SPARQ-SGD: Event-Triggered and Compressed Communication in Decentralized Optimization

Navjot Singh[†], Deepesh Data[†], Jemin George[‡], and Suhas Diggavi[†]

***Abstract*— In this paper, we propose and analyze SPARQ-SGD, an event-triggered and compressed algorithm for decentralized training of large-scale machine learning models over a graph. Each node can locally compute a condition (event) which triggers a communication where quantized and sparsified local model parameters are sent. In SPARQ-SGD, each node first takes a fixed number of local gradient steps and then checks if the model parameters have significantly changed compared to its last update; it communicates further compressed model parameters only when there is a significant change, as specified by a (design) criterion. We prove that SPARQ-SGD converges as $O(\frac{1}{nT})$ and $O(\frac{1}{\sqrt{nT}})$ in the strongly-convex and non-convex settings, respectively, demonstrating that aggressive compression, including event-triggered communication, model sparsification and quantization does not affect the overall convergence rate compared to uncompressed decentralized training; thereby theoretically yielding communication efficiency for 'free'. We evaluate SPARQ-SGD over real datasets to demonstrate significant savings in communication bits over the state-of-the-art.**

I. Introduction

There has been a recent interest in communication efficient *decentralized* training of large-scale machine learning models *e.g.,* [1]–[3]. In decentralized training, the nodes do not have a central coordinator, and are not directly connected to all other nodes, but are connected through a communication graph. This implies that the communication is inherently more efficient, as the local connection (degree) of such graphs could be a small constant, independent of the network size. In this paper, we propose SPARQ-SGD[1] to improve communication efficiency of decentralized training through *event-driven* exchange of quantized and sparsified model parameters between the nodes.

Over the past few years, a number of different methods have been developed to achieve communication efficiency in *distributed* SGD, where there exists a central coordinator. These can be broadly divided into 2 categories. In the first category, to reduce communication, workers send *compressed* updates either with sparsification [4]–[8] or quantization [9]–[12] or a combination of both [13].[2] Another class of algorithms that are based on the idea of *infrequent communication*, workers do not communicate in each iteration; rather, they send the updates after performing a *fixed* number of local gradient steps [13]–[16]. The idea of compressed communication, using quantization or sparsification, has been extended to the setting of *decentralized* optimization [2], [3], [17].

In this paper, we propose SPARQ-SGD with *event-triggered* communication, where a node initiates a (communication) action regulated by a locally computable triggering condition (event), thereby further reducing the communication among nodes. In particular, the proposed triggering condition is such that at least a fixed number of local gradient steps or iterations (say, H local iterations) are first completed and after that the condition checks if there is a significant change (beyond a certain threshold) in its local model parameter vector since the last time communication occurred. Only if the change in model parameter exceeds the prescribed threshold, does a node trigger compressed communication. As far as we know, such an idea of event-triggered and compressed communication has not been proposed and analyzed in the context of decentralized (stochastic) training of large-scale machine learning models.

As mentioned earlier, in addition to event-triggered communication, we also incorporate compression of the model parameters, when a node communicates; *i.e.,* when a node communicates its model parameters, it sends a quantized and sparsified version of the model parameters. We therefore combine the recent ideas applied to communication efficient training (quantization and sparsification) with our event-triggered communication to propose SPARQ-SGD[3]; see Algorithm 1. We analyze the performance of our algorithm for both convex and (smooth) non-convex objective functions, in terms of its convergence rate as a function of the number of iterations T (and also the number of communication rounds) and the amount of communication bits exchanged to learn a model to a certain accuracy. We prove that the SPARQ-SGD converges as $O(\frac{1}{nT})$ and $O(\frac{1}{\sqrt{nT}})$ in strongly-convex and non-convex settings, respectively, demonstrating that such aggressive compression, including event-triggered communication does not affect the overall convergence rate as compared to a uncompressed decentralized training [1]. Moreover, we show that SPARQ-SGD yields significant

[†]Department of Electrical and Computer Engineering, University of California, Los Angeles, USA Email: {navjotsingh, suhasdiggavi}@ucla.edu, deepesh.data@gmail.com

[‡]US Army Research Lab, Maryland, USA; Email: jemin.george.civ@mail.mil

[1]Acronym stands for SParsified Action Regulated Quantized SGD.

[2]In sparsification, the vector sparsification is done by selecting either its top k entries (in terms of the absolute value) or random k entries, where k is less than the dimension of the vector. Quantization consists of discretization of the vector by rounding off its entries either randomly or deterministically (in the extreme case, this can be just the sign operator).

[3]The idea of combining compression and *fixed* number of local iterations has been carried out in a *distributed* setting (the master-worker architecture) in [13]. In this work, in addition to *extending* this combination to the *decentralized* setting, we also propose and analyze event-triggered communication.

amount of saving in communication over the state-of-the-art; see Section V for more details.

Related work. In decentralized setting, [2], [18], propose unbiased stochastic compression for gradient exchange. [19], [20] analyze Stochastic Gradient Push algorithm for non-convex objectives which approximates distributed averaging instead of compressing the gradients. Our work most closely relates to [3] which proposed CHOCO-SGD, which uses compressed (sparsified *or* quantized) updates; the distinction is that we propose an *event-triggered* communication where sparsified *and* quantized model parameters are transmitted only if they have changed significantly after performing some fixed number of local iterations, further reducing communication. The idea of event-triggered communication has been explored previously in the control community [21]–[24], [25] and in optimization literature [26]–[28]. These papers focus on continuous-time, deterministic optimization algorithms for convex problems; in contrast, we propose event-driven *stochastic* gradient descent algorithms for both convex and non-convex problems. [29] propose an adaptive scheme to skip gradient computations in a *distributed* setting for *deterministic* gradients; moreover, their focus is on saving communication rounds, and do not have any compressed communication. Sub-gradient descent with quantization for *deterministic* decentralized optimization has been studied in [30] and [31] for convex objectives only, with the former showing convergence only within a neighborhood of the optimum and the latter employing an adaptive quantization scheme to recover rates attained by un-quantized schemes. Decentralized consensus with quantization over time varying topology has been analyzed in [32]. [33] considers inexact proximal gradient with quantization in decentralized optimization for strongly convex objectives, showing convergence to the global optimum. As far as we know, ours is the first paper which uses event-triggered (incorporating infrequent communication) and compressed communication for decentralized *stochastic* optimization of both strongly convex and non-convex objectives.

Contributions. We study optimization in a decentralized setup, where n different workers, each having a different dataset $\mathcal{D}_i$ (the dataset $\mathcal{D}_i$ has an associated objective function $f_i : \mathbb{R}^d \to \mathbb{R}$), are linked through a connected graph $\mathcal{G} = ([n], \mathcal{E})$, where $[n] := \{1, 2, \ldots, n\}$. Vertex i in $\mathcal{G}$ is associated with the ith worker who can only communicate with its neighbors $\mathcal{N}_i = \{j \in [n] : \{i, j\} \in \mathcal{E}\}$. We consider the empirical risk minimization of the loss function:

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where $f_i(\mathbf{x}) = \mathbb{E}_{\xi_i \in_U \mathcal{D}_i}[F_i(\mathbf{x}, \xi_i)]$, where the notation $\xi_i \in_U \mathcal{D}_i$ denotes that ξ_i is a uniformly random data sample from the dataset $\mathcal{D}_i$ and $F_i(\mathbf{x}, \xi_i)$ denotes the risk associated with the data sample ξ_i with respect to (w.r.t.) $\mathbf{x}$ at the ith worker node. We solve the decentralized optimization in (1) using SPARQ-SGD. Our theoretical results are the convergence analyses for both strongly convex and non-convex objectives in the synchronous setting; see Theorem 1 and 2, respectively. In the strongly-convex setting, we show a convergence rate of $\mathcal{O}\left(\frac{1}{nT}\right) + \mathcal{O}\left(\frac{c_0}{\delta^2 T^{(1+\epsilon)}}\right) + \mathcal{O}\left(\frac{H^2}{\delta^4 \omega^2 T^2}\right) + \mathcal{O}\left(\frac{H^3}{\omega^3 \delta^6 T^3}\right)$ for some $\epsilon \in (0, 1)$, the factors (c_0 for triggering threshold, H for number of local iterations, and ω for compression) for communication efficiency, and δ, the spectral-gap of the connectivity matrix W, appear in the higher order terms. Thus, for large enough T, they do not affect the dominating term $\mathcal{O}\left(\frac{1}{nT}\right)$, which, in fact, is the convergence rate of centralized vanilla SGD with mini-batch size of n. Similar observation is also made in the non-convex setting, where we get a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{nT}})$; see Corollary 1 and 2 and the following remarks for more details. Hence, for both the objectives, we get essentially the same convergence rate as that of vanilla SGD, even after applying SPARQ-SGD to gain communication efficiency; and hence, we get communication efficiency essentially "for free". We compare our algorithm against CHOCO-SGD [17], which is the state-of-the-art in compressed decentralized training and provide theoretical justification for communication efficiency of SPARQ-SGD over CHOCO-SGD to achieve the same target accuracy. We corroborate our theoretical understanding with numerical results in Section V where we demonstrate that SPARQ-SGD yields significant savings in communication bits. For a convex objective simulated on the MNIST dataset, SPARQ-SGD saves total communicated bits by a factor of $15\times$ compared to CHOCO-SGD [3] and by $1000\times$ compared to vanilla SGD to converge to the same target accuracy. Similarly, for a non-convex objective simulated on the CIFAR-10 dataset [34], we save total bits by a factor of $40\times$ compared to CHOCO-SGD [17] and around $3\text{K}\times$ compared to vanilla SGD to reach the same target accuracy.

Paper organization. We describe SPARQ-SGD, our proposed algorithm, in Section II. In Section III, we state our main results for strongly-convex and non-convex objectives, and give proof outlines of these theorems in Section IV. We validate our theoretical findings with numerical experiments in Section V. The complete proofs for our theorems can be found in the full paper [35].

II. Our Algorithm: SPARQ-SGD

In this section, we describe SPARQ-SGD, our decentralized SGD algorithm with compression and event-triggered communication. First we need to define its main ingredients.

Definition 1 (Compression, [7])**.** *A (possibly randomized) function $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is called a* compression *operator, if there exists a constant $\omega \in (0, 1]$, such that the following holds for every $\mathbf{x} \in \mathbb{R}^d$:*

$$\mathbb{E}_{\mathcal{C}}[\|\mathbf{x} - \mathcal{C}(\mathbf{x})\|_2^2] \leq (1 - \omega)\|\mathbf{x}\|_2^2, \tag{2}$$

where expectation is taken over the randomness of $\mathcal{C}$. We assume $\mathcal{C}(\mathbf{0}) = \mathbf{0}$.

It is known that some important sparsifiers as well as quantizers are examples of compression operators: (i)

Top_k and $Rand_k$ sparsifiers (in which we select k entries; see Footnote 2) with $\omega = k/d$ [7], (ii) Stochastic quantizer Q_s from [9][4] with $\omega = (1 - \beta_{d,s})$ for $\beta_{d,s} < 1$, and (iii) Deterministic quantizer $\frac{\|\mathbf{x}\|_1}{d} Sign(\mathbf{x})$ from [12] with $\omega = \frac{\|\mathbf{x}\|_1^2}{d\|\mathbf{x}\|_2^2}$. It was shown in [13] that if we compose these sparsifiers and quantizers, the resulting operator also gives compression and outperforms their individual components. For example, for any $Comp_k \in \{Top_k, Rand_k\}$, the following are compression operators: (iv) $\frac{1}{(1+\beta_{k,s})} Q_s(Comp_k)$ with $\omega = \left(1 - \frac{k}{d(1+\beta_{k,s})}\right)$ for any $\beta_{k,s} \geq 0$, and (v) $\frac{\|Comp_k(\mathbf{x})\|_1 SignComp_k(\mathbf{x})}{k}$ with $\omega = \max\left\{\frac{1}{d}, \frac{k}{d}\left(\frac{\|Comp_k(\mathbf{x})\|_1^2}{d\|Comp_k(\mathbf{x})\|_2^2}\right)\right\}$.

Event-triggered communication. As mentioned in Section I, our proposed event-triggered communication consists of two phases: in the first phase, nodes perform a fixed number H of local iterations, and in the second phase, they check for the communication-triggering condition (event), if satisfied, then they send the (compressed) updates. Let $\mathcal{I}_T \subseteq [T]$ denote a set of indices at which workers check for the triggering condition. Since we are in the synchronous setting, we assume that $\mathcal{I}_T$ is same for all workers. Let $\mathcal{I}_T = \{I_{(1)}, I_{(2)}, \ldots, I_{(k)}\}$. The gap of $\mathcal{I}_T$ is defined as $gap(\mathcal{I}_T) := \max_{i \in [k-1]}\{(I_{(i+1)} - I_{(i)})\}$, [14], which is equal to the maximum number of local iterations a worker performs before checking for the triggering condition. Note that $gap(\mathcal{I}_T) = 1$ is equivalent to the case when workers check for the triggering condition in every iteration.

Our algorithm, SPARQ-SGD, for optimizing (1) in a decentralized setting is presented in Algorithm 1. For designing this, in addition to combining sparsification *and* quantization, we carefully incorporate local iterations and event-triggered[5] communication into the CHOCO-SGD algorithm [3], which uses only sparsified *or* quantized updates. This poses several technical challenges in proving the convergence; see proofs of Theorems 1, 2, and in particular, the proof of Lemma 1.

In SPARQ-SGD, each node $i \in [n]$ maintains a local parameter vector $\mathbf{x}_i^{(t)}$, and their goal is to achieve consensus among themselves on the value of $\mathbf{x}$ that minimizes (1), while allowing only for compressed and infrequent communication. Node i updates $\mathbf{x}_i^{(t)}$ in each iteration t by a stochastic gradient step (line 4). An estimate $\hat{\mathbf{x}}_i^{(t)}$ of $\mathbf{x}_i^{(t)}$ is also maintained at each neighbor $j \in \mathcal{N}_i$ and at i itself. Thus, each node maintains an estimate of all its neighbors' local parameter vectors and of itself. In our algorithm, $\mathcal{I}_T$ is the set of indices for which the workers check for the triggering condition and take a consensus step. We also allow the triggering threshold c_t to vary with t with the requirement that $c_t = o(t)$. At time-step t, if $(t+1) \in \mathcal{I}_T$, the nodes check for the triggering condition (line 7), if satisfied, then each node $i \in [n]$ sends to all its neighbors, the compressed difference between its local parameter vector and its estimate that its neighbors have (line 8). If this condition is not satisfied, then the node does not communicate (written as 'Send **0**') (line 11). Then, based on the messages received from its neighbors, the ith node updates $\hat{\mathbf{x}}_j^{(t)}$ – the estimate of the jth node's local parameter vector (line 13), and then every node performs the consensus step (line 15).

In SPARQ-SGD, observe that every worker node initializes its estimate $\hat{\mathbf{x}}_i^{(0)}$ of the ith node's local parameter vector $\mathbf{x}_i^{(0)}$ to be $\hat{\mathbf{x}}_i^{(0)} := 0$, whereas, in principle, it should have been equal to $\mathbf{x}_i^{(0)}$. To ensure this, in the first round of our algorithm, every worker sends its (compressed) local parameter vector to all its neighbors.

Algorithm 1 SPARQ-SGD: SParsified Action Regulated Quantized SGD

1: Initial values $\mathbf{x}_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$, consensus stepsize γ, SGD stepsizes $\{\eta_t\}_{t \geq 0}$, threshold sequence $\{c_t\}_{t \geq 0}$, compression operator $\mathcal{C}$ having parameter ω, communication graph $G = ([n], E)$ and mixing matrix W, set of synchronization indices $\mathcal{I}_T$, initialize $\hat{\mathbf{x}}_i^{(0)} := 0$ for all i
2: **for** $t = 0$ **to** $T-1$ in parallel for all workers $i \in [n]$ **do**
3: Sample $\xi_i^{(t)}$ and compute stochastic gradient $\mathbf{g}_i^{(t)} := \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
4: $\mathbf{x}_i^{(t+\frac{1}{2})} := \mathbf{x}_i^{(t)} - \eta_t \mathbf{g}_i^{(t)}$
5: **if** $(t+1) \in I_T$ **then**
6: **for** neighbors $j \in \mathcal{N}_i \cup \{i\}$ **do**
7: **if** $\|\mathbf{x}_i^{(t+\frac{1}{2})} - \hat{\mathbf{x}}_i^{(t)}\|_2^2 > c_t \eta_t^2$ **then**
8: Compute $\mathbf{q}_i^{(t)} := \mathcal{C}(\mathbf{x}_i^{(t+\frac{1}{2})} - \hat{\mathbf{x}}_i^{(t)})$
9: Send $\mathbf{q}_i^{(t)}$ and receive $\mathbf{q}_j^{(t)}$
10: **else**
11: Send $\mathbf{0}$ and receive $\mathbf{q}_j^{(t)}$
12: **end if**
13: $\hat{\mathbf{x}}_j^{(t+1)} := \mathbf{q}_j^{(t)} + \hat{\mathbf{x}}_j^{(t)}$
14: **end for**
15: $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+\frac{1}{2})} + \gamma \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{\mathbf{x}}_j^{(t+1)} - \hat{\mathbf{x}}_i^{(t+1)})$
16: **else**
17: $\hat{\mathbf{x}}_i^{(t+1)} = \hat{\mathbf{x}}_i^{(t)}$, $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+\frac{1}{2})}$ for all $i \in [n]$
18: **end if**
19: **end for**

III. Main Results

Our main results are under the following assumptions:

Assumptions. (i) **L-Smoothness:** Each local function f_i for $i \in [n]$ is L-smooth, i.e, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have $f_i(\mathbf{y}) \leq f_i(\mathbf{x}) + \langle \nabla f_i(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2$. (ii) **Bounded variance:** For every $i \in [n]$, we have $\mathbb{E}_{\xi_i}\|\nabla F_i(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma_i^2$, for some finite σ_i, where $\nabla F_i(\mathbf{x}, \xi_i)$ is the unbiased gradient at worker i such that $\mathbb{E}_{\xi_i}[\nabla F_i(\mathbf{x}, \xi_i)] = \nabla f_i(\mathbf{x})$. We define the average variance across all workers as $\bar{\sigma}^2 := \frac{1}{n}\sum_{i=1}^n \sigma_i^2$.

[4] $Q_s : \mathbb{R}^d \to \mathbb{R}^d$ is a stochastic quantizer, if for every $\mathbf{x} \in \mathbb{R}^d$, we have (i) $\mathbb{E}[Q_s(\mathbf{x})] = \mathbf{x}$ and (ii) $\mathbb{E}[\|\mathbf{x} - Q_s(\mathbf{x})\|_2^2] \leq \beta_{d,s}\|\mathbf{x}\|_2^2$. Q_s from [9] satisfies this definition with $\beta_{d,s} = \min\left\{\frac{d}{s^2}, \frac{\sqrt{d}}{s}\right\}$.

[5] The Zeno phenomenon [21] does not occur in our setup as we have a discrete sampling period as well as a fixed number of local iterations, giving a lower bound to the event intervals of atleast H times the sampling period.

(iii) **Bounded second moment:** For every $i \in [n]$, we have $\mathbb{E}_{\xi_i}\|\nabla F_i(\mathbf{x}, \xi_i)\|^2 \leq G^2$, for some finite G.[6]

Let $W \in \mathbb{R}^{n\times n}$ denote the weighted connectivity matrix of our underlying graph $\mathcal{G}$, with w_{ij} for every $i, j \in [n]$ being its (i, j)th entry, which denotes the weight on the link between worker i and j. W is assumed to be symmetric and doubly stochastic, which implies that all its eigenvalues $\lambda_i(W), i = 1, 2, \ldots, n$, lie in $[-1, 1]$. Without loss of generality, assume that $|\lambda_1(W)| > |\lambda_2(W)| \geq \ldots \geq |\lambda_n(W)|$. Since W is doubly stochastic, we have $\lambda_1(W) = 1$, and since $\mathcal{G}$ is connected, we have $\lambda_2(W) < \lambda_1(W)$. Let the spectral gap of W be defined as $\delta := 1-|\lambda_2(W)|$. Since $|\lambda_2(W)| \in [0, 1)$ we have $\delta \in (0, 1]$. It is known that simple matrices W with $\delta > 0$ exist for every connected graph, [3].

Now we state the main results of this paper both for strongly-convex and non-convex objectives.

Theorem 1 (Smooth and strongly-convex objective with decaying learning rate)**.** *Suppose f_i, for all $i \in [n]$ is L-smooth and μ-strongly convex. Let $\mathcal{C}$ be a compression operator with parameter equal to $\omega \in (0, 1]$. Let $\mathcal{I}_T = \{I_{(1)}, I_{(2)}, \ldots, I_{(k)}\}$ and $H = \max_{i\in[k-1]}\{(I_{(i+1)} - I_{(i)})\}$. If we run SPARQ-SGD with consensus step-size $\gamma = \frac{2\delta\omega}{64\delta+\delta^2+16\beta^2+8\delta\beta^2-16\delta\omega}$, (where $\beta = max_i\{1-\lambda_i(W)\}$), an increasing threshold function $c_t \leq c_0 t^{(1-\epsilon)}$ for all t where constant $c_0 \geq 0$ and $\epsilon \in (0, 1)$ and decaying learning rate $\eta_t = \frac{8}{\mu(a+t)}$, where $a \geq \max\{\frac{5H}{p}, \frac{32L}{\mu}\}$ for $p = \frac{\gamma\delta}{8}$, and let the algorithm generate $\{\mathbf{x}_i^{(t)}\}_{t=0}^{T-1}$ for $i \in [n]$, then*

$$\mathbb{E}f(\mathbf{x}_{avg}^{(T)}) - f^* \leq \frac{\mu a^3}{8S_T}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|^2 + \frac{4T(T+2a)}{\mu S_T}\frac{\bar{\sigma}^2}{n} + \frac{Z_1 T}{\mu^2 S_T}(2L+\mu)\frac{G^2H^2}{p^2} + \frac{Z_2 c_0 \omega T^{(2-\epsilon)}}{\mu^2(2-\epsilon)S_T}\left(\frac{2L+\mu}{p}\right)$$

where $\bar{\mathbf{x}}_{avg}^{(T)} = \frac{1}{S_T}\sum_{t=0}^{T-1} w_t \bar{\mathbf{x}}^{(t)}$, where $\bar{\mathbf{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^{(t)}$, weights $w_t = (a+t)^2$, and $S_T = \sum_{t=0}^{T-1} w_t \geq \frac{1}{3}T^3$ and Z_1, Z_2 are universal constants.

The analysis provided also works for any $c_t = o(t)$, however we provide it for $c_t \leq c_0 t^{(1-\epsilon)}$ to highlight the main idea. The consensus step-size γ does not appear explicitly in the above rate expression, but affects the convergence indirectly through $p = \gamma\delta/8$. Note that $\delta \in (0, 1]$, $\beta \leq 2$, and $\omega \geq 0$. Substituting these in the expression of γ and p gives $\gamma \geq \frac{2\delta\omega}{161}$ and $p \geq \frac{\delta^2\omega}{644}$. Now we simplify the above expression to gain further insights as to how our techniques for reducing communication affect the convergence rate.

Corollary 1. *Using $\mathbb{E}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 \leq \frac{4G^2}{\mu^2}$ (from [36, Lemma 2]) and $p \geq \frac{\delta^2\omega}{644}$, hiding constants (including L) in $\mathcal{O}$ notation, the rate expression in Theorem 1 is simplified as:*

$$\mathbb{E}[f(\bar{\mathbf{x}}_{avg}^{(T)})] - f^* \leq \mathcal{O}\left(\frac{\bar{\sigma}^2}{\mu n T}\right) + \mathcal{O}\left(\frac{c_0}{\mu^2\delta^2 T^{(1+\epsilon)}}\right) + \mathcal{O}\left(\frac{G^2H^2}{\mu^2\delta^4\omega^2T^2}\right) + \mathcal{O}\left(\frac{G^2H^3}{\mu\omega^3\delta^6T^3}\right)$$

[6]Bounded second moment is a standard assumption in stochastic optimization with *compressed* communication [7], [8].

Remark 1. *Observe that the dominating term $\mathcal{O}\left(\frac{\bar{\sigma}^2}{\mu n T}\right)$ is not affected by the compression factor ω, the number of local iterations H, the factor c_0 in the triggering condition, and the topology of the underlying communication graph (which is controlled by the spectral gap δ) – they all appear in the higher order terms (note that $\epsilon > 0$). In order to ensure that they do not affect the dominating term while converging at a rate of $\mathcal{O}\left(\frac{\bar{\sigma}^2}{\mu n T}\right)$, we would require $T \geq T_0 := C \times \max\left\{\left(\frac{nc_0}{\mu\delta^2\bar{\sigma}^2}\right)^{\frac{1}{\epsilon}}, \left(\frac{nH^2G^2}{\mu\bar{\sigma}^2\delta^4\omega^2}\right)\right\}$ for sufficiently large constant C. Thus, for large enough T, we get benefits of all these techniques in saving communication bits, without affecting the convergence rate significantly.*

Now we analyze the effect of ω, H, c_0, δ on the threshold T_0: (i) *if we compress the communication more, i.e., smaller ω, then T_0 increases, as expected;* (ii) *if we take more number of local iterations H, T_0 would again increase, as expected, because increasing H means communicating less frequently;* (iii) *if we increase c_0, which means that triggering threshold has become bigger, we expect less frequent communication, thus T_0 increases, as expected;* (iv) *if the spectral gap $\delta \in (0, 1]$ is closer to 1, implying that the graph is well-connected, then the threshold T_0 decreases, which is expected, as good connectivity results in faster consensus.*

Remark 2. *Observe that after a large enough $T \geq T_0$, we get the same rate as that of distributed vanilla SGD and also a distributed gain of n with the number of nodes (workers). Thus, we essentially converge at the same rate as that of vanilla SGD, while significantly saving in terms of communication bits among all the workers; refer numerical results in Section V.*

Theorem 2 (Smooth and non-convex objective with fixed learning rate)**.** *Suppose f_i, for all $i \in [n]$ be L-smooth. Let $\mathcal{C}$ be a compression operator with parameter equal to $\omega \in (0, 1]$. Let $\mathcal{I}_T = \{I_{(1)}, I_{(2)}, \ldots, I_{(k)}\}$ and $H = \max_{i\in[k-1]}\{(I_{(i+1)} - I_{(i)})\}$. If we run SPARQ-SGD for $T \geq 64nL^2$ iterations with fixed learning rate $\eta = \sqrt{\frac{n}{T}}$, an increasing threshold function c_t such that $c_t < \frac{1}{\eta}$ for all t and consensus step-size $\gamma = \frac{2\delta\omega}{64\delta+\delta^2+16\beta^2+8\delta\beta^2-16\delta\omega}$, (where $\beta = \max_i\{1 - \lambda_i(W)\}$), and let the algorithm generate $\{\mathbf{x}_i^{(t)}\}_{t=0}^{T-1}$ for $i \in [n]$, then the averaged iterates $\bar{\mathbf{x}}^{(t)} := \frac{1}{n}\sum_{i=0}^n \mathbf{x}_i^{(t)}$ satisfy:*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{(t)})\|_2^2}{T} \leq \frac{4\left(f(\bar{\mathbf{x}}_0) - f^* + L\bar{\sigma}^2\right)}{\sqrt{nT}} + \frac{\tilde{Z}_1 G^2H^2L^2n}{Tp^2}\left(1+\frac{2p}{\omega}\right) + \frac{\tilde{Z}_2L^2\omega\sqrt{n^{(1+\epsilon)}}}{p\sqrt{T^{(1+\epsilon)}}} + \frac{\tilde{Z}_3G^2H^2L^3n^{3/2}}{T^{3/2}p^2}\left(1+\frac{2p}{\omega}\right) + \frac{\tilde{Z}_4L^3\omega\sqrt{n^{(2+\epsilon)}}}{p\sqrt{T^{(2+\epsilon)}}}$$

Here $p = \frac{\gamma\delta}{8}$, $c_t \leq \frac{1}{\eta^{(1-\epsilon)}}$ for all t where $\epsilon \in (0, 1)$ and $\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3$ and $\tilde{Z}_4$ are universal constants.

Though the consensus step-size γ does not appear in the rate expression, it affects it through the parameter p. As

argued after Theorem 1, we can show that $p \geq \frac{\delta^2\omega}{644}$.

Corollary 2. *Let $f(\bar{\mathbf{x}}^{(0)}) - f^* \leq J^2$, where $J^2 < \infty$ is a constant. Using $p \geq \frac{\delta^2\omega}{644}$, and hiding constants (including L) in the $\mathcal{O}$ notation, we can simplify the rate expression in Theorem 2 to the following:*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{(t)})\|_2^2}{T} \leq \mathcal{O}\left(\left(\frac{n}{T}\right)^{\frac{1+\epsilon}{2}}\left(1+\sqrt{\frac{n}{T}}\right)\left[\frac{1}{\delta^2}\right]\right) + \mathcal{O}\left(\left(\frac{n}{T}+\left(\frac{n}{T}\right)^{3/2}\right)\left[\frac{(1+\delta^2)G^2H^2}{\omega^2\delta^4}\right]\right) + \mathcal{O}\left(\frac{J^2+\bar{\sigma}^2}{\sqrt{nT}}\right)$$

Remark 3. *Observe that ω, H, δ do not affect the dominating term $\mathcal{O}\left(\frac{J^2+\bar{\sigma}^2}{\sqrt{nT}}\right)$. Since Theorem 2 provides non-asymptotic guarantee, we need to decide the horizon T before running the algorithm; so, to ensure that the dominating term does not get affected by these different factors, while converging at a rate of $\mathcal{O}\left(\frac{J^2+\bar{\sigma}^2}{\sqrt{nT}}\right)$, we would be required to fix $T \geq T_1 := C_1 \times \max\left\{\left(\frac{n^{(2+\epsilon)}}{(J^2+\bar{\sigma}^2)^2\delta^4}\right)^{1/\epsilon}, \frac{n^3G^4H^4}{(J^2+\bar{\sigma}^2)^2\omega^4\delta^4}\right\}$ for sufficiently large constant C_1. This implies that for large enough T, we get the benefits of all these techniques in saving on the communication bits, essentially for "free", without affecting the convergence rate by too much. The rest of Remark 1 and Remark 2 are also applicable here.*

Theoretical justification for communication gain. The convergence result for SPARQ-SGD highlights savings in communication compared to CHOCO-SGD [3]. For the sake of argument, consider the case when SPARQ-SGD only performs local iterations and no threshold based triggering ($c_t = 0, \forall t$) . For the same compression operator ω used for both SPARQ and CHOCO, to transmit the same number of bits (i.e., having same number of communication rounds), T iterations of CHOCO would correspond to $T \times H$ iterations of SPARQ (due to H local SGD steps). Thus for the same number of bits transmitted, the bound on sub-optimality for convex objective for CHOCO is $\mathcal{O}(1/\mu nT) + \mathcal{O}(G^2/\omega^2\delta^4\mu^2T^2)$ while for SPARQ it is $\mathcal{O}(1/\mu nHT) + \mathcal{O}(G^2/\omega^2\delta^4\mu^2T^2)$. Thus for the same amount of communication, SPARQ-SGD has a better performance compared to CHOCO-SGD (the first dominant term is affected by H). Similarly, for the same number of communication rounds, the bound on sub-optimality for CHOCO-SGD for non-convex objectives is $\mathcal{O}(1/\sqrt{T}) + \mathcal{O}(1/T)$ while for SPARQ-SGD it is $\mathcal{O}(1/\sqrt{HT}) + \mathcal{O}(H/T)$. Thus, it can be seen that for large values of T, the performance of SPARQ-SGD is better than that of CHOCO-SGD for the number of communicated bits. Thus there is theoretical justification for our algorithm to have a better performance while using less bits for communication and this claim is also supported through our experiments.

IV. Proof Outlines

In this section, we give proof outlines of Theorem 1 and 2 and provide complete proofs in the full paper [35]. Our proof outlines have been adapted from [3], [17], with significant changes in the proof details arising due to event-triggered communication.

Since we are in a decentralized setting, in order to do a global optimization, i.e., optimizing (1), workers will have to reach to a consensus. That is in fact a main ingredient in our convergence analyses; see Lemma 1 and Lemma 2, and also Remark 5.

A. Proof Outline of Theorem 1

Consider the collection of iterates $\{\mathbf{x}_i^{(t)}\}_{t=0}^{T-1}$, $i \in [n]$ generated by Algorithm 1 at time t. For any time $t \geq 0$, we have from line 15 of Algorithm 1 that

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+\frac{1}{2})} + \mathbb{1}_{(t+1)\in\mathcal{I}_T}\gamma\sum_{j=1}^n w_{ij}(\hat{\mathbf{x}}_j^{(t+1)} - \hat{\mathbf{x}}_i^{(t+1)}),$$

where $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta_t\nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$ (line 4). Note that we changed the summation from $j \in \mathcal{N}_i$ to $j = 1$ to n; this is because $w_{ij} = 0$ whenever $j \notin \mathcal{N}_i$.

Let $\bar{\mathbf{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^{(t)}$ denote the average of the local iterates at time t. Now we argue that $\bar{\mathbf{x}}^{(t+1)} = \bar{\mathbf{x}}^{(t+\frac{1}{2})}$. This trivially holds when $(t+1) \notin \mathcal{I}_T$. For the other case, i.e., $(t+1) \in \mathcal{I}_T$, this follows because $\sum_{i=1}^n\sum_{j=1}^n w_{ij}(\hat{\mathbf{x}}_j^{(t+1)} - \hat{\mathbf{x}}_i^{(t+1)}) = 0$, which uses the fact that W is a doubly stochastic matrix. Thus, we have

$$\bar{\mathbf{x}}^{(t+1)} = \bar{\mathbf{x}}^{(t)} - \frac{\eta_t}{n}\sum_{j=1}^n \nabla F_j(\mathbf{x}_j^{(t)}, \xi_j^{(t)}). \tag{3}$$

Subtracting $\mathbf{x}^*$ (the minimizer of (1)) from both sides gives

$$\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^* = \bar{\mathbf{x}}^{(t)} - \frac{\eta_t}{n}\sum_{j=1}^n \nabla F_j(\mathbf{x}_j^{(t)}, \xi_j^{(t)}) - \mathbf{x}^* \tag{4}$$

Using $\eta_t \leq \frac{1}{4L}$ (which follows from substituting $a \geq \frac{32L}{\mu}$ in $\eta_t = \frac{8}{\mu(a+t)}$), together with some algebraic manipulations, we have the following sequence relation for $\{\bar{x}^{(t)}\}$:

$$\mathbb{E}\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2 \leq \left(1 - \frac{\eta_t\mu}{2}\right)\mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2 + \frac{\eta_t^2\bar{\sigma}^2}{n} - \eta_t e_t + \eta_t\left(\frac{2L+\mu}{n}\right)\sum_{j=1}^n \mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\|^2 \tag{5}$$

where $e_t := \mathbb{E}f(\bar{\mathbf{x}}^{(t)}) - f^*$ and expectation is taken w.r.t. the entire process. We need to bound the last term of (5). For this, let $I_{(t_0)}$ denote the last synchronization index in $\mathcal{I}_T$ before time t. This, together with the assumption that $gap(\mathcal{I}_T) \leq H$, implies $t - I_{(t_0)} \leq H$. Using this and the bounded gradient assumption, we can easily bound the last term in the RHS of (5) (calculations are done in the full paper [35] in a more general matrix form):

$$\sum_{j=1}^n \mathbb{E}\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\right\|^2 \leq 2\mathbb{E}\sum_{j=1}^n\left\|\bar{\mathbf{x}}^{I_{(t_0)}} - \mathbf{x}_j^{I_{(t_0)}}\right\|^2 + 2n\eta_{I_{(t_0)}}^2H^2G^2 \tag{6}$$

In the following lemma, we show that the local iterates $\mathbf{x}_j^{(t)}, j \in [n]$ asymptotically approach to the average iterate $\bar{\mathbf{x}}^{(t)}$, thereby proving the contraction of the first term on the RHS of (6). In other words, the *consensus* occurs eventually.

Lemma 1 (Consensus – contracting deviation of local iterates and the averaged iterates)**.** *Under the assumptions of Theorem 1, for any $I_{(t)}$ such that $I_{(t)} \in \mathcal{I}_T$, we have*

$$\sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{I_{(t)}} - \mathbf{x}_j^{I_{(t)}}\right\|^2 \leq \frac{20 A_{I_{(t)}} \eta_{I_{(t)}}^2}{p^2},$$

where $A_{I_{(t)}} = 2nG^2H^2 + \frac{p}{2}\left(\frac{8nG^2H^2}{\omega} + \frac{5\omega n c_{I_{(t)}}}{4}\right)$ with $c_{I_{(t)}}$ denoting the threshold function evaluated at timestep $I_{(t)}$.

We give a proof sketch of Lemma 1 in Section IV-A.1.

Note that $\eta_{I_{(t_0)}} \leq 2\eta_t$, which follows from the following set of inequalities: $\frac{\eta_{I_{(t_0)}}}{\eta_t} = \frac{a+t}{a+I_{(t_0)}} \leq \frac{a+I_{(t_0)}+H}{a+I_{(t_0)}} \overset{(a)}{\leq} \frac{2(a+I_{(t_0)})}{a+I_{(t_0)}} = 2$, where (a) follows from our assumption that $a \geq H$. Now, substituting the bound from Lemma 1 in (6) and using $\eta_{I_{(t_0)}} \leq 2\eta_t$ gives $\sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\right\|^2 \leq 4\eta_t^2\left(\frac{40A_t}{p^2} + 2nH^2G^2\right)$. Putting this back in (5) yields

$$\mathbb{E}\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^*\|^2 \leq \left(1 - \frac{\eta_t\mu}{2}\right)\mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2 + \frac{\eta_t^2\bar{\sigma}^2}{n} - \eta_t e_t + 4\eta_t^3\left(\frac{2L+\mu}{n}\right)\left(\frac{40A_t}{p^2} + 2nH^2G^2\right)$$

Substituting the value of $A_t = 2nG^2H^2 + \frac{p}{2}\left(\frac{8nG^2H^2}{\omega} + \frac{5\omega n c_t}{4}\right)$ and defining $a_t = \mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}^*\|^2$, $Q = \frac{\bar{\sigma}^2}{n}$, $R = 8(2L+\mu)\left(\frac{40}{p^2} + \frac{80}{p\omega} + 1\right)G^2H^2$, $U = 100\left(\frac{2L+\mu}{p}\right)\omega$ and $U_t = Uc_t$, we get the recursion:

$$a_{t+1} \leq \left(1 - \frac{\mu\eta_t}{2}\right)a_t - \eta_t e_t + \eta_t^2 Q + \eta_t^3 R + \eta_t^3 U_t$$

Employing a modified version of [7, Lemma 3.3], which is provided in the full paper [35], gives

$$\frac{1}{S_T}\sum_{t=0}^{T-1} w_t e_t \leq \frac{\mu a^3}{8S_T}a_0 + \frac{4T(T+2a)}{\mu S_T}Q + \frac{64T}{\mu^2 S_T}R + \frac{64c_0T^{(2-\epsilon)}}{\mu^2(2-\epsilon)S_T}U,$$

where we've used that $c_t \leq c_0 t^{(1-\epsilon)}$ for $c_0 \geq 0$ and some $\epsilon \in (0,1)$, $w_t = (a+t)^2$ and $S_T = \sum_{t=0}^{T-1} w_t \geq \frac{T^3}{3}$. Using convexity of global objective f in the above inequality gives

$$\mathbb{E}f(\bar{\mathbf{x}}_{avg}^{(T)}) - f^* \leq \frac{\mu a^3}{8S_T}a_0 + \frac{4T(T+2a)}{\mu S_T}Q + \frac{64T}{\mu^2 S_T}R + \frac{64c_0T^{(2-\epsilon)}}{\mu^2(2-\epsilon)S_T}U,$$

where $\bar{\mathbf{x}}_{avg}^{(T)} = \frac{1}{S_T}\sum_{t=0}^{T-1} w_t \bar{\mathbf{x}}^{(t)}$. Substituting the values of a_0, Q, R, U in the above inequality gives the result of Theorem 1.

1) Proof sketch of Lemma 1: Note that Lemma 1 states that $e_{I_{(t)}}^{(1)} := \sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{I_{(t)}} - \mathbf{x}_j^{I_{(t)}}\right\|^2$ – the difference between local and the average iterates at the synchronization indices – decays asymptotically to zero for decaying learning rate η_t. We show this by setting up a contracting recursion for $e_{I_{(t)}}^{(1)}$. First we prove that

$$e_{I_{(t+1)}}^{(1)} \leq (1-\alpha_1)e_{I_{(t)}}^{(1)} + (1-\alpha_1)e_{I_{(t)}}^{(2)} + c_1\eta_{I_{(t)}}^2, \quad (7)$$

where $e_{I_{(t)}}^{(2)} := \sum_{j=1}^{n} \mathbb{E}\left\|\hat{\mathbf{x}}^{I_{(t+1)}} - \mathbf{x}_j^{I_{(t)}}\right\|^2$, $\alpha_1 \in (0,1)$, and c_1 is a constant that depends on n, δ, H, G. Note that (7) gives a contracting recursion in $e_{I_{(t)}}^{(1)}$, but it also gives the other term $e_{I_{(t)}}^{(2)}$, which we have to bound. It turns out that we can prove a similar inequality for $e_{I_{(t)}}^{(2)}$ as well:

$$e_{I_{(t+1)}}^{(2)} \leq (1-\alpha_2)e_{I_{(t)}}^{(1)} + (1-\alpha_2)e_{I_{(t)}}^{(2)} + c_2(t)\eta_{I_{(t)}}^2, \quad (8)$$

where $\alpha_2 \in (0,1)$; furthermore, we can choose α_1, α_2 such that $\alpha_1 + \alpha_2 > 1$. In (8), $c_2(t)$, in addition to n, δ, H, G, also depends on the compression factor ω and c_t which is the triggering threshold at timestep t.

Remark 4. *Note that [3] also proved analogous inequalities* (7) *and* (8) *with constants $c_1 = c_2 = 0$. Here $c_1, c_2(t)$ are non-zero (with $c_2(t)$ possibly varying with t) and arise due to the use of local iterations and event-triggered communication, which make the proof of these inequalities (in particular, the inequality* (8)*) significantly more involved than the corresponding inequalities in [3].*

Define $e_{I_{(t)}} := e_{I_{(t)}}^{(1)} + e_{I_{(t)}}^{(2)}$. Adding (7) and (8) gives the following recursion with $\alpha \in (0,1)$:

$$e_{I_{(t+1)}} \leq (1-\alpha)e_{I_{(t)}} + c_3(t)\eta_{I_{(t)}}^2. \quad (9)$$

From (9), we can show that $e_{I_{(t)}} \leq c(t)\eta_{I_{(t)}}^2$ for some $c(t)$ that depends on $n, \delta, H, G, \omega, c_t$. Lemma 1 follows from this because $\sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{I_{(t)}} - \mathbf{x}_j^{I_{(t)}}\right\|^2 = e_{I_{(t)}}^{(1)} \leq e_{I_{(t)}}$. □

B. Proof Outline of Theorem 2

Note that (3) holds irrespective to the learning rate schedule. So, by substituting η_t with η in (3), we get

$$\bar{\mathbf{x}}^{(t+1)} = \bar{\mathbf{x}}^{(t)} - \frac{\eta}{n}\sum_{j=1}^{n} \nabla F_j(\mathbf{x}_j^{(t)}, \xi_j^{(t)}).$$

With some algebraic manipulations, we get:

$$\mathbb{E}[f(\bar{\mathbf{x}}^{(t+1)})] \leq \mathbb{E}f(\bar{\mathbf{x}}^{(t)}) - \frac{\eta}{4}\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{(t)})\|_2^2 + \frac{L\eta^2\bar{\sigma}^2}{n} + \left[\frac{\eta L^2}{2n} + \frac{2L^3\eta^2}{n}\right]\sum_{j=1}^{n}\mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\|^2 \quad (10)$$

where expectation is taken over the entire process. Let $I_{(t_0)}$ be the last synchronization index in $\mathcal{I}_T$ before time t. Note that $t - I_{(t_0)} \leq H$. Similar to (6), we can also bound the last term on the RHS of (10) as (by replacing $\eta_{I_{t_{(0)}}}$ in (6) by η)

$$\sum_{j=1}^{n}\mathbb{E}\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\|^2 \leq 2\sum_{j=1}^{n}\mathbb{E}\|\bar{\mathbf{x}}^{I_{(t_0)}} - \mathbf{x}_j^{I_{(t_0)}}\|^2 + 2n\eta^2H^2G^2 \quad (11)$$

We can use the following lemma to bound the first term in the RHS of (11). This lemma is analogous to Lemma 1 in the

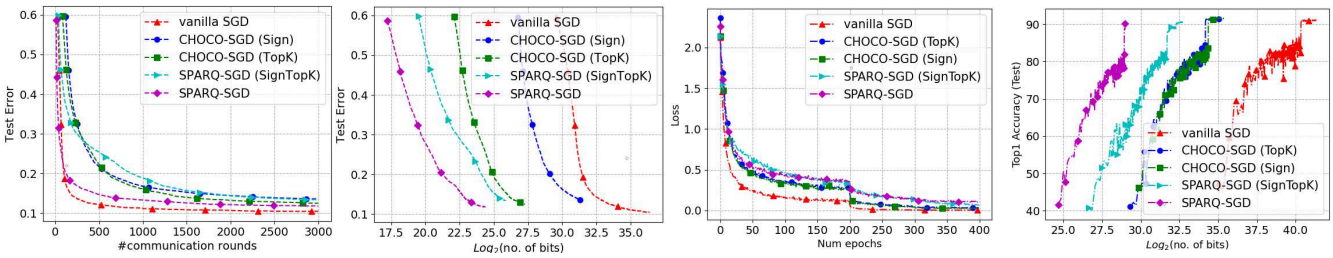

Fig. 1 Figure 1a and 1b are for convex objective and we plot test error vs number of communication rounds and test error vs total number of bits communicated, respectively, for different algorithms. Figure 1c and 1d are for non-convex objective where we plot training loss vs epochs and Top-1 accuracy vs total number of bits communicated, respectively.

fixed learning rate. Observe that if we simply replace $\eta_{I_{(t_0)}}$ with η in the bound of Lemma 1, we would get a slightly weaker bound than what we obtain in the following lemma, which we prove in the full paper [35].

Lemma 2 (Bounded deviation of local iterates and the averaged iterates)**.** *Under the assumptions of Theorem 2, for any $I_{(t)}$ such that $I_{(t)} \in \mathcal{I}_T$, we have*

$$\sum_{j=1}^{n} \mathbb{E}\|\bar{\mathbf{x}}^{I_{(t)}} - \mathbf{x}_j^{I_{(t)}}\|^2 \leq \frac{4A\eta^2}{p^2},$$

where $A = 2nG^2H^2 + \frac{p}{2}\left(\frac{8nG^2H^2}{\omega} + \frac{5\omega n}{4\eta^{(1-\epsilon)}}\right)$.

Remark 5. *Lemma 2 is essentially about consensus with bounded error, i.e., the nodes do not reach to a consensus, but within an error that is proportional to the learning rate η. Note that if we take a decaying learning rate η_t (as in the strongly-convex case), then, as shown in Lemma 1, different nodes will exactly reach to a consensus, however, the convergence rate of our algorithm will no longer be $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$, but we will only get a rate of $\mathcal{O}\left(\frac{1}{\log T}\right)$, which, though matches the convergence rate of running vanilla SGD with decaying learning rate on non-convex objectives, is much slower than what we can get with a fixed learning rate, as considered in this paper.*

Using the bound from Lemma 2 in (11) gives $\sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\right\|^2 \leq C := \frac{8A}{p^2}\eta^2 + 2n\eta^2H^2G^2$. Note that for the case of fixed learning rate η, we have to fix the time horizon (the number of iterations) T before the algorithm begins. By setting $\eta = \sqrt{\frac{n}{T}}$ and $T \geq 64nL^2$, we get $\eta \leq \frac{1}{8L}$. Now, substituting the bound on $\sum_{j=1}^{n} \mathbb{E}\left\|\bar{\mathbf{x}}^{(t)} - \mathbf{x}_j^{(t)}\right\|^2$ and $\eta \leq \frac{1}{8L}$ in (10), rearranging terms, and then summing from $t = 0$ to $T-1$ gives:

$$\sum_{t=0}^{T-1} \eta\mathbb{E}\|\nabla f(\bar{\mathbf{x}}^{(t)})\|_2^2 \leq 4\left(f(\bar{\mathbf{x}}^{(0)}) - \mathbb{E}f(\bar{\mathbf{x}}^{(t)})\right) + \frac{2L^2C}{n}\sum_{t=0}^{T-1}\eta^3 + \frac{8L^3C}{n}\sum_{t=0}^{T-1}\eta^4 + \frac{4L\bar{\sigma}^2}{n}\sum_{t=0}^{T-1}\eta^2$$

Dividing both sides by ηT, setting $\eta = \sqrt{\frac{n}{T}}$ and substituting the value of A from Lemma 2 proves Theorem 2. □

V. Experiments

In this section, we compare SPARQ-SGD with CHOCO-SGD [3], [17], which only employs compression (sparsification *or* quantization) and is state-of-the-art in communication efficient decentralized training.

Convex. We run SPARQ-SGD on MNIST dataset and use multi-class cross-entropy loss to model the local objectives $f_i, i \in [n]$. We consider $n = 60$ nodes connected in a ring topology, each processing a mini-batch size of 5 per iteration and having heterogeneous distribution of data across classes.

We work with $\eta_t = 1/(t + 100)$ (based on grid search) and synchronization index $H = 5$. For SPARQ-SGD, we use the composed operator $SignTopK$ [13] with $k = 10$ (out of 7840 length vector for MNIST dataset) For our experiments, we set the triggering constant $c_0 = 5000$ in SPARQ-SGD (line 7) and keep it unchanged until a certain number of iterations and then increase it periodically; while still maintaining that $c_t\eta_t^2$ decreases with t (as c_t is $o(t)$) .

• **Results.** In Figure 1a, we observe SPARQ-SGD can reach a target test error in fewer communication rounds while converging at a rate similar to that of vanilla SGD. The advantage to SPARQ-SGD comes from the significant savings in the number of bits communicated to achieve a desired test error, as seen in Figure 1b: to achieve a test error of around 0.12, SPARQ-SGD gets 250× savings as compared to CHOCO-SGD with $Sign$ quantizer, around 10-15× savings than CHOCO-SGD with $TopK$ sparsifier, and around 1000× savings than vanilla decentralized SGD. We also provide a plot for using the composed $SignTopK$ operator without event-triggering titled 'SPARQ-SGD (SignTopK)' for comparison.

Non-convex. We match the setting in CHOCO-SGD [17] and perform our experiments on the CIFAR-10 [34] dataset and train a ResNet20 [37] model with $n = 8$ nodes connected in a ring topology. Learning rate is initialized to 0.1, following a schedule consisting of a warmup period of 5 epochs followed by piecewise decay of 5 at epoch 200 and 300 and we stop training at epoch 400. The SGD algorithm is implemented with momentum with a factor of 0.9 and mini-batch size of 256. SPARQ-SGD consists of $H = 5$ local iterations followed by checking for a triggering condition, and then communicating with the composed $SignTopK$ operator, where we take top 1% elements of each tensor and

only transmit the sign and norm of the result. The triggering threshold follows a schedule piecewise constant: initialized to 2.5 and increases by 1.5 after every 20 epochs till 350 epochs are complete; while maintaining that $c_t < 1/\eta$ for all t. We compare performance of SPARQ-SGD against CHOCO-SGD with $Sign$, $TopK$ compression (taking top 1% of elements of the tensor) and decentralized vanilla SGD [1]. We also provide a plot for using the composed $SignTopK$ operator without event-triggering titled 'SPARQ-SGD (Sign-TopK)' for comparison.

• **Results.** We plot the global loss function evaluated at the average parameter vector across nodes in Figure 1c, where we observe SPARQ-SGD converging at a similar rate as CHOCO-SGD and vanilla decentralized SGD. Figure 1d shows the performance for a given bit-budget, where we show the Top-1 test accuracy as a function of the total number of bits communicated. For Top-1 test-accuracy of around 90%, SPARQ-SGD requires about 40× less bits than CHOCO-SGD with $Sign$ or $TopK$ compression, and around 3K× less bits than vanilla decentralized SGD to achieve the same Top-1 accuracy.

VI. ACKNOWLEDGMENTS

This work was partially supported by NSF grants #1740047, #1514531, #2007714, by UC-NL grant LFR-18-548554 and by Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *NIPS*, 2017, pp. 5330–5340.

[2] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *NeurIPS*, 2018, pp. 7663–7673.

[3] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication," in *ICML*, 2019.

[4] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *INTERSPEECH*, 2015, pp. 1488–1492.

[5] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *EMNLP*, 2017, pp. 440–445.

[6] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *ICLR*, 2018.

[7] S. U. Stich, J. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *NeurIPS*, 2018, pp. 4452–4463.

[8] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *NeurIPS*, 2018, pp. 5977–5987.

[9] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: communication-efficient SGD via gradient quantization and encoding," in *NIPS*, 2017, pp. 1709–1720.

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

[11] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *ICML*, 2017, pp. 3329–3337.

[12] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," in *ICML*, 2019, pp. 3252–3261.

[13] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-sgd: Distributed SGD with quantization, sparsification and local computations," in *NeurIPS*, 2019, pp. 14 668–14 679.

[14] S. U. Stich, "Local SGD converges fast and communicates little," in *ICLR*, 2019.

[15] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication:demystifying why model averaging works for deep learning," in *AAAI*, 2019, pp. 5693–5700.

[16] G. F. Coppola, "Iterative parameter mixing for distributed large-margin training of structured predictors for natural language processing," Ph.D. dissertation, University of Edinburgh, UK, 2015.

[17] A. Koloskova*, T. Lin*, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *ICLR*, 2020.

[18] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized decentralized consensus optimization," in *CDC*, 2018, pp. 5838–5843.

[19] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *ICML*, 2019, pp. 344–353.

[20] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[21] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *CDC*, 2012, pp. 3270–3285.

[22] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Automat. Contr.*, vol. 57, no. 5, pp. 1291–1297, 2012.

[23] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[24] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Trans. Autom. Contr.*, vol. 60, no. 7, pp. 1992–1997, 2015.

[25] Y. Liu, C. Nowzari, Z. Tian, and Q. Ling, "Asynchronous periodic event-triggered coordination of multi-agent systems," in *CDC*, 2017, pp. 6696–6701.

[26] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[27] W. Chen and W. Ren, "Event-triggered zero-gradient-sum distributed consensus optimization over directed networks," *Automatica*, vol. 65, pp. 90–97, 2016.

[28] W. Du, X. Yi, J. George, K. H. Johansson, and T. Yang, "Distributed optimization with dynamic event-triggered mechanisms," in *CDC*, 2018, pp. 969–974.

[29] T. Chen, G. Giannakis, T. Sun, and W. Yin, "Lag: Lazily aggregated gradient for communication-efficient distributed learning," in *NeurIPS*, 2018, pp. 5050–5060.

[30] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *CDC*, 2008, pp. 4177–4184.

[31] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast convergence rates of distributed subgradient methods with adaptive quantization," *IEEE Trans. Automat. Contr.*, 2020, DOI: 10.1109/TAC.2020.3014095.

[32] A. Nedich, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Automat. Contr.*, vol. 54, no. 11, pp. 2506–2517, 2009.

[33] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization design for distributed optimization," *IEEE Trans. Automat. Contr.*, vol. 62, no. 5, pp. 2107–2120, 2016.

[34] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[35] N. Singh, D. Data, J. George, and S. N. Diggavi, "SPARQ-SGD: event-triggered and compressed communication in decentralized stochastic optimization," 2019, arXiv: 1910.14280.

[36] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *ICML*, 2012.

[37] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *NIPS*, 2016, pp. 2074–2082.