FINITE BIT QUANTIZATION FOR DECENTRALIZED LEARNING UNDER SUBSPACE CONSTRAINTS

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

[1] Université Côte d'Azur, France

[2] Imperial College London, UK

[3] University of Salerno, Italy

[4] École Polytechnique Fédérale de Lausanne, Switzerland

ABSTRACT

In this paper, we consider decentralized optimization problems where agents have individual cost functions to minimize subject to subspace constraints that require the minimizers across the network to lie in low-dimensional subspaces. This constrained formulation includes consensus optimization as special case, and allows for more general task relatedness models such as multitask smoothness and coupled optimization. In order to cope with communication constraints, we propose and study a quantized differential based approach where the communicated estimates among agents are quantized. The analysis shows that, under some general conditions on the quantization noise, and for sufficiently small step-sizes μ, the strategy is stable in the mean-square error sense. The analysis also reveals the influence of the gradient and quantization noises on the performance.

***Index Terms*—Decentralized stochastic optimization, subspace projection, quantization effects.**

I. INTRODUCTION

Mobile phones, wearable devices, and autonomous vehicles are examples of modern distributed networks generating massive amounts of data each day. Due to the growing computational power in these devices and the increasing size of the datasets, coupled with concerns over sharing private data, federated and decentralized training of statistical models have become desirable and often necessary [1]–[7]. 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. Training data is kept locally on users' devices, and the devices are used as agents performing computation on their local data in order to update global models of interest. In applications where communication to a server becomes a bottleneck, *decentralized* topologies (where agents only communicate with their neighbors) are potential alternatives to federated topologies (where a server connects with all remote devices). Decentralized implementations can reduce the high communication cost on the central server since, in this case, model updates are exchanged between agents without relying on a central coordinator [5]–[8].

There have been significant works in the literature on solving optimization and inference problems in a decentralized manner [5]–[19]. However, with some exceptions [8], [12]–[19], the large majority of these works is not tailored to the specific challenge of limited communication capabilities (due to tight energy and bandwidth constraints) encountered in decentralized settings. In this work, we study the effects of quantization on the performance of the following decentralized stochastic gradient approach that has been recently proposed and studied in [10], [11]:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1}) & \text{(1a)} \\ \boldsymbol{w}_{k,i} = \displaystyle\sum_{\ell\in\mathcal{N}_k} A_{k\ell}\boldsymbol{\psi}_{\ell,i} & \text{(1b)} \end{cases}$$

where $\mu > 0$ is a small step-size parameter, $\mathcal{N}_k$ is the neighborhood set of agent k (i.e., the set of nodes connected to agent k by a communication link or edge), $A_{k\ell}$ is an $M_k \times M_\ell$ matrix associated with the link (k,ℓ), $w_k \in \mathbb{R}^{M_k}$ is the parameter vector at agent k, and $J_k(w_k) : \mathbb{R}^{M_k} \to \mathbb{R}$ 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 $J_k(w_k) = \mathbb{E}\, L_k(w_k; \boldsymbol{y}_k)$, where $\boldsymbol{y}_k$ denotes the random data (throughout the paper, random quantities are denoted in boldface). The expectation is computed over the data. In the stochastic setting, when the distribution of the data $\boldsymbol{y}_k$ is unknown, the risks $J_k(\cdot)$ and their gradients $\nabla_{w_k} J_k(\cdot)$ are unknown. In this case, and instead of using the true gradient, it is common to use approximate gradient vectors by using $\widehat{\nabla_{w_k} J_k}(w_k) = \nabla_{w_k} L_k(w_k; \boldsymbol{y}_{k,i})$ where $\boldsymbol{y}_{k,i}$ represents the data observed at iteration i [5]. We note that a unique feature of algorithm (1) is the utilization of *matrix* valued combination weights, as opposed to scalar weighting as is commonly employed in conventional consensus and diffusion optimization [5], [9]. As explained in the following, this generalization allows the network to solve a broader class of multitask optimization problems beyond classical consensus.

Let N denote the total number of agents and let $\mathcal{w} = \text{col}\{w_1, \ldots, w_N\}$ denote the collection of parameter vectors from across the network. Let $\mathcal{A}$ denote the $N \times N$ block matrix whose (k,ℓ)-th block is $A_{k\ell}$ if $\ell \in \mathcal{N}_k$ and 0 otherwise. It was shown in [10, Theorem 1] that, for sufficiently small μ and for a combination matrix $\mathcal{A}$ satisfying:

$$\mathcal{A}\mathcal{U} = \mathcal{U}, \quad \mathcal{U}^\top\mathcal{A} = \mathcal{U}^\top, \quad \text{and} \quad \rho(\mathcal{A} - \mathcal{P}_\mathcal{U}) < 1, \qquad (2)$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument,

$\mathcal{U}$ is an $M \times P$ full-column rank matrix (with $P \ll M$) that is assumed to be semi-orthogonal ($\mathcal{U}^\top \mathcal{U} = I_P$), and $\mathcal{P}_{\mathcal{U}}$ is the orthogonal projection matrix onto Range($\mathcal{U}$), strategy (1) will converge in the mean-square-error sense to the solution of the following problem:

$$\begin{aligned} \mathcal{W}^o = \arg\min_{\mathcal{W}} \; & J^{\text{glob}}(\mathcal{W}) \triangleq \sum_{k=1}^{N} J_k(w_k) \\ & \text{subject to } \mathcal{W} \in \text{Range}(\mathcal{U}) \end{aligned} \tag{3}$$

In particular, it was shown that $\limsup_{i\to\infty} \mathbb{E}\|w_k^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu)$ for all k, where w_k^o is the k-th $M_k \times 1$ subvector of $\mathcal{W}^o$. As explained in [10, Sec. II], by properly selecting $\mathcal{U}$ and $\mathcal{A}$, strategy (1) can be employed to solve different decentralized (single-task and multitask) optimization problems such as consensus optimization [5], [9], decentralized coupled optimization [10], and multitask inference under smoothness [10].

The first step (1a) in algorithm (1) is the *self-learning* step corresponding to the stochastic gradient descent step on the individual cost $J_k(\cdot)$. This step is followed by the *social learning* step (1b) where agent k receives the intermediate estimates $\{\boldsymbol{\psi}_{\ell,i}\}$ from its neighbors $\ell \in \mathcal{N}_k$ and combines them through $\{A_{k\ell}\}$ to form $\boldsymbol{w}_{k,i}$, which corresponds to the estimate of w_k^o at agent k and iteration i. To alleviate the communication bottleneck resulting from the exchange of the intermediate estimates among agents over many iterations, quantized communication can be considered. In this paper, we study the effect of quantization on the convergence properties of the decentralized learning approach (1). First, we propose in Sec. II a *differential quantization* based algorithm for solving problem (3), and then we establish in Sec. III that, under some general conditions on the quantization noise, and for sufficiently small step-sizes μ, the proposed decentralized quantized approach is stable in the mean-square error sense. Our analysis reveals explicitly the influence of the gradient and quantization noises on the network performance. The analysis also shows that, by properly designing the quantization operator, the iterates generated by the quantized decentralized adaptive implementation can still lead to small estimation errors on the order of μ.

II. DECENTRALIZED LEARNING IN THE PRESENCE OF QUANTIZED COMMUNICATIONS

Motivated by the approaches proposed in [8], [15]–[19], we equip strategy (1) with a quantization mechanism by proposing the following decentralized learning approach:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1}) & \text{(4a)} \\ \boldsymbol{\phi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \boldsymbol{Q}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1}) & \text{(4b)} \\ \boldsymbol{w}_{k,i} = \zeta \boldsymbol{\phi}_{k,i} + (1-\zeta) \sum_{\ell \in \mathcal{N}_k} A_{k\ell} \boldsymbol{\phi}_{\ell,i} & \text{(4c)} \end{cases}$$

where $\zeta \in (0,1)$ and $\boldsymbol{Q}_k(\cdot)$ is the quantization operator (a map from real valued vectors to a finite set of quantized vectors) at agent k. *Differential quantization* is used. In this case, instead of communicating compressed versions of the estimates $\boldsymbol{\psi}_{k,i}$, the *prediction* error $\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1}$ is quantized at agent k and then transmitted [8], [15]–[19]. At each iteration i, agent k performs *quantization* by mapping the real-valued vector $\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1}$ into a quantized vector $\boldsymbol{\psi}'_{k,i} = \boldsymbol{Q}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1})$, sends $\boldsymbol{\psi}'_{k,i}$ to its neighbors through the communication links, receives $\{\boldsymbol{\psi}'_{\ell,i}\}$ from its neighbors $\ell \in \mathcal{N}_k$, and computes $\{\boldsymbol{\phi}_{\ell,i}\}$ according to step (4b):

$$\boldsymbol{\phi}_{\ell,i} = \boldsymbol{\phi}_{\ell,i-1} + \boldsymbol{\psi}'_{\ell,i}, \qquad \ell \in \mathcal{N}_k. \tag{5}$$

Observe that implementing (5) requires storing the previous estimates $\{\boldsymbol{\phi}_{\ell,i-1}\}_{\ell\in\mathcal{N}_k}$ by agent k. The reconstructed vectors $\{\boldsymbol{\phi}_{\ell,i}\}$ are then combined according to (4c) to produce the estimate $\boldsymbol{w}_{k,i}$.

We shall analyze algorithm (4) under the following general assumption on the quantizers $\{\boldsymbol{Q}_k(\cdot)\}$, which relaxes the condition on the mean-square error from [3], [8], [15]–[18].

Assumption 1. *The quantizers $\{\boldsymbol{Q}_k(\cdot)\}$ are random and satisfy:*

$$\mathbb{E}[\boldsymbol{x}_k - \boldsymbol{Q}_k(\boldsymbol{x}_k)|\boldsymbol{x}_k] = 0, \tag{6}$$

$$\mathbb{E}[\|\boldsymbol{x}_k - \boldsymbol{Q}_k(\boldsymbol{x}_k)\|^2|\boldsymbol{x}_k] \le \beta_{z,k}^2 \|\boldsymbol{x}_k\|^2 + \sigma_{z,k}^2, \tag{7}$$

for some $\beta_{z,k}^2 \ge 0$, $\sigma_{z,k}^2 \ge 0$, and where the expectations are evaluated w.r.t. the randomness of $\boldsymbol{Q}_k(\cdot)$. When $\boldsymbol{x}_k$ is random, the quantizer is statistically independent of $\boldsymbol{x}_k$.

Note that the conditions in Assumption 1 are satisfied by many random quantization operators of interest in decentralized learning such as the the rand-k and dit-k quantizers [17]. While many existing works focus on studying decentralized learning approaches in the presence of random quantizers that satisfy the *unbiasedness* condition (6) and the *variance bound* condition (7) with the *absolute noise* term $\sigma_{z,k}^2 = 0$ [3], [8], [15], [16], the analysis in the current work is general and does not require $\sigma_{z,k}^2$ to be zero. As explained in [19], neglecting the effect of $\sigma_{z,k}^2$ requires that some quantities (e.g., the norm of the vector to be quantized [8]) are represented with no quantization error, in practice at the machine precision. In the following, we illustrate how a finite-bit random quantizer $\boldsymbol{Q}_k(\cdot)$ satisfying conditions (6) and (7) can be designed.

Example 1 (*Randomized quantizer* [19]): Given a scalar ξ, and two design parameters $\omega \in [0,1)$ and $\eta > 0$, the quantized value $\boldsymbol{Q}(\xi)$ is computed as follows:

$$j = \left\lceil \frac{\ln\left(1 + \frac{\omega}{\eta}|\xi|\right)}{2\ln\left(\omega + \sqrt{1+\omega^2}\right)} \right\rceil \tag{8a}$$

$$q_j = \frac{\eta}{\omega}\left[\left(\omega + \sqrt{1+\omega^2}\right)^{2j} - 1\right] \tag{8b}$$

$$\boldsymbol{q} = \begin{cases} q_{j-1} & \text{with probability } \frac{q_j - |\xi|}{q_j - q_{j-1}} \\ q_j & \text{with probability } \frac{\xi - q_{j-1}}{q_j - q_{j-1}} \end{cases} \tag{8c}$$

$$\boldsymbol{Q}(\xi) = \boldsymbol{q} \cdot \text{sign}(\xi) \tag{8d}$$

Step (8a) is a rounding operation, which when coupled with step (8b), produces a reproduction level of the quantizer. Step (8c) exploits probabilistic quantization and selects randomly one of the nearest quantization points so as to guarantee the unbiasedness condition $\mathbb{E}[\boldsymbol{Q}(\xi)] = \xi$. Finally, step (8d) is used to handle negative inputs in a symmetric way. Vector-valued inputs $x \in \mathbb{R}^L$ are quantized component-wise according to the rule (8). Note that the design parameters ω and η in (8) are used to tune the degree of non-uniformity and the resolution of the quantizer. It is shown in [19] that the quantizer (8) satisfies Assumption 1 with $\sigma_{z,k}^2 = 2L\eta^2$ and $\beta_{z,k}^2 = 2\omega^2$. Observe that the described quantizer would require an infinite number of bits since the quantization range is unbounded. To overcome this issue, we can resort to a variable-rate scheme that uses a different number of bits depending on the value x to be quantized. To this aim, let us consider an encoder alphabet $\mathcal{S} \bigcup \{s_0\}$, where $\mathcal{S}$ is a certain alphabet with $|\mathcal{S}| \geq 2$, and s_0 is a special symbol reserved to parse the received encoded string. Given a sequence of samples $x_1, x_2, \ldots$, the variable-rate encoder: $i)$ determines the number of symbols in $\mathcal{S}$ necessary to represent index j in (8a); $ii)$ adds the special symbol s_0 to denote termination of the information string; and $iii)$ repeats the procedure on x_2, x_3, and so on. A strategy to implement this type of variable-rate quantizer is proposed in [19]. While the number of bits required to encode $x \in \mathbb{R}^L$ can be evaluated numerically (as we will do in the simulations), the following upper bound on the number of bits is derived in [19]:

$$B(x) = M \log_2(S+1) \times \left[3 + \log_S \left(2 + \frac{\ln\left(1 + \frac{\omega}{\eta} \frac{\|x\|}{\sqrt{L}}\right)}{2\ln\left(\omega + \sqrt{1+\omega^2}\right)} \right) \right]. \quad (9)$$

□

III. STOCHASTIC PERFORMANCE ANALYSIS

We analyze strategy (4) with a matrix $\mathcal{A}$ satisfying (2) by examining the average squared distance between $\boldsymbol{w}_{k,i}$ and w_k^o, namely, $\limsup_{i\to\infty} \mathbb{E}\|w_k^o - \boldsymbol{w}_{k,i}\|^2$, under Assumption 1 and the following assumptions on the risks $\{J_k(\cdot)\}$ and on the gradient noise processes $\{\boldsymbol{s}_{k,i}(\cdot)\}$ defined as [5]:

$$\boldsymbol{s}_{k,i}(w) \triangleq \nabla_{w_k} J_k(w) - \widehat{\nabla_{w_k} J_k}(w). \quad (10)$$

Assumption 2. *The individual costs $J_k(w_k)$ are assumed to be twice differentiable and convex such that:*

$$\lambda_{k,\min} I_{M_k} \leq \nabla_{w_k}^2 J_k(w_k) \leq \lambda_{k,\max} I_{M_k}, \quad (11)$$

where $\lambda_{k,\min} \geq 0$ for $k = 1, \ldots, N$. It is further assumed that, for any $\{w_k \in \mathbb{R}^{M_k}\}$ the individual costs satisfy:

$$0 < \lambda_{\min} I_P \leq \mathcal{U}^\top \mathrm{diag}\left\{\nabla_{w_k}^2 J_k(w_k)\right\}_{k=1}^N \mathcal{U} \leq \lambda_{\max} I_P, \quad (12)$$

for some positive parameters $\lambda_{\min} \leq \lambda_{\max}$.

Assumption 3. *The gradient noise process defined in* (10) *satisfies for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for $k = 1, \ldots, N$:*

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0, \quad (13)$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \leq \beta_{s,k}^2 \|\boldsymbol{w}\|^2 + \sigma_{s,k}^2, \quad (14)$$

for some $\beta_{s,k}^2 \geq 0$, $\sigma_{s,k}^2 \geq 0$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration generated by the random processes $\{\boldsymbol{w}_{\ell,j}, \boldsymbol{\phi}_{\ell,j}\}$ for all $\ell = 1, \ldots, N$ and $j \leq i-1$.

Before proceeding, it should be noted that there exist several useful works in the literature that study decentralized variations of stochastic gradient descent in the presence of differential quantization [8], [15]–[18], under different assumptions on the gradient noise, quantization operator, and cost functions. For instance, while the works [8], [17], [18] require strongly convex costs at each agent, the works [15], [16] relax this assumption by requiring network global strong convexity. In the current work, condition (12) requires the costs to be strongly convex in the range space of $\mathcal{U}$. In terms of consensus optimization, this is equivalent to requiring global strong convexity, namely, $\sum_{k=1}^N \nabla_{w_k}^2 J_k(w_k) > 0$. While we also assume global strong convexity, it should be noted that the analysis of strategy (4) differs from the analysis conducted in [15], [16] in three different ways. First, approach (4) uses *matrix-valued combination* coefficients $A_{k\ell}$ instead of scalar valued coefficients $a_{k\ell}$. As we previously explained, this generalization allows us to solve general constrained problems of the form (3). Second, instead of using the *damping coefficient* ζ in the quantization step (4b) as in [15], [16], it is used in the social learning step (4c). Consequently, under Assumption 1, the network quantization error vector $\boldsymbol{z}_i$ (defined in (28)), which collects the individual quantization error vectors, will be zero mean in the subsequent analysis. Third, and unlike the previous works [8], [15]–[18], the current analysis is conducted in the presence of the *absolute noise* term $\sigma_{z,k}^2$.

Theorem 1. ***(Network mean-square-error stability)*** *Consider a network of N agents running the quantized decentralized strategy* (4) *with a matrix $\mathcal{A}$ satisfying* (2)*. Let $1 - \zeta = O(\mu^\gamma)$ with $\gamma \in (0, 0.5)$. Under Assumptions 1, 2, and 3, the network is mean-square-error stable for sufficiently small step-size μ, namely, it holds that:*

$$\limsup_{i\to\infty} \mathbb{E}\|w_k^o - \boldsymbol{w}_{k,i}\|^2 = \beta_{z,\max}^2 O(\mu^{1-\gamma}) + \sigma_s^2 O(\mu) + \varphi^2 O(\mu) + \sigma_z^2 O(\mu^{-1}), \quad (15)$$

for $k = 1, \ldots, N$ and where

$$\sigma_s^2 \triangleq \sum_{k=1}^N \left(2\beta_{s,k}^2 \|w_k^o\|^2 + \sigma_{s,k}^2\right), \beta_{z,\max}^2 \triangleq \max_{1\leq k\leq N} \{\beta_{z,k}^2\},$$

$\sigma_z^2 \triangleq \sum_{k=1}^N \sigma_{z,k}^2$*, and φ^2 is a term that depends on $\beta_{z,\max}^2$ and $\|b\|^2 = O(1)$ where the vector b is given by* (17)*.*

Proof. Let $\widetilde{\boldsymbol{w}}_{k,i} = w_k^o - \boldsymbol{w}_{k,i}$, $\widetilde{\boldsymbol{\psi}}_{k,i} = w_k^o - \boldsymbol{\psi}_{k,i}$, and $\widetilde{\boldsymbol{\phi}}_{k,i} = w_k^o - \boldsymbol{\phi}_{k,i}$. Using similar arguments as in [10], we can show that the vector $\widetilde{\boldsymbol{\psi}}_i = \mathrm{col}\{\widetilde{\boldsymbol{\psi}}_{k,i}\}_{k=1}^N$ evolves according to:

$$\widetilde{\boldsymbol{\psi}}_i = (I_M - \mu \boldsymbol{\mathcal{H}}_{i-1})\, \widetilde{\boldsymbol{\mathcal{W}}}_{i-1} - \mu \boldsymbol{s}_i + \mu b, \quad (16)$$

where

$$b \triangleq \operatorname{col}\left\{\nabla_{w_k} J_k(w_k^o)\right\}_{k=1}^N, \tag{17}$$

$$\boldsymbol{s}_i \triangleq \operatorname{col}\left\{\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\right\}_{k=1}^N, \tag{18}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \triangleq \operatorname{diag}\left\{\boldsymbol{H}_{k,i-1}\right\}_{k=1}^N, \tag{19}$$

with $\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla_{w_k}^2 J_k(w_k^o - t\widetilde{\boldsymbol{w}}_{k,i-1})dt$. By subtracting w_k^o from both sides of (4c), by replacing w_k^o by $\zeta w_k^o + (1-\zeta)w_k^o$, and by using $w_k^o = \sum_{\ell\in\mathcal{N}_k} A_{k\ell} w_\ell^o$ [10], we obtain:

$$\widetilde{\boldsymbol{w}}_{k,i} = \zeta\widetilde{\boldsymbol{\phi}}_{k,i} + (1-\zeta)\sum_{\ell\in\mathcal{N}_k} A_{k\ell}\widetilde{\boldsymbol{\phi}}_{\ell,i}. \tag{20}$$

From (20), we can show that the network error vector $\widetilde{\boldsymbol{\mathcal{W}}}_{i-1} = \operatorname{col}\{\widetilde{\boldsymbol{w}}_{k,i-1}\}_{k=1}^N$ evolves according to:

$$\widetilde{\boldsymbol{\mathcal{W}}}_{i-1} = \zeta\widetilde{\boldsymbol{\phi}}_{i-1} + (1-\zeta)\mathcal{A}\widetilde{\boldsymbol{\phi}}_{i-1}, \tag{21}$$

where $\widetilde{\boldsymbol{\phi}}_i = \operatorname{col}\{\widetilde{\boldsymbol{\phi}}_{k,i}\}_{k=1}^N$. By subtracting w_k^o from both sides of (4b) and by adding and subtracting w_k^o to the difference $\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1}$, we can write:

$$\widetilde{\boldsymbol{\phi}}_{k,i} = \widetilde{\boldsymbol{\phi}}_{k,i-1} - \boldsymbol{Q}_k(\widetilde{\boldsymbol{\phi}}_{k,i-1} - \widetilde{\boldsymbol{\psi}}_{k,i}). \tag{22}$$

Let $\boldsymbol{\delta}_{k,i} = \widetilde{\boldsymbol{\phi}}_{k,i-1} - \widetilde{\boldsymbol{\psi}}_{k,i}$. By introducing the quantization error vector:

$$\boldsymbol{z}_{k,i}(\boldsymbol{\delta}_{k,i}) \triangleq \boldsymbol{\delta}_{k,i} - \boldsymbol{Q}_k(\boldsymbol{\delta}_{k,i}), \tag{23}$$

we can write:

$$\widetilde{\boldsymbol{\phi}}_{k,i} = \widetilde{\boldsymbol{\psi}}_{k,i} + \boldsymbol{z}_{k,i}(\boldsymbol{\delta}_{k,i}). \tag{24}$$

By using (16)–(24), we can show that the network error vector $\widetilde{\boldsymbol{\phi}}_i$ evolves according to the following dynamics:

$$\boxed{\widetilde{\boldsymbol{\phi}}_i = \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{\phi}}_{i-1} - \mu\boldsymbol{s}_i + \mu b + \boldsymbol{z}_i} \tag{25}$$

where

$$\boldsymbol{\mathcal{B}}_{i-1} \triangleq (I_M - \mu\boldsymbol{\mathcal{H}}_{i-1})\,\mathcal{A}', \tag{26}$$

$$\mathcal{A}' \triangleq \zeta I_M + (1-\zeta)\mathcal{A}, \tag{27}$$

$$\boldsymbol{z}_i \triangleq \operatorname{col}\left\{\boldsymbol{z}_{k,i}(\boldsymbol{\delta}_{k,i})\right\}_{k=1}^N. \tag{28}$$

For a semi-orthogonal $\mathcal{U}$, the matrix $\mathcal{A}$ satisfying the conditions in (2) has a Jordan decomposition of the form $\mathcal{A} = \mathcal{V}_\epsilon\Lambda_\epsilon\mathcal{V}_\epsilon^{-1}$ with [10, Lemma 2]:

$$\mathcal{V}_\epsilon = \left[\begin{array}{c|c} \mathcal{U} & \mathcal{V}_{R,\epsilon} \end{array}\right],\ \Lambda_\epsilon = \left[\begin{array}{c|c} I_P & 0 \\ \hline 0 & \mathcal{J}_\epsilon \end{array}\right],\ \mathcal{V}_\epsilon^{-1} = \left[\begin{array}{c} \mathcal{U}^\top \\ \hline \mathcal{V}_{L,\epsilon}^\top \end{array}\right], \tag{29}$$

where $\mathcal{J}_\epsilon$ is a Jordan matrix with eigenvalues λ (which may be complex but have magnitude less than one) on the diagonal and $\epsilon > 0$ on the first lower sub-diagonal. Consequently, the matrix $\mathcal{A}'$ in (27) has a Jordan decomposition of the form $\mathcal{A}' = \mathcal{V}_\epsilon\Lambda_\epsilon'\mathcal{V}_\epsilon^{-1}$ where

$$\Lambda_\epsilon' = \left[\begin{array}{c|c} I_P & 0 \\ \hline 0 & \mathcal{J}_\epsilon' \end{array}\right], \quad \text{with } \mathcal{J}_\epsilon' \triangleq \zeta I_{M-P} + (1-\zeta)\mathcal{J}_\epsilon. \tag{30}$$

It can be shown that, for ϵ small enough, $\|\mathcal{J}_\epsilon'\| \in (0,1)$. In fact, the block diagonal matrix $\mathcal{J}_\epsilon'$, which is given by (30) satisfies:

$$\|\mathcal{J}_\epsilon'\|^2 \le (\rho(\mathcal{J}_\epsilon') + (1-\zeta)\epsilon)^2. \tag{31}$$

Using the fact that $\rho(\mathcal{J}_\epsilon) \in (0,1)$ and $\zeta \in (0,1)$, we obtain $\rho(\mathcal{J}_\epsilon') \in (0,1)$. We can also show that $\rho(\mathcal{J}_\epsilon') \le \zeta + (1-\zeta)\rho(\mathcal{J}_\epsilon)$, which implies that $\rho(\mathcal{J}_\epsilon') \in (\zeta, 1)$ and that:

$$\begin{aligned} \|\mathcal{J}_\epsilon'\| &\le \zeta + (1-\zeta)\rho(\mathcal{J}_\epsilon) + (1-\zeta)\epsilon \\ &= 1 - (1-\zeta)(1-\rho(\mathcal{J}_\epsilon) - \epsilon) \end{aligned} \tag{32}$$

By multiplying both sides of (25) by $\mathcal{V}_\epsilon^{-1}$ and by partitioning the transformed iterate $\mathcal{V}_\epsilon^{-1}\widetilde{\boldsymbol{\phi}}_i$ into $\operatorname{col}\{\overline{\boldsymbol{\phi}}_i, \breve{\boldsymbol{\phi}}_i\}$ with $\overline{\boldsymbol{\phi}}_i = \mathcal{U}^\top\widetilde{\boldsymbol{\phi}}_i$ and $\breve{\boldsymbol{\phi}}_i = \mathcal{V}_{L,\epsilon}^\top\widetilde{\boldsymbol{\phi}}_i$, we obtain:

$$\overline{\boldsymbol{\phi}}_i = (I_P - \boldsymbol{\mathcal{D}}_{11,i-1})\overline{\boldsymbol{\phi}}_{i-1} - \boldsymbol{\mathcal{D}}_{12,i-1}\breve{\boldsymbol{\phi}}_{i-1} + \overline{\boldsymbol{z}}_i - \overline{\boldsymbol{s}}_i \tag{33}$$

$$\breve{\boldsymbol{\phi}}_i = (\mathcal{J}_\epsilon' - \boldsymbol{\mathcal{D}}_{22,i-1})\breve{\boldsymbol{\phi}}_{i-1} - \boldsymbol{\mathcal{D}}_{21,i-1}\overline{\boldsymbol{\phi}}_{i-1} + \breve{\boldsymbol{z}}_i + \breve{b} - \breve{\boldsymbol{s}}_i \tag{34}$$

where $\overline{\boldsymbol{s}}_i \triangleq \mu\mathcal{U}^\top\boldsymbol{s}_i$, $\overline{\boldsymbol{z}}_i \triangleq \mathcal{U}^\top\boldsymbol{z}_i$, $\breve{\boldsymbol{s}}_i \triangleq \mu\,\mathcal{V}_{L,\epsilon}^\top\boldsymbol{s}_i$, $\breve{\boldsymbol{z}}_i \triangleq \mathcal{V}_{L,\epsilon}^\top\boldsymbol{z}_i$, $\breve{b} = \mu\mathcal{V}_{L,\epsilon}^\top b$, and:

$$\boldsymbol{\mathcal{D}}_{11,i-1} \triangleq \mu\mathcal{U}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{U}, \quad \boldsymbol{\mathcal{D}}_{12,i-1} \triangleq \mu\mathcal{U}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{V}_{R,\epsilon}\mathcal{J}_\epsilon',$$
$$\boldsymbol{\mathcal{D}}_{21,i-1} \triangleq \mu\mathcal{V}_{L,\epsilon}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{U}, \quad \boldsymbol{\mathcal{D}}_{22,i-1} \triangleq \mu\mathcal{V}_{L,\epsilon}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{V}_{R,\epsilon}\mathcal{J}_\epsilon',$$

and where we used the fact that $\mathcal{U}^\top b = 0$ as shown in [10].

Using similar arguments as in [5, Theorem 9.1] and [10, Theorem 1], we can show that, under Assumptions 1, 2, and 3, when $1-\zeta = O(\mu^\gamma)$ with $\gamma \in (0, 0.5)$, the variances of $\overline{\boldsymbol{\phi}}_i$ and $\breve{\boldsymbol{\phi}}_i$ are coupled and recursively bounded as:

$$\left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\phi}}_i\|^2 \\ \mathbb{E}\|\breve{\boldsymbol{\phi}}_i\|^2 \end{array}\right] \preceq \Gamma \left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\phi}}_{i-1}\|^2 \\ \mathbb{E}\|\breve{\boldsymbol{\phi}}_{i-1}\|^2 \end{array}\right] + \left[\begin{array}{c} e + v_1^2\sigma_z^2 \\ f + v_1^2\sigma_z^2 \end{array}\right] \tag{35}$$

where Γ is a stable matrix given by:

$$\Gamma = \left[\begin{array}{cc} a & b \\ c & d \end{array}\right] \tag{36}$$

with $v_1 = \|\mathcal{V}_\epsilon^{-1}\|$, $a = 1 - O(\mu)$, $b = O(\mu^{2\gamma})$, $c = O(\mu^{2-\gamma})$,

$$\begin{aligned} d &= \|\mathcal{J}_\epsilon'\| + \beta_{z,\max}^2 \cdot O(\mu^{2\gamma}) + O(\mu^{2-\gamma}) = 1 - O(\mu^\gamma) \\ e &= O(\mu^{2-\gamma})\cdot\beta_{z,\max}^2 + O(\mu^2)\cdot\sigma_s^2 \\ f &= O(\mu^{2-\gamma})\cdot\varphi^2 + O(\mu^2)\cdot\sigma_s^2. \end{aligned}$$

The φ^2-term depends on the bias $(\|\breve{b}\|/\mu)^2$ and $\beta_{z,\max}^2$. It follows that,

$$\begin{aligned} &\limsup_{i\to\infty} \left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\phi}}_i\|^2 \\ \mathbb{E}\|\breve{\boldsymbol{\phi}}_i\|^2 \end{array}\right] \preceq \\ &\left[\begin{array}{c} \beta_{z,\max}^2 O(\mu^{1-\gamma}) + \sigma_s^2 O(\mu) + \varphi^2 O(\mu) + \sigma_z^2 O(\mu^{-1}) \\ \beta_{z,\max}^2 O(\mu^{3-3\gamma}) + \sigma_s^2 O(\mu^{2-\gamma}) + \varphi^2 O(\mu^{2(1-\gamma)}) + \sigma_z^2 O(\mu^{-\gamma}) \end{array}\right] \end{aligned} \tag{37}$$

Using (21) and the fact that $\mathcal{A}' = \mathcal{V}_\epsilon\Lambda_\epsilon'\mathcal{V}_\epsilon^{-1}$, we can write:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 \le \limsup_{i\to\infty} \|\mathcal{V}_\epsilon\|^2[\mathbb{E}\|\overline{\boldsymbol{\phi}}_i\|^2 + \mathbb{E}\|\breve{\boldsymbol{\phi}}_i\|^2]. \tag{38}$$

Combining (37) and (38), we obtain (15). The proof of (35)–(37) is omitted due to space limitations. □

Expression (15) reveals the influence of the *step-size* μ, the *relative quantization noise* term (captured by $\beta_{z,\max}^2$), the *absolute quantization noise* term (captured by σ_z^2), and

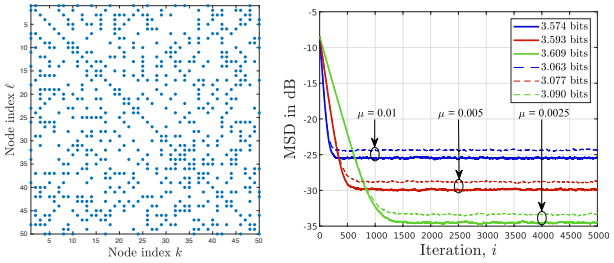

Fig. 1. *(Left)* Link matrix. *(Right)* Performance of (4) when quantizers (8) are used with $\beta_{z,k}^2 = 0$ (solid curve) and $\beta_{z,k}^2 \neq 0$ (dashed curve).

the *gradient noise* term (captured by σ_s^2), on the steady-state mean-square error. Several conclusions can be drawn from our analysis. For instance, if the quantizers are designed such that $\beta_{z,k}^2 = O(\mu^\gamma)$ and $\sigma_{z,k}^2 = O(\mu^2)$, we obtain $\limsup_{i\to\infty} \mathbb{E}\|w_k^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu)$. This means that, compared with (1), approach (4) can reduce the communication cost whilst still ensuring small estimation errors on the order of μ. Furthermore, in the absence of the *absolute noise*, i.e., when $\sigma_z^2 = 0$, small estimation errors of $O(\mu)$ can also be ensured by choosing γ small enough.

IV. SIMULATION RESULTS

We apply strategy (4) to a network of $N = 50$ nodes, generated randomly with the link matrix shown in Fig. 1 (left). 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 $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}^\top w_k^\star + \boldsymbol{v}_k(i)$ for some unknown $L \times 1$ vector $w_k^\star$ with $\boldsymbol{v}_k(i)$ denoting a zero-mean measurement noise and $L = 5$. The processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ are generated using similar settings as in [11, Sec. IV]. The signal $\mathcal{w}^\star = \text{col}\{w_1^\star, \ldots, w_N^\star\}$ is generated by smoothing a signal $\mathcal{w}_o$ with $\tau = 2$ and $\mathcal{w}_o$ randomly generated from the Gaussian distribution $\mathcal{N}(0.2 \times \mathbb{1}_{NL}, I_{NL})$–see [11, Sec. IV]. The matrix $\mathcal{U}$ is generated according to $\mathcal{U} = U \otimes I_L$ (U is chosen as the first two eigenvectors of the graph Laplacian). We use randomized quantizers of the form (8). We set the *absolute noise* parameter $\sigma_{z,k}^2 = \mu^2\ \forall k$, and $1-\zeta = \mu^{\frac{1}{45}}$. We report the network MSD learning curves $\frac{1}{N}\sum_{k=1}^N \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ in Fig. 1 (right) for 3 different values of the step-size μ. The results are averaged over 100 Monte-Carlo runs. Solid curves refer to the case $\beta_{z,k}^2 = 0$, whereas dashed curves refer to the case where $\beta_{z,k}^2 = C_\beta \cdot \mu^\gamma$, with the constant C_β chosen such that $\beta_{z,k}^2 = 0.25$ when $\mu = 0.01$. In each case, we report the average number of bits per agent/dimension/iteration. By comparing solid and dashed curves, we observe that, for a fixed value of $\sigma_{z,k}^2$, setting the compression parameter $\beta_{z,k}^2$ to a non-zero value allows for reducing the number of bits, at the expense of an increase in the steady-state MSD.

V. REFERENCES

[1] 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.

[2] 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.

[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] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, vol. 30.

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

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

[7] 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.

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

[10] 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.

[11] R. Nassif, S. Vlaski, and A. H. Sayed, "Adaptation and learning over networks under subspace constraints–Part II: Performance analysis," *IEEE Trans. Signal Process.*, vol. 68, pp. 2948–2962, 2020.

[12] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. IEEE Conf. Decis. Control*, Cancun, Mexico, Dec. 2008, pp. 4177–4184.

[13] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, 2012.

[14] D. Thanou, E. Kokiopoulou, Y. Pu, and P. Frossard, "Distributed average consensus with quantization refinement," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 194–205, 2013.

[15] M. Carpentiero, V. Matta, and A. H. Sayed, "Distributed adaptive learning under communication constraints," Available as arXiv:2112.02129, Dec. 2021.

[16] M. Carpentiero, V. Matta, and A. H. Sayed, "Adaptive diffusion with compressed communication," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2022, pp. 1–5.

[17] 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.

[18] H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized decentralized stochastic learning over directed graphs," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2020, vol. 119, pp. 9324–9333.

[19] C.-S. Lee, G. Scutari, and N. Michelusi, "Finite-bit quantization for distributed algorithms with linear convergence," Available as arXiv:2107.11304v1, Jul. 2021.