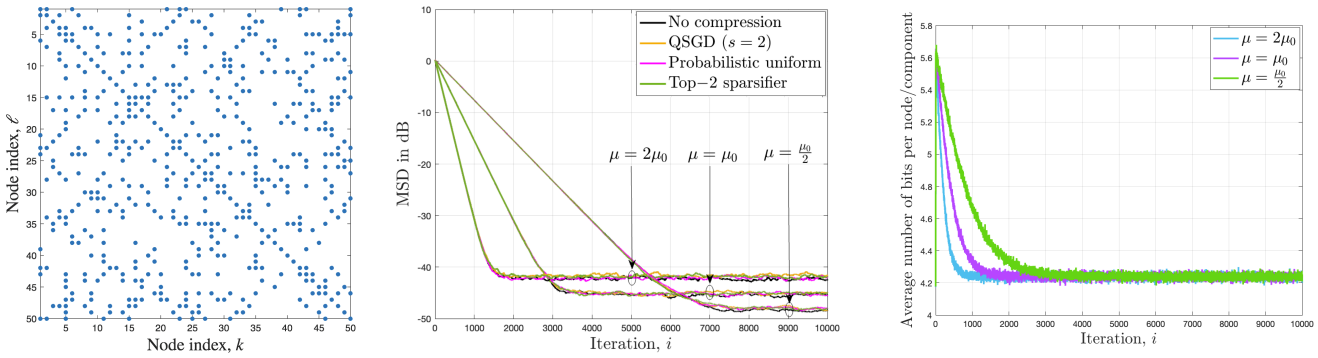

Fig. 1. *(Left)* Communication link matrix. *(Middle)* Performance of DEF-ATC for three different values of μ ($\mu_0 = 0.0015$) when QSGD, variable-rate probabilistic uniform ($\sigma^2_{q,k} = \mu^2$), and top-2 compression operators are used. Black curves correspond to the standard diffusion ATC approach. *(Right)* Evolution of the average number of bits per node, per component, when the variable-rate probabilistic uniform quantizer is used.

where $\mathcal{J}''_\epsilon = [(1-\gamma\zeta)I_{K-1} + \gamma\zeta J_\epsilon] \otimes I_M$, $v_1 \triangleq \|\mathcal{V}_\epsilon^{-1}\|$, $v_2 \triangleq \|\mathcal{V}_\epsilon\|$, $\beta^2_{q,\max} \triangleq \max_{1\le k\le K}\{\beta^2_{q,k}\}$, σ_{11} *is some positive constant that depends on* ν, *and* $\|\cdot\|$ *represents the 2-induced matrix norm. Then, the network is mean-square-error stable for sufficiently small step-size* μ, *namely, it holds that:*

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = \overline{\sigma}_s^2 \cdot O(\mu) + \kappa \cdot O(\mu^2) + \overline{\sigma}_q^2 \cdot O(1), \quad (30)$$

where $\overline{\sigma}_s^2 \triangleq \sum_{k=1}^K \sigma^2_{s,k}$, κ *is a constant depending on the term* $b = O(1)$ *in* (16)*, the absolute gradient noise term* $\overline{\sigma}_s^2$, *and the relevant quantization noise term* $\beta^2_{q,\max}$, *and* $\overline{\sigma}_q^2 \triangleq \sum_{k=1}^K \sigma^2_{q,k}$.

Proof. Due to space limitations, the proof is omitted. □

While expression (30) reveals the influence of the *step-size* μ, the *quantization noise* (captured by $\{\overline{\sigma}_q^2, \beta^2_{q,\max}\}$), and the *gradient noise* (captured by $\overline{\sigma}_s^2$) on the steady state mean-square-error, expressions (28) and (29) reveal the influence of the *relative quantization noise* term (captured by $\beta^2_{q,\max}$) on the network stability, and how this influence can be mitigated by properly choosing the mixing parameter γ. One main conclusion stemming from Theorem 1 is that the mean-square-error contains: i) the gradient noise term $\overline{\sigma}_s^2 \cdot O(\mu)$, which is classically encountered in the uncompressed case; ii) an $O(\mu^2)$ term that has a negligible effect in the small step-size regime (i.e., when $\mu \to 0$); and iii) an $O(1)$ term that depends on the quantizers' absolute noise components $\{\sigma^2_{q,k}\}$. *Interestingly, by choosing compression schemes with* $\sigma^2_{q,k} \propto \mu^2$, *we obtain the classical result* $\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = \overline{\sigma}_s^2 \cdot O(\mu)$ *observed in the uncompressed case studied in [5].*

This setup, however, requires a careful inspection since small values of $\sigma^2_{q,k}$ imply small quantization errors, which might in principle require large bit rates. Consequently, in the small step-size regime ($\mu \to 0$), the bit rate might increase indefinitely when $\sigma^2_{q,k} \propto \mu^2$. Remarkably, we will illustrate in the simulation section that this is not the case, implying that, for sufficiently small step-sizes, the same performance as the uncompressed case can be attained with a finite number of bits.

V. SIMULATION RESULTS

We apply the DEF-ATC strategy listed in Algorithm 1 to a network of $K = 50$ nodes with the communication link matrix shown in Fig. 1 (left), where the (k,ℓ)-th entry is equal to 1 if there is a link between k and ℓ and is 0 otherwise. Each agent is subjected to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ assumed to satisfy a linear regression model of the form $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}^\top w_k^o + \boldsymbol{v}_k(i)$ for some $M \times 1$ vector w_k^o with $\boldsymbol{v}_k(i)$ denoting a zero-mean measurement noise and $M = 5$. A mean-square-error cost of the form $J_k(w) = \frac{1}{2}\mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}^\top w|^2$ is associated with each agent k. The processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ and the model parameters $\{w_k^o\}$ are generated using similar settings as in [15, Sec. VI]. The matrix A satisfying the conditions in (2) is found by following the same approach as in [22]. We implement two *unbiased* compression schemes: i) the QSGD scheme [3], which transmits the norm with high precision and randomly rounds the components to s-bit representations [3] (in this case, $\mathbb{E}[\boldsymbol{\mathcal{C}}_k(x)] = x$, $\beta^2_{q,k} = \min(\frac{M}{s^2}, \frac{\sqrt{M}}{s})$, $\sigma^2_{q,k} = 0$ [15]); and ii) the variable-rate probabilistic uniform quantizer [14], which incorporates dithering into the uniform quantization scheme (in this case, $\mathbb{E}[\boldsymbol{\mathcal{C}}_k(x)] = x$, $\beta^2_{q,k} = 0$, $\sigma^2_{q,k} = \frac{M\Delta^2}{4}$, where Δ is the quantization step). We further implement the top-c compression scheme, which is a *biased deterministic* sparsification rule that consists of selecting the c largest magnitude components and setting the other ones to zero (in this case, $\beta^2_{q,k} = \frac{M-c}{M}$ and $\sigma^2_{q,k} = 0$ [7]). For the variable-rate probabilistic uniform quantizer, we set the damping coefficient $\zeta = 1$, the mixing parameter $\gamma = 1$ and the quantization step Δ such that $\sigma^2_{q,k} = \mu^2$. For the QSGD and top-c compression operators, we set $s = 2$ and $c = 2$, respectively, $\zeta = 1$ and $\gamma = 0.7$. We report the network MSD learning curves $\frac{1}{K}\sum_{k=1}^K \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2$ in Fig. 1 (middle) for three different values of the step-size. Results are averaged over 100 Monte-Carlo runs. As it can be observed, despite compression, the DEF-ATC approach achieves a performance that is almost identical to the uncompressed ATC approach (which can be obtained from Algorithm 1 by setting $\gamma = \zeta = 1$ and replacing the compression operator by identity). In Fig. 1 (right), we report the average number of bits per node, per component[1] as a function of the iteration i, when the variable-rate probabilistic uniform quantizer is employed. As it can be observed, for the three different values of the step-size, we approximately obtain the same finite average number of bits in steady state (approximately 4.2 bits/component/iteration are required on average in steady state). For the QSGD scheme, the bit-budget required to encode an $M \times 1$ vector is given by $B_{\text{HP}} + M + M\lceil\log_2(s)\rceil$ where B_{HP} denotes the number of bits required to encode a scalar with high precision [15, Table 1]. On the other hand, the top-c scheme requires $cB_{\text{HP}} + c\lceil\log_2(M)\rceil$ bits to encode an $M \times 1$ vector. If we use $B_{\text{HP}} \approx 32$ bits, we find that the QSGD scheme with $s = 2$ requires $42/5 = 8.4$ bits/node/component/iteration, which is almost two times higher that the one obtained when the variable-rate probabilistic uniform quantizer is used. The top-2 performs worse since it would require $2(32+3)/5 = 14$ bits/node/component/iteration.

[1]Defined as $R(i) = \frac{1}{K}\sum_{k=1}^K \frac{1}{M}\mathbb{E}[\boldsymbol{r}_k(i)]$, where $\boldsymbol{r}_k(i)$ is the (random) number of bits associated with the encoding of $\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}$, at agent k, iteration i.

VI. REFERENCES

[1] T. Li, A. K. Sahu, A. S. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, pp. 50–60, May 2020.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, Ft. Lauderdale, FL, USA, 2017, vol. 54, pp. 1273–1282.

[3] 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.*, Long Beach, CA, USA, 2017, pp. 1709–1720.

[4] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[5] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4-5, pp. 311–801, 2014.

[6] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 14–25, 2020.

[7] D. Kovalev, A. Koloskova, M. Jaggi, P. Richtarik, and S. Stich, "A linearly convergent algorithm for decentralized optimization: Sending less bits for free!," in *Proc. Int. Conf. Artif. Intell. Stat.*, Virtual, 2021, pp. 4087–4095.

[8] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 3478–3487.

[9] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," Available as arXiv:2301.01313v1, 2023.

[10] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4905–4918, 2008.

[11] A. Beznosikov, S. Horvath, P. Richtárik, and M. H. Safaryan, "On biased compression for distributed learning," *J. Mach. Learn. Res.*, vol. 24, no. 276, pp. 1–50, 2023.

[12] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934–4947, 2019.

[13] M. Carpentiero, V. Matta, and Ali H. Sayed, "Distributed adaptive learning under communication constraints," *IEEE Open J. Signal Process.*, vol. 5, pp. 321–358, 2024.

[14] N. Michelusi, G. Scutari, and C.-S. Lee, "Finite-bit quantization for distributed algorithms with linear convergence," *IEEE Trans. Inf. Theory.*, vol. 68, no. 11, pp. 7254–7280, 2022.

[15] R. Nassif, S. Vlaski, M. Carpentiero, V. Matta, M. Antonini, and A. H. Sayed, "Quantization for decentralized learning under subspace constraints," *IEEE Trans. Signal Process.*, vol. 71, pp. 2320–2335, 2023.

[16] H. Zhao, B. Li, Z. Li, P. Richtarik, and Y. Chi, "BEER: Fast O(1/T) rate for decentralized nonconvex optimization with communication compression," in *Proc. Adv. Neural Inf. Process. Syst.*, New Orleans, Louisiana, USA, 2022, vol. 35, pp. 31653–31667.

[17] N. Singh, D. Data, J. George, and S. Diggavi, "SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization," *IEEE Trans. Automat. Contr.*, vol. 68, no. 2, pp. 721–736, 2023.

[18] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes SignSGD and other gradient compression schemes," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 2019, vol. 97, pp. 3252–3261.

[19] H. Tang, X. Lian, S. Qiu, L. Yuan, C. Zhang, T. Zhang, and J. Liu, "DeepSqueeze: Decentralization meets error-compensated compression," Available as arXiv:1907.07346, 2019.

[20] A. H. Sayed, *Inference and Learning from Data*, 3 vols., Cambridge University Press, 2022.

[21] R. Nassif, S. Vlaski, and A. H. Sayed, "Adaptation and learning over networks under subspace constraints–Part I: Stability analysis," *IEEE Trans. Signal Process.*, vol. 68, pp. 1346–1360, 2020.

[22] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control. Lett.*, vol. 53, no. 1, pp. 65–78, 2004.