Efficient Randomized Subspace Embeddings for Distributed Optimization under a Communication Budget

Rajarshi Saha, Mert Pilanci, and Andrea J. Goldsmith

***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 $O\left(n^2\right)$ 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., $O\left(n \log n\right)$ 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 here.**

***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 $\min_{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n} f(\mathbf{x})$ in the parameter-server framework [3], the *server* maintains an iterate $\mathbf{x}_t$ at time t, which is an estimate of the minimizer $\mathbf{x}^*_f = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$. We consider a setting in which at every iteration, the worker(s) receives the current iterate $\mathbf{x}_t$ from the server, computes $\nabla f(\mathbf{x}_t)$, 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 $\{\mathbf{x}_t\}_{t=1,2,\ldots}$ 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:

(i) When the objective function $f(\mathbf{x})$ is L-smooth and μ-strongly convex.
(ii) When $f(\mathbf{x})$ 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 Π_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 $\{\mathbf{x}_t\}$ to $\mathbf{x}^*_f$. With this in mind, to quantify the performance of any protocol $\pi \in \Pi_R$, consider the asymptotic worst-case linear convergence rate:

$$C(\pi) = \limsup_{T \to \infty} \sup_{f \in \mathcal{F}_{\mu,L,D}} \left(\frac{\|\mathbf{x}_T(\pi) - \mathbf{x}^*_f\|_2}{D} \right)^{\frac{1}{T}}. \quad (1)$$

Here, $\mathcal{F}_{\mu,L,D}$ is the class of L-smooth and μ-strongly convex functions which satisfy $\|\mathbf{x}^*_f\|_2 \leq D$ ($D \geq 0$), and $\mathbf{x}_T(\pi)$ denotes the output of the protocol $\pi \in \Pi_R$ after T iterations. Note that (1) implies π achieves a convergence of the iterates $\{\mathbf{x}_t\}$ to $\mathbf{x}^*_f$ with a guarantee $\|\mathbf{x}_T(\pi) - \mathbf{x}^*_f\|_2 \lesssim C(\pi)^T D$ for $C(\pi) \leq 1$. This implies that algorithms in the class Π_R require $O\left(\log\left(\frac{D}{\epsilon}\right) / \log\left(C(\pi)^{-1}\right)\right)$ iterations to achieve a suboptimality gap of ϵ. Since a smaller value of $C(\pi)$ is desirable for faster convergence, we can characterize the set

This work was partially supported by the Office of Naval Research under grant ONR N00014-18-1-2191, National Science Foundation under grants DMS-2134248 and ECCS-2037304, Army Research Office Early Career Award W911NF-21-1-0242, Intel Research, Huawei Technologies, Facebook Research, Adobe Research, and Stanford SystemX Alliance.

Rajarshi Saha and Mert Pilanci are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

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

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes a document containing further analyses, proofs, discussions and additional simulations. Contact Rajarshi Saha (email: rajsaha@stanford.edu) for further questions about this work.

of protocols Π_R according to the following *minimax rate*:

$$C(R) \triangleq \inf_{\pi \in \Pi_R} C(\pi)$$
$$= \inf_{\pi \in \Pi_R} \limsup_{T \to \infty} \sup_{f \in \mathcal{F}_{\mu,L,D}} \left(\frac{\|\mathbf{x}_T(\pi) - \mathbf{x}_f^*\|_2}{D} \right)^{\frac{1}{T}}. \quad (2)$$

It has been shown [6] that the information-theoretic lower bound on (2) can be obtained as $C(R) \geq \max\{\sigma, 2^{-R}\}$ where $\sigma = \frac{L-\mu}{L+\mu}$ is the convergence rate in the absence of any bit-budget constraints. This provides a theoretical limit to the performance of **any** protocol in Π_R. 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 $O\left(\sqrt{\log n}\right)$ 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 $C(R) \geq \max\{\sigma, 2^{-R}\}$.

On the other hand for setting (ii), when $f(\mathbf{x})$ 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 π, we consider the *expected suboptimality gap*,

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

and study how it scales with the number of iterations T. Here, $\mathbf{x}_T(\pi)$ is the output of protocol π, 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 $\sup_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_2 \leq D$, and stochastic subgradient oracles $\mathcal{O}$ whose outputs are uniformly bounded by some parameter B. We consider algorithms π from the class $\Pi_{T,R}$ 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 $\Pi_{T,R}$ can be characterized according to the following *minimax expected suboptimality gap*,

$$\mathcal{E}(T,R) \triangleq \inf_{\pi \in \Pi_{T,R}} \mathcal{E}(\pi) = \inf_{\pi \in \Pi_{T,R}} \sup_{(f,\mathcal{O})} \mathbb{E}[f(\mathbf{x}_T(\pi))] - f(\mathbf{x}^*). \quad (4)$$

A lower bound on (4) can be obtained [7] as $\mathcal{E}(T,R) \geq \frac{cDB}{\sqrt{T \cdot \min\{1,R\}}}$. We propose **DQ-PSGD** and its relaxed version, that respectively, attain this lower bound to within constant and mild logarithmic factors while requiring $O(n^2)$ multiplications and $O(n \log n)$ additions.

A minimax optimal protocol (or algorithm) π^* 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* $\mathsf{E} : \mathbb{R}^n \to \{0,1\}^{nR}$ is done by the worker to quantize the information it wants to send to the server. The *decoding* map $\mathsf{D} : \{0,1\}^{nR} \to \mathbb{R}^n$ 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 $C(R) \geq \max\{\sigma, 2^{-R}\}$ implies that we cannot hope to achieve the convergence rate of unquantized setting if $R < \log\left(\frac{1}{\sigma}\right)$. Whereas naive quantizers [8] would require $R_{thr} \gtrsim \log\left(\frac{\sqrt{n}}{\sigma}\right)$ bits to attain the unquantized convergence rate, **DGD-DEF** gets rid of the dimension dependence, and requires just $R_{thr} = O\left(\log\left(\frac{1}{\sigma}\right)\right)$ bits. Moreover, it achieves this while entailing only a polynomial complexity of $O\left(n^2\right)$ as opposed to Roger's quantizer [9] used in [6] that demands exponential complexity. For setting (ii), the lower bound of $\mathcal{E}(T,R) \geq \frac{cDB}{\sqrt{T \cdot \min\{1,R\}}}$ 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 $R_{thr} = O\left(\sqrt{n}\right)$ for naive quantizers or $R_{thr} = O\left(\log \log n\right)$ 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 Kashin 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**: **D**istributed **G**radient **D**escent with **D**emocratically **E**ncoded **F**eedback, 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**: **D**emocratically **Q**uantized **P**rojected **S**tochastic sub**G**radient **D**escent that achieves the lower bound on (4).

(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**: **N**ear **D**emocratic **S**ource **C**oding, which achieves optimality to within a mild logarithmic factor. We observe that in simulations, **NDSC** performs at par with **DSC**.

(e) Finally, in §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 (§V and Supp. §2).

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]. [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-PSGD** 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 is beyond the algorithm designer's control 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 ℓ_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 $\mathbf{y} \in \mathbb{R}^n$, is a vector $\mathbf{x} \in \mathbb{R}^N$ $(N \geq n)$, which has the property that the components of $\mathbf{x}$ 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 $\mathbf{y}$ may be arbitrarily varying in magnitude, Kashin embeddings have an effect of evenly distributing this variation across different components of $\mathbf{x}$. Subsequently, applying lossy compression schemes (eg. quantization) to the democratic embedding $\mathbf{x}$ instead of the original vector $\mathbf{y}$ incurs less error. The inverse embedding map $\mathbf{S} : \mathbb{R}^N \to \mathbb{R}^n$ is linear, i.e., $\mathbf{y} = \mathbf{S}\mathbf{x}$. Usually, $\mathbf{S}$ is randomly generated and the properties of the democratic embeddings are very closely related to *Restricted Isometry Property (RIP)* parameters of $\mathbf{S}$ [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]–[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 $\mathbf{S}$, which is not readily available. To this end, [27] introduced the notion of **democratic embeddings (DE)**. **DE** of a vector $\mathbf{y} \in \mathbb{R}^n$ 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 $\mathbf{S} \in \mathbb{R}^{n \times N}$, where $n \leq N$. For any given vector $\mathbf{y} \in \mathbb{R}^n$, the system of equations $\mathbf{y} = \mathbf{S}\mathbf{x}$ is under-determined in $\mathbf{x} \in \mathbb{R}^N$, with the set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^N \mid \mathbf{y} = \mathbf{S}\mathbf{x}\}$ as the solution space. The vector $\mathbf{x}^* \in \mathcal{S}$ which has the minimum ℓ_∞-norm in this solution space is referred to as the **Democratic Embedding** of $\mathbf{y}$ with respect to $\mathbf{S}$. In other words, $\mathbf{x}^*$ is obtained by solving,

$$\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_\infty \quad \text{subject to} \quad \mathbf{y} = \mathbf{S}\mathbf{x}. \tag{5}$$

The constraint set $\mathcal{S}$ can be relaxed to a larger set $\mathcal{S}' = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{y} - \mathbf{S}\mathbf{x}\|_2 \leq \epsilon\}$ 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 $\mathbf{S} \in \mathbb{R}^{n \times N}$ with $n \leq N$ is called a **frame** if $A\|\mathbf{y}\|_2^2 \leq \|\mathbf{S}^\top \mathbf{y}\|_2^2 \leq B\|\mathbf{y}\|_2^2$ holds for any vector $\mathbf{y} \in \mathbb{R}^n$ with $0 < A \leq B < \infty$, where A and B are called **lower** and **upper frame bounds** respectively.*

Compression Scheme	No. of Bits	Error	Complexity
Sign quantization [14], [15]	$O(n)$	$O(n)$	$O(n)$
QSGD [8]	$O(2^R(2^R+\sqrt{n}))$	$\min\{\sqrt{n}2^{-R}, n2^{-2R}\}$	$O(n)$
Ternary quantization [16]	$O(n\log_2 3)$	$O(n)$	$O(n)$
vqSGD-Gaussian [17]	$O\left(c\right), c > \log n$	$O\left(\frac{n}{c}\right)$	$O(\exp(c))$
vqSGD-Cross Polytope [17]	$O\left(\log n\right)$	$O\left(n\right)$	$O(n)$
Top-k sparsification [18]	$O\left(k+\log_2\binom{n}{k}\right)$	$(n-k)/n$	$O(k{+}(n{-}k)\log_2 k)$
Random sparsification [19]	$O\left(k+\log_2\binom{n}{k}\right)$	$O\left(n/k\right)$	$O(n)$
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***)	$nR+O(1)$	$O(2^{-2R/\lambda}\log n)$	$O(n\log n)$

TABLE I: Comparison of various compression schemes

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

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

$$\frac{K_l}{\sqrt{N}}\|\mathbf{y}\|_2 \le \|\mathbf{x}_d\|_\infty \le \frac{K_u}{\sqrt{N}}\|\mathbf{y}\|_2, \tag{6}$$

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

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

The value of upper Kashin constant K_u depends on the choice of frame construction $\mathbf{S}$, as well as its aspect ratio $\lambda = N/n$. [10], [27] show that if $\mathbf{S}$ is a *random Haar orthonormal matrix*, then $K = K\left(\lambda\right)$, 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 λ is up to us, Lemma 1 implies that for random orthonormal frames, democratic embeddings satisfy $\|\mathbf{x}_d\|_\infty = \Theta(1/\sqrt{N})$ w.h.p. As we will see in §III, for large n (or equivalently large N since $N \ge n$), this remarkably improves the robustness of our proposed compression schemes. A comprehensive comparison of different choices for $\mathbf{S}$ is given in Supp. §4.

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 η, δ which is not readily available. We propose a simpler relaxation of (5):

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_2^2 \ \text{ subject to } \ \mathbf{y} = \mathbf{S}\mathbf{x}. \tag{7}$$

The solution of the ℓ_2-minimization (7) can be found in closed form (cf. Supp. §1) as:

$$\mathbf{x}_{nd} = \mathbf{S}^\dagger\mathbf{y} = \mathbf{S}^\top\left(\mathbf{S}\mathbf{S}^\top\right)^{-1}\mathbf{y} \in \mathbb{R}^n\,, \tag{8}$$

where $(\cdot)^\dagger$ (defined as above) is the pseudo-inverse. For Parseval frames $\mathbf{S}$, this boils down to $\mathbf{x}_{nd} = \mathbf{S}^\top\mathbf{y}$. We refer to $\mathbf{x}_{nd} = \mathbf{S}^\dagger\mathbf{y}$ as the **Near-Democratic** embedding of $\mathbf{y} \in \mathbb{R}^n$ with respect to $\mathbf{S} \in \mathbb{R}^{n\times N}$, and show that the solution $\mathbf{x}_{nd}$ of (7) satisfies $\|\mathbf{x}_{nd}\|_\infty = O((\sqrt{\log N}/\sqrt{N})\|\mathbf{y}\|_2)$ 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 λ approaches 1, the solution space $\mathcal{S}$ 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 $\mathbf{G} \in \mathbb{R}^{N\times N}$ with i.i.d. entries, $\mathbf{G}_{ij} \overset{iid}{\sim} \mathcal{N}(0,1)$, computing its singular-value decomposition $\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, letting $\widetilde{\mathbf{S}} = \mathbf{U}\mathbf{V}^\top$ and generating $\mathbf{S} \in \mathbb{R}^{n\times N}$ by randomly selecting n rows of $\widetilde{\mathbf{S}}$, i.e., $\mathbf{S} = \mathbf{P}\widetilde{\mathbf{S}}$ where $\mathbf{P} \in \mathbb{R}^{n\times N}$ is a sampling matrix obtained by randomly selecting n rows of $\mathbf{I}_N$.

Lemma 2. *(Near-Democratic Embeddings with Random Orthonormal Frames) For a random orthonormal frame $\mathbf{S} \in \mathbb{R}^{n\times N}$ generated as described above, with probability (w.p.) at least $1 - \frac{1}{2N}$, the solution of (7) satisfies:*

$$\|\mathbf{x}_{nd}\|_\infty \le 2\sqrt{\frac{\lambda\log(2N)}{N}}\|\mathbf{y}\|_2. \tag{9}$$

The proof of Lemma 2 is delegated to App. A. It utilizes the observation that each coordinate of $\mathbf{S}^\top\mathbf{y} \in \mathbb{R}^N$ is isotropically distributed, and subsequently exploits measure concentration. Random orthonormal matrices prove quite beneficial in this regard. Nevertheless, computing the near-democratic embeddings $\mathbf{x}_{nd} = \mathbf{S}^\top\mathbf{y}$, for random orthonormal frames still requires $O\left(n^2\right)$ time, and moreover, even storing $\mathbf{S}$, comprising of 32-bit floating-point entries can be memory intensive. To address this, we further propose a randomized

Hadamard construction for $\mathbf{S}$. 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 $\mathbf{H}$ whose entries are normalized, i.e., $\mathbf{H}_{ij} = \pm 1/\sqrt{N}$, $\mathbf{H} = \mathbf{H}^\top$, and $\mathbf{H}\mathbf{H}^\top = \mathbf{I}_N$. Let $\mathbf{D} \in \mathbb{R}^{N \times N}$ be a diagonal matrix whose entries are randomly chosen to be ± 1 with equal probability. Let $\mathbf{P} \in \mathbb{R}^{n \times N}$ be the sampling matrix as before. We define our frame to be $\mathbf{S} = \mathbf{PDH} \in \mathbb{R}^{n \times N}$. Note that $\mathbf{S}\mathbf{S}^\top = \mathbf{PDH}\mathbf{H}^\top\mathbf{D}\mathbf{P}^\top = \mathbf{P}\mathbf{D}^2\mathbf{P}^\top = \mathbf{P}\mathbf{P}^\top = \mathbf{I}_n$. i.e.,, our randomized Hadamard construction is a Parseval frame. Storing the 1-bit signs is enough to store the matrix $\mathbf{S} = \mathbf{PDH}$ in the memory. Furthermore, the near-democratic embedding $\mathbf{x}_{nd} = \mathbf{S}^\top \mathbf{y} = \mathbf{H}\mathbf{D}\mathbf{P}^\top \mathbf{y}$ can be computed with just $O(n \log n)$ additions, subtractions and sign-flips as $\mathbf{S}_{ij} = \pm 1/\sqrt{N}$. 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 $\mathbf{S} = \mathbf{PDH}$.

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

$$\|\mathbf{x}_{nd}\|_\infty \le 2\sqrt{\frac{\log(2N)}{N}}\|\mathbf{y}\|_2. \tag{10}$$

Its proof is provided in App. B, and upper bounds the tail probability of each coordinate of $\mathbf{S}^\top \mathbf{y}$ using a Chernoff-type argument, followed by a union bound. In §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 §III-A and derive upper bounds on the ℓ_2 quantization errors, which are relevant for the convergence analysis of our proposed algorithms in §IV. Furthermore, in §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 ℓ_∞-ball of radius r centered at the origin of $\mathbb{R}^N$ by $\mathcal{B}_\infty^N(r)$. Finite length source coding schemes map its inputs to a discrete set of finite cardinality. An R-**bit uniform scalar quantizer** is a mapping $\mathsf{Q}(\cdot) : \mathcal{B}_\infty^N(1) \to S$ with $S \subset \mathbb{R}^N$ and $|S| \le 2^{\lfloor nR \rfloor}$. 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 $v_i = -1 + (2i-1)\Delta/2$, for $i = 1, \ldots, M$, where $\Delta = 2/M$ is the **resolution**. $\mathsf{Q}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{B}_\infty^N(1)$ is defined as $\mathsf{Q}(\mathbf{x}) = [x_1', \ldots, x_N']^\top$; $x_j' \triangleq \arg\min_{y \in \{v_1, \ldots, v_M\}} |y - x_j|$. The maximum possible quantization error is given by,

$$d = \sup_{\mathbf{x} \in \mathcal{B}_\infty^N(1)} \|\mathbf{x} - \mathsf{Q}(\mathbf{x})\|_2 \le \frac{\Delta}{2}\sqrt{N}. \tag{11}$$

A. Proposed Quantization Strategy

Given a frame $\mathbf{S} \in \mathbb{R}^{n \times N}$, for any $\mathbf{y} \in \mathbb{R}^n$, denote its *democratic* and *near-democratic embeddings* (i.e., the solutions of (5) and (7) respectively) by $\mathbf{x}_d$ and $\mathbf{x}_{nd}$, both in $\mathbb{R}^N$. The **democratic** and **near-democratic encoders** are mappings $\mathsf{E}_d(\cdot), \mathsf{E}_{nd}(\cdot) : \mathbb{R}^n \to S \subset \mathbb{R}^N$, $|S| \le 2^{\lfloor nR \rfloor}$ defined as:

$$\mathsf{E}_d(\mathbf{y}) = \mathsf{Q}\left(\frac{\mathbf{x}_d}{\|\mathbf{x}_d\|_\infty}\right), \; \mathsf{E}_{nd}(\mathbf{y}) = \mathsf{Q}\left(\frac{\mathbf{x}_{nd}}{\|\mathbf{x}_{nd}\|_\infty}\right). \tag{12}$$

$\mathsf{E}_d(\cdot)$ and $\mathsf{E}_{nd}(\cdot)$ 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 $\mathsf{D}(\cdot) : S \to \mathbb{R}^n$, $\mathsf{D}(\mathbf{x}') = \|\mathbf{x}\|_\infty \mathbf{S}\mathbf{x}'$, where $\mathbf{x}$ is either $\mathbf{x}_d$ or $\mathbf{x}_{nd}$, and $\mathbf{x}' = \mathsf{E}_d(\mathbf{y})$ or $\mathsf{E}_{nd}(\mathbf{y})$. Normalization by $\|\mathbf{x}\|_\infty$ is needed to ensure that the input to $\mathsf{Q}(\cdot)$ lies in $\mathcal{B}_\infty^N(1)$. In the following Thm. 1 we show the independence/weak-logarithmic dependence of **DSC** and **NDSC**. For simplicity of exposition, here we have assumed that the scalar magnitude $\|\mathbf{x}\|_\infty$ is known exactly at the receiver. We can quantize $\|\mathbf{x}\|_\infty$ 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 $R + \frac{O(1)}{n}$ bits per dimension, which $\to R$ as $n \to \infty$. In App. 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 $\mathbf{S} \in \mathbb{R}^{n \times N}$ and an R-bit uniform scalar quantizer $\mathsf{Q}(\cdot)$, for any $\mathbf{y} \in \mathbb{R}^n$, let $\mathsf{Q}_d(\mathbf{y}) = \mathsf{D}(\mathsf{E}_d(\mathbf{y}))$ and $\mathsf{Q}_{nd}(\mathbf{y}) = \mathsf{D}(\mathsf{E}_{nd}(\mathbf{y}))$. Then, with probability at least $1 - e^{-\Omega(n)}$,*

$$\|\mathbf{y} - \mathsf{Q}_d(\mathbf{y})\|_2 \le 2^{\left(1 - \frac{R}{\lambda}\right)} K_u \|\mathbf{y}\|_2, \tag{13}$$

and with probability at least $1 - 1/\Omega(n)$,

$$\|\mathbf{y} - \mathsf{Q}_{nd}(\mathbf{y})\|_2 \le 2^{\left(2 - \frac{R}{\lambda}\right)} \sqrt{\log(2N)}\|\mathbf{y}\|_2. \tag{14}$$

The proof of Thm. 1 is a direct consequence of Lemmas 1, 2 and 3 and is provided in App. C. In the above Thm. 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 $\lambda = 1$, Thm. 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.

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]–[30]. We review certain definitions to precisely characterize this notion. Let $\mathcal{B}_2^n(a)$ denote the Euclidean ball of radius a centered at the origin. The **dynamic range** (r) of an R-bit quantizer $\mathsf{Q} : \mathcal{R} \to \mathcal{R}' \subset \mathbb{R}^n$, $|\mathcal{R}'| \le 2^{\lfloor nR \rfloor}$ is defined to be the radius of the largest Euclidean ball which fits inside the domain of Q, i.e., $r \triangleq \sup\{a \mid \mathcal{B}_2^n(a) \subseteq \mathcal{R}\}$. The **covering**

[1] The exact expression depends on the choice of $\mathbf{S}$ (with its UP parameters) and is given in Supp. §4.

radius of Q is defined as the maximum possible quantization error when any $\mathbf{x} \in \mathcal{B}_2^n(r)$ is quantized to its nearest neighbor, i.e., $d(\mathsf{Q}) \triangleq \inf\{d > 0 \mid \forall \mathbf{x} \in \mathcal{B}_2^n(r), \|\mathbf{x} - \mathsf{Q}(\mathbf{x})\|_2 \leq d\}$. The **covering efficiency** $\rho_n(\mathsf{Q})$ of $\mathsf{Q} : \mathcal{R} \to \mathcal{R}' \subset \mathbb{R}^n$ is defined as:

$$\rho_n(\mathsf{Q}) = \left(|\mathcal{R}'| \frac{\mathrm{vol}\left(\mathcal{B}_2^n(d(\mathsf{Q}))\right)}{\mathrm{vol}\left(\mathcal{B}_2^n(r)\right)} \right)^{\frac{1}{n}} = |\mathcal{R}'|^{\frac{1}{n}} \frac{d(\mathsf{Q})}{r}. \quad (15)$$

If we consider Euclidean balls of radius $d(\mathsf{Q})$ around each quantization point, the total volume of these balls must cover $\mathcal{B}_2^n(r)$. Covering efficiency formalizes how well this covering is and $\rho_n(\mathsf{Q}) \geq 1$ is a natural lower bound. [6] notes that for Roger's quantizer [9], $\rho_n \to 1$ as $n \to \infty$, 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 Thm. 1 and is given in Supp. §5.

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

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

where $\lambda = N/n$ is the aspect ratio of the frame $\mathbf{S} \in \mathbb{R}^{n \times N}$, and K_u 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 μ-strongly convex objective functions that satisfy $\|\mathbf{x}_f^*\| \leq D$ for some known $D \geq 0$, where $\mathbf{x}_f^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$. For a starting point $\widehat{\mathbf{x}}_0 \in \mathbb{R}^n$ and step-size $\alpha > 0$, we consider Π_R to be the class of R-bit **Quantized Gradient Descent (QGD)** algorithms that iterate the descent rule $\widehat{\mathbf{x}}_{t+1} \leftarrow \widehat{\mathbf{x}}_t - \alpha \mathbf{q}_t$, 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 $2^{\lfloor nR \rfloor}$. We consider the class of algorithms Π_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, $\mathbf{z}_t \in \widehat{\mathbf{x}}_t + \mathrm{span}\{\mathbf{e}_0, \ldots, \mathbf{e}_{t-1}\}$ and $\mathbf{u}_t \in \nabla f(\mathbf{z}_t) + \mathrm{span}\{\mathbf{e}_0, \ldots, \mathbf{e}_t - 1\}$, where $\mathbf{e}_i \triangleq \mathbf{q}_i - \mathbf{u}_i, i = 0, \ldots, t-1$ are the past quantization errors. From Thm. IV.1 of [6], for the class Π_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 $C(R) \geq \max\left\{\sigma, 2^{-R}\right\}$, where $\sigma \triangleq \frac{L-\mu}{L+\mu}$. Here, σ is the worst-case linear convergence rate of unquantized gradient descent over the same function class [31]. $C(R)$ has a sharp transition at a threshold budget $R_* = \log(1/\sigma)$. [6] shows that for their proposed algorithm, using scalar quantizers yields a convergence rate of $\leq \max\left\{\sigma, \sqrt{n} 2^{-R}\right\}$. This means that we require $R \geq \log(\sqrt{n}/\sigma)$ to achieve the convergence rate of unquantized GD, which is far from R_* for large n. We propose **DGD-DEF**: ***D**istributed **G**radient **D**escent with **D**emocratically **E**ncoded **F**eedback* (Alg. 1) which resolves this issue. Here, $\mathsf{E}(\cdot)$ 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). Thm. 2 characterizes the convergence rate of **DGD-DEF**. Its proof is similar to [6, Thm. 7] and is deferred to App. D.

Algorithm 1 DGD-DEF

```
Initialize: $\widehat{\mathbf{x}}_0 \leftarrow \mathbf{0}$ and $\mathbf{e}_{-1} \leftarrow \mathbf{0}$
for $t = 0$ to $T - 1$ do
    Worker:
    $\mathbf{z}_t \leftarrow \widehat{\mathbf{x}}_t + \alpha \mathbf{e}_{t-1}$            (gradient access point)
    $\mathbf{u}_t \leftarrow \nabla f(\mathbf{z}_t) - \mathbf{e}_{t-1}$                 (error feedback)
    $\mathbf{v}_t = \mathsf{E}(\mathbf{u}_t)$                                     (source encoding)
    $\mathbf{e}_t \leftarrow \mathsf{D}(\mathbf{v}_t) - \mathbf{u}_t$                      (error for next step)
    Server:
    $\mathbf{q}_t = \mathsf{D}(\mathbf{v}_t)$                                     (source decoding)
    $\widehat{\mathbf{x}}_{t+1} \leftarrow \widehat{\mathbf{x}}_t - \alpha \mathbf{q}_t$            (gradient descent step)
end for
Output: $\widehat{\mathbf{x}}_T$
```

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

$$\|\widehat{\mathbf{x}}_T - \mathbf{x}^*\|_2 \leq \begin{cases} \max\left\{\nu, \beta\right\}^T \left(1 + \beta \frac{\alpha L}{|\beta - \nu|}\right) D, & \text{if } \nu \neq \beta, \\ \nu^T \left(1 + \alpha L T\right) D, & \text{otherwise}, \end{cases}$$

where β is the normalized error as in Thm. 1, i.e., $\beta \triangleq 2^{(1-R/\lambda)} K_u$ if $\mathsf{E} = \mathsf{E}_d$, and $\beta \triangleq 2^{(2-R/\lambda)} \sqrt{\log(2N)}$ if $\mathsf{E} = \mathsf{E}_{nd}$, and $\nu \triangleq (1 - (\alpha^* L \mu)\alpha)^{1/2}$ is the convergence rate of unquantized gradient descent with stepsize α.

With $\alpha = \alpha^*$, $\limsup_{T \to \infty} \sup_{f \in \mathcal{F}_{\mu,L,D}} \left(\|\mathbf{x}_T - \mathbf{x}_f^*\|/D\right)^{1/T} = \max\left\{\sigma, 2^{-R}\beta\right\}$. For **DSC**, $\beta = O(1)$ w.h.p., implying **DGD-DEF** achieves the lower bound of $\max\{\sigma, 2^{-R}\}$ to within constant factors, and since $\beta = O(\sqrt{\log n})$ for **NDSC** 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 $R_{thr} = \log\left(\beta/\sigma\right)$ is much less than $\log\left(\sqrt{n}/\sigma\right)$ 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., $O(n^3)$ or $O(n^2)$.

Algorithm 2 DQ-PSGD

```
Initialize: $\widehat{\mathbf{x}}_0 \in \mathcal{X}$, $\alpha \in \mathbb{R}_+$ and $T$
for $t = 0$ to $T - 1$ do
    Worker:
    $\widehat{\mathbf{g}}_t = \widehat{\mathbf{g}}(\widehat{\mathbf{x}}_t)$                          (noisy subgradient)
    $\mathbf{v}_t = \mathsf{E}_{Dith}(\widehat{\mathbf{g}}_t)$                        (source encoding)
    Server:
    $\mathbf{q}_t = \mathsf{D}_{Dith}(\mathbf{v}_t)$                          (source decoding)
    $\underline{\widehat{\mathbf{x}}}_{t+1} \leftarrow \widehat{\mathbf{x}}_t - \alpha\mathbf{q}_t$                      (subgradient step)
    $\widehat{\mathbf{x}}_{t+1} = \Gamma_{\mathcal{X}}\left(\underline{\widehat{\mathbf{x}}}_{t+1}\right)$                       (projection step)
end for
Output: $\mathbf{x}_T = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{x}}_t$
```

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 $\widehat{\mathbf{g}}(\mathbf{x})$ for any input query $\mathbf{x} \in \mathcal{X}$ is assumed to be *unbiased*, i.e.,

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

and *uniformly bounded*, i.e.,

$$\|\widehat{\mathbf{g}}(\mathbf{x})\|_2 \leq B \quad \text{for some } B > 0$$

In this case, an **R-bit quantizer** is defined to be a set of (possibly randomized) mappings $(\mathsf{Q}^e, \mathsf{Q}^d)$ with the encoder $\mathsf{Q}^e : \mathbb{R}^n \to \{0,1\}^{nR}$ and the decoder $\mathsf{Q}^d : \{0,1\}^{nR} \to \mathbb{R}^n$. 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,

$$\mathsf{Q}(\mathbf{y}) \triangleq \mathsf{Q}_G(\|\mathbf{y}\|_2) \cdot \mathsf{Q}_S(\mathbf{y}/\|\mathbf{y}\|_2),$$

where $\mathsf{Q}_G : \mathbb{R} \to \mathbb{R}$ and $\mathsf{Q}_S : \mathbb{R}^n \to \mathbb{R}^n$ 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 $(\mathsf{E}_{Dith}, \mathsf{D}_{Dith})$ in Alg. 2 (cf. App. E for detailed description of this quantizer design) for Q_S, and propose **DQ-PSGD**: **D**emocratically **Q**uantized **P**rojected **S**tochastic sub**G**radient **D**escent (Alg. 2). We use a dithered version of **DSC** instead of the nearest neighbor scheme of §III because for stochastic oracles, it enables us to attain the optimal minimax rate even without error-feedback. Thm. 3 characterizes the expected suboptimality gap of **DQ-PSGD**. Its proof is similar to [7, Corollary 3.4] (ref. App. E).

Theorem 3. *(**DQ-PSGD** convergence guarantee) For any general objective f with access to the oracle $\mathsf{Q} \circ \mathcal{O}$ which outputs quantized noisy subgradients $\mathsf{Q}(\widehat{\mathbf{g}}(\mathbf{x}))$, where Q employs **DSC** for the shape quantizer, with a step-size choice of $\alpha = \frac{D}{BK_u}\sqrt{\frac{\min\{R,1\}}{T}}$, the worst case expected suboptimality gap of the output $\mathbf{x}_T$ of* **DQ-PSGD** *after T iterations is*

$$\sup_{(f,\mathcal{O})} \mathbb{E}f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{K_u DB}{\sqrt{T \cdot \min\{1, R\}}}. \tag{16}$$

Since $K_u = O(1)$ w.h.p., Thm. 3 shows that **DQ-PSGD** achieves the minimax lower bound [7, Thm. 2.3, 3.1], using only $R + o_n(1)$ bits per dimension. The additional $o_n(1)$ bits is for transmitting the scalar magnitude. Compared to [7], **DQ-PSGD** attains the minimax optimal $O(1/\sqrt{T})$ 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**. Supp. §2 (Thm. 1) shows that **DSC** & **NDSC** improve performance when used in conjunction with existing general 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):

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \equiv \arg\min_{\mathbf{x}\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x}). \tag{17}$$

Here, the objective $f(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ is the sum of multiple $f_i(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$, 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, Alg. 2 can be extended to multiple workers by incorporating an additional consensus step at the PS. The pseudocode is provided in Alg. 3. We analyze this setting in more detail in Supp. §3. With a budget of R-bits per dimension per worker, we show that using a naïve quantizer, the worst-case convergence rate scales as:

$$\sup_{f,\mathcal{O}} \mathbb{E}f(\mathbf{x}_T) - f(\mathbf{x}^*) \lesssim O\left(\frac{1}{\sqrt{mT}} \cdot \frac{\sqrt{n}B}{(2^R - 1)}\right).$$

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 $O\left(\frac{1}{\sqrt{mT}} \cdot \frac{K_u}{(2^R-1)}\right)$ and $O\left(\frac{1}{\sqrt{mT}} \cdot \frac{\sqrt{\log n}}{(2^R-1)}\right)$ with **DSC** and **NDSC** respectively.

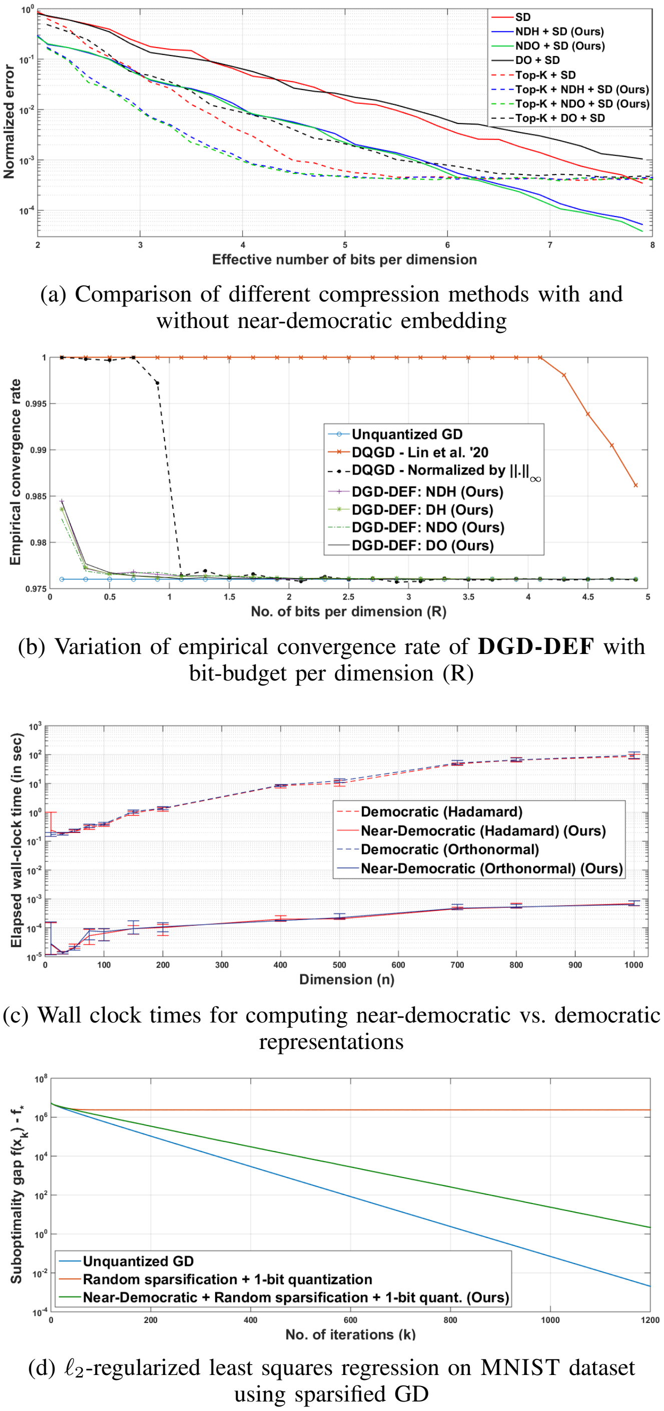
Algorithm 3 DQ-PSGD (Multiple workers)

```
Initialize: $\widehat{\mathbf{x}}_0 \in \mathcal{X}$ (at the PS), $\alpha \in \mathbb{R}_+$ and $T$.
for $t = 0$ to $T - 1$ do
    Server: Broadcasts $\widehat{\mathbf{x}}_t$ to all workers $\mathrm{Wk}_i$, $i \in [m]$.
    for $i = 1$ to $n$ at $\mathrm{Wk}_i$ do
        Compute $\widehat{\mathbf{g}}_t^i = \widehat{\mathbf{g}}^i(\widehat{\mathbf{x}}_t)$              (noisy subgradient)
        Encode $\mathbf{v}_t^i = \mathsf{E}_{Dith}(\widehat{\mathbf{g}}_t^i)$               (source encoding)
        $\mathrm{Wk}_i$ sends $\mathbf{v}_t^i$ to the PS.               (Communication)
    end for
    Server:
    $\mathbf{q}_t^i = \mathsf{D}_{Dith}(\mathbf{v}_t^i)$ for all $i \in [n]$           (source decoding)
    $\mathbf{q}_t = \frac{1}{n}\sum_{i=1}^{n}\mathbf{q}_t^i$                        (consensus step)
    $\underline{\widehat{\mathbf{x}}}_{t+1} \leftarrow \widehat{\mathbf{x}}_t - \alpha\mathbf{q}_t$                      (subgradient step)
    $\widehat{\mathbf{x}}_{t+1} = \Gamma_{\mathcal{X}}\left(\underline{\widehat{\mathbf{x}}}_{t+1}\right)$                       (projection step)
end for
Output: $\mathbf{x}_T = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{x}}_t$
```

V. Numerical Simulations

We validate our theoretical claims with numerical simulations. Fig. 1a plots the normalized compression errors i.e., $\mathbb{E}\left[\|\mathsf{Q}(\mathbf{y}) - \mathbf{y}\|\right]_2 / \|\mathbf{y}\|_2$ 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 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 $\mathbf{y}$ 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 η, δ. For the two plots labelled **Kashin** (with random orthonormal frame), we choose $\lambda = 1.5$ and 1.8, which implies availability of R/λ 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 λ 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. 1b compares the empirical convergence rate, defined as $\|\widehat{\mathbf{x}}_T - \mathbf{x}^*_f\|_2 \Big/ \|\widehat{\mathbf{x}}_0 - \mathbf{x}^*_f\|_2$ versus the bit-budget constraint, i.e., R bits per dimension, for solving the least squares problem $\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$, where $n = 116$, and the entries of $\mathbf{A}$ 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 $\frac{L-\mu}{L+\mu}$ [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 App. 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. 1c 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 $\mathbf{x} = \mathbf{S}^\top\mathbf{y}$. 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. 1d, we solve the ℓ_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 α^* given by Thm. 2.



(a) Comparison of different compression methods with and without near-democratic embedding

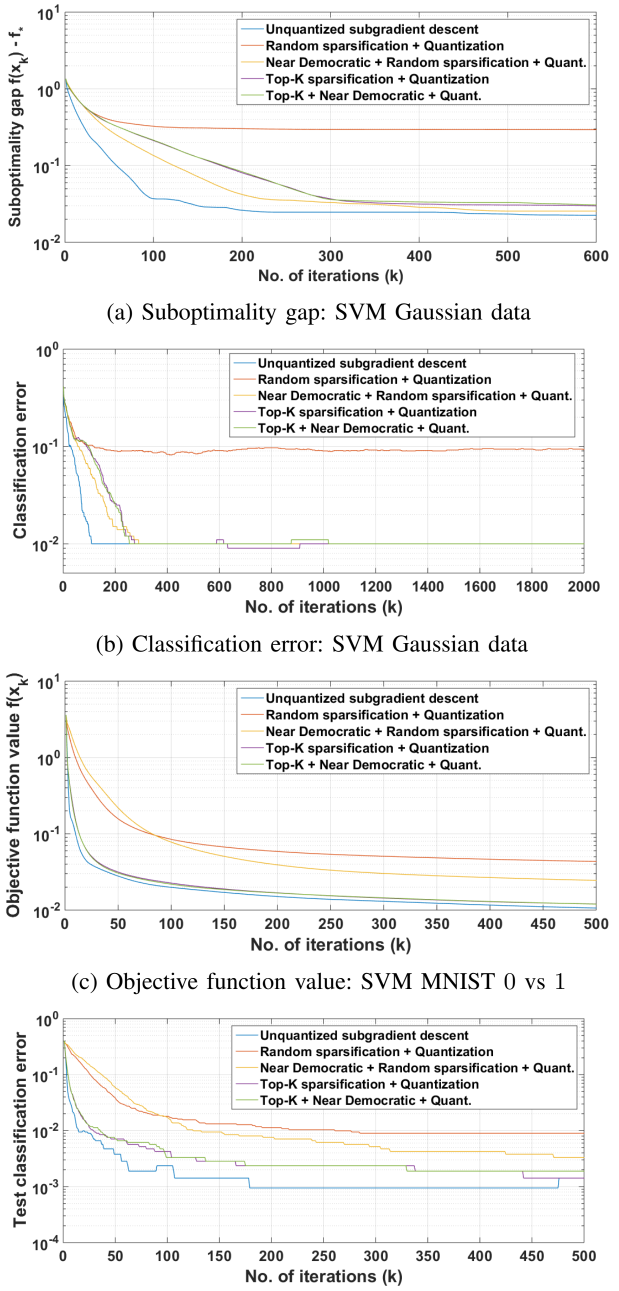
(b) Variation of empirical convergence rate of **DGD-DEF** with bit-budget per dimension (R)

(c) Wall clock times for computing near-democratic vs. democratic representations

(d) ℓ_2-regularized least squares regression on MNIST dataset using sparsified GD

Fig. 1: Simulations on smooth and strongly convex objectives

For general convex & non-smooth objectives, we train a support vector machine where the subgradients are



(a) Suboptimality gap: SVM Gaussian data

(b) Classification error: SVM Gaussian data

(c) Objective function value: SVM MNIST 0 vs 1

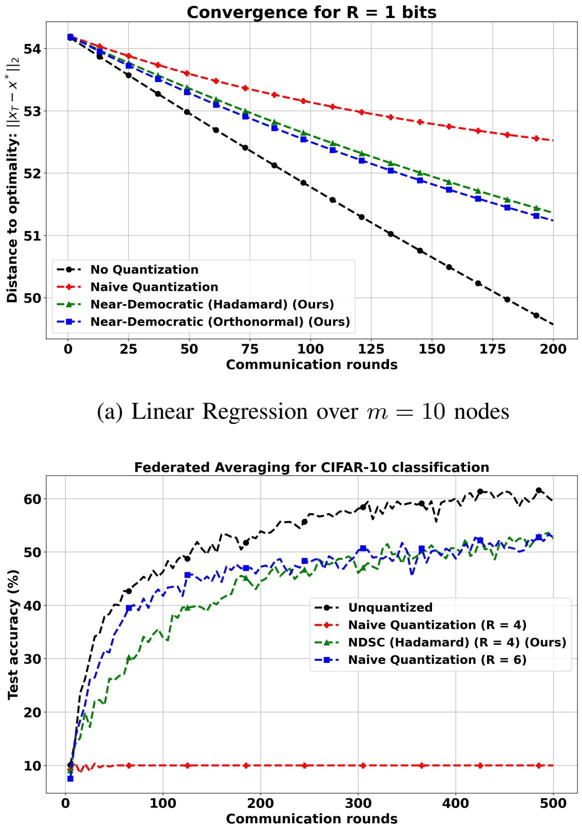
(d) Classification error: SVM MNIST 0 vs 1

Fig. 2: General convex and non-smooth: Training an SVM

quantized using R-bits per dimension. Each worker has m datapoints $\{(\mathbf{a}_i, b_i)\} \in \mathbb{R}^n \times \{-1, +1\}$ for $i = 1, \ldots, m$. We want to solve the following optimization problem in which our aim is to minimize the hinge loss: $\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m} \max\left(0, 1 - b_i \cdot \mathbf{x}^\top \mathbf{a}_i\right)$. 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. 2a and 2b, 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 2a 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. 2a 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 tradeoff 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. 2b and observe a similar trend for the different sparsification and quantization schemes.

Figs. 2c and 2d consider the MNIST dataset [34], and the problem of training an SVM to distinguish the digit 0 from digit 1. Fig. 2c shows how the objective function value decreases with the number of iterations. Fig. 2d 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 $\lfloor nR \rfloor = \lfloor 784 \times 0.1 \rfloor = 78$ 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 sparsification in this setting i.e., 78, top-K performs better as expected. For this set of simulations, we have chosen a nominal step-size $\alpha = 1$ empirically, and kept it constant for a fair comparison of different algorithms.

Multi-worker simulations. We consider two problems. Fig. 3a considers a regression model that solves: $\mathbf{x}^* \equiv \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{s}\sum_{j=1}^{s}\frac{1}{2}\left(b_{ij} - \mathbf{a}_{ij}^\top \mathbf{x}\right)^2\right)$. Here, $\{\mathbf{a}_{ij}, b_{ij}\}_{j=1}^{s}$ denotes the local dataset at node i, for $i \in [m]$. The dimension of the problem is $n = 30$, $m = 10$ workers



(a) Linear Regression over $m = 10$ nodes

(b) Training a CNN over $m = 10$ nodes

Fig. 3: Parameter-Server with multiple workers

with each worker having $s = 10$ local datapoints. The dataset is generated synthetically from a model $\mathbf{x}^*$ according to the noisy planted model $\mathbf{b} = \mathbf{A}\mathbf{x}^*$, where $\mathbf{b} \in \mathbb{R}^{ms}$ is the regression output and the rows of $\mathbf{A} \in \mathbb{R}^{ms \times n}$, i.e. $\{\mathbf{a}_1^\top, \ldots, \mathbf{a}_{ms}^\top\}$ are the data vectors. We let $\mathbf{x}^* \sim$ Student-t (df = 1) and the entries of the data matrix $\mathbf{A} \overset{iid}{\sim} \mathcal{N}(0, 1)$.

Although our theoretical analysis is for convex functions, we also provide simulations for non-convex settings. In Fig. 3b, 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 Supp. §3.

VI. Conclusions

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.

References

[1] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning: Parallel and Distributed Approaches*. USA: Cambridge University Press, 2011.

[2] P. Kairouz and et. al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation*, 2014, p. 583–598.

[4] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[5] R. Saha, S. Rini, M. Rao, and A. J. Goldsmith, "Decentralized optimization over noisy, rate-constrained networks: Achieving consensus by communicating differences," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 449–467, 2022.

[6] C.-Y. Lin, V. Kostina, and B. Hassibi, "Differentially quantized gradient descent," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1200–1205.

[7] P. Mayekar and H. Tyagi, "RATQ: A universal fixed-length quantizer for stochastic optimization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1399–1409.

[8] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 1709–1720.

[9] C. A. Rogers, "Covering a sphere with spheres," *Mathematika*, vol. 10, p. 157–164, 1963.

[10] Y. Lyubarskii and R. Vershynin, "Uncertainty principles and vector quantization," *IEEE Trans. Inf. Theor.*, vol. 56, no. 7, p. 3491–3501, 2010.

[11] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3329–3337.

[12] A. Abdi and F. Fekri, "Quantized compressive sampling of stochastic gradients for efficient communication in distributed deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34(04), 2020, pp. 3105–3112.

[13] R. Hadad and U. Erez, "Dithered quantization via orthogonal transformations," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5887–5900, 2016.

[14] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 559–568.

[15] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, “Error feedback fixes SignSGD and other gradient compression schemes,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 3252–3261.

[16] 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 *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 1509–1519.

[17] V. Gandikota, D. Kane, R. Kumar Maity, and A. Mazumdar, “vqSGD: Vector quantized stochastic gradient descent,” in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021, pp. 2197–2205.

[18] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, “Sparsified SGD with memory,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS’18. Curran Associates Inc., 2018, p. 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*. Curran Associates, Inc., 2018.

[20] P. Mayekar and H. Tyagi, “Limits on gradient compression for stochastic optimization,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2658–2663.

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

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

[23] W.-N. Chen, P. Kairouz, and A. Ozgur, “Breaking the communication-privacy-accuracy trilemma,” in *Neural Information Processing Systems (NeurIPS)*, 2020.

[24] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, “Expanding the reach of federated learning by reducing client resource requirements,” 2018. [Online]. Available: https://arxiv.org/abs/1812.07210

[25] M. Safaryan, E. Shulgin, and P. Richtá rik, “Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor,” *Information and Inference: A Journal of the IMA*, 2021.

[26] R. Saha, M. Pilanci, and A. J. Goldsmith, “Minimax optimal quantization of linear models: Information-theoretic limits and efficient algorithms,” 2022. [Online]. Available: https://arxiv.org/abs/2202.11277

[27] C. Studer, T. Goldstein, W. Yin, and R. G. Baraniuk, “Democratic representations,” 2014. [Online]. Available: https://arxiv.org/abs/1401.3420

[28] I. Dumer, “Covering spheres with spheres,” *Discrete & Computational Geometry*, vol. 38, pp. 665–679, 2016.

[29] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer International, 1992.

[30] A. D. Wyner, “Random packings and coverings of the unit n-sphere,” *The Bell System Technical Journal*, vol. 46, no. 9, pp. 2111–2118, 1967.

[31] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed. Springer Publishing Company, Incorporated, 2014.

[32] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, “The convergence of sparsified gradient methods,” in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[33] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” http://cvxr.com/cvx, Mar. 2014.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” http://yann.lecun.com/exdb/mnist/, pp. 2278–2324, 1998.

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

Appendix A
Proof of Lemma 2: Near-Democratic Embeddings with Random Orthonormal Frames

Let $\{\mathbf{s}_i\}_{i=1}^N \in \mathbb{R}^n$ and $\{\widetilde{\mathbf{s}}_i\}_{i=1}^N \in \mathbb{R}^N$ denote the columns of $\mathbf{S}$ and $\widetilde{\mathbf{S}}$ respectively, where $\mathbf{S}$, $\widetilde{\mathbf{S}}$ are defined in §II-A. For $i \in [N]$, since $\widetilde{\mathbf{S}}^\top \widetilde{\mathbf{S}} = \mathbf{I}_N$, we have,

$$\|\mathbf{s}_i\|_2 \;\leq\; \|\mathbf{P}\widetilde{\mathbf{s}}_i\|_2 \;\leq\; \|\widetilde{\mathbf{s}}_i\|_2 \;=\; 1.$$

For any $\mathbf{y} \in \mathbb{R}^n$, let $\widehat{\mathbf{y}} = \mathbf{y}/\|\mathbf{y}\|_2$. Then,

$$\begin{aligned}\|\mathbf{S}^\top\mathbf{y}\|_\infty = \max_{i\in[N]} |\mathbf{s}_i^\top \mathbf{y}| &= \|\mathbf{y}\|_2 \max_{i\in[N]} \|\mathbf{s}_i\|_2 |\widehat{\mathbf{s}}_i^\top \widehat{\mathbf{y}}| \\ &\leq \|\mathbf{y}\|_2 \max_{i\in[N]} |\widehat{\mathbf{s}}_i^\top \widehat{\mathbf{y}}|,\end{aligned}$$

where $\widehat{\mathbf{s}}_i = \mathbf{s}_i/\|\mathbf{s}_i\|_2$. Note that $\widehat{\mathbf{s}}_i \in \mathbb{R}^n$ is uniformly random on the unit sphere in $\mathbb{R}^n$, i.e., $\widehat{\mathbf{s}}_i$ has identical distribution as $\mathbf{g}/\|\mathbf{g}\|_2$ where $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. Due to rotational invariance of Gaussian distribution, for any fixed $\widehat{\mathbf{y}} \in \mathbb{R}^n$ such that $\|\widehat{\mathbf{y}}\|_2 = 1$, $\widehat{\mathbf{s}}_i^\top \widehat{\mathbf{y}}$ has identical distribution as $\widehat{\mathbf{s}}_i^\top \mathbf{e}_1$, where $\mathbf{e}_1$ is the first canonical basis vector. From concentration of measure for uniform distribution on the unit sphere,

$$\mathbb{P}\left[|\widehat{\mathbf{s}}_i^\top \widehat{\mathbf{y}}| \geq t\right] = \mathbb{P}\left[|s_1| \geq t\right] \leq 2e^{-nt^2/2}.$$

Using a union bound,

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

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

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

which completes the proof.

Appendix B
Proof of Lemma 3: Near-Democratic Embeddings with Randomized Hadamard Frames

Denote $\mathbf{z} = \mathbf{P}^\top\mathbf{y} = [z_1, \ldots, z_N]^\top \in \mathbb{R}^N$, and let $\mathbf{u} = \mathbf{HDz} = [u_1, \ldots, u_N]^\top$. 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,

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

Now,

$$\begin{aligned}\mathbb{E}\left[e^{\lambda a_i z_i}\right] &= \frac{1}{2}e^{\frac{\lambda}{\sqrt{N}}z_i} + \frac{1}{2}e^{-\frac{\lambda z_i}{\sqrt{N}}} \\ &= \cosh\left(\frac{\lambda}{\sqrt{N}z_i}\right) \leq e^{\lambda^2 z_i^2/(2N)},\end{aligned}$$

where the last inequality follows from a bound on hyperbolic cosine. This gives us $\mathbb{P}\left[u_j > t\right] \leq e^{\frac{\lambda^2}{2N}\|\mathbf{z}\|_2^2 - \lambda t}$. Setting $\lambda = tN/\|\mathbf{z}\|_2^2$ gives the tightest bound,

$$\mathbb{P}\left[u_j > t\right] \leq e^{-t^2N/(2\|\mathbf{z}\|_2^2)}.$$

Similarly, one can show that,

$$\mathbb{P}\left[u_j < -t\right] \leq e^{-t^2N/(2\|\mathbf{z}\|_2^2)}.$$

Since $\|\mathbf{u}\|_\infty = \max_{j\in[N], s\in\{\pm1\}} su_j$, union bound gives,

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

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

$$\mathbb{P}\left[\|\mathbf{HDz}\|_\infty \leq 2\|\mathbf{z}\|_2\sqrt{\frac{\log(2N)}{N}}\right] \geq 1 - \frac{1}{2N}.$$

Since $\mathbf{z} = \mathbf{P}^\top\mathbf{y} \implies \|\mathbf{z}\|_2 = \|\mathbf{y}\|_2$, this completes the proof.

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

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

$$\|\mathbf{y}' - \mathbf{y}\|_2 \le \|\mathbf{x}\|_\infty \|\mathbf{S}\,(\mathbf{x}' - \widetilde{\mathbf{x}})\|_2 \le \|\mathbf{x}\|_\infty \|\mathbf{x}' - \widetilde{\mathbf{x}}\|_2$$

The last inequality follows since,

$$\|\mathbf{S}\,(\mathbf{x}' - \widetilde{\mathbf{x}})\|_2 \le \|\mathbf{S}\|_2 \|\mathbf{x} - \widetilde{\mathbf{x}}\|_2.$$

Since $\mathbf{S}$ is a Parseval frame, and non-zero eigenvalues of $\mathbf{S}^\top\mathbf{S}$ are the same as those of $\mathbf{S}\mathbf{S}^\top = \mathbf{I}_n$, we have $\|\mathbf{S}\|_2 = 1$. To upper bound the quantization error $\|\mathbf{x}' - \widetilde{\mathbf{x}}\|_2$, note that if we originally had a total budget of nR-bits, the number of bits per dimension to uniformly quantize $\widetilde{\mathbf{x}} \in \mathbb{R}^N$ is now nR/N, i.e., $2^{nR/N}$ quantization points per dimension. From (11),

$$\|\mathbf{x}' - \widetilde{\mathbf{x}}\|_2 \le 2^{1 - nR/N}\sqrt{N}.$$

So, if we use Q_d from Lemma 1,

$$\|\mathbf{y}' - \mathbf{y}\|_2 \le \frac{K_u}{\sqrt{N}} \|\mathbf{y}\|_2 2^{1 - \frac{nR}{N}} \sqrt{N} = 2^{\left(1 - \frac{R}{\lambda}\right)} K_u \|\mathbf{y}\|_2.$$

Similarly, for Q_{nd}, using Lemma 3,

$$\|\mathbf{y}' - \mathbf{y}\|_2 \le 2\sqrt{\frac{\log(2N)}{N}} \|\mathbf{y}\|_2 2^{\left(1 - \frac{nR}{N}\right)} \sqrt{N}.$$

This completes the proof.

APPENDIX D
PROOF OF THM. 2: CONVERGENCE RATE OF **DGD-DEF**

For an R**-bit Quantized Gradient Descent (QGD)** algorithm (defined in §IV and [6, Def. IV.1]), the minimax convergence rate (2) over the function class $\mathcal{F}_{\mu,L,D}$ is lower bounded as $C(R) \ge \max\{\sigma, 2^{-R}\}$, where $\sigma \triangleq \frac{L-\mu}{L+\mu}$. 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 α, starting at the same location $\widehat{\mathbf{x}}_0 = \mathbf{x}_0$. Then using triangle inequality, at each iteration $i \in \mathbb{N}$ we have,

$$\begin{aligned} &\widehat{\mathbf{x}}_t = \mathbf{x}_t - \alpha\widehat{\mathbf{e}}_{t-1} \\ \implies &\|\widehat{\mathbf{x}}_t - \mathbf{x}^*\|_2 \le \|\mathbf{x}_t - \mathbf{x}^*\|_2 + \alpha\|\widehat{\mathbf{e}}_{t-1}\|_2, \end{aligned}$$

where $\widehat{\mathbf{e}}_{-1} = \mathbf{0}$. From algorithm pseudocode 1, note that $\mathbf{z}_t = \mathbf{x}_t$, i.e., **DGD-DEF** computes the gradient at the unquantized trajectory $\{\mathbf{x}_t\}_{t=0}^\infty$. Decay of the first term $\|\mathbf{x}_t - \mathbf{x}^*\|_2$ is given by the convergence guarantee of unquantized GD [31], which states that $\|\mathbf{x}_T - \mathbf{x}^*\|_2 \le \nu^T\|\mathbf{x}_0 - \mathbf{x}^*\|_2$, where $\nu \triangleq (1 - (\alpha^* L\mu)\alpha)^{1/2}$ is the convergence rate for unquantized GD with step size α. An upper bound to the second term $\|\widehat{\mathbf{e}}_{t-1}\|_2$ 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 $\|\mathbf{u}_t\|_2 \le r_t$, where the sequence $\{r_t\}$ is given by $r_t = LD\sum_{j=0}^t \nu^j\beta^{t-j}$. Here $\beta \triangleq 2^{(1-R/\lambda)}K_u$ if democratic embeddings are used, and $\beta \triangleq 2^{(2-R/\lambda)}\sqrt{\log(2N)}$ if near-democratic embeddings are used.*

Proof. This is proved using induction. For $t = 0$, we have $\mathbf{u}_0 = \nabla f(\mathbf{x}_0) - \mathbf{e}_{-1}$. Since $\mathbf{e}_{-1} = \mathbf{0}$, recalling that $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and f is L-smooth, we have,

$$\begin{aligned} \|\mathbf{u}_0\|_2 = \|\nabla f(\mathbf{x}_0)\|_2 &= \|\nabla f(\mathbf{x}_0) - \nabla f(\mathbf{x}^*)\|_2 \\ &\le L\|\mathbf{x}_0 - \mathbf{x}^*\|_2 \le LD, \end{aligned}$$

and so the lemma holds true for $t = 0$. From triangle inequality,

$$\mathbf{u}_t = \nabla f(\mathbf{x}_t) - \mathbf{e}_{t-1} \implies \|\mathbf{u}_t\|_2 \le \|\nabla f(\mathbf{x}_t)\|_2 + \|\mathbf{e}_{t-1}\|_2.$$

The first term can be upper bounded as,

$$\begin{aligned} \|\nabla f(\mathbf{x}_t)\|_2 &= \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}^*)\|_2 \\ &\le L\|\mathbf{x}_t - \mathbf{x}^*\|_2 \le L\nu^t\|\mathbf{x}_0 - \mathbf{x}^*\|_2 \le L\nu^t D. \end{aligned}$$

Furthermore, from Thm. 1,

$$\begin{aligned} \|\mathbf{e}_{t-1}\|_2 &= \|\mathbf{u}_{t-1} - \mathsf{D}\left(\mathsf{E}(\mathbf{u}_{t-1})\right)\|_2 \\ &\le \beta\|\mathbf{u}_{t-1}\|_2 \le \beta LD\sum_{j=0}^{t-1}\nu^j\beta^{t-1-j} \\ &= LD\sum_{j=0}^{t-1}\nu^j\beta^{t-j}, \end{aligned}$$

where β 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,

$$\|\mathbf{u}_t\|_2 \le LD\left(\nu^t + \sum_{j=0}^{t-1}\nu^j\beta^{t-j}\right) = LD\sum_{j=0}^{t}\nu^j\beta^{t-j}.$$

This completes the proof. □

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

$$\|\widehat{\mathbf{x}}_t - \mathbf{x}^*\|_2 \le \|\mathbf{x}_t - \mathbf{x}^*\|_2 + \alpha\|\widehat{\mathbf{e}}_{t-1}\|_2,$$

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

$$\|\mathbf{x}_T - \mathbf{x}^*\|_2 \le \nu^T\|\mathbf{x}_0 - \mathbf{x}^*\|_2 \le \nu^T D.$$

The second term can be upper bounded as

$$\|\mathbf{e}_{T-1}\|_2 \le \beta\|\mathbf{u}_{T-1}\|_2 \le \beta r_{T-1} = \beta LD\sum_{j=0}^{T-1}\nu^j\beta^{T-1-j},$$

where the inequalities follow from the definition of β and Lemma 5. So we have, $\|\widehat{\mathbf{x}}_T - \mathbf{x}^*\|_2 \le bD$, where $b = \nu^T + \beta\alpha L\sum_{j=0}^{T-1}\nu^j\beta^{T-1-j}$. There are now three possibilities:

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

$$b = \nu^T + \beta\alpha L\nu^{T-1}\frac{1 - (\beta/\nu)^T}{1 - \beta/\nu} \le \nu^T\left(1 + \beta\frac{\alpha L}{\nu - \beta}\right).$$

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

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

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

$$\begin{aligned} b &= \nu^T + \beta \alpha L \beta^{T-1} \sum_{j=0}^{T-1} \left(\frac{\nu}{\beta}\right)^j \\ &= \nu^T + \alpha L \beta^T \frac{1 - (\nu/\beta)^T}{1 - \nu/\beta} \leq \beta^T \left(1 + \beta \frac{\alpha L}{\beta - \nu}\right) \end{aligned}$$

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

$$\|\widehat{\mathbf{x}}_T - \mathbf{x}^*\|_2 \leq \begin{cases} \max\{\nu, \beta\}^T \left(1 + \beta \frac{\alpha L}{|\beta - \nu|}\right) D, & \text{if } \nu \neq \beta, \\ \nu^T \left(1 + \alpha L T\right) D & \text{otherwise.} \end{cases} \tag{18}$$

Appendix E
Proof of Thm. 3: Convergence Rate of DQ-PSGD

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

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

$$\mathcal{A}_{\mathsf{Q}} \triangleq \sup_{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_2 \leq B} \sqrt{\mathbb{E}[\|\mathsf{Q}(\mathbf{y})\|_2^2]},$$

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

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

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

$$\sup_{(f, \mathcal{O})} \mathbb{E} f(\mathbf{x}) - f(\mathbf{x}^*) \leq D \left(\frac{\mathcal{A}_{\mathsf{Q}}}{\sqrt{T}} + \mathcal{B}_{\mathsf{Q}}\right). \tag{19}$$

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

$$\mathsf{Q}(\mathbf{y}) \triangleq \mathsf{Q}_G(\|\mathbf{y}\|_2) \cdot \mathsf{Q}_S\left(\frac{\mathbf{y}}{\|\mathbf{y}\|_2}\right),$$

where $\mathsf{Q}_G : \mathbb{R} \rightarrow \mathbb{R}$, and $\mathsf{Q}_S : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

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

$$\mathcal{B}_{\mathsf{Q}} \leq \sup_{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_2 \leq B} |\mathbb{E}[\mathsf{Q}_G(\|\mathbf{y}\|_2) - \|\mathbf{y}\|_2]|.$$

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 $v \in [u_j, u_{j+1}) \subseteq [0, B]$, the output of the gain quantizer $\mathsf{Q}_G(v)$ is defined to be:

$$\mathsf{Q}_G(v) = \begin{cases} u_j & \text{with probability r,} \\ u_{j+1} & \text{with probability 1-r,} \end{cases} \tag{20}$$

where, $r \triangleq \frac{u_{j+1} - v}{(B/(m+1))}$. If a fixed number of $b = \log_2 m$ bits (typically 32) are used, it can be easily shown that Q_G 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 Thm. 3 is completed after combining the results of §E-A and §E-B.

A. High-budget regime

Let the input to Q_S be $\mathbf{y}$ such that $\|\mathbf{y}\|_2 \leq 1$. For $\mathbf{S} \in \mathbb{R}^{n \times N}$, if $\mathbf{x}_d$ denotes the democratic embedding of $\mathbf{y}$ with respect to $\mathbf{S} \in \mathbb{R}^{n \times N}$, then $\|\mathbf{x}_d\|_\infty \leq \frac{K_u}{\sqrt{N}} \|\mathbf{y}\|_2 = \frac{K_u}{\sqrt{N}}$. Let the **coordinate-wise uniformly dithered quantizer** (Q_{CUQ}) be as in [7], in which we do dithered quantization (20) of each coordinate of $\mathbf{x}_d$ independently, using R bits per dimension and a dynamic range of $\left[-\frac{K_u}{\sqrt{N}}, +\frac{K_u}{\sqrt{N}}\right]$. The output is $\mathsf{Q}_S(\mathbf{y}) = \mathbf{S}\mathsf{Q}_{CUQ}(\mathbf{x}_d)$. Since Q_{CUQ} is unbiased, the output of Q_S is also unbiased, i.e., $\mathbb{E}[\mathsf{Q}_S(\mathbf{y})] = \mathbb{E}[\mathbf{S}\mathsf{Q}_{CUQ}(\mathbf{x}_d)] = \mathbf{S}\mathbb{E}[\mathsf{Q}_{CUQ}(\mathbf{x}_d)] = \mathbf{S}\mathbf{x}_d = \mathbf{y}$. Since both Q_G and Q_S are conditionally independent of each other, the bias of $\mathsf{Q}(\cdot) = \mathsf{Q}_G(\cdot) \cdot \mathsf{Q}_S(\cdot) = 0$, i.e., the worst-case bias $\mathcal{B}_{\mathsf{Q}} = 0$. To evaluate the worst-case second moment of Q, from [7, Thm. 4.2], $\mathcal{A}_{\mathsf{Q}} \leq \mathcal{A}_{\mathsf{Q}_G} \mathcal{A}_{\mathsf{Q}_S}$. Since the dynamic range of Q_G is B, the worst-case bias $\mathcal{A}_{\mathsf{Q}_G} \leq B$. For Q_S and any $\mathbf{y}$ such that $\|\mathbf{y}\|_2 \leq 1$, we have $\|\mathsf{Q}_S(\mathbf{y})\|_2^2 = \|\mathbf{S}\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2 \leq \sigma_{max}^2(\mathbf{S}) \cdot \|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2 \leq \|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2$. The final inequality follows since $\mathbf{S}\mathbf{S}^\top = \mathbf{I}_n$. Moreover, since the dynamic range for Q is $\frac{K_u}{\sqrt{N}}$, we have $\|\mathbf{x}_d\|_\infty \leq \frac{K_u}{\sqrt{N}} \implies \|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2 \leq K_u^2$ for all $\mathbf{y} \implies \mathbb{E}\|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2 \leq K_u^2 \implies \mathcal{A}_{\mathsf{Q}_S} \leq K_u$. So, the worst-case second moment is $\mathcal{A}_{\mathsf{Q}} = \sqrt{\mathbb{E}\|\mathsf{Q}_S(\mathbf{y})\|_2^2} \leq BK_u$. Substituting these values for $\mathcal{A}_{\mathsf{Q}}$ and $\mathcal{B}_{\mathsf{Q}}$ in (19), we get the result. A similar result with a $O(\sqrt{\log n})$ 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 $\mathbb{R}^{nR}$, and subsequently doing a 1-bit quantization of the vector in $\mathbb{R}^{nR}$, 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 democratic representation + subsampling + 1-bit quantization scheme as is $\mathsf{Q}_S(\mathbf{y}) = \frac{1}{R}\mathbf{S}\sum_{i\in\mathcal{S}}\mathsf{Q}_{CUQ}(\mathbf{x}_d)\mathbf{e}_i = \frac{1}{R}\mathbf{S}\sum_{i=1}^{n}\mathsf{Q}_{CUQ}(\mathbf{x}_d)\mathbf{e}_i\mathbf{1}_{i\in\mathcal{S}}$. Here, $\mathbf{e}_i \in \mathbb{R}^n$ is the i^{th} canonical basis vector, $\mathcal{S}$ with $|\mathcal{S}| = nR$ denotes the random nR indices chosen in the subsampling step, and $\mathbf{1}_{(\cdot)}$ denotes the indicator function, since the coordinates not selected in the subsampling step are decoded as 0. Unbiasedness ensures that $\mathcal{B}_{\mathsf{Q}} = 0$. Moreover,

$$\begin{aligned}\mathbb{E}\|\mathsf{Q}_S(\mathbf{y})\|_2^2 &\leq \frac{1}{R^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{S}}\mathsf{Q}_{CUQ}(\mathbf{x}_d)\mathbf{e}_i\right\|_2^2\right]\\ &= \frac{1}{R^2}\sum_{i=1}^{n}\mathbb{E}[\mathsf{Q}_{CUQ}(\mathbf{x}_d)_i^2]\cdot\mathbb{E}[\mathbf{1}_{i\in\mathcal{S}}]\\ &= \frac{1}{R}\mathbb{E}\|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2\end{aligned}$$

The last equality follows as sampling nR coordinates uniformly at random from n coordinates implies $\mathbb{E}[\mathbf{1}_{i\in\mathcal{S}}] = R$. Since $\mathbb{E}\|\mathsf{Q}_{CUQ}(\mathbf{x}_d)\|_2^2 \leq K_u^2$, the worst-case second moment is $\mathcal{A}_{\mathsf{Q}} \leq \sqrt{\mathbb{E}\|\mathsf{Q}(\mathbf{y})\|_2^2} \leq \frac{BK_u}{\sqrt{R}}$. Substituting these values of $\mathcal{A}_{\mathsf{Q}}$ and $\mathcal{B}_{\mathsf{Q}}$ in (19), the convergence rate when $R < 1$ is $\frac{K_uDB}{\sqrt{RT}}$. This completes the proof.

Appendix F
Quantizing the ℓ_∞ norm

For simplicity, the statement of Thm. 2 assumes that the ℓ_∞ 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 $\|\mathbf{x}\|_\infty$, the total number of bits required to quantize the vector is $nR + O(1) \implies R + \frac{O(1)}{n} \to R$ bits per dimension as $n \to \infty$. 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 $\|\mathbf{x}\|_\infty$, note that if $\mathbf{y}_S = \frac{\mathbf{y}}{\|\mathbf{y}\|_\infty}$,

$$\begin{aligned}\|\mathsf{Q}(\mathbf{y}) - \mathbf{y}\|_2 &= \|\mathsf{Q}_G(\|\mathbf{y}\|_\infty)\mathsf{Q}_S(\mathbf{y}_S) - \|\mathbf{y}\|_\infty\mathbf{y}_S\|_2\\ &= \|(\|\mathbf{y}\|_\infty + \epsilon)\,\mathsf{Q}_S(\mathbf{y}_S) - \|\mathbf{y}\|_\infty\mathbf{y}_S\|_2\\ &\leq \|\mathbf{y}\|_\infty\|\mathsf{Q}_S(\mathbf{y}_S) - \mathbf{y}_S\|_2 + \epsilon\|\mathsf{Q}_S(\mathbf{y}_S)\|_2\end{aligned}$$

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

$$\epsilon\|\mathsf{Q}_S(\mathbf{y}_S)\|_2 \leq \epsilon\sqrt{N}\|\mathsf{Q}(\mathbf{y}_S)\|_\infty \leq \epsilon\sqrt{N}\frac{K_u}{\sqrt{N}} \leq \epsilon K_u,$$

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