Matching centralized learning performance via compressed decentralized learning with error feedback

Roula Nassif, Marco Carpentiero, Stefan Vlaski, Vincenzo Matta, Ali Sayed

Matching centralized learning performance via compressed decentralized learning with error feedback

Roula Nassif [1], Marco Carpentiero [2], Stefan Vlaski [3], Vincenzo Matta [2], Ali H. Sayed [4]
[1] Université Côte d'Azur, I3S Laboratory, CNRS, France
[2] University of Salerno, Italy
[3] Imperial College London, UK
[4] Ecole Polytechnique Fédérale de Lausanne, Switzerland

***Abstract*—The *DEF-ATC (Differential Error Feedback – Adapt Then Combine)* approach is a novel strategy to address decentralized learning and optimization problems under communication constraints. The strategy blends *differential quantization* and *error feedback* to mitigate the negative impact of exchanging *compressed updates* between neighboring agents. While differential quantization leverages correlations between subsequent iterates, error feedback (which consists of incorporating the compression error into subsequent steps) allows to compensate for the bias caused by compression. In this work, we examine the steady-state mean-square-error performance of the DEF-ATC approach in order to uncover the influence of several factors, including the gradient noise, the network topology, the learning step-size, and the compression schemes, on the network performance. The theoretical findings indicate that, under some general conditions on the compression error, and in the small step-size regime, it is possible to achieve performance levels comparable to those obtained without compression. This implies that, despite using compression techniques to reduce communication overheads, the performance of the decentralized compressed approach can still match that of its uncompressed counterpart, which in turn can match that of centralized learning where all data is aggregated and processed in a centralized manner.**

***Index Terms*—Error feedback, differential quantization, communication-efficient learning, decentralized learning, steady-state performance.**

I. Introduction

In recent years, there has been a growing interest in distributed machine learning frameworks across both academic research and industrial applications. This interest is closely related to their potential to address a multitude of challenges encountered in various domains, including handling large-scale data collected in a distributed and streaming manner, privacy concerns, and the need for collaborative learning in various fields ranging from healthcare and finance to smart cities and autonomous systems. One approach to distributed learning is *federated learning* where the model parameters are learned locally from local datasets distributed across various devices or nodes (such as smartphones), and then these parameters are transmitted to a central server for aggregation [1]–[3]. On the other hand, in *decentralized learning* frameworks, participating clients communicate directly with their neighboring nodes in an arbitrary topology without relying on a central server. This decentralized approach offers advantages in scenarios where communication with a server becomes a bottleneck [4]–[8].

In traditional decentralized implementations, agents need to exchange parameter vectors at every iteration of the learning algorithm, leading to high communication costs. To reduce these costs, a variety of methods have been proposed including: i) *skipping communication rounds* while performing a certain number of local updates in between [2], [9], [10], and ii) *compressing information* by employing either quantization (e.g., dithered quantizers [11]), sparsification (e.g., rand-k sparsifiers [7]), or both (e.g., top-c sparsifiers combined with dithering [12]). In the latter case, compression operators and learning algorithms are 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], [13]–[18].

In this work, we examine the steady-state mean-square-error performance of the *DEF-ATC (Differential Error Feedback – Adapt Then Combine) diffusion* approach, which is a communication-efficient decentralized approach that was recently proposed in [17]. The approach is characterized by the combination of two different concepts, namely, *differential quantization* and *error feedback*, in order to mitigate the negative impact of compressed communications, instead of using either of these concepts in isolation as in [8], [10], [14]–[16], [19]. Differential quantization, which consists of communicating compressed versions of the differences between current estimates and their predictions based on previous iterations, allows to leverage correlations between subsequent iterates. On the other hand, error feedback, which consists of locally storing the compression error and incorporating it back into the next iteration, allows to compensate for the bias caused by compression. While the work [17] establishes the mean-square-error stability of the DEF-ATC and shows that, for sufficiently small step-sizes μ, and under some general conditions on the compression error, it is possible to keep the estimation errors small (on the order of μ), the current work examines its steady-state performance. The aim here is to understand how the main factors, such as *gradient noise, network topology, learning step-size, compression schemes*, or *other relevant parameters*, affect the network steady-state performance. The analysis shows that, in the small step-size regime, the iterates generated by the DEF-ATC achieve the same steady-state performance as the decentralized baseline full-precision approach,

The work of R. Nassif was supported by ANR JCJC grant ANR-22- CE23-0015-01 (CEDRO project).

i.e., the diffusion *ATC (Adapt-Then-Combine)* approach [4], [5], where no communication compression is performed. Now, since the uncompressed decentralized ATC approach achieves the same network steady-state mean-square-error performance as the centralized solution (where all data is aggregated and processed in a centralized manner) [5, Chap. 12], this implies that the DEF-ATC approach matches the performance of centralized learning in the small step-size regime. Simulations illustrate the theoretical findings and the effectiveness of DEF-ATC, revealing that, in the small step-size regime, it is possible to achieve the same performance as the uncompressed centralized system without relying on a central processor and by using a finite number of bits.

II. Problem Setup and Algorithmic Framework

We consider the same decentralized optimization problem and algorithmic framework as in [17].

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

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

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), we employ the DEF-ATC approach proposed in [17] and listed in Algorithm 1. At each iteration i, each agent k in the network performs three steps. In the first step, which corresponds to the *adaptation* 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 μ/ζ 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 (5) to control the network stability in scenarios where compression leads to network instability. The second step is the *compression* step where 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 *randomized* compression operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ satisfying Property 1), and broadcasts it to its neighbors. Note that $\mathcal{N}_k$ denotes the set of nodes connected to agent k by a communication link (including node k itself). Then, agent k updates the compression error vector $\boldsymbol{z}_{k,i}$ according to (4) and produces, for each neighbor ℓ, an estimate $\boldsymbol{\phi}_{\ell,i}$ by using the received vector $\boldsymbol{\delta}_{\ell,i}$ according to (5). 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 (6) where agent k combines the reconstructed vectors $\{\boldsymbol{\phi}_{\ell,i}\}_{\ell\in\mathcal{N}_k}$ using a set of 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 (6) can also be used to control the network stability. The combination coefficients $\{a_{k\ell}\}$ are chosen such that, by collecting them in a matrix $A = [a_{k\ell}]$, the (k,ℓ)-th entry of the matrix is zero if nodes k and ℓ are not neighbors, i.e., $a_{k\ell} = 0$ if $\ell \notin \mathcal{N}_k$, and A 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.

Observe that, in the absence of compression (i.e., when the operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ is replaced by the identity operator and the stability parameters ζ and γ in (5) and (6) are set to 1), we obtain the diffusion ATC-type approach for solving (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 approach uses *differential quantization* and *error-feedback*.

Compression operators: Compression is performed through the application of a mapping $\mathcal{C} : \mathbb{R}^M \to \mathbb{R}^M$, where $\mathcal{C}(x)$ rep-

---

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, damping coefficient ζ, combination matrix A.

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

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

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \frac{\mu}{\zeta}\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1}) \quad (3)$$

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:

$$\boldsymbol{z}_{k,i} = (\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) - \boldsymbol{\delta}_{k,i} \quad (4)$$

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

$$\boldsymbol{\phi}_{\ell,i} = \boldsymbol{\phi}_{\ell,i-1} + \zeta\boldsymbol{\delta}_{\ell,i}, \qquad \ell \in \mathcal{N}_k \quad (5)$$

Combine: Update local model according to:

$$\boldsymbol{w}_{k,i} = (1-\gamma)\boldsymbol{\phi}_{k,i} + \gamma\sum_{\ell\in\mathcal{N}_k} a_{k\ell}\boldsymbol{\phi}_{\ell,i} \quad (6)$$

---

resents 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. ***(Unbiasedness and bounded variance).*** *The randomized compression operator $\boldsymbol{\mathcal{C}}_k(\cdot)$ at agent k satisfies the following conditions:*

$$\mathbb{E}[x - \boldsymbol{\mathcal{C}}_k(x)] = 0, \tag{7}$$

$$\mathbb{E}\|x - \boldsymbol{\mathcal{C}}_k(x)\|^2 \leq \beta_{c,k}^2 \|x\|^2 + \sigma_{c,k}^2, \tag{8}$$

for some $\beta_{c,k}^2 \geq 0$ and $\sigma_{c,k}^2 \geq 0$, and where the expectation is evaluated with respect to the randomness of $\boldsymbol{\mathcal{C}}_k(\cdot)$. □

Property 1 is satisfied by many compression operators of interest in decentralized learning such as rand-c, gradient sparsifier, QSGD, probabilistic ANQ, and probabilistic uniform quantizer – see Table 1 in [16] for details.

III. Mean-square-error performance

A. Mean-square-error stability

The work [17] examined the steady-state average squared distance between the random estimates $\boldsymbol{w}_{k,i}$ generated by Algorithm 1 and w^o, namely, $\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2$, under the following assumptions on the risks $\{J_k(\cdot)\}$ and on the gradient noise processes $\{\boldsymbol{s}_{k,i}(\cdot)\}$ defined as [5]:

$$\boldsymbol{s}_{k,i}(w) \triangleq \nabla_w J_k(w) - \widehat{\nabla_w J_k}(w). \tag{9}$$

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 $J^{\text{glob}}(w)$ is twice-differentiable and strongly convex. It is further assumed to satisfy:*

$$0 < \nu I_M \leq \nabla_w^2 J^{\text{glob}}(w) \leq \delta I_M, \tag{10}$$

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 (9) satisfies for $k = 1, \ldots, K$:*

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) | \{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^K] = 0, \tag{11}$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2 | \{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^K] \leq \beta_{s,k}^2 \|\widetilde{\boldsymbol{w}}_{k,i-1}\|^2 + \sigma_{s,k}^2, \tag{12}$$

for some $\beta_{s,k}^2 \geq 0$, $\sigma_{s,k}^2 \geq 0$, and where $\widetilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$. □

Before providing a brief description of the theoretical findings in [17], it is worth mentioning that the mean-square-error stability and performance analyses exploit the eigenstructure of the combination matrix A satisfying (2). It can be shown that this matrix has a Jordan decomposition of the form $A = V_\epsilon J V_\epsilon^{-1}$, where [5]:

$$V_\epsilon = [\alpha \mathbb{1}_K | V_R], \;\; J = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & J_\epsilon \end{array}\right], \;\; V_\epsilon^{-1} = \left[\begin{array}{c} \alpha \mathbb{1}_K^\top \\ \hline V_L^\top \end{array}\right], \tag{13}$$

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 $\rho(J_\epsilon) + \epsilon < 1$.

Under Assumptions 1 and 2, a network of K agents running the compressed decentralized Algorithm 1 with:

1) combination matrix A satisfying (2),
2) compression operators $\{\boldsymbol{\mathcal{C}}_k(\cdot)\}$ satisfying Property 1 with the absolute compression noise terms $\sigma_{c,k}^2 \propto \mu^2$,
3) *in the presence of the relative quantization noise (i.e., $\beta_{c,k}^2 \neq 0$):* stability parameters $\zeta \in (0, 1]$ and $\gamma \in (0, 1]$ satisfying the two following conditions:

$$\|\mathcal{J}_\epsilon''\| + 4 v_1^2 v_2^2 \beta_{c,\max}^2 \zeta^2 \|I - \mathcal{J}_\epsilon'\|^2 < 1, \tag{14}$$

$$\frac{2\zeta^2 \|I - \mathcal{J}_\epsilon'\|^2}{1 - \|\mathcal{J}_\epsilon''\|} + 2\beta_{c,\max}^2 \zeta^2 v_1^2 v_2^2 \left(1 + \|2I - \mathcal{J}_\epsilon'\|^2\right) < 1, \tag{15}$$

where:

$$\mathcal{J}_\epsilon' \triangleq [(1-\gamma) I_{K-1} + \gamma J_\epsilon] \otimes I_M, \tag{16}$$

$$\mathcal{J}_\epsilon'' \triangleq [(1-\gamma\zeta) I_{K-1} + \gamma\zeta J_\epsilon] \otimes I_M, \tag{17}$$

$v_1 \triangleq \|\mathcal{V}_\epsilon^{-1}\|$, $v_2 \triangleq \|\mathcal{V}_\epsilon\|$, $\beta_{c,\max}^2 \triangleq \max_{1\leq k\leq K}\{\beta_{c,k}^2\}$, σ_{11} is some positive constant that depends on ν, $\|\cdot\|$ and $\otimes$ represent the 2-induced matrix norm and the Kronecker product operation, respectively.

4) *in the absence of the relative quantization noise:* stability parameters ζ and γ set to 1,

is mean-square-error stable for sufficiently small step-size μ, namely, it holds that [17, Theorem 1]:

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu), \qquad k = 1, \ldots, K, \tag{18}$$

for small μ. This result shows that, despite the gradient and compression noise processes, the error variance of the communication-efficient decentralized Algorithm 1 relative to the solution w^o enters a bounded region whose size is on the order of μ. In the current work, we shall assess the size of this mean-square-error by deriving a closed-form expression for the network mean-square-deviation (MSD) defined by [5]:

$$\text{MSD} \triangleq \mu \lim_{\mu\to 0} \left(\limsup_{i\to\infty} \frac{1}{\mu} \left(\frac{1}{K} \sum_{k=1}^K \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 \right) \right). \tag{19}$$

That is, we shall assess the size of the constant multiplying μ in the $O(\mu)$ term. This closed form expression will reveal how the step-size μ, the data characteristics (captured by the second-order properties of the costs and second order moments of the gradient noises), the compression error (captured by the parameters $\beta_{c,k}^2$ and $\sigma_{c,k}^2$), and the network topology (captured by the combination coefficients $a_{k\ell}$) influence the network mean-square-error performance. Furthermore, by analyzing this expression, we will be able to compare the performance of different approaches, such as the DEF-ATC, the decentralized uncompressed ATC, and the centralized uncompressed approach given by [5, Chap. 5]:

$$\boldsymbol{w}_i^{\text{cent}} = \boldsymbol{w}_{i-1}^{\text{cent}} - \frac{\mu}{K}\sum_{k=1}^{K}\widehat{\nabla_w J_k}(\boldsymbol{w}_{i-1}^{\text{cent}}), \tag{20}$$

where $\boldsymbol{w}_i^{\text{cent}}$ is the estimate of w^o at iteration i and where the superscript "cent" is used to indicate that the iterate is for the centralized solution. Observe that each agent at each iteration in (20) needs to send its data (or its gradient approximation) to a fusion center, which performs the aggreagation in (20) and then sends the resulting estimate $\boldsymbol{w}_i^{\text{cent}}$ back to the agents.

B. Mean-square-error performance

Due to space limitations, we only list the main results of the analysis without showing the proof details. The arguments are along the lines developed in [5, Chaps. 9–11], [22] for uncompressed diffusion strategies with proper adjustments to handle communication-efficient learning. We focus on the steady-state mean-square-error performance of Algorithm 1 under Assumptions 1 and 2 and the following two smoothness assumptions on the risks $\{J_k(\cdot)\}$ and the gradient noise processes $\{\boldsymbol{s}_{k,i}(\cdot)\}$.

Assumption 3. *It is assumed that each $J_k(w)$ satisfies the following smoothness condition:*

$$\|\nabla_w J_k(w^o + \Delta w) - \nabla_w J_k(w^o)\| \leq \kappa\|\Delta w\|, \tag{21}$$

for small perturbations $\|\Delta w\|$ and $\kappa \geq 0$.

We refer to the individual gradient noise process in (9) and denote its conditional covariance matrix by:

$$R_{s,k,i}(\boldsymbol{w}_{k,i-1}) \triangleq \mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\boldsymbol{s}_{k,i}^{\top}(\boldsymbol{w}_{k,i-1})|\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^{K}\right]. \tag{22}$$

We assume that, in the limit, the following moment matrices tend to constant values when evaluated at w^o:

$$R_{s,k} \triangleq \lim_{i\to\infty} R_{s,k,i}(w^o). \tag{23}$$

Assumption 4. *It is assumed that each $R_{s,k,i}(w)$ satisfies the following smoothness condition close to the limit point w^o:*

$$\|R_{s,k,i}(w^o + \Delta w) - R_{s,k,i}(w^o)\| \leq \kappa_s\|\Delta w\|^{\theta}, \tag{24}$$

for small perturbations $\|\Delta w\|$, and for some constant $\kappa_s \geq 0$ and exponent $0 < \theta \leq 4$.

Theorem 1. (Mean-square-error performance). *Consider a network of K agents running the compressed decentralized DEF-ATC Algorithm 1 with a combination matrix A satisfying (2). Assume that the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 1 and 3. Assume further that the gradient noise processes satisfy Assumption 4 and Assumption 2 with the fourth-order moment condition:*

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^4|\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell=1}^{K}] \leq \beta_{s4,k}^4\|\widetilde{\boldsymbol{w}}_{k,i-1}\|^4 + \sigma_{s4,k}^4, \tag{25}$$

where $\beta_{s4,k}^4 \geq 0$ and $\sigma_{s4,k}^4 \geq 0$. Assume that the compression operators $\{\boldsymbol{\mathcal{C}}_k(\cdot)\}$ satisfy Property 1 with the fourth-order condition:

$$\mathbb{E}\|x - \boldsymbol{\mathcal{C}}_k(x)\|^4 \leq \beta_{c4,k}^4\|x\|^4 + \sigma_{c4,k}^4, \tag{26}$$

where $\beta_{c4,k}^4 \geq 0$ and $\sigma_{c4,k}^4 \propto \mu^4 \geq 0$. In the absence of the relative quantization noise term (i.e., $\beta_{c,k}^2 = 0$, $\beta_{c4,k}^4 = 0$, $\forall k$), let the stability parameters be such that $\gamma = \zeta = 1$. In the presence of the relative quantization noise, let $\zeta \in (0, 1]$ and $\gamma \in (0, 1]$ be such that the two conditions in (14) and (15), and the following two conditions are satisfied:

$$\|\mathcal{J}_\epsilon''\| + 128v_1^4v_2^4\beta_{c4,\max}^4\zeta^4\|I - \mathcal{J}_\epsilon'\|^4 < 1, \tag{27}$$

$$\frac{8\zeta^4\|I - \mathcal{J}_\epsilon'\|^4}{(1 - \|\mathcal{J}_\epsilon''\|)^3} + 16\beta_{c4,\max}^4v_1^4v_2^4\zeta^4\left(1 + \|2I - \mathcal{J}_\epsilon'\|^4\right) < 1, \tag{28}$$

where $\beta_{c4,\max}^4 \triangleq \max_{1\leq k\leq K}\{\beta_{c4,k}^4\}$. Then, it holds that:

$$\text{MSD} \overset{(19)}{=} \frac{\mu}{2K}\text{Tr}\left(\left(\sum_{k=1}^{K}H_k^o\right)^{-1}\left(\sum_{k=1}^{K}R_{s,k}\right)\right), \tag{29}$$

where each H_k^o is given by the value of the Hessian matrix at the minimizer, namely, $H_k^o = \nabla_w^2 J_k(w^o)$, and where $\text{Tr}(\cdot)$ denotes the trace operator.

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

While expressions (14), (15), (27), and (28) reveal the influence of the *relative quantization noise* term (captured by $\{\beta_{c,\max}^2, \beta_{c4,\max}^4\}$) on the network stability, and how this influence can be mitigated by properly choosing the damping coefficient ζ and the mixing parameter γ, expression (29) reveals explicitly the influence of the step-size μ, the data characteristics (through the Hessian matrices H_k^o), and the gradient noise (through the covariance matrices $R_{s,k}$) on the network MSD performance. Note first that the performance of the uncompressed diffusion ATC approach (with a combination matrix satisfying (2)) is also equal to (29) [5, Sec. 12.1]. This implies that, despite using compression techniques to reduce the communication overhead, the performance of the DEF-ATC matches that of its uncompressed counterpart. From [5, Sec. 12.1], we also know that the performance of the centralized implementation (20) is equal to (29). This implies that, for sufficiently small step-sizes, the decentralized compressed DEF-ATC algorithm can achieve the same performance as the uncompressed centralized solution (20) without relying on a fusion center, and by reducing considerably the communication costs compared to the uncompressed decentralized variant.

IV. Simulation Results

To illustrate the theoretical findings, we adopt in our simulations the same experimental setup as in [17]. In particular, we consider the same 50-node mean-square-error (MSE) network as in [17, Fig. 1]. Each agent is subjected to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ related according to 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 5×1 vector w_k^o with $\boldsymbol{v}_k(i)$ denoting a zero-mean measurement noise. The cost over the MSE network is defined by $J_k(w) = \frac{1}{2}\mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}^{\top}w|^2$. The processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$, the model parameters $\{w_k^o\}$, and the matrix A satisfying the conditions in (2) are generated using similar settings as in [17]. We implement two *unbiased* compression schemes: i) the QSGD scheme [3], which

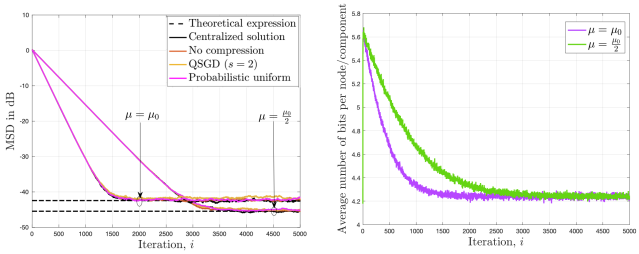

Fig. 1. *(Left)* Performance of DEF-ATC for two different values of μ ($\mu_0 = 0.003$) when QSGD and variable-rate probabilistic uniform ($\sigma^2_{c,k} = \mu^2$) compression operators are used. Red and black curves correspond to the standard diffusion ATC approach and the centralized solution (20), respectively. Dashed curves correspond to the analytical expression (29). *(Right)* Evolution of the average number of bits per node, per component, when the variable-rate probabilistic uniform quantizer is used.

transmits the norm with high precision and randomly rounds the components to s-bit representations [3] (in this case, $\beta^2_{c,k} = \min(\frac{M}{s^2}, \frac{\sqrt{M}}{s})$, $\sigma^2_{c,k} = 0$ [16]); and $ii)$ the variable-rate probabilistic uniform quantizer [15], which incorporates dithering into the uniform quantization scheme (in this case, $\beta^2_{c,k} = 0$, $\sigma^2_{c,k} = \frac{M\Delta^2}{4}$, where Δ is the quantization step). For the variable-rate probabilistic uniform quantizer, we set the stability parameters $\gamma = \zeta = 1$ and the quantization step Δ such that $\sigma^2_{c,k} = \mu^2$. For the QSGD compression operator, we set $s = 2$, $\zeta = 1$, 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 (left) for two different values of the step-size. Results are averaged over 100 Monte-Carlo runs. We observe that the theoretical model (29) matches well the actual performance of the network running the decentralized compressed DEF-ATC approach. Furthermore, we report in Fig. 1 (left) the performance of the uncompressed ATC approach (which can be obtained from Algorithm 1 by setting $\zeta = \gamma = 1$ and replacing the compression operator by identity) and the performance of the centralized solution (20). As predicted by Theorem 1, in the slow step-size regime, the compressed decentralized solution is able to attain the same performance (29) as the centralized one.

To illustrate the efficiency of the DEF-ATC in terms of communication savings, we report in Fig. 1 (right) the average number of bits per node, per component, when the variable-rate probabilistic uniform quantizer is employed. As it can be observed, for the two different values of μ, we approximately obtain the same finite average number of bits in steady state (approximately 4.2 bits/component/iteration are required). We recall from [17] that the QSGD scheme with $s = 2$ would require 8.4 bits/node/component/iteration (assuming that the number of bits required to encode a scalar with high precision is 32), which is almost two times higher that the one obtained when the variable-rate probabilistic uniform quantizer is used.

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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] R. Nassif, S. Vlaski, M. Carpentiero, V. Matta, and A. H. Sayed, "Differential error feedback for communication-efficient decentralized optimization," in *Proc. IEEE Sens. Array Multichannel Signal Process. Workshop*, Corvallis, OR, USA, Jul. 2024.

[18] 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.

[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] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part II: Performance analysis," *IEEE Trans. Inf. Theory.*, vol. 61, no. 6, pp. 3518–3548, 2015.