Efficient Randomized Subspace Embeddings for Distributed Optimization Under a Communication Budget

Rajarshi Saha\textsuperscript{O}, Student Member, IEEE, Mert Pilanci\textsuperscript{O}, Member, IEEE, and Andrea J. Goldsmith\textsuperscript{O}, Fellow, IEEE

Abstract—We study first-order optimization algorithms under the constraint that the descent direction is quantized using a pre-specified budget of \( R \)-bits per dimension, where \( R \in (0, \infty) \). We propose computationally efficient optimization algorithms with convergence rates matching the information-theoretic performance lower bounds for: (i) Smooth and Strongly-Convex objectives with access to an Exact Gradient oracle, as well as (ii) General Convex and Non-Smooth objectives with access to a Noisy Subgradient oracle. The crux of these algorithms is a polynomial complexity source coding scheme that embeds a vector into a random subspace before quantizing it. These embeddings are such that with high probability, their projection along any of the canonical directions of the transform space is small. As a consequence, quantizing these embeddings followed by an inverse transform to the original space yields a source coding method with optimal covering efficiency while utilizing just \( R \)-bits per dimension. Our algorithms guarantee optimality for arbitrary values of the bit-budget \( R \), which includes both the sub-linear budget regime \((R < 1)\), as well as the high-budget regime \((R \geq 1)\), while requiring \( O(n^2) \) multiplications, where \( n \) is the dimension. We also propose an efficient relaxation of this coding scheme using Hadamard subspaces that requires a near-linear time, i.e., \( O(n \log n) \) additions. Furthermore, we show that the utility of our proposed embeddings can be extended to significantly improve the performance of gradient sparsification schemes. Numerical simulations validate our theoretical claims. Our implementations are available at https://github.com/rajarshisaha95/DistOptConstrComm.

Index Terms—Kashin embeddings, random orthonormal subspace, hadamard subspace, distributed optimization, bit-budget constraint, gradient quantization, error feedback.

I. INTRODUCTION

DISTRIBUTED optimization algorithms that leverage edge computation of remote devices have proved promising for training large-scale machine learning models [1], [2]. To solve an optimization problem \( \min_{x \in \mathcal{X} \subseteq \mathbb{R}^n} f(x) \) in the parameter-server framework [3], the server maintains an iterate \( x_t \) at time \( t \), which is an estimate of the minimizer \( x^*_t = \arg \min_{x \in \mathcal{X}} f(x) \). We consider a setting in which at every iteration, the worker(s) receives the current iterate \( x_t \) from the server, computes \( \nabla f(x_t) \), and communicates information about the computed gradient back to the server, allowing it to take a descent step in that direction. This process is repeated till the sequence of iterates \( \{x_t\}_{t=1,2,\ldots} \) converges.

Communication overhead is invariably the primary bottleneck of such distributed systems. When distributed optimization algorithms are implemented over severely communication constrained environments, for instance in wireless networks [4], [5], the bandwidth of the channel (over which the workers send information to the server) is an expensive resource, and is often a pre-specified constraint beyond the algorithm designers’ control. In this work, we resolve the question: How to design optimal quantization schemes when the worker(s) is constrained to communicate its (sub) gradient information to the server with a strict budget of \( R \) bits per dimension? We consider the following settings:

1) When the objective function \( f(x) \) is \( L \)-smooth and \( \mu \)-strongly convex.

2) When \( f(x) \) is a general convex function that is not necessarily smooth.

We first consider a single-worker single-server setting and extend it later to multiple workers. Let \( \Pi_R \) denote the set of all optimization protocols with access to an exact first-order oracle for which the information exchange between the worker and the server at every iteration is limited to \( R \)-bits per dimension. For setting (i), it is possible to achieve a linear convergence of the iterates \( \{x_t\} \) to \( x^*_t \). With this in mind, to quantify the performance of any protocol \( \pi \in \Pi_R \), consider the asymptotic worst-case linear convergence rate:

\[
C(\pi) = \limsup_{T \to \infty} \sup_{f \in \mathcal{F}_{L,D}} \left( \frac{\|x_T(\pi) - x^*_T\|_2}{D} \right)^\frac{1}{7}.
\]

Here, \( \mathcal{F}_{L,D} \) is the class of \( L \)-smooth and \( \mu \)-strongly convex functions which satisfy \( \|x^*_T\|_2 \leq D \) \((D \geq 0)\), and \( x_T(\pi) \) denotes the output of the protocol \( \pi \in \Pi_R \) after \( T \) iterations.

Manuscript received 31 December 2021; revised 6 May 2022 and 13 July 2022; accepted 4 August 2022. Date of publication 16 August 2022; date of current version 14 November 2022. This work was supported in part by the Office of Naval Research under Grant ONR N00014-18-1-2191; in part by the National Science Foundation under Grant DMS-2134248 and Grant ECCS-2037304; in part by the Army Research Office Early Career Award under Grant W911NF-21-1-0242; in part by Intel Research; in part by Huawei Technologies; in part by Facebook Research; in part by Adobe Research; and in part by Stanford SystemX Alliance. (Corresponding author: Rajarshi Saha.)

Rajarshi Saha and Mert Pilanci are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: rajarshi@stanford.edu).

Andrea J. Goldsmith is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/JSATIT.2022.3198412, provided by the authors.

Digital Object Identifier 10.1109/JSATIT.2022.3198412

2641-8770 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Note that (1) implies $\pi$ achieves a convergence of the iterates $\{x_t\}$ to $x^\ast$ with a guarantee $\|x_T(\pi) - x^\ast\|_2 \lesssim C(\pi)^T_D$ for $C(\pi) \leq 1$. This implies that algorithms in the class $P_{\pi}$ require $O(\log(\frac{D}{\epsilon})/\log(C(\pi)^{-1}))$ iterations to achieve a suboptimality gap of $\epsilon$. Since a smaller value of $C(\pi)$ is desirable for faster convergence, we can characterize the set of protocols $P_{\pi}$ according to the following minimax rate:

$$C(R) \triangleq \inf_{\pi \in P_{\pi}} C(\pi) = \inf_{\pi \in P_{\pi}} \lim_{T \to \infty} \sup_{f \in F_{\mu,L,D}} \left( \frac{\|x_T(\pi) - x^\ast\|_2}{D} \right)^{\frac{1}{T}}. \quad (2)$$

It has been shown [6] that the information-theoretic lower bound on (2) can be obtained as $C(R) \geq \max[\sigma, 2^{-K}]$ where $\sigma = \frac{L}{\sqrt{T}}$ is the convergence rate in the absence of any bit-budget constraints. This provides a theoretical limit to the performance of any protocol in $P_{\pi}$. In this work, we propose an algorithm DGD-DEF that achieves this lower bound to within constant factors while requiring only $O(n^3)$ (or $O(n^2)$, if additional information is available) multiplications. We also propose a relaxed, near-linear time version that requires only $O(n\log n)$ additions, saving significantly on the computation requirement while attaining a performance that is a mild $O(\sqrt{\log n})$ factor away from the lower bound. To the best of our knowledge, DGD-DEF is the first polynomial complexity algorithm whose performance matches the lower bound of $C(R) \geq \max[\sigma, 2^{-K}]$.

On the other hand for setting (ii), when $f(x)$ is a general convex function (not necessarily smooth) and we have access to a noisy subgradient oracle [7], it is not possible to achieve a linear convergence. In this case, to measure the performance of any protocol $\pi$, we consider the expected suboptimality gap,

$$\mathcal{E}(\pi) = \sup_{(f,\mathcal{O})} \mathbb{E}[f(x_T(\pi))] - f(x^\ast), \quad (3)$$

and study how it scales with the number of iterations $T$. Here, $x_T(\pi)$ is the output of protocol $\pi$, and we consider the worst-case performance over all objectives $f : \mathcal{X} \to \mathbb{R}$ with compact, convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ satisfying $\sup_{x,y \in \mathcal{X}} \|x-y\|_2 \leq L$, and stochastic subgradient oracles $O$ whose outputs are uniformly bounded by some parameter $B$. We consider algorithms $\pi$ from the class $P_{T,R}$ of all optimization protocols that execute at most $T$ iterations with a communication budget of $R$-bits per dimension per iteration. This set of protocols $P_{T,R}$ can be characterized according to the following minimax expected suboptimality gap,

$$\mathcal{E}(T, R) \triangleq \inf_{\pi \in P_{T,R}} \inf_{\pi \in P_{T,R}} \mathbb{E}[f(x_T(\pi))] - f(x^\ast). \quad (4)$$

A lower bound on (4) can be obtained [7] as $\mathcal{E}(T, R) \geq \frac{C_{DB}}{\sqrt{T\min[1,R]}}$. We propose DQ-PSGD and its relaxed version, that respectively, attain this lower bound to within constant and mild logarithmic factors while requiring $O(n^3)$ multiplications and $O(n\log n)$ additions.

A minmax optimal protocol (or algorithm) $\pi^\ast$ in either setting requires designing an optimal source coding scheme that quantizes the gradient information efficiently. A source coding scheme is a pair of mappings $(E, D)$, where the encoding $E : \mathbb{R}^n \to \{0,1\}^{\epsilon R}$ is done by the worker to quantize the information it wants to send to the server. The decoding map $D : \{0,1\}^{\epsilon R} \to \mathbb{R}^n$ recovers an estimate of the input to the encoder and is implemented at the server. In this work, we present Democratic Source Coding (DSC), an efficient polynomial-time fixed-length vector quantization scheme, which when used with suitably designed first-order optimization algorithms, can achieve the respective lower bounds on the minimax rates (2) and (4) to within constant factors, establishing the minimax optimality of the algorithms.

An alternative way to look at the bit-budget constrained optimization problem is to consider the minimum threshold budget $R_{thr}$ required to attain the convergence rate achievable in the absence of any budget constraint. For setting (i), the lower bound of $C(R) \geq \max[\sigma, 2^{-K}]$ implies that we cannot hope to achieve the convergence rate of unquantized setting if $R < \log(\frac{1}{\sigma})$. Whereas naive quantizers [8] would require $R_{thr} \geq \log(\frac{\pi}{1})$ bits to attain the unquantized convergence rate, DGD-DEF gets rid of the dimension dependence, and requires just $R_{thr} = O(\log(\frac{1}{\sigma}))$ bits. Moreover, it achieves this while entail only a polynomial complexity of $O(n^2)$ as opposed to Roger’s quantizer [9] used in [6] that demands exponential complexity. For setting (ii), the lower bound of $\mathcal{E}(T, R) \geq \frac{C_{DB}}{\sqrt{T\min[1,R]}}$ implies that for the sub-linear budget regime, i.e., when $R < 1$, we can expect the suboptimality gap to scale as $\frac{1}{\sqrt{T}}$ as long as we have a constant bit-budget, i.e., $R_{thr} = O(1)$. DQ-PSGD ensures this while requiring only $R_{thr} = O(1)$ bits, which establishes its optimality, as opposed to $R_{thr} = O(\sqrt{n})$ for naive quantizers or $R_{thr} = O(\log\log n)$ for RATQ [7].

A. Our Contributions

In this work, we consider algorithms that attain the information-theoretic lower bounds to the minimax performance metrics of bit-budget constrained optimization. Existing works [6], [7] have characterized the precise lower bounds to (2), (4), and we provide optimal algorithms that achieve these minimax lower bounds to within constant factors while requiring $O(n^2)$ computation. Our contributions are as follows:

(a) We first propose Democratic Source Coding (DSC) which use Kashim embeddings [10] to compress a vector in $\mathbb{R}^n$ subject to a constraint of $R$ bits per dimension. DSC is a polynomial-time source coding scheme and its error is independent of the dimension $n$; a crucial property for efficiently compressing high-dimensional vectors.

(b) For strongly convex smooth objectives, we propose DGD-DEF: Distributed Gradient Descent with Democratically Encoded Feedback, an algorithm that uses DSC to quantize the feedback-corrected gradients and show that it achieves the lower bound on (2).

(c) For general convex non-smooth objectives, we propose DQ-PSGD: Democratically Quantized Projected Stochastic subGradient Descent that achieves the lower bound on (4).
TABLE I

| COMPRESSION SCHEME          | NO. OF BITS       | ERROR               | COMPLEXITY        |
|-----------------------------|-------------------|---------------------|-------------------|
| SIGN QUANTIZATION [14], [15]| $O(n)$            | $O(n)$              | $O(n)$            |
| QSGD [8]                    | $O(2^R(2^R + \sqrt{n}))$ | $\min\{\sqrt{n}2^{-n}, n2^{-3R}\}$ | $O(n)$            |
| TERNARY QUANTIZATION [16]   | $O(n \log n)$     | $O(n)$              | $O(n)$            |
| vQSGD-GAUSSIAN [17]         | $O(\log n)$       | $O(\{\varpi\})$   | $O(\exp(c))$      |
| vQSGD-CROSS POLYTOPE [17]  | $O(n)$            | $O(n/k)$            | $O(n)$            |
| Top-$k$ SPARSIFICATION [18]| $O(k + \log_2(n^2))$ | $(n-k)/n$           | $O(k+\log_2k)$    |
| RANDOM SPARSIFICATION [19] | $O(k + \log_2(n^2))$ | $O(\{\varpi\})$   | $O(n)$            |
| SIM-Q+ [20]                 | $O(3n)$           | $O(1)$              | $O(n^2)$          |
| DSC (Ours)                  | $nR + O(1)$       | $O\left(2^{-2R/\lambda}\right)$ | $O(n^2)$          |
| NDSC (Ours)                 | $nR + O(1)$       | $O\left(2^{-2R/\lambda} \log n\right)$ | $O(n \log n)$    |

(d) Since even the $O(n^2)$ complexity of DSC can be computationally demanding for large $n$, we further propose a computationally simpler relaxation, referred to as NDSC: Near Democratic Source Coding, which achieves optimality to within a mild logarithmic factor. We observe that in simulations, NDSC performs at par with DSC.

(e) Finally, in Section IV-C, we show how our algorithms can be extended to multi-worker setups. We also show that DSC or NDSC consistently improve the performance when used in conjunction with other existing compression strategies (Section V and Section II in the supplementary material).

B. Significance and Related Work

Communication-Constrained Distributed Optimization: Much work has been done in recent years to address the communication bottleneck in distributed optimization. Variable-length coding schemes were proposed in [8]. The bit-requirement of these quantization schemes are optimal in expectation, but their worst-case performance is not. Our work considers fixed-length quantizers for the setting where precision constraints are imposed as a pre-specified bit-budget of $R$-bits that needs to be strictly respected even for worst case inputs. The problem of distributed optimization under bit-budget constraints is considered in [6], [7]. Reference [6] considers smooth and strongly convex objectives, and derive a lower bound on the minimax convergence rate defined in (2), along with a matching upper bound. However, their upper bounding algorithm has exponential complexity and hence, practically infeasible; whereas, our proposed algorithm DGD-DEF, which uses DSC for quantization has polynomial complexity and achieves the minimax lower bound to within constant factors. For the setting of general convex and non-smooth objectives, [7] provides a lower bound to the minimax suboptimality gap defined in (4). Using their proposed quantizer RATQ, they also give an upper bound which characterizes the minimum bit-budget required to attain this minimax optimal lower bound to within an iterated logarithmic factor in $d$. Compared to this, DQ-PSPGD uses $R + o_n(1)$ bits per dimension, and attains a suboptimality gap within constant factors of the minimax lower bound. Here, $R$ is specified as a constraint and $o_n(1)$ is a term that goes to zero as $n \to \infty$. A fixed length nearly optimal coding scheme that employs random rotations was used in [11], [12]. Orthonormal transforms for random rotations were also used in [13]. However, their goal was to design quantizers that achieve low statistical correlation between signal and quantization error rather than minimizing the $\ell_2$ quantization error, which is more relevant for quantizing gradients in distributed optimization, when the distribution of quantizer input is not known. In our work, we also propose a computationally simpler relaxation of DSC, namely NDSC that achieves the minimax lower bounds to within a logarithmic factor. We note that our proposed near-democratic embeddings boil down to random rotations when square orthonormal transforms are used, i.e., NDSC is a generalization of random rotations. When Hadamard transforms are considered, these works assume that the dimension $n$ is such that a Hadamard matrix can be constructed. However, it might not necessarily be true, and naive heuristics like partitioning the vector or zero-padding in order to make the dimension equal to the nearest power of 2 can be suboptimal. NDSC performs better than random rotations in such cases. Another popular strategy to reduce the communication requirement is gradient sparsification that reduces the dimension of the vector being exchanged. Our coding strategies, DSC and NDSC can be used in conjunction with these sparsification methods. We provide a comparison of our work with existing quantization and sparsification strategies in Table I.

Kashin Embeddings and Random Matrix Theory: Kashin embeddings were studied in the random matrix theory literature [10], [21] for their relation to convex geometry and vector quantization. From a high level perspective, Kashin embedding of a vector $y \in \mathbb{R}^n$, is a vector $x \in \mathbb{R}^N (N \geq n)$, which has the property that the components of $x$ are similar to each other in magnitude, i.e., for all $i \in [N]$, $|x_i| = \Theta(1/\sqrt{N})$ with high probability (w.h.p.). Even if components of $y$ may be arbitrarily varying in magnitude, Kashin embeddings have an effect of evenly distributing this variation across different components of $x$. Subsequently, applying lossy compression schemes (e.g., quantization) to the democratic embedding $x$ instead of the
original vector $y$ incurs less error. The inverse embedding map $S : \mathbb{R}^N \rightarrow \mathbb{R}^n$ is linear, i.e., $y = Sx$. Usually, $S$ is randomly generated and the properties of the democratic embeddings are very closely related to Restricted Isometry Property (RIP) parameters of $S$ [22]. We study different classes of random matrices (subgaussian, orthonormal, and Hadamard) and the pros and cons of using them for constructing respective DSC schemes. The efficacy of Kashin embeddings for various learning problems have been studied in [23], [24], [25], [26]. In our work, we go even further in using them for designing general source coding schemes (both stochastic and deterministic) and show that they yield minimax optimal optimization algorithms. Kashin embedding of a vector is not unique, and [10] proposed an iterative-projection type algorithm to compute a Kashin embedding. However, their algorithm requires explicit knowledge of RIP parameters of $S$, which is not readily available. To this end, [27] introduced the notion of democratic embeddings (DE). DE of a vector $y \in \mathbb{R}^n$ is a Kashin embedding too, and is obtained by solving a linear program. We propose a simple relaxation of this linear program, and show that its solution yields a near-democratic embedding.

II. DEMOCRATIC EMBEDDINGS

Consider a wide matrix $S \in \mathbb{R}^{n \times N}$, where $n \leq N$. For any given vector $y \in \mathbb{R}^n$, the system of equations $y = Sx$ is underdetermined in $x \in \mathbb{R}^N$, with the set $S = \{ x \in \mathbb{R}^N | y = Sx \}$ as the solution space. The vector $x^* \in S$ which has the minimum $\ell_2$-norm in this solution space is referred to as the Democratic Embedding of $y$ with respect to $S$. In other words, $x^*$ is obtained by solving,

$$\min_{x \in \mathbb{R}^n} \|x\|_\infty \text{ subject to } y = Sx. \quad (5)$$

The constraint set $S$ can be relaxed to a larger set $S' = \{ x \in \mathbb{R}^n \| y - Sx \|_2 \leq \epsilon \}$ as in [27]. In the rest of our work, we consider $\epsilon = 0$, i.e., exact representations. In order to characterize the solution of (5) (cf. Lemma 1), we review certain definitions from [10], [27].

Definition 1 (Frame): A matrix $S \in \mathbb{R}^{n \times N}$ with $n \leq N$ is called a frame if $A\|y\|_2^2 \leq \|S^Ty\|_2^2 \leq B\|y\|_2^2$ holds for any vector $y \in \mathbb{R}^n$ with $0 < A \leq B < \infty$, where $A$ and $B$ are called lower and upper frame bounds respectively.

Definition 2 [Uncertainty Principle (UP)]: A frame $S \in \mathbb{R}^{n \times N}$ satisfies the Uncertainty Principle with parameters $\eta, \delta$, with $\eta > 0, \delta \in (0, 1)$ if $\|Sx\|_2 \leq \eta \|x\|_2$ holds for all (sparse) vectors $x \in \mathbb{R}^n$ satisfying $\|x\|_0 \leq \delta N$, where $\|x\|_0$ denotes the number of non-zero elements in $x$.

Lemma 1 (Democratic Embeddings) [27]: Let $S \in \mathbb{R}^{n \times N}$ be a frame with bounds $A, B$ (cf. Def. 1) that satisfies the uncertainty principle (UP) (cf. Def. 2) with parameters $\eta, \delta$ such that $A > \eta B$. Then for any vector $y \in \mathbb{R}^n$, the solution $x_d$ of (5) satisfies

$$K_I \sqrt{N} \|y\|_2 \leq \|x_d\|_\infty \leq K_u \sqrt{N} \|y\|_2, \quad (6)$$

where $K_I = \frac{1}{\sqrt{B}}$ and $K_u = \frac{\eta}{(A-\eta \sqrt{B}) \sqrt{A}}$ are called lower and upper Kashin constants respectively.

We are interested in Parseval frames ($A = B = 1$), i.e., they satisfy $SS^T = I_n$ (where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix), implying $K_I = 1$ and $K_u = \eta(1-\eta)^{-1}\delta^{-1/2}$. $K_u$ depends only on the choice of $S$ and nothing else. Lemma 1 shows that none of the coordinates of the democratic embedding is too large, and the information content of $y$ is distributed evenly.

The value of upper Kashin constant $K_u$ depends on the choice of frame construction $S$, as well as its aspect ratio $\lambda = N/n$. [10], [27] show that if $S$ is a random Haar orthonormal matrix, then $K = K(\lambda)$, where $\lambda > 1$ can be arbitrarily close to 1. Such frames can be obtained by generating a random $N \times N$ orthonormal matrix sampled from the Haar distribution, and randomly selecting $n$ of its rows. Since choosing $\lambda$ is up to us, Lemma 1 implies that for random orthonormal frames, democratic embeddings satisfy $\|x_d\|_\infty = \Theta(1/\sqrt{N})$ w.h.p. As we will see in Section III, for large $n$ (or equivalently large $N$ since $N \geq n$), this remarkably improves the robustness of our proposed compression schemes. A comprehensive comparison of different choices for $S$ is given in Section IV in the supplementary material.

A. Near-Democratic Embeddings

Although the linear program (5) can be solved with $O(n^3)$ multiplications using simplex or Newton’s method, it can still be computationally intensive. A projected gradient descent type algorithm with $O(n^2)$ complexity was presented in [10], but implementing it requires explicit knowledge of $\eta, \delta$ which is not readily available. We propose a simpler relaxation of (5):

$$\min_{x \in \mathbb{R}^n} \|x\|_2^2 \text{ subject to } y = Sx. \quad (7)$$

The solution of the $\ell_2$-minimization (7) can be found in closed form (cf. Section I in the supplementary material) as:

$$x_{nd} = S^T y = S^T \left( SS^T \right)^{-1} y \in \mathbb{R}^n, \quad (8)$$

where $(\cdot)^+$ (defined as above) is the pseudo-inverse. For Parseval frames $S$, this boils down to $x_{nd} = S^Ty$. We refer to $x_{nd} = S^Ty$ as the Near-Democratic embedding of $y \in \mathbb{R}^n$ with respect to $S \in \mathbb{R}^{n \times N}$ and show that the solution $x_{nd}$ of (7) satisfies $\|x_{nd}\|_\infty = O((\sqrt{\log N}/\sqrt{N})\|y\|_2)$ w.h.p. The additional $\sqrt{\log N}$ factor instead of the constant $K_u$ is a very modest price to pay compared to the computational savings, even for dimensions as large as $N \sim 10^6$. Note that as $\lambda$ approaches 1, the solution space $S$ of (5) becomes smaller, and for $\lambda = 1$, the solutions of (5) and (7) coincide. Lemma 2 characterizes our result explicitly. A random orthonormal matrix is obtained by generating random Gaussian matrix $G \in \mathbb{R}^{n \times N}$ with i.i.d. entries, $G_{ij}$ are $N(0, 1)$, computing its singular-value decomposition $G = USV^T$, letting $S = UV^T$ and generating $S \in \mathbb{R}^{n \times N}$ by randomly selecting $n$ rows of $\tilde{S}$, i.e., $S = PS$, where $P \in \mathbb{R}^{n \times n}$ is a sampling matrix obtained by randomly selecting $n$ rows of $I_N$.

Lemma 2 (Near-Democratic Embeddings With Random Orthonormal Frames): For a random orthonormal frame $S \in \mathbb{R}^{n \times n}$ generated as described above, with probability (w.p.) at
The proof of Lemma 2 is delegated to Appendix A. It utilizes the observation that each coordinate of $S^T y \in \mathbb{R}^N$ is isotropically distributed, and subsequently exploits measure concentration. Random orthonormal matrices prove quite beneficial in this regard. Nevertheless, computing the near-democratic embeddings $x_{nd} = S^T y$, for random orthonormal frames still requires $O(n^2)$ time, and moreover, even storing $S$, comprising of 32-bit floating-point entries can be memory intensive. To address this, we further propose a randomized Hadamard construction for $S$. Storing a randomized Hadamard matrix amounts to only storing the signs, and near-democratic embeddings using such matrices can be computed in near-linear time. Consider the $N \times N$ Hadamard matrix $H$ whose entries are normalized, i.e., $H_{ij} = \pm 1/\sqrt{N}$, $H = H^T$, and $HH^T = I_N$. Let $D \in \mathbb{R}^{N \times N}$ be a diagonal matrix whose entries are randomly chosen to be $\pm 1$ with equal probability. Let $P \in \mathbb{R}^{N \times N}$ be the sampling matrix as before. We define our frame to be $S = PDH \in \mathbb{R}^{N \times N}$. Note that $SS^T = PDH H^T D^T = PD^T D^T = I_N$, i.e., our randomized Hadamard construction is a Parseval frame. Storing the 1-bit signs is enough to store the matrix $S = PDH$ in the memory. Furthermore, the near-democratic embedding $x_{nd} = S^T y = HD^T y$ can be computed with just $O(n \log n)$ additions, subtractions and sign-flips as $S y_j = \pm 1/\sqrt{N}$. Unlike random orthonormal matrices, it does not require any explicit floating-point multiplications. Lemma 3 characterizes the $\| \cdot \|_\infty$ of the solution of (7) with $S = PDH$.

Lemma 3 (Near-Democratic Embeddings With Randomized Hadamard Frames): For $S = PDH \in \mathbb{R}^{N \times N}$, with $P, D, H$ defined as above, with probability at least $1 - \frac{1}{2N}$, the solution of (7) satisfies:

$$\|x_{nd}\|_\infty \leq 2\sqrt{\log(2N)} \|y\|_2. \hspace{1cm} (9)$$

Its proof is provided in Appendix B, and upper bounds the tail probability of each coordinate of $S^T y$ using a Chernoff-type argument, followed by a union bound. In Section III, we employ our democratic and near-democratic embeddings for source coding and show that they respectively yield efficient optimal and near-optimal vector quantizers.

III. DEMOCRATIC SOURCE CODING

We introduce our proposed random-embedding based quantization algorithms in Section III-A and derive upper bounds on the $\ell_2$ quantization errors, which are relevant for the convergence analysis of our proposed algorithms in Section IV. Furthermore, in Section III-B, we discuss covering efficiency, which is an alternative notion of characterizing the efficiency of vector quantizers.

We first start with the definition of uniform scalar quantizer. Denote the $\ell_\infty$-ball of radius $r$ centered at the origin of $\mathbb{R}^N$ by $B_\infty^N(r)$. Finite length source coding schemes map its inputs to a discrete set of finite cardinality. An $R$-bit uniform scalar quantizer is a mapping $Q(\cdot) : B_\infty^N(1) \rightarrow S$ with $S \subseteq \mathbb{R}^N$ and $|S| \leq 2^{\log |S|}$. With a bit-budget of $R$-bits per dimension, the $M = 2^R$ quantization points $\{v_i\}_{i=1}^M$ along any dimension are given by

$$v_i = -1 + (2i - 1)\Lambda/2, \text{ for } i = 1, \ldots, M, \text{ where } \Lambda = 2/M \text{ is the resolution. }$$

The proof of Lemma 2 is delegated to Appendix A. It utilizes the observation that each coordinate of $S^T y \in \mathbb{R}^N$ is isotropically distributed, and subsequently exploits measure concentration. Random orthonormal matrices prove quite beneficial in this regard. Nevertheless, computing the near-democratic embeddings $x_{nd} = S^T y$, for random orthonormal frames still requires $O(n^2)$ time, and moreover, even storing $S$, comprising of 32-bit floating-point entries can be memory intensive. To address this, we further propose a randomized Hadamard construction for $S$. Storing a randomized Hadamard matrix amounts to only storing the signs, and near-democratic embeddings using such matrices can be computed in near-linear time. Consider the $N \times N$ Hadamard matrix $H$ whose entries are normalized, i.e., $H_{ij} = \pm 1/\sqrt{N}$, $H = H^T$, and $HH^T = I_N$. Let $D \in \mathbb{R}^{N \times N}$ be a diagonal matrix whose entries are randomly chosen to be $\pm 1$ with equal probability. Let $P \in \mathbb{R}^{N \times N}$ be the sampling matrix as before. We define our frame to be $S = PDH \in \mathbb{R}^{N \times N}$. Note that $SS^T = PDH H^T D^T = PD^T D^T = I_N$, i.e., our randomized Hadamard construction is a Parseval frame. Storing the 1-bit signs is enough to store the matrix $S = PDH$ in the memory. Furthermore, the near-democratic embedding $x_{nd} = S^T y = HD^T y$ can be computed with just $O(n \log n)$ additions, subtractions and sign-flips as $S y_j = \pm 1/\sqrt{N}$. Unlike random orthonormal matrices, it does not require any explicit floating-point multiplications. Lemma 3 characterizes the $\| \cdot \|_\infty$ of the solution of (7) with $S = PDH$.

Lemma 3 (Near-Democratic Embeddings With Randomized Hadamard Frames): For $S = PDH \in \mathbb{R}^{N \times N}$, with $P, D, H$ defined as above, with probability at least $1 - \frac{1}{2N}$, the solution of (7) satisfies:

$$\|x_{nd}\|_\infty \leq 2\sqrt{\log(2N)} \|y\|_2. \hspace{1cm} (9)$$

Its proof is provided in Appendix B, and upper bounds the tail probability of each coordinate of $S^T y$ using a Chernoff-type argument, followed by a union bound. In Section III, we employ our democratic and near-democratic embeddings for source coding and show that they respectively yield efficient optimal and near-optimal vector quantizers.

A. Proposed Quantization Strategy

Given a frame $S \in \mathbb{R}^{N \times N}$ for any $y \in \mathbb{R}^n$, denote its democratic and near-democratic embeddings (i.e., the solutions of (5) and (7) respectively) by $x_d$ and $x_{nd}$, both in $\mathbb{R}^N$. The democratic and near-democratic encoders are mappings $E_d(\cdot), E_{nd}(\cdot) : \mathbb{R}^n \rightarrow S \subseteq \mathbb{R}^N, |S| \leq 2^{\log |S|}$ defined as:

$$E_d(y) = Q\left( \frac{x_d}{\|x_d\|_\infty} \right), E_{nd}(y) = Q\left( \frac{x_{nd}}{\|x_{nd}\|_\infty} \right). \hspace{1cm} (10)$$

$E_d(\cdot)$ and $E_{nd}(\cdot)$ are quantized outputs and sent over the channel from source (worker) to the destination (parameter server). The corresponding decoder is the same for both, and is defined as the mapping $D(\cdot) : S \rightarrow \mathbb{R}^n$, $D(x) = |x|\|x\|S_x$, where $x$ is either $x_d$ or $x_{nd}$, and $x = E_d(y)$ or $E_{nd}(y)$. Normalization by $\|x\|_\infty$ is needed to ensure that the input to $Q(\cdot)$ lies in $B_\infty^N(1)$. In the following Theorem 1 we show the independence/weak-logarithmic dependence of DSC and NDSC. For simplicity of exposition, here we have assumed that the scalar magnitude $\|x\|_\infty$ is known exactly at the receiver. We can quantize $\|x\|_\infty$ using a constant number of bits. In that case, the total number of bits required to quantize the vector is $nR + O(1)$, which implies $R + \frac{Q(1)}{n}$ bits per dimension, which $\rightarrow R$ as $n \rightarrow \infty$. In Appendix F, we show that this just introduces a small additive constant quantization error and all the results still hold true.

Theorem 1 (Quantization Error of DSC and NDSC): Given $S \in \mathbb{R}^{N \times N}$ and an $R$-bit uniform scalar quantizer $Q(\cdot)$, for any $y \in \mathbb{R}^n$, let $Q_d(y) = D(E_d(y))$ and $Q_{nd}(y) = D(E_{nd}(y))$. Then, with probability at least $1 - e^{-\Omega(n)}$, $\|y - Q_d(y)\|_2 \leq 2^{(1 - \frac{1}{2})\frac{\log(2N)}{n}} \|y\|_2$, and with probability at least $1 - \frac{1}{\Omega(n)}$, $\|y - Q_{nd}(y)\|_2 \leq 2^{(1 - \frac{1}{2})\frac{\log(2N)}{n}} \|y\|_2$. \hspace{1cm} (11)

The proof of Theorem 1 is a direct consequence of Lemmas 1, 2 and 3 and is provided in Appendix C. In the above Theorem 1 we consider a randomized Hadamard frame for near-democratic representation (i.e., Lemma 3). For random orthonormal frames, a $\sqrt{\lambda \log(2N)}$ factor appears instead of $\sqrt{\log(2N)}$. For $\lambda = 1$, Theorem 1 holds for both classes of frames. Choosing $\lambda = 1$ is possible for random orthonormal frames, but not in the case of Hadamard frames for which the dimension $N$ must be such that Hadamard matrix can be constructed. Democratic and near-democratic embeddings provide a unified way of looking at basis transforms for quantization, and can be applied with any general compression scheme.

1 The exact expression depends on the choice of $S$ (with its UP parameters) and is given in Section IV in the supplementary material.
B. Optimal Covering Efficiency of Democratic Source Coding

The notion of covering efficiency is a measure of how close a fixed-length quantizer is to being optimal. Quantizer efficiency is related to how effectively a Euclidean ball of unit radius can be covered with a finite number of smaller balls [28], [29], [30]. We review certain definitions to precisely characterize this notion. Let $B_n^0(\alpha)$ denote the Euclidean ball of radius $\alpha$ centered at the origin. The dynamic range ($r$) of an $R$-bit quantizer $Q : \mathcal{R} \rightarrow \mathcal{R}' \subset \mathbb{R}^n$, $|\mathcal{R}'| \leq 2^{[nR]}$ is defined to be the radius of the largest Euclidean ball which fits inside the domain of $Q$, i.e., $r = \sup\{a|B_n^0(\alpha) \subseteq \mathcal{R}\}$. The covering radius of $Q$ is defined as the maximum possible quantization error when any $x \in B_n^0(r)$ is quantized to its nearest neighbor, i.e., $d(Q) \triangleq \inf\{d > 0|\forall x \in B_n^0(r), \|x - Q(x)\|_2 \leq d\}$. The covering efficiency $\rho_n(Q)$ of $Q : \mathcal{R} \rightarrow \mathcal{R}' \subset \mathbb{R}^n$ is defined as:

$$\rho_n(Q) = \left(\frac{|\mathcal{R}'|}{\text{vol}(B_n^0(d(Q)))}\right)^{\frac{1}{n}} = \|r\|^\frac{1}{n} \cdot \frac{d(Q)}{r}. \quad (15)$$

If we consider Euclidean balls of radius $d(Q)$ around each quantization point, the total volume of these balls must cover $B_n^0(r)$. Covering efficiency formalizes how well this covering is and $\rho_n(Q) \geq 1$ is a natural lower bound. [6] notes that for Roger’s quantizer [9], $\rho_n \rightarrow 1$ as $n \rightarrow \infty$, and is hence asymptotically optimal. However, it is practically infeasible for large $n$ as it cannot be implemented in polynomial time. Popular quantization schemes [8] use uniform scalar quantizers that have $\rho_n = \sqrt{n}$, which grows significantly far away from the lower bound of 1 for large $n$ and are quite suboptimal. The following Lemma 4 quantifies the efficiency of our proposed quantization scheme. Proof is a direct consequence of Theorem 1 and is given in Section V in the supplementary material.

**Lemma 4 [Covering Efficiency of (Near) Democratic Source Coding]:** For the (near) democratic source coding schemes described in Section III-A, with probability at least $1 - \frac{1}{2N}$, the covering efficiencies are given by

$$\rho_d = 2^{1+R \left(1 - \frac{1}{n}\right)} K_a, \text{ and } \rho_{nd} = 2^{2+R \left(1 - \frac{1}{n}\right)} \sqrt{\log(2N)},$$

where $\lambda = N/n$ is the aspect ratio of the frame $S \in \mathbb{R}^{n \times N}$, and $K_a$ is its upper Kashin constant.

Lemma 4 shows that when compared to naive uniform scalar quantizers, DSC and NDSC have remarkably better covering efficiency for large $n$, since it is either dimension independent or has a weak logarithmic dependence. In the next section, we will show that this gives us independence/weak-logarithmic dependence on dimension for distributed optimization under bit-budget constraints.

IV. PROPOSED OPTIMIZATION ALGORITHMS

A. Smooth and Strongly Convex With Exact Gradient Oracle

Consider the class of $L$-smooth and $\mu$-strongly convex objective functions that satisfy $\|\mathbf{x}\|_2 \leq D$ for some known $D \geq 0$, where $\mathbf{x}^* = \arg\min_{x \in \mathcal{F}} f(x)$. For a starting point $\hat{x}_0 \in \mathbb{R}^n$ and step-size $\alpha > 0$, we consider $\Pi_R$ to be the class of $R$-bit Quantized Gradient Descent (QGD) algorithms that iterate the descent rule $\mathbf{x}_t+1 = \mathbf{x}_t - \alpha \mathbf{q}_t$, where the descent direction $\mathbf{q}_t$ is a function of the computed gradients up until iteration $t$ [6]. Due to a bit-budget constraint, $\mathbf{q}_t$ can only take values from a finite set of cardinality $2^{[nR]}$. We consider the class of algorithms $\Pi_R$ to be those in which the worker determines the point $\mathbf{z}_t$ at which the gradient is evaluated, and the quantizer input $\mathbf{u}_t$, taking into account error feedback so that, $\mathbf{z}_t \in \hat{x}_t + \text{span}\{\mathbf{e}_0, \ldots, \mathbf{e}_{t-1}\}$ and $\mathbf{u}_t \in \nabla f(\mathbf{z}_t) + \text{span}\{\mathbf{e}_0, \ldots, \mathbf{e}_{t-1}\}$, where $\mathbf{e}_i = \mathbf{q}_i - \mathbf{u}_i$, $i = 0, \ldots, t-1$ are the past quantization errors. From [6, Th. IV.1], for the class $\Pi_R$ of $R$-bit QGD algorithms as described above, the minimax rate over the function class $\mathcal{F}_{\mu,L,D}$ defined in (2) is lower bounded by $C(\mathcal{R}) \geq \max\{\sigma, 2^{-R}\}$, where $\sigma \triangleq \frac{L-\mu}{L+\mu}$. Here, $\sigma$ is the worst-case linear convergence rate of unquantized gradient descent over the same function class [31]. $C(\mathcal{R})$ has a sharp transition at a threshold budget $R_s = \log(1/\sigma)$. [6] shows that for their proposed algorithm, using scalar quantizers yields a convergence rate of $\leq \max\{\sigma, \sqrt{n}2^{-R}\}$. This means that we require $R \geq \log(\sqrt{n}/\sigma)$ to achieve the convergence rate of unquantized GD, which is far from $R_s$ for large $n$. We propose **DGD-DEF: Distributed Gradient Descent With Democratically Encoded Feedback** (Algorithm 1) which resolves this issue. Here, $E(\cdot)$ can be either $E_d$ or $E_{nd}$. **DGD-DEF** is essentially a modification of the algorithm in [6], with the quantization scheme replaced by our coding scheme(s). Theorem 2 characterizes the convergence rate of **DGD-DEF**. Its proof is similar to [6, Th. 7] and is deferred to Appendix D.

**Theorem 2 (DGD-DEF Convergence Guarantee):** For an objective function $f \in \mathcal{F}_{\mu,L,D}$, a bit-budget of $R$-bits per dimension, with high probability, **DGD-DEF** (Algorithm 1) with step-size $\alpha \leq \alpha^* \triangleq \frac{2}{\mu L^2}$, employing a frame $S \in \mathbb{R}^{n \times N}$ for DSC or NDSC achieves

$$\|\hat{x}_T - \mathbf{x}\|_2 \leq \begin{cases} \max\{v, \beta\} \frac{T}{1 + \beta} \left(1 + \frac{aL}{\|\mathbf{v}\|} D\right), & \text{if } v \neq \beta, \\ vT \left(1 + aLT\right)D, & \text{otherwise}, \end{cases}$$

where $\beta$ is the normalized error as in Theorem 1, i.e., $\beta \triangleq 2^{(1-\gamma/R)} K_a$ if $E = E_d$, and $\beta \triangleq 2^{(2-\gamma/R)} \sqrt{\log(2N)}$ if $E = E_{nd}$, and $v \triangleq \left(1 - (\alpha^* L^2)\alpha^*\right)^{1/2}$ is the convergence rate of unquantized gradient descent with stepsize $\alpha$. 

---

**Algorithm 1 DGD-DEF**

- **Initialize:** $\hat{x}_0 \leftarrow 0$ and $e_{-1} \leftarrow 0$
- **for** $t = 0$ to $T - 1$ **do**
  - **Worker:**
    - $\mathbf{z}_t \leftarrow \hat{x}_t + \alpha e_{t-1}$ (gradient update)
    - $\mathbf{u}_t \leftarrow \nabla f(\mathbf{z}_t) - e_{t-1}$ (error feedback)
    - $\mathbf{v}_t \leftarrow E(\mathbf{u}_t)$ (source encoding)
    - $e_t \leftarrow D(\mathbf{v}_t) - \mathbf{u}_t$ (error for next step)
  - **Server:**
    - $\mathbf{q}_t = D(\mathbf{v}_t)$ (source decoding)
    - $\hat{x}_{t+1} = \hat{x}_t - \alpha \mathbf{q}_t$ (gradient descent step)
- **end for**
- **Output:** $\hat{x}_T$
Algorithm 2 DQ-PSGD

Initialize: \( \tilde{\mathbf{x}}_0 \in \mathcal{X}, \alpha \in \mathbb{R}_+ \) and \( T \)

\[ \text{for } t = 0 \text{ to } T - 1 \text{ do} \]

\[ \text{Worker:} \]
\[ \mathbf{\tilde{g}}_t = \mathbf{g}(\mathbf{\tilde{x}}_t) \quad \text{(noisy subgradient)} \]
\[ \mathbf{v}_t = \mathbf{D}_{\text{dith}}(\mathbf{\tilde{g}}_t) \quad \text{(source encoding)} \]

\[ \text{Server:} \]
\[ \mathbf{q}_t = \mathbf{D}_{\text{dith}}(\mathbf{v}_t) \quad \text{(source decoding)} \]
\[ \mathbf{\tilde{x}}_{t+1} = \mathbf{\tilde{x}}_t - \alpha \mathbf{q}_t \quad \text{(subgradient step)} \]
\[ \mathbf{\tilde{x}}_{t+1} \leftarrow \mathcal{G}(\mathbf{\tilde{x}}_{t+1}) \quad \text{(projection step)} \]

\[ \text{end for} \]

Output: \( \mathbf{x}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{\tilde{x}}_t \)

With \( \alpha = \alpha^*, \lim_{T \to \infty} \sup_{f \in \mathcal{F}_{m,L,D}} \mathbb{E}[\mathbf{x}_T - x^*_T]^{\|/D}^{1/T} = \max\{\|, 2^{-R\beta}\} \). For DSC, \( \beta = \mathcal{O}(1) \) w.h.p., implying DGD-DEF achieves the lower bound of \( \max\{\|, 2^{-R}\} \) to within constant factors, and since \( \beta = \mathcal{O}(\sqrt{\log n}) \) for DSC w.h.p., it is just a weak logarithmic factor away, which is better than \( \sqrt{n} \) scaling of uniform scalar quantizers. In other words, the threshold budget \( R_{\text{th}} = \log(1/\alpha) \) is much less than \( \log(\sqrt{n}/\alpha) \) for large \( n \). Furthermore, compared to [6], which used Roger’s quantizer [9] (exponential complexity), the worst-case complexity of DGD-DEF is polynomial w.r.t. dimension, i.e., \( \mathcal{O}(n^3) \) or \( \mathcal{O}(n^2) \).

B. General Convex and Non-Smooth Objectives With Stochastic Subgradient Oracle

Consider \( f \) to be convex, but not necessarily smooth. The stochastic subgradient oracle output \( \mathbf{g}(\mathbf{x}) \) for any input query \( \mathbf{x} \in \mathcal{X} \) is assumed to be unbiased, i.e.,

\[ \mathbb{E}[\mathbf{g}(\mathbf{x})|\mathbf{x}] \in \partial f(\mathbf{x}), \]

and uniformly bounded, i.e.,

\[ \|\mathbf{g}(\mathbf{x})\|_2 \leq B \text{ for some } B > 0 \]

In this case, an R-bit quantizer is defined to be a set of (possibly randomized) mappings \( \mathbf{Q}^e, \mathbf{Q}^d \) with the encoder \( \mathbf{Q}^e : [0, 1]^n \to [0, 1]^m \) and the decoder \( \mathbf{Q}^d : [0, 1]^m \to \mathbb{R}^n \). To design the source coding scheme for a stochastic subgradient oracle, we consider the class of gain-shape quantizers. For any vector input \( \mathbf{y} \in \mathbb{R}^n \), gain-shape quantizers are of the form,

\[ \mathbf{Q}(\mathbf{y}) \equiv \mathbf{Q}_G(|\mathbf{y}|_2) \cdot \mathbf{Q}_S(\mathbf{y}/|\mathbf{y}|_2), \]

where \( \mathbf{Q}_G : \mathbb{R} \to \mathbb{R} \) and \( \mathbf{Q}_S : \mathbb{R}^n \to \mathbb{R}^n \) quantize the magnitude and shape separately, and multiply the estimates to obtain the quantized output. We consider a uniformly dithered variant of DSC which we denote as \( (\mathbf{E}_{\text{dith}}, \mathbf{D}_{\text{dith}}) \) in Algorithm 2 (cf. Appendix E for detailed description of this quantizer design) for \( \mathbf{Q}_S \), and propose DQ-PSGD: Democratically Quantized Projected Stochastic SubGradient Descent (Algorithm 2). We use a dithered version of DSC instead of the nearest neighbor scheme of Section III because for stochastic oracles, it enables us to attain the optimal minimax rate even without error-feedback. Theorem 3 characterizes the expected suboptimality gap of DQ-PSGD. Its proof is similar to [7, Corollary 3.4] (ref. Appendix E).

**Theorem 3 (DQ-PSGD Convergence Guarantee):** For any general objective \( f \) with access to the oracle \( \mathbf{Q} \circ \mathcal{O} \) which outputs quantized noisy subgradients \( \mathbf{Q}(\mathbf{g}(\mathbf{x})) \), where \( \mathbf{Q} \) employs DSC for the shape quantizer, with a step-size choice of \( \alpha = \frac{2000}{\mathbb{E}\left[\mathcal{O}^{-1} \cdot \frac{K_u dB}{\sqrt{T} \cdot \min[1, T]}\right]} \), the worst case expected suboptimality gap of the output \( \mathbf{x}_T \) of DQ-PSGD after \( T \) iterations is

\[
\sup_{f \in \mathcal{O}} \mathbb{E}[f(x_T) - f(x^*)] \leq \frac{K_u dB}{\sqrt{T} \cdot \min[1, T]}.
\]

Since \( K_u = \mathcal{O}(1) \) w.h.p., Theorem 3 shows that DQ-PSGD achieves the minimax lower bound [7, Th. 2.3, 3.1], using only \( R + o(n) \) bits per dimension. The additional \( o(n) \) bits is for transmitting the scalar magnitude. Compared to [7], DQ-PSGD attains the minimax optimal \( \mathcal{O}(1/\sqrt{T}) \) rate without additional logarithmic multiplicative factors in the bit-budget requirement. A similar result with a weak logarithmic dependence on \( n \) can be derived for NDSC. Section II in the supplementary material (Theorem 1) shows that DSC & NDSC improve performance when used in conjunction with existing compression schemes, such as random sparsification.

C. Extension to Multiple Workers

To extend our algorithm to a setup with multiple workers, consider the following optimization problem over \( m \) workers and a parameter-server (PS):

\[ \mathbf{x}^* \equiv \arg \min_{\mathbf{x} \in \mathcal{X}} \frac{1}{m} \sum_{i=1}^{m} f_i(\mathbf{x}). \]

Here, the objective \( f(x) : \mathcal{X} \to \mathbb{R} \) is the sum of multiple \( f_i(x) : \mathcal{X} \to \mathbb{R} \), each known privately to a corresponding node \( i \). Node \( i \) can compute the gradient \( \nabla f_i(\mathbf{x}) \) (or a subgradient, \( \mathbf{g}_i(\mathbf{x}) \in \partial f_i(\mathbf{x}) \)) for any \( \mathbf{x} \in \mathcal{X} \), and communicate it to the PS. For general convex, non-smooth functions with stochastic subgradient oracle, Algorithm 2 can be extended to multiple workers by incorporating an additional consensus step at the PS. The pseudocode is provided in Algorithm 3. We analyze this setting in more detail in Section III in the supplementary material. With a budget of \( R \) bits per dimension per worker, we show that using a naïve quantizer, the worst-case convergence rate scales as:

\[
\sup_{f,\mathcal{O}} \mathbb{E}[f(x_T) - f(x^*)] \leq \mathcal{O} \left( \frac{1}{\sqrt{mt}} \cdot \frac{\sqrt{n}B}{(2^n - 1)} \right).
\]

The linear dependence of this convergence rate on the dimension \( n \) can be detrimental for high dimensional problems. With our proposed source coding schemes, we can get rid of this and get rates \( \mathcal{O} \left( \frac{1}{\sqrt{mt}} \cdot \frac{K_u dB}{(2^n - 1)} \right) \) and \( \mathcal{O} \left( \frac{1}{\sqrt{mt}} \cdot \frac{\sqrt{\log n}}{(2^n - 1)} \right) \) with DSC and NDSC respectively.

V. Numerical Simulations

We validate our theoretical claims with numerical simulations. Fig. 1(a) plots the normalized compression errors, i.e., \( \mathbb{E}[\|\mathbf{Q}(\mathbf{y}) - \mathbf{y}\|_2 / \|\mathbf{y}\|_2] \) for different compression schemes with and without the presence of near-democratic source coding. The vectors \( \mathbf{y} \in \mathbb{R}^{1000} \) chosen for compression are
Algorithm 3 DQ-PSGD (Multiple Workers)

Initialize: $\hat{x}_0 \in \mathcal{X}$ (at the PS), $\alpha \in \mathbb{R}^+$ and $T$.

for $t = 0$ to $T - 1$ do

Server: Broadcasts $\hat{x}_t$ to all workers $W_k$, $i \in [m]$.

for $i = 1$ to $n$ at $W_k$ do

Compute $\bar{g}^i = \bar{g}(\hat{x}_t)$ (noisy subgradient)

Encode $v^i_t = D_{\text{Dith}}(\bar{g}^i)$ (source encoding)

$W_k$ sends $v^i_t$ to the PS. (Communication)

end for

Server:

$q^i_t = D_{\text{Dith}}(v^i_t)$ for all $i \in [n]$ (source decoding)

$q_t = \frac{1}{n}\sum_{i=1}^n q^i_t$ (consensus step)

$x_{\hat{t}+1} \leftarrow \hat{x}_t - \alpha q_t$ (subgradient step)

$x_{\hat{t}+1} = \Gamma(x_{\hat{t}+1})$ (projection step)

end for

Output: $x_T = \frac{1}{T}\sum_{t=1}^T \hat{x}_t$

---

generated from a standard Gaussian distribution, and then raised to the power of 3 element-wise, and averaged over 50 realizations. This ensures a heavier tail, and hence the entries of $y$ have very different magnitudes. In the legend, SD denotes Standard Dithering [8], Top-K denotes Top-K sparsification [32], and NDH, NDO are abbreviations for Near-Democratic Hadamard/Orthogonal, specifying the type of randomized frame chosen for our coding scheme. Note that for $n = 1000$ dimensions, solving (5) to compute the democratic representation using standard optimization packages like CVX [33] is computationally demanding. Hence, we used [10]'s algorithm to compute Kashin representations, which require explicit knowledge of UP parameters $\eta, \delta$. For the two plots labelled Kashin (with random orthonormal frame), we choose $\lambda = 1.5$ and 1.8, which implies availability of $R/\lambda$ bits per dimension to quantize. Due to the fixed bit-budget, the desired effect of even distribution of information in Kashin representation, is offset by the poorer quantization resolution per coordinate, which results in no net benefit (if not worse). For this reason, in our near democratic representation with orthonormal frame, we choose $\lambda = 1$. We observe that $\lambda$ is desired to be as close to 1 as possible, and for Hadamard frame, we let $N = 2^{\lceil\log_2 n\rceil} = 1024$.

Fig. 1(b) compares the empirical convergence rate, defined as $\|x_T - x^*_T\|/\|x_0 - x^*_T\|_2$, versus the bit-budget constraint, i.e., $R$ bits per dimension, for solving the least squares problem $\min_{x \in \mathbb{R}^n} \frac{1}{2}\|y - Ax\|^2_2$, where $n = 116$, and the entries of $A$ are drawn from Gaussian-cubed as before. If the algorithm does not converge, the empirical rate is clipped at 1. Unquantized GD has a constant rate equal to $\frac{16}{64} [31]$. DQGD proposed in [6] used a predefined sequence of dynamic ranges, and nearest-neighbor scalar quantization. In comparison, we normalize the input to the quantizer by $\|\cdot\|_\infty$ norm, and since it is a scalar quantity, we assume that it is transmitted with infinite precision. A more comprehensive justification for sending scalars can be found in Appendix F. We observe that Near-Democratic Embeddings (NDE) perform at par with Democratic Embeddings (DE), and both ensure convergence at very low bit-budgets. Sometimes, it may even perform better because NDE allows us to choose $N = n$, and hence as seen before, no resolution is lost due to the fixed bit-budget. Moreover, the computational advantage of NDE is evident from Fig. 1(c) where we plot the wall-clock time (in seconds) (averaged over 10 realizations) vs. dimension to find these embeddings. The DE’s are obtained by solving (5) using CVX and the NDE’s are obtained from the closed form expression $x = S^T y$. Here, for each $n$, the value of $N$ is chosen to be $N = 2^{\lceil\log_2 n\rceil}$. This plot was obtained on a Dell Vostro with an Intel i5 1.60GHz processor running MATLAB R2014b. Finally, in Fig. 1(d), we solve the $\ell_2$-regularized least squares
problem for the MNIST dataset [34]. We use gradient descent where the gradients are compressed, first by random sparsification followed by an aggressive 1-bit quantization for the retained coordinates, so that effectively \( R = 0.5 \) bits are used per dimension. We note that NDE’s using random orthonormal frames converge for \( R = 0.5 \), whereas the vanilla compression scheme fails. For least-square simulations, we use the step-size \( \alpha^* \) given by Theorem 2.

For general convex & non-smooth objectives, we train a support vector machine where the subgradients are quantized using \( R \)-bits per dimension. Each worker has \( m \) datapoints \( \{(a_i, b_i)\} \in \mathbb{R}^n \times \{-1, +1\} \) for \( i = 1, \ldots, m \). We want to solve the following optimization problem in which our aim is to minimize the hinge loss: \( \min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \max(0, 1 - b_i \cdot x^\top a_i) \). We compare the performance of our proposed DQ-PSGD algorithm with naive scalar quantization as well as unquantized projected stochastic subgradient descent. The stochasticity in the subgradient oracle evaluation arises from randomly subsampling the dataset to compute the subgradient at every iteration. We consider the number of datapoints \( m = 100 \), and dimension of the problem \( n = 30 \). For Figs. 2(a) and 2(b), the data corresponding to each class is generated independently from Gaussian distributions with different means. We consider random orthonormal frames for computing NDE’s. Fig. 2(a) plots the suboptimality gap (averaged over 10 different realizations) versus the number of iterations. The optimal value \( f^* \) for computing the suboptimality gap in Fig. 2(a) is obtained by using the interior point solver provided by CVX [33]. We effectively have \( R = 0.5 \), i.e., less than one bit per dimension. In other words, since we only have a total of \( nR = 15 \) bits available, the subgradient is randomly sparsified by making certain coordinates zero, and the remaining vector is quantized using 1-bit per dimension. There is a significant difference in performance when randomly sparsifying 50% of the coordinates with and without NDE’s. We also consider top-K sparsification [18] with and without NDE’s. We choose \( K = 3 \), i.e., we decide to retain only the top 10% of the coordinates. When we employ both sparsification and quantization techniques simultaneously to compress the gradient but only have a fixed total number of bits available, there arises a trade-off between how many coordinates we want to retain and the number of bits allotted for quantizing each retained coordinate. Smaller \( K \) means more bits per coordinate, i.e., better resolution for scalar quantization of the retained coordinates, and vice versa. In random sparsification, we retain 15 coordinates with 1-bit allotted for each. For top-K, we retain 3 coordinates and allot 5 bits for quantizing each of them. Although top-K is expected to perform better than random sparsification, choosing the value of \( K \) heuristically may yield poorer performance (despite the additional computation required for determining the top K coordinates) as in this case. We also plot the classification error, that is the percentage of misclassified samples in the training set at every iteration in Fig. 2(b) and observe a similar trend for the different sparsification and quantization schemes.

Figs. 2(c) and 2(d) consider the MNIST dataset [34], and the problem of training an SVM to distinguish the digit 0 from digit 1. Fig. 2(c) shows how the objective function value decreases with the number of iterations. Fig. 2(d) plots the classification error on the hold-out test set for each iteration. We consider only 1 realization for this setting and let \( R = 0.1 \). For top-K, we retain the top 10% coordinates, while ensuring that the total bit-budget remains same for all the schemes, i.e., a total of \( |\nu R| = [784 \times 0.1] = 78 \) bits. For random sparsification with and without NDE’s, 78 coordinates are chosen randomly from the gradient which \( \in \mathbb{R}^{784} \) and 1 bit is allotted to each of them. For top-10%, we now choose to retain the top 78 coordinates of maximum magnitude, allot 1 bit to quantize each of them, and make the rest zero. Since the number of retained coordinates is the same for both random and top-K
Multi-Worker Simulations: We consider two problems. Fig. 3(a) considers a regression model that solves: \( x^* = \arg \min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \left( \frac{1}{s} \sum_{j=1}^{s} (b_{ij} - a^T_{ij} x)^2 \right). \) Here, \( [a_{ij}, b_{ij}]_{j=1}^{s} \) denotes the local dataset at node \( i \), for \( i \in [m] \). The dimension of the problem is \( n = 30, m = 10 \) workers with each worker having \( s = 10 \) local datapoints. The dataset is generated synthetically from a model \( y = x^* \) according to the noisy planted model \( b = Ax^* \), where \( b \in \mathbb{R}^{ms} \) is the regression output and the rows of \( A \in \mathbb{R}^{ms \times n} \), i.e., \( [a_1, \ldots, a_m] \) are the data vectors. We let \( x^* \sim \text{Student-t}(df = 1) \) and the entries of the data matrix \( A \) are iid \( \mathcal{N}(0, 1) \).

Although our theoretical analysis is for convex functions, we also provide simulations for non-convex settings. In Fig. 3(b), we train a convolutional neural network (CNN) over \( m = 10 \) workers to do multi-class classification on the CIFAR-10 [35] image classification dataset that contains 50,000 training and 10,000 test images from 10 classes. The entire dataset is distributed across these workers in a non-i.i.d. fashion, so that each worker has images from at most 2 out of the 10 classes. As can be seen from the plot, with a bit-budget of \( R = 4 \) bits per dimension per worker, our proposed near-democratic source coding (NDSC) scheme with randomized Hadamard frame outperforms naïve quantization with the same bit-budget (\( R = 4 \)) which fails to even converge. As a matter of fact, naïve quantization requires a higher budget of \( R = 6 \) bits per dimension per worker to achieve a performance comparable to that of NDSC. Further detailed simulations are provided in Section III in the supplementary material.

VI. Conclusion

In this work, we show that democratic embeddings can yield minimax optimal distributed optimization algorithms under communication constraints when employed in source coding schemes. For smooth & strongly convex objectives, we propose **DGD-DEF**, which employs error feedback to achieve linear convergence. For the case of general convex & non-smooth objectives, when the output of the stochastic subgradient oracle is quantized using a democratic source coding scheme, **DQ-PSGD** attains minimax optimal convergence rate. We note that although **DSC** theoretically attains minimax optimal performance to within constant factors, computing democratic embeddings can be computation and memory intensive. We also propose a randomized Hadamard construction for fast near-democratic embeddings. Finally, we extend the analysis and simulate our algorithms for multi-worker setups. A potential limitation of the proposed optimization approaches is that the curvature information is not utilized, which we leave as future work.

**APPENDIX A**

**Proof of Lemma 2: Near-Democratic Embeddings With Random Orthonormal Frames**

Let \( [s_i]_{i=1}^{N} \in \mathbb{R}^n \) and \( [\hat{s}_i]_{i=1}^{N} \in \mathbb{R}^N \) denote the columns of \( S \) and \( \hat{S} \) respectively, where \( S, \hat{S} \) are defined in Section II-A. For \( i \in [N] \), since \( \hat{S}^\top S = I_N \), we have,

\[
\|s_i\|_2 \leq \|\hat{Ps_i}\|_2 \leq \|s_i\|_2 = 1.
\]

For any \( y \in \mathbb{R}^n \), let \( \hat{y} = y/\|y\|_2 \). Then,

\[
\|S^\top y\|_\infty = \max_{i \in [N]} |s_i^\top y| = \|y\|_2 \max_{i \in [N]} \|s_i\|_2 \|\hat{s}_i^\top \hat{y}\|
\leq \|y\|_2 \max_{i \in [N]} \|\hat{s}_i^\top \hat{y}\|
\]

where \( \hat{s}_i = s_i/\|s_i\|_2 \). Note that \( \hat{s}_i \in \mathbb{R}^n \) is uniformly random on the unit sphere in \( \mathbb{R}^n \), i.e., \( \hat{s}_i \) has identical distribution as \( g/\|g\|_2 \) where \( g \sim \mathcal{N}(0, I_n) \). Due to rotational invariance of Gaussian distribution, for any fixed \( \hat{y} \in \mathbb{R}^n \) such that \( \|\hat{y}\|_2 = 1 \), \( \hat{s}_i^\top \hat{y} \) has identical distribution as \( \hat{s}_i^\top e_i \), where \( e_i \) is the first canonical basis vector. From concentration of measure for uniform distribution on the unit sphere,

\[
\mathbb{P}\left[ |\hat{s}_i^\top \hat{y}| \geq t \right] = \mathbb{P}[|t_1| \geq t] \leq 2e^{-nt^2/2}.
\]

Using a union bound,

\[
\mathbb{P}\left[ \max_{i \in [N]} |\hat{s}_i^\top \hat{y}| \geq t \right] \leq 2Ne^{-nt^2/2}.
\]

Setting \( t = \sqrt{\frac{\log(2N)}{n}} \) yields,

\[
\mathbb{P}\left[ \|S^\top y\|_\infty \geq 2 \frac{\sqrt{\log(2N)}}{N} \|y\|_2 \right] \leq \frac{1}{2N},
\]

which completes the proof.
APPENDIX B
PROOF OF LEMMA 3: NEAR-DemOCRATIC EMBEDDINGS WITH RANDOMIZED HADAMARD FRAMES

Denote \( z = P^T y = [z_1, \ldots, z_N]^T \in \mathbb{R}^N \), and let \( u = \text{HD}x = [u_1, \ldots, u_N]^T \). Here, \( u_j \) is of the form \( \sum_{i=1}^{N} a_i z_i \), with each \( a_i = \pm \frac{1}{\sqrt{N}} \) chosen i.i.d. For any \( t \in \mathbb{R} \) and \( \lambda > 0 \), a Chernoff-type argument gives,

\[
P[u_j > t] = P[e^{u_j} > e^{t}] \leq e^{-\lambda t} \prod_{i=1}^{N} E[e^{\lambda a_i z_i}].
\]

Now,

\[
E[e^{\lambda a_j z_j}] = \frac{1}{2} e^{\frac{\lambda}{\sqrt{N}}} + \frac{1}{2} e^{-\frac{\lambda}{\sqrt{N}}},
\]

which is upper bounded by hyperbolic cosine. This gives us \( P[u_j > t] \leq e^{rac{\lambda^2}{4N} \| z \|_2^2 - \lambda t} \). Setting \( \lambda = tN/\| z \|_2^2 \) gives the tightest bound,

\[
P[u_j > t] \leq e^{-\frac{t^2}{4}N/(2\| z \|_2^2)}.
\]

Similarly, one can show that,

\[
P[u_j < -t] \leq e^{-\frac{t^2}{4}N/(2\| z \|_2^2)}.
\]

Since \( \| u \|_\infty = \max_{j \in \{1,\ldots,N\}, \nu \in \{\pm 1\}} |u_j| \), union bound gives,

\[
P\left[ \| \text{HD}z \|_\infty > t \right] \leq e^{-2\frac{2N}{\| z \|_2^2} + \log(2N)}.
\]

Setting \( t = 2\| z \|_2 \sqrt{\log(2N)/N} \) yields,

\[
P\left[ \| \text{HD}z \|_\infty \leq 2\| z \|_2 \sqrt{\log(2N)/N} \right] \geq 1 - \frac{1}{N}.
\]

Since \( z = P^T y \implies \| z \|_2 = \| y \|_2 \), this completes the proof.

APPENDIX C
PROOF OF THEOREM 1: QUANTIZATION ERROR: (N)-DSC

Let \( E \) denote either \( E_d \) or \( E_{nd} \). Then, given an input \( y \in \mathbb{R}^n \) to \( E(\cdot) \), let \( x \in \mathbb{R}^N \) be the (near) democratic representation, \( \tilde{x} = x/\| x \|_\infty \) be the normalized input to \( Q(\cdot) \), \( \tilde{x}' \in \mathbb{R}^N \) be the encoder output, and \( y' = D(\tilde{x}') \in \mathbb{R}^n \) be the decoder output. The error incurred after encoding and subsequent decoding is

\[
\| y' - y \|_2 \leq \| x \|_\infty \| S(x' - \tilde{x}) \|_2 \leq \| x \|_\infty \| x' - \tilde{x} \|_2
\]

The last inequality follows since,

\[
\| S(x' - \tilde{x}) \|_2 \leq \| S \|_2 \| x - \tilde{x} \|_2.
\]

Since \( S \) is a Parseval frame, and non-zero eigenvalues of \( S^T S \) are the same as those of \( SS^T = I_n \), we have \( \| S \|_2 = 1 \). To upper bound the quantization error \( \| x' - \tilde{x} \|_2 \), note that if we originally had a total budget of \( nR \)-bits, the number of bits per dimension to uniformly quantize \( \tilde{x} \in \mathbb{R}^N \) is now \( nR/N \), i.e., \( 2^{nR/N} \) quantization points per dimension. From (11),

\[
\| x' - \tilde{x} \|_2 \leq 2^{1-nR/N} \sqrt{N}.
\]

So, if we use \( Q_d \) from Lemma 1,

\[
\| y' - y \|_2 \leq \frac{K_u}{\sqrt{N}} \| y \|_2 2^{1-nR/N} \sqrt{N} = 2^{1-nR/N} \frac{K_u}{\sqrt{N}} \| y \|_2.
\]

Similarly, for \( Q_{nd} \), using Lemma 3,

\[
\| y' - y \|_2 \leq 2 \sqrt{\log(2N)/N} \| y \|_2 2^{1-nR/N} \sqrt{N}.
\]

This completes the proof.

APPENDIX D
PROOF OF THEOREM 2: CONVERGENCE RATE OF DGD-DEF

For an \( R \)-bit Quantized Gradient Descent (QGD) algorithm (defined in Section IV and [6, Definition IV.1]), the minimax convergence rate (2) over the function class \( F_{\mu,L,D} \) is lower bounded as \( C(R) \geq \max(\sigma, 2^{-R}) \), where \( \sigma \triangleq \frac{L}{\mu + \nu} \). The convergence analysis makes use of a recursive invariant satisfied by the trajectory of DGD-DEF. Consider the two descent trajectories: DGD-DEF and unquantized GD with the same step size \( \alpha \), starting at the same location \( \tilde{x}_0 = x_0 \). Then using triangle inequality, at each iteration \( i \in \mathbb{N} \) we have,

\[
\tilde{x}_i = x_i - \alpha \tilde{e}_{i-1} \implies \| \tilde{x}_i - x^* \|_2 \leq \| x_i - x^* \|_2 + \alpha \| \tilde{e}_{i-1} \|_2.
\]

where \( \tilde{e}_{i-1} = 0 \). From algorithm pseudocode 1, note that \( z_i = x_i \), i.e., DGD-DEF computes the gradient at the unquantized trajectory \( \{x_i\}_{i=0}^{\infty} \). Decay of the first term \( \| x_i - x^* \|_2 \) is given by the convergence guarantee of unquantized GD [31], which states that \( \| x_i - x^* \|_2 \leq \nu^i \| x_0 - x^* \|_2 \), where \( \nu \triangleq (1 - (\alpha^2 \mu + \lambda))^1/2 \) is the convergence rate for unquantized GD with step size \( \alpha \). An upper bound to the second term \( \| \tilde{e}_{i-1} \|_2 \) is obtained from our quantization scheme, as per the following auxiliary lemma.

**Lemma 5:** For \( f \in \mathcal{F}_{\mu,L,D} \), at any iteration \( t \), the quantizer input satisfies \( \| u_t \|_2 \leq \| r_t \|_2 \), where the sequence \( \{r_t\} \) is given by \( r_t = LD \sum_{j=0}^{t-1} \nu_{t-j} \). Here \( \nu = 2^{1-R/2} \mu \) and \( \beta = 2^{1-R/2} \mu \) if democratic embeddings are used, and \( \beta = 2^{(2-R/\lambda)} \sqrt{\log(2N)} \) if near-democratic embeddings are used.

**Proof:** This is proved using induction. For \( t = 0 \), we have \( u_0 = \nabla f(x_0) - e_{-1} \). Since \( e_{-1} = 0 \), recalling that \( \nabla f(x^*) = 0 \) and \( f \) is \( L \)-smooth, we have,

\[
\| u_0 \|_2 = \| \nabla f(x_0) \|_2 = \| \nabla f(x_0) - \nabla f(x^*) \|_2 \leq L \| x_0 - x^* \|_2 \leq LD,
\]

and so the lemma holds true for \( t = 0 \). From triangle inequality,

\[
\| u_t \|_2 = \| \nabla f(x_t) - e_{t-1} \|_2 \leq \| u_0 \|_2 + \| e_{t-1} \|_2.
\]

The first term can be upper bounded as,

\[
\| \nabla f(x_t) \|_2 \leq \| \nabla f(x_t) - \nabla f(x^*) \|_2 \leq L \| x_t - x^* \|_2 \leq \nu^t \| x_0 - x^* \|_2 \leq \nu^t LD.
\]

Furthermore, from Theorem 1,

\[
\| e_{t-1} \|_2 \leq \| u_{t-1} - D(E(u_{t-1})) \|_2.
\]
where $\beta$ depends on whether we choose democratic or near-democratic embeddings for our source coding scheme, and the second inequality is the induction hypothesis. Using these, we can upper bound the magnitude of the quantizer input as,

$$
\|u_t\|_2 \leq LD \left( v^T + \sum_{j=0}^{t-1} v^j \beta^{t-1-j} \right) = LD \sum_{j=0}^{t-1} v^j \beta^{t-1-j}.
$$

This completes the proof.

The proof of Theorem 2 is similar to the [6, Th. III.1] except for appropriate modifications due to our proposed source coding schemes: DSC and NDSC. Since for $t \in [N],$

$$
\|\hat{x}_t - x^*\|_2 \leq \|x_t - x^*\|_2 + \alpha \|\hat{e}_{t-1}\|_2,
$$

the first term can be upper bounded from the descent guarantee of unquantized GD trajectory as,

$$
\|x_t - x^*\|_2 \leq v^T \|x_0 - x^*\|_2 \leq v^T D.
$$

The second term can be upper bounded as

$$
\|e_{t-1}\|_2 \leq \beta \|u_{t-1}\|_2 \leq \beta r_{t-1} = \beta LD \sum_{j=0}^{t-1} v^j \beta^{t-1-j},
$$

where the inequalities follow from the definition of $\beta$ and Lemma 5. So we have, $\|\hat{x}_t - x^*\|_2 \leq bD,$ where $b = v^T + \beta aL \sum_{j=0}^{t-1} v^j \beta^{t-1-j}.$ There are now three possibilities:

(i) $v > \beta$: The geometric sum is computed as,

$$
b = v^T + \beta aL v^{-1} \sum_{j=0}^{T-1} \frac{(\beta/v)^j}{1 - \beta/v} \leq v^T \left( 1 + \beta \frac{aL}{v - \beta} \right).
$$

(ii) $v = \beta$: In this case,

$$
b = v^T + \alpha L v \cdot v^{T-1} T = v^T (1 + \alpha L T).
$$

(iii) $v < \beta$: The case parallels the first case by interchanging the role of $v$ and $\beta$, and we get,

$$
b = v^T + \beta aL \beta^{-1} \sum_{j=0}^{T-1} \left( \frac{v}{\beta} \right)^j \leq v^T + \alpha L \beta \frac{1 - v^{T-1}}{1 - v/\beta} \leq \beta^T \left( 1 + \beta \frac{aL}{\beta - v} \right)
$$

the proof is complete by concisely expressing the above three cases as:

$$
\|\hat{x}_t - x^*\|_2 \leq \begin{cases} 
\max\{v, \beta\}^T \left( 1 + \beta \frac{aL}{\beta - v} \right) D, & \text{if } v \neq \beta, \\
v^T (1 + \alpha L T) D & \text{otherwise}.
\end{cases}
$$

### Appendix E

#### PROOF OF THEOREM 3: CONVERGENCE RATE OF DQ-PSGD

Consider the optimization problem: $\min_{x \in \mathcal{X}} f(x),$ where, $\mathcal{X} \subseteq \mathbb{R}^n$ is a convex set, and $f$ is convex, but not necessarily smooth. $\mathcal{X}$ satisfies $\sup_{x \in \mathcal{X}} \|x - y\|_2 \leq D$ for some known $D \geq 0.$ We assume oracle access to noisy subgradients of $f.$ The oracle output $\hat{g}(x)$ for any input query point $x \in \mathcal{X}$ to be unbiased, i.e., $\mathbb{E}[\hat{g}(x)] = \partial f(x),$ where, $\partial f(x)$ denotes the subdifferential of $f$ at the point $x,$ and uniformly bounded, i.e., $\|\hat{g}(x)\|_2 \leq B$ for all $x,$ for some $B > 0.$ An R-bit quantizer is defined to be a (possibly randomized) pair of mappings $(Q^r, Q^d),$ with the encoder mapping $Q^r : \mathbb{R}^n \to [0, 1]^n,$ and the decoder mapping $Q^d : [0, 1]^n r \to \mathbb{R}^n.$ Let $Q_R$ denote the set of all such R-bit quantizers. For any pair $(f, \mathcal{O})$ of objective function $f$ and oracle $\mathcal{O},$ and an R-bit quantizer $Q,$ let $Q \circ \mathcal{O}$ denote the composition oracle that outputs $Q(\hat{g}(x))$ for each query $x \in \mathcal{X}.$ Let $\pi \in \Pi_f$ be an optimization protocol as defined in Section I. We consider the minimax expected suboptimality gap (4). The convergence rate of DQ-PSGD depends on the quantizer design of $Q \in Q_R.$

The performance of any quantizer is determined by the following two quantities: The worst-case second moment, i.e.,

$$
A_Q \triangleq \sup_{y \in \mathbb{R}^n : \|y\|_2 \leq B} \mathbb{E}[(\|Q(y)\|_2)^2],
$$

and the worst-case bias, i.e.,

$$
B_Q \triangleq \sup_{y \in \mathbb{R}^n : \|y\|_2 \leq B} \mathbb{E}[\|y - Q(y)\|_2].
$$

For any such quantizer $Q,$ from [7, Th. 2.4], the worst-case expected suboptimality gap of quantized projected subgradient algorithm after $T$ iterations, with step-size $\alpha = \frac{D}{A_Q \sqrt{T}},$ is

$$
\sup_{Q \in Q_R} \mathbb{E}[f(x) - f(x^*)] \leq D \left( \frac{A_Q}{\sqrt{T}} + B_Q \right). \tag{19}
$$

We design $Q$ so that for any input $y \in \mathbb{R}^n,$ $Q$ encodes the magnitude (gain) of the input $\|y\|_2$ and the direction (shape) $y_S = \frac{y}{\|y\|_2}$ of $y,$ separately, and forms the estimate of $y$ by multiplying the estimates for the magnitude and direction. In other words,

$$
Q(y) \triangleq Q_G(\|y\|_2) \cdot Q_S \left( \frac{y}{\|y\|_2} \right),
$$

where $Q_G : \mathbb{R} \to \mathbb{R},$ and $Q_S : \mathbb{R}^n \to \mathbb{R}^n.$

It is assumed that given $y,$ $Q_G$ and $Q_S$ are independent of each other. From [7, Th. 4.2], if $Q_S$ is unbiased, i.e., $\mathbb{E}[Q_S(y_S)] = y_S$ for all $y_S$ satisfying $\|y_S\|_2 \leq 1,$ then,

$$
B_Q \leq \sup_{y \in \mathbb{R}^n : \|y\|_2 \leq B} \mathbb{E}[Q_G(\|y\|_2) - \|y\|_2].
$$

The uniformly dithered quantizer for $Q_G$ is described next. Let the dynamic range of $Q_G$ be the known uniform upper bound, $B.$ Consider $m$ quantization points $\{u_1, \ldots, u_m\}$ uniformly spaced along the interval $[0, B]$ and let $u_0 = 0$ and $u_{m+1} = B.$ For any input $v \in [u_i, u_{i+1}) \subseteq [0, B],$ the output of the gain quantizer $Q_G(v)$ is defined to be:

$$
Q_G(v) = \begin{cases} 
u_i & \text{with probability } r, \\
u_{i+1} & \text{with probability } 1 - r,
\end{cases}
$$

(20)
where, $r \triangleq \frac{s_{n+1} \cdots s_m}{(b/m+1)^n}$. If a fixed number of $b = \log_2 m$ bits (typically 32) are used, it can be easily shown that $Q_S$ is unbiased. For designing $Q_S$, we consider two separate cases: The high-budget regime ($R > 1$) and the sub-linear budget regime ($R < 1$). The proof Theorem 3 is completed after combining the results of Appendices E-A and E-B.

A. High-Budget Regime

Let the input to $Q_S$ be $y$ such that $\|y\|_2 \leq 1$. For $S \in \mathbb{R}^{m \times N}$, if $x_j$ denotes the diagonal embedding of $y$ with respect to $S \in \mathbb{R}^{m \times N}$, then $\|x_j\|_\infty \leq \frac{1}{\sqrt{N}} \|y\|_2 = \frac{1}{\sqrt{N}}$. Let the coordinate-wise uniformly dithered quantizer ($Q_{CUQ}$) be as in [7], in which we do dithered quantization (20) of each coordinate of $x_j$ independently, using $R$ bits per dimension and a dynamic range of $[-\frac{K_u}{\sqrt{N}}, +\frac{K_u}{\sqrt{N}}]$. The output is $Q_S(y) = SQ_{CUQ}(x_j)$. Since $Q_{CUQ}$ is unbiased, the output of $Q_S$ is also unbiased, i.e., $E[Q_S(y)] = E[SQ_{CUQ}(x_j)] = \mathbb{E}[Q_{CUQ}(x_j)]= x_d = y$. Since both $Q_G$ and $Q_S$ are conditionally independent of each other, the bias of $Q(\cdot) = Q_G(\cdot) + Q_S(\cdot) = 0$, i.e., the worst-case bias $B_Q = 0$. To evaluate the worst-case second moment of $Q$, from [7, Th. 4.2], $A_Q \leq A_{Q_G}A_{Q_S}$. Since the dynamic range of $Q_S$ is $B$, the worst-case bias $A_{Q_S} \leq B$. For $Q_S$ and any $y$ such that $\|y\|_2 \leq 1$, we have $\|Q_S(y)\|_2^2 = \|SQ_{CUQ}(x_j)\|_2^2 \leq \sigma_{\text{max}}^2(S) \cdot \|Q_{CUQ}(x_j)\|_2^2 \leq \|Q_{CUQ}(x_j)\|_2^2$. The final inequality follows since $SS^\top = I_n$. Thus, the dynamic range for $Q_S$ is $\frac{K_u}{\sqrt{N}}$, we have $\|Q_S(y)\|_2^2 \leq \frac{K_u}{\sqrt{N}} \Rightarrow \|Q_{CUQ}(x_j)\|_2^2 \leq K_u^2$ for all $y$ $\Rightarrow \|Q_{CUQ}(x_j)\|_2 \leq K_u \Rightarrow A_{Q_S} \leq K_u$. So, the worst-case second moment is $A_Q = \sqrt{\mathbb{E}[\|Q_S(y)\|_2^2]} \leq BK_u$. Substituting these values for $A_Q$ and $B_Q$ in (19), we get the result. A similar result with a $O(\sqrt{\log_2 n})$ dependence on dimension can be proved for NDSC.

B. Sub-Linear Budget Regime

When $R < 1$, the total bit-budget is $r = nR \leq n$, i.e., we have less than 1-bit per coordinate. For brevity, let $N = n$. The statements here can be easily generalized to the case of $N > n$. To allocate $r = nR < n$ bits to each coordinate of a vector in $\mathbb{R}^n$ so that on an average $R$-bits per dimension is utilized, we choose $r = nR$ coordinates uniformly at random; allocate 1-bit to each of these coordinates, and allocate 0-bit to the remaining coordinates. This essentially subsamples the vector in $\mathbb{R}^n$ to a smaller dimensional vector in $\mathbb{R}^R$, and subsequently doing a 1-bit quantization of the vector in $\mathbb{R}^R$, while decoding the other coordinates as 0. Since and we are subsampling and want $Q_S$ to be unbiased, we need to scale the quantized output by a factor of $\frac{1}{R}$. So, the coordinate representation + subsampling + 1-bit quantization scheme is $Q_S(y) = \frac{1}{R} S \sum_{i \in S} Q_{CUQ}(x_j)e_i = \frac{1}{R} S \sum_{i=1}^N Q_{CUQ}(x_j)e_i I_{i \in S}$. Here, $e_i \in \mathbb{R}^n$ is the $i^{th}$ canonical basis vector, $S$ with $|S| = nR$ denotes the random $nR$ indices chosen in the subsampling step, and $I_{i \in S}$ denotes the indicator function, since the coordinates not selected in the subsampling step are decoded as 0. Unbiasedness ensures that $B_Q = 0$. Moreover, $\mathbb{E}[\|Q_S(y)\|_2^2] = \frac{1}{R^2} \mathbb{E} \left[ \left\| \sum_{i \in S} Q_{CUQ}(x_j)e_i \right\|^2 \right] = \frac{1}{R^2} \sum_{i=1}^n \mathbb{E}[Q_{CUQ}(x_j)^2] \cdot \mathbb{E}[I_{i \in S}] = \frac{1}{R^2} \mathbb{E}[\|Q_{CUQ}(x_j)\|_2^2]$. The last equality follows as sampling $nR$ coordinates uniformly at random from $n$ coordinates implies $\mathbb{E}[I_{i \in S}] = R$. Since $\mathbb{E}[\|Q_{CUQ}(x_j)\|_2^2] \leq K_u^2$, the worst-case second moment is $A_Q \leq \sqrt{\mathbb{E}[\|Q(y)\|_2^2]} \leq \frac{BK_u}{\sqrt{R}}$. Substituting these values of $A_Q$ and $B_Q$ in (19), the convergence rate when $R < 1$ is $K_u DB / \sqrt{R}$. This completes the proof.

APPENDIX F

QUANTIZING THE $\ell_\infty$ NORM

For simplicity, the statement of Theorem 2 assumes that the $\ell_\infty$ norm of the input to the quantizer can be transmitted perfectly without lossy quantization. If we use a constant number $O(1)$ of bits (typically, 32 bits depending on the machine precision) to uniformly quantize $\|x\|_\infty$, the total number of bits required to quantize the vector is $nR + O(1) \rightarrow R + O(1) \rightarrow R$ bits per dimension as $n \rightarrow \infty$. Hence, the bit-budget is respected asymptotically and the additional constant number of bits is negligible for high-dimensional problems.

To take account the error due to quantizing $\|x\|_\infty$, note that if $y = \frac{1}{\sqrt{n}}x$,

$\|Q(y) - y\|_2 = \|Q_G(\|y\|_\infty)Q_S(y_S) - \|y\|_\infty y_S\|_2 = \|Q_G(\|y\|_\infty + \epsilon)Q_S(y_S) - \|y\|_\infty y_S\|_2 \leq \|y\|_\infty \|Q_S(y_S) - y_S\|_2 + \epsilon \|Q_S(y_S)\|_2$.

The last inequality follows from triangle inequality. Since we employ DSC for the shape quantizer, the first term can be upper bounded using Theorem 1. The second term can be upper bounded as,

$\epsilon \|Q_S(y_S)\|_2 \leq \epsilon \sqrt{n} \|Q_S(y_S)\|_\infty \leq \epsilon \sqrt{n} \frac{K_u}{\sqrt{N}} \leq \epsilon K_u$,

which is an additional constant error. The whole convergence analysis follows through with this constant additive error too.

REFERENCES

[1] R. Bekkerman, M. Bilenko, and J. Langford, Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge, MA, USA: Cambridge Univ. Press, 2011.
[2] P. Kairouz et al., “Advances and open problems in federated learning,” Found. Trends Mach. Learn., vol. 14, nos. 1–2, pp. 1–210, 2021.
[3] M. Li et al., “Scaling distributed machine learning with the parameter server,” in Proc. 11th USENIX Conf. Oper. Syst. Des. Implement., 2014, pp. 583–598.
[4] M. M. Amiri and D. Gündüz, “Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,” IEEE Trans. Signal Process., vol. 68, pp. 2155–2169, Mar. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9042352
[5] R. Saha, S. Rini, M. Rao, and A. J. Goldsmith, “Decentralized optimization over noisy, rate-constrained networks: Achieving consensus by communicating differences,” IEEE J. Sel. Areas Commun., vol. 40, no. 2, pp. 449–467, Feb. 2022.
[6] C.-Y. Lin, V. Kostina, and B. Hassibi, “Differentially quantized gradient descent,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), 2021, pp. 1200–1205.
[7] P. Mayekar and H. Tyagi, “RATQ: A universal fixed-length quantizer for stochastic optimization,” in Proc. 23rd Int. Conf. Artif. Intell. Stat., 2020, pp. 1399–1409.
[8] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via gradient Quantization and encoding,” in Advances in Neural Information Processing Systems, vol. 30. Red Hook, NY, USA: Curran Assoc., Inc., 2017, pp. 1709–1720.

[9] C. A. Rogers, “Covering a sphere with spheres,” Mathematika, vol. 10, no. 2, pp. 157–164, 2010.

[10] Y. Lyubarskii and R. Vershynin, “Uncertainty principles and vector quantization,” IEEE Trans. Inf. Theory, vol. 56, no. 7, pp. 3491–3501, Jul. 2010.

[11] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, “Distributed mean estimation with limited communication,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 3329–3337.

[12] A. Abdi and F. Fekri, “Quantized compressive sampling of stochastic gradients for efficient communication in distributed deep learning,” in Proc. 36th Int. Conf. Inf. Sci. and Signal Process. (CISS), 2020, pp. 3105–3110.

[13] R. Hadad and U. Erez, “Dithered quantization via orthogonal transformations,” IEEE Trans. Signal Process., vol. 64, no. 22, pp. 5887–5900, Nov. 2016.

[14] J. Bernstein, Y.-X. Wang, K. Azzizadeneshali, and A. Anandkumar, “signSGD: Compressed optimisation for non-convex problems,” in Proc. 35th Int. Conf. Mach. Learn., 2018, pp. 559–568.

[15] I. Gottlieb, E. Rehman, S. Stich, and M. Jaggi, “Error feedback fixes signSGD and other gradient compression schemes,” in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 3252–3261.

[16] W. Wen et al., “TernGrad: Ternary gradients to reduce communication in distributed deep learning,” in Advances in Neural Information Processing Systems, vol. 30. Red Hook, NY, USA: Curran Assoc., Inc., 2017, pp. 1509–1519.

[17] V. GuptGoyal, D. Kane, R. K. Maity, and A. Mazumdar, “vqSGD: Vector quantized stochastic gradient descent,” in Proc. 24th Int. Conf. Artif. Intell. Stat., 2021, pp. 2197–2205.

[18] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, “Sparserfied SGD with memory,” in Proc. 32nd Int. Conf. Neural Inf. Process. Syst., 2019, pp. 4452–4463.

[19] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” in Advances in Neural Information Processing Systems. Red Hook, NY, USA: Curran Assoc., Inc., 2018.

[20] P. Mayekar and H. Tyagi, “Limits on gradient compression for stochastic optimization,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), 2020, pp. 2658–2663.

[21] B. S. Kashin, “Diameters of some finite-dimensional sets and classes of smooth functions,” Math. USSR-Izvestiya, vol. 11, no. 2, pp. 317–333, Apr. 1977.

[22] E. J. Candes and T. Tao, “Decoding by linear programming,” IEEE Trans. Inf. Theory, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[23] W.-N. Chen, P. Kairouz, and A. Özgür, “Breaking the communication-privacy-accuracy trilemma,” in Proc. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 3316–3327.

[24] P. Safaryan, A. G. Markopoulou, and P. Richtárik, “Uncertainty principle for communication compression,” in Proc. 31st Int. Conf. Mach. Learn., 2014, pp. 1009–1017.

[25] A. J. Goldsmith (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, where he is currently the Harris Professor Emerita. She is also the Dean of Engineering and Applied Science and the Arthur LeGrand Doty Professor of Electrical Engineering with Princeton University. Her research interests broadly lie in identifying and addressing problems related to distributed learning over resource-constrained edge devices. During his time at IIT Kharagpur, he worked on topics of MIMO radar imaging which received the Best Undergraduate Thesis Award. He was the recipient of the Prime Minister of India Gold Medal for being the class valedictorian.

Rajarshee Saha (Student Member, IEEE) received the bachelor’s and master’s degrees from the Indian Institute of Technology (IIT) Kharagpur, with a major in electronics and electrical communications engineering, a minor in computer science, and a micro-specialization in embedded wireless systems. He is currently pursuing the Ph.D. degree in electrical engineering from Stanford University. His research interests broadly lie in identifying and addressing problems related to distributed learning over resource-constrained edge devices. During his time at IIT Kharagpur, he worked on topics of MIMO radar imaging which received the Best Undergraduate Thesis Award. He was the recipient of the Prime Minister of India Gold Medal for being the class valedictorian.

Mert Pilanci (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Bilkent University, and the Ph.D. degree in electrical engineering and computer science from UC Berkeley in 2016. He is an Assistant Professor with the Department of Electrical Engineering, Stanford University. Prior to joining Stanford, he was an Assistant Professor of Electrical Engineering and Computer Science, the University of Michigan. In 2017, he was a Math+X Postdoctoral Fellow working with Emmanuel Candès, Stanford University. His Ph.D. advisors were Laurent El Ghaoui and Martin Wainwright, and his studies were supported partially by a Microsoft Research Ph.D. Fellowship.

Andrea J. Goldsmith (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from UC Berkeley. She was a Stephen Harris Professor of Engineering and a Professor of Electrical Engineering with Stanford University, where she is currently the Harris Professor Emerita. She is also the Dean of Engineering and Applied Science and the Arthur LeGrand Doty Professor of Electrical Engineering with Princeton University. Her research interests include information theory, communication theory, and signal processing, and their application to wireless communications, interconnected systems, and neuroscience. She has received several awards for her work, including the Marconi Prize, the IEEE Sumner Technical Field Award, the ACM Athena Lecturer Award, the ComSoC Armstrong Technical Achievement Award, the Kirchmayer Graduate Teaching Award, the WCE Mentoring Award, and the Silicon Valley/San Jose Business Journal’s Women of Influence Award. She founded and served as the Chief Technical Officer of Plume WiFi (formerly Accela, Inc.) and Quantenna, Inc. At Stanford University, she has served as the Chair of Stanford’s Faculty Senate and multiple terms as a Senator, and on its Academic Council Advisory Board, Budget Group, Committee on Research, Planning, and Policy Board, Commissions on Graduate and Undergraduate Education, Faculty Women’s Forum Steering Committee, and Task Force on Women and Leadership. She served as the President of the IEEE Information Theory Society in 2009 and the Founding Chair of the IEEE Information Theory Society’s Student Committee. She currently serves the Board of Directors of Intel, Medtronic, and Crown Castle Inc. She is the Founding Chair of the IEEE Board of Directors Committee on Diversity, Inclusion, and Ethics.