Wyner-Ziv Estimators for Distributed Mean Estimation With Side Information and Optimization

Prathamesh Mayekar*, Shubham Jha*, Member, IEEE, Ananda Theertha Suresh, and Himanshu Tyagi, Senior Member, IEEE

Abstract—Communication efficient distributed mean estimation is an important primitive that arises in many distributed learning and optimization scenarios such as federated learning. Without any probabilistic assumptions on the underlying data, we study the problem of distributed mean estimation where the server has access to side information. We propose Wyner-Ziv estimators, which are communication and computationally efficient and near-optimal when an upper bound for the distance between the side information and the data is known. As a corollary, we also show that our algorithms provide efficient schemes for the classic Wyner-Ziv problem in information theory. In a different direction, when there is no knowledge assumed about the distance between side information and the data, we present an alternative Wyner-Ziv estimator that uses correlated sampling. This latter setting offers universal recovery guarantees, and perhaps will be of interest in practice when the number of users is large and keeping track of the distances between the data and the side information may not be possible. With this mean estimator at our disposal, we revisit basic problems in decentralized optimization and compression where our Wyner-Ziv estimator yields algorithms with almost optimal performance. First, we consider the problem of communication constrained distributed optimization and compression where our Wyner-Ziv estimator yields algorithms with almost optimal performance.

Index Terms—Federated learning, Wyner-Ziv compression, distributed mean estimation.

I. INTRODUCTION

A. Background

Consider the problem of distributed mean estimation for \( n \) vectors \( \{x_i\}_{i=1}^n \) in \( \mathbb{R}^d \), where \( x_i \) is available to client \( i \). Each client communicates to a server using a few bits to enable the server to compute the empirical mean

\[
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.
\]

This estimation problem has become a crucial primitive for distributed optimization scenarios such as federated learning, where the data is distributed across multiple clients (see [4], [8], [10], [11], [18], [20], [21], [26], [28], [31], [35], [39], [49], [50], [52], [53], [54], [58], [59], [60], [61]). One of the main bottlenecks in such distributed scenarios is the significant communication cost incurred due to client communication at each iteration of the distributed algorithm. This has spurred a recent line of work which seeks to design quantizers to express \( x_i \)'s using a low precision and, yet, enable the server to compute a high accuracy estimate of \( \bar{x} \) (see [7], [14], [24], [25], [30], [44], [51], [56], and the references therein).

Most of the recent works on distributed mean estimation focus on the setting where the server must estimate the sample mean based on the client vectors, and nothing else. However, in practice, the server may also have access to some side information. For example, consider the task of training a machine learning model based on remote client data as well as some publicly accessible data [9]. At each iteration, the server communicates its global model to the client, based on which the clients compute their updates (the gradient estimates based on their local data), compress them, and then send them to the server. The server may choose to compute its own update using the publicly available dataset to complement the updates from the client. In a related setting, the server can use the previously received gradients as side information for the next gradients expected from the clients. Alternatively, the server may ‘simulate’ side information from some client updates. It can then use this side information to form much more accurate estimates of other clients’ updates, leading to a faster distributed training algorithm. We discuss this application in detail in Section A. Similarly, distributed mean estimation with side information can be used for variance reduction in other problems such as power iteration or parallel SGD (cf. [15]).

Motivated by these observations, for the distributed mean estimation problem described at the start of the section,
we study the setting in which the server has access to the side information \( \{y_i\}_{i=1}^n \) in \( \mathbb{R}^d \), in addition to the communication from clients. Here, \( y_i \) can be viewed as server’s initial estimate (guess) of \( x_i \). We emphasize that the side information \( y_i \) is available only to the server and can, therefore, be used for estimating the mean at the server, but is not available to the clients while quantizing the updates \( \{x_i\}_{i=1}^n \).

### B. The Model

Consider the input \( x := (x_1, \ldots, x_n) \) and the side information \( y := (y_1, \ldots, y_n) \). The clients use a communication protocol to send \( r \) bits each about their observed vector to the server. For the ease of implementation, we restrict to non-interactive protocols. Specifically, we allow simultaneous message passing (SMP) protocols \( \pi = (\pi_1, \ldots, \pi_n) \) where the communication \( C_i = \pi_i(x_i, U) \in \{0, 1\}^r \) of client \( i \), \( i \in [n] \), can only depend on its local observation \( x_i \) and public randomness \( U \). Note that the clients are not aware of side information \( y \), which is available only to the server. In effect, the message \( C_i \) is obtained by quantizing \( x_i \) using an appropriately chosen randomized quantizer. Denoting the overall communication by \( C^n := (C_1, C_2, \ldots, C_n) \), the server uses the transcript \( (C^n, U) \) of the protocol and the side information \( y \) to form the estimate of the sample mean \( \hat{x} = \hat{x}(C^n, U, y) \); see Figure 1 for a depiction of our setting. We call such a \( \pi \) an \( r \)-bit SMP protocol with input \( (x, y) \) and output \( \hat{x} \).

We measure the performance of protocol \( \pi \) for inputs \( x \) and \( y \) and output \( \hat{x} \) using mean squared error (MSE) given by

\[
\text{MSE}(\pi, x, y) := \mathbb{E} \left[ ||\hat{x} - \bar{x}||_2^2 \right],
\]

where the expectation is over the public randomness \( U \) and \( \bar{x} \) is given in (1). We study the MSE of protocols for \( x \) and \( y \) such that the Euclidean distance between \( x_i \) and \( y_i \) is at most \( \Delta_i \), i.e.,

\[
||x_i - y_i||_2 \leq \Delta_i, \quad \forall \ i \in [n].
\]

Denoting \( \Delta := (\Delta_1, \ldots, \Delta_n) \), we are interested in the performance of our protocols for the following two settings:

1. **The known \( \Delta \) setting**, where \( \Delta_i \) is known to client \( i \) and the server;

2. **The unknown \( \Delta \) setting**, where \( \Delta_i \)'s are unknown to everyone.

In both these settings, we seek to find efficient \( r \)-bit quantizers for \( x_i \) that will allow accurate sample mean estimation. In the known \( \Delta \) setting, the quantizers of different clients can be chosen using the knowledge of \( \Delta \); in the unknown \( \Delta \) setting, they must be fixed irrespective of \( \Delta \).

In another direction, we distinguish the low-precision setting of \( r \leq d \) from the high-precision setting of \( r > d \). The former is perhaps of more relevance for federated learning and high-dimensional distributed optimization, while the latter has received a lot of attention in the information theory literature on rate-distortion theory. Moreover, the distributed estimation problem is a lot more interesting in the low-precision setting. We, therefore, focus more on this regime while also providing extensions of our protocols to the high-precision regime.

As a benchmark, we recall the result for distributed mean estimation with no side-information from [56]. When all \( x_i \)'s lie in the Euclidean ball of radius 1, [56] showed that the minmax MSE in the no-side-information case is

\[
\Theta\left(\frac{d}{nr}\right).
\]

### C. Our Contributions

Drawing on ideas from distributed quantization problem in information theory (cf. [62]), specifically the Wyner-Ziv problem, we present Wyner-Ziv estimators for distributed mean estimation. In the known \( \Delta \) setting, for a fixed \( \Delta \), and the low-precision setting of \( r \leq d \), we propose an \( r \)-bit SMP protocol \( \pi^*_k \) which satisfies

\[
\text{MSE}(\pi^*_k, x, y) = O\left(\sum_{i=1}^n \frac{\Delta_i^2}{n} \cdot \frac{d \log \log n}{nr}\right),
\]

for all \( x \) and \( y \) satisfying (2). Thus, in the case where all \( x_i \)'s lie in the Euclidean ball of radius 1, we improve upon the optimal estimator for distributed mean estimation (3) in the regime \( \sum_{i=1}^n \frac{\Delta_i^2 \log \log n}{n} \leq 1 \). Our estimator is motivated by the classic Wyner-Ziv problem, and hence, we refer to it as the Wyner-Ziv estimator. The details of the algorithm are given in Section III-C.

Our protocol uses the same (randomized) \( r \)-bit quantizer for each client’s data and simply uses the sample mean of the quantized vectors as the estimate for \( \bar{x} \). Furthermore, the common quantizer used by the clients is efficient and has nearly linear time-complexity of \( O(d \log d) \). Our proposed quantizer first applies a random rotation (proposed in [6]) to the input vectors \( x_i \) at client \( i \) and the side information vector \( y_i \) at the server. This ensures that the \( \Delta_i \) upper bound on the \( \ell_2 \) distance of \( x_i \) and \( y_i \) is converted to roughly a \( \Delta_i/\sqrt{d} \) upper bound on the \( \ell_\infty \) distance between \( x_i \) and \( y_i \). This then enables us to use efficient one-dimensional quantizers for each coordinate of the \( x_i \)'s, which can now operate with the knowledge that the server knows a \( y_i \) with each coordinate within roughly \( \Delta_i/\sqrt{d} \) of \( x_i \)'s coordinates.

---

1. \([n] := \{1, \ldots, n\}\).
2. While side information \( y_i \) is associated with client \( i \), we do not enforce this association in our general formulation at this point.
3. We denote by \( \log(\cdot) \) logarithm to the base 2 and by \( \ln(\cdot) \) logarithm to the base \( e \).
Moreover, we show that this protocol $\pi^*_r$ has optimal (worst-case) MSE up to an $O(\log \log n)$ factor. That is, we show that for any other $r$-bit SMP protocol $\pi$ for $r \leq d$, we can find $x$ and $y$ satisfying (2) such that

$$\text{MSE}(\pi, x, y) = \Omega\left(\min_{i \in \{1, \ldots, n\}} \Delta^2 \cdot \frac{d}{nr}\right).$$

In the unknown $\Delta$ setting, we propose a protocol $\pi^*_r$ which adapts to the unknown distance $\Delta_i$ between $x_i$ and $y_i$ and, remarkably, provides MSE guarantees dependent on $\Delta$. Specifically, for the low-precision setting of $r \leq d$, the protocol satisfies

$$\text{MSE}(\pi^*_r, x, y) = O\left(\sum_{i=1}^{n} \Delta_i \cdot \frac{d \log d}{nr}\right),$$

for all $x$ and $y$ in the unit Euclidean ball $B := \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ and satisfying (2). Thus, we improve upon the optimal estimator for the no side information counterpart (3) in the regime $\sum_{i=1}^{n} \Delta_i \cdot \frac{\ln d}{n} \leq 1$. Once again, the quantizer employed by the protocol is efficient and has nearly linear time-complexity of $O(d \log d)$. At the heart of our proposed quantizer is the technique of correlated sampling from [22] which enables to derive a $\Delta$ dependent MSE bound.

Furthermore, both our quantizers can be extended to the high-precision regime of $r > d$. The quantizer for the known $\Delta$ setting directly extends by using $r/d$ bits per dimension. The MSE of the SMP protocol using this quantizer for all the clients is only a factor of $\log n + r/d$ from the lower bound derived in [15] for the high-precision regime. The quantizer for the unknown $\Delta$ setting can be extended by sending the “type” of the communication vector, following an idea proposed in [43]. The MSE of the SMP protocol using this quantizer for all the clients falls as $2^{-r/d} \cdot \ln d$ as opposed to $d/r$ that can be obtained using naive extensions of our quantizer.

As remarked at the outset, mean estimator is a basic primitive that can be used in problems related to decentralized optimization. Indeed, we apply our Wyner-Ziv estimator to a basic communication-constrained optimization problem and show that it leads to much faster algorithms for a large number of remote clients. We also propose a universal distributed optimization algorithm UWZ-SGD, where the remote clients can operate without the knowledge of the stochastic gradient’s variance. UWZ-SGD, too, improves the performance of the baseline Parallel SGD algorithm for large enough remote clients.

Finally, in a different direction, we revisit the classic Gaussian rate-distortion problem (cf. [47]) in information theory. In this problem, the encoder observing an Gaussian vector $X$ wants to send it to a decoder observing a correlated Gaussian vector $Y$ using $r$ bits. Using the quantizer developed in the known $\Delta$ setting, we obtain an efficient scheme for this classic problem which requires a minuscule excess rate over the optimal asymptotic rate. Our scheme for this classic problem is interesting for two reasons: The first that it gives almost optimal result while using “covering” for each coordinate separately and hence is computationally efficient. All the existing schemes rely on high-dimensional covering constructed using structured codes and most of them are computationally inefficient. The second reason is that we do not require the distribution to be exactly Gaussian and subgaussianity suffices.

### D. Prior Work

The known $\Delta$ setting described above was first considered in [15]. The scheme of [15] relies on lattice quantizers with information theoretically optimal covering radius. Explicit lattices to be used and computationally efficient decoding is not provided.

In contrast, we provide explicit computationally efficient protocols for both low- and high-precision settings. Also, we establish lower bounds showing the optimality of our quantizer up to a multiplicative factor of $\log \log n$ in the low-precision regime of $r \leq d$. In comparison, the scheme of [15] is off by a factor of $\frac{d}{r}$ from this lower bound. Thus, when $r \ll d$, our scheme performs significantly better than that in [15]. We remark that the unknown $\Delta$ setting, which is perhaps more important in certain applications where estimating the distance of side information of each client is infeasible, has not been considered before.

In the classic information theoretic setting, related problems of quantization with side information at the decoder have been considered in rate-distortion theory starting with the seminal work of Wyner and Ziv [62]. Practical codes for settings where the observations are generated from known distributions have been constructed using channel codes; see, for instance, [32], [36], [38], [48], [63]. However, these codes are computationally too expensive for our setting, cannot be directly used for our distribution-free setup, and are designed for the high-precision setting of $r > d$. We remark that the scheme proposed in [15] is similar to lattice schemes in [36], [38], and [63].

The version of the distributed mean estimation problem with no side information at the server has been extensively studied. For any protocol in this setting operating with a precision constraint of $r \leq d$ bits per client, using a strong data processing inequality from [16] and [56] shows a lower bound on MSE of $\Omega\left(\frac{d}{nr}\right)$, when all $x_i$s lie in the Euclidean ball of radius one. Reference [56] propose a rotation based uniform quantization scheme which matches this lower bound up to a factor of $\log \log d$ for any precision constraint $r$. This upper bound is further improved by a random rotation based adaptive quantizer in [44] to a much tighter $\log^* d$ factor. For a precision constraint of $r = \Theta(d)$, the variable-length quantizers proposed in [8], [49], and [56] as well as the fixed-length quantizers in [20] and [43] are order-wise optimal.

A recent work on distributed mean estimation [27], which came after the conference version of our paper [42], proposed two different schemes for distributed mean estimation. The first scheme improves the performance of the standard Rand-k (cf. [54]) estimator when data across the clients are correlated.
The second scheme uses previous gradient updates to improve the performance of the standard scheme. Using previous gradient updates can be seen as a special case of our setup when we use a historical gradient as side information. Interestingly, the second scheme in [27] uses the idea of centering the gradient estimate around the side information [27, Equation 12], which is similar to the decoding rule used in our second Wyner-Ziv estimate (14). A follow-up work of [33] and [42] also proposed using correlation amongst clients to improve over standard sample mean estimators. Another recent work [55], which also came up after our conference version [42], proposed using correlated randomness for stochastic quantization across clients to improve the performance of the standard scheme.

Reference [34] and an application considered in [55] are closest to the application of communication distributed optimization considered in Section A. Reference [34] builds on the distributed mean estimation schemes in [42] and proposes an algorithm for non-convex distributed optimization. However, unlike our proposed schemes, [34] suggests using historical gradients as side information, and its optimality is unclear.

Reference [55] considers the same setting for communication-constrained distributed optimization as considered in this paper. The proposed scheme, too, is similar to UWZ – SGD, one of the schemes proposed in this paper. In more detail, both schemes leverage the fact that the stochastic estimates of the gradients across clients are close to each other to reduce the compression error. Moreover, they do this by using correlated randomness, and the compression can operate without knowing how close the stochastic gradients are across clients. However, there are crucial differences between the two schemes. At a high level, our scheme is designed for the low precision setting (where per client precision is less than the dimension) and only uses a fixed length code, the scheme in [55] is designed for the high precision setting and uses a variable length code in this setting. Our results for the low-precision regime in known $\Delta$ setting are provided in Section III and in the unknown $\Delta$ setting are provided in Section IV. In Section V, we extend our results to the high-precision regime. In Appendix A, we derive new algorithms for communication-constrained distributed optimization using our distributed mean estimation protocols. In Appendix B, we provide an application of the quantizer developed for the known-setting to the Gaussian Wyner-Ziv problem. Finally, we close with all the proofs in Appendix C. Before presenting these results, we review some preliminaries in the next section.

II. PRELIMINARIES AND THE STRUCTURE OF OUR PROTOCOLS

While our lower bound for the known $\Delta$ setting holds for an arbitrary SMP protocol, both the protocols we propose in this paper, for the known $\Delta$ and the unknown $\Delta$ settings, have a common structure. We use $r$-bit quantizers to form estimates of $x_i$ at the server and then compute the sample mean of the estimates of $x_i$. To describe our protocols and facilitate our analysis, we begin by concretely defining the distributed quantizers needed for this problem. Further, we present a simple result relating the performance of the resulting protocol to the parameters of the quantizer.

An $r$-bit quantizer $Q$ for input vectors in $\mathcal{X} \subset \mathbb{R}^d$ and side information $\mathcal{Y} \subset \mathbb{R}^e$ consists of randomized mappings $^5$ ($Q^a, Q^b$) with the encoder mapping $Q^a : \mathcal{X} \rightarrow \{0, 1\}^r$ used by the client to quantize and the decoder mapping $Q^b : \{0, 1\}^r \times \mathcal{Y} \rightarrow \mathcal{X}$ used by the server to aggregate quantized vectors. The overall quantizer $Q$ is given by the composition mapping $Q(x, y) = Q^b((Q^a(x, y))$.

In our protocols, for input $x$ and side information $y$, client $i$ uses the encoder $Q_i^a$ for the $r$-bit quantizer $Q_i$ to send $Q_i(x_i)$. The server uses $Q_i(x_i)$ and $y_i$ to form the estimate $\hat{x}_i = Q_i(x_i, y_i)$ of $x_i$. We assume that the randomness used in quantizers $Q_i$ for different $i$ is independent, whereby $\hat{x}_i$ are independent of each other for different $i$. Then server finally forms the estimate of the sample mean as

$$\hat{x} := \frac{1}{n} \sum_{i=1}^n \hat{x}_i,$$  \hspace{1cm} (4)

For any quantizer $Q$, the following two quantities will determine its performance when used in our distributed mean estimation protocol:

$$\alpha(Q; \Delta) := \sup_{x \in \mathcal{X}, y : \|x - y\|_2 \leq \Delta} \mathbb{E} \left[ \|Q(x) - x\|_2^2 \right],$$

$$\beta(Q; \Delta) := \sup_{x \in \mathcal{X}, y : \|x - y\|_2 \leq \Delta} \mathbb{E} \left[ \|Q(x) - x\|_2^2 \right],$$

where the expectation is over the randomization of the quantizer. Note that $\alpha(Q; \Delta)$ can be interpreted as the worst-case MSE and $\beta(Q; \Delta)$ the worst-case bias over $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\|x - y\|_2 \leq \Delta$.

The result below will be very handy for our analysis.

**Lemma 1:** For $x \in \mathcal{X}^n$ and $y \in \mathcal{Y}^n$ satisfying (2) and $r$-bit quantizers $Q_i, i \in [n]$, using independent randomness for different $i \in [n]$, the estimate $\hat{x}$ in (4) and the sample mean $\bar{x}$ in (1) satisfy

$$\mathbb{E} \left[ \|\hat{x} - \bar{x}\|_2^2 \right] \leq \sum_{i=1}^n \frac{\alpha(Q_i; \Delta_i)}{n^2} + \sum_{i=1}^n \frac{\beta(Q_i; \Delta_i)}{n}.$$

III. DISTRIBUTED MEAN ESTIMATION WITH KNOWN $\Delta$

In this section, we present our Wyner-Ziv estimator for the known $\Delta$ setting. As described in Section II, we use the same (randomized) quantizer across all the clients and form the estimate of sample mean as in (4). We only need to define the common quantizer used by all the clients, which we do in Section III-C. In Sections III-A and III-B, we provide the basic building blocks of our final quantizer. Further, in Section III-D, we derive a lower bound for the worst-case MSE that establishes the near-optimality of our protocol. Throughout we restrict to the low-precision setting of $r \leq d$.

A. Modulo Quantizer (MQ)

The first subroutine used by our larger quantizer is the Modulo Quantizer (MQ). MQ is a one dimensional distributed

---

5We can use public randomness $U$ for randomizing.
quantizer that can be applied to the input $x \in \mathbb{R}$ with side information $y \in \mathbb{R}$. We give an input parameter $\Delta'$ to MQ where $|x - y| \leq \Delta'$. In addition to $\Delta'$, MQ also has the resolution parameter $k$ and the lattice parameter $\varepsilon$ as inputs.

For an appropriate $\varepsilon$ to be specified later, we consider the lattice $\mathbb{Z}_\varepsilon = \{z : z \in \mathbb{Z}\}$. For a given input $x$, the encoder $Q^a_k$ finds the closest points in $\mathbb{Z}_\varepsilon$ larger and smaller than $x$. Then, one of these points is sampled randomly to get an unbiased estimate of $x$. The sampled point will be of the form $\tilde{z} \varepsilon$, where $\tilde{z}$ is in $\mathbb{Z}$. We note that the chosen point $\tilde{z} \varepsilon$ satisfies

$$\varepsilon \mathbb{E} [\tilde{z}] = x \quad \text{and} \quad |x - \tilde{z} \varepsilon| < \varepsilon, \quad \text{almost surely.} \tag{5}$$

The encoder sends $w = \tilde{z} \mod k$ to the decoder, which requires $\log k$ bits.

Upon receiving this $w$, the decoder $Q^d_k$ looks at the set $Z_{w,\varepsilon} = \{(z k + w) : \varepsilon : z \in \mathbb{Z}\}$ and decodes the point closest to $y$, which we denote by $Q_b(x, y)$. Note that declaring $y$ will already give a MSE of less than $\Delta$. A useful property of this decoder is that its output is always within a bounded distance from $y$; namely, since in Step 1 of Alg. 2 we look for the closest point to $y$ in the lattice $Z_{w,\varepsilon} := \{(z k + w) : \varepsilon : z \in \mathbb{Z}\}$, the output $Q_b(x, y)$ satisfies

$$|Q_b(x, y) - y| \leq k \varepsilon, \quad \text{almost surely.} \tag{6}$$

We summarize MQ in Alg. 1 and 2.

### Algorithm 1 Encoder $Q^a_k(x)$ of MQ

**Require:** Input $x \in \mathbb{R}$, Parameters $k$, $\Delta'$, and $\varepsilon$

1. Compute $z_u = \lfloor x/\varepsilon \rfloor$, $z_i = \lfloor z/\varepsilon \rfloor$
2. Generate $\tilde{z} = \begin{cases} z_u, & \text{w.p. } x/\varepsilon - z_i \\ z_i, & \text{w.p. } z_u - x/\varepsilon \end{cases}$
3. **Output:** $Q^a_k(x) = \tilde{z} \mod k$

### Algorithm 2 Decoder $Q^d_k(w, y)$ of MQ

**Require:** Input $w \in \{0, \ldots, k-1\}$, $y \in \mathbb{R}$

1. Compute $\tilde{z} = \arg \min \{(z k + w) : \varepsilon : z \in \mathbb{Z}\}$
2. **Output:** $Q^d_k(w, y) = (\tilde{z} k + w) \varepsilon$

### Algorithm 3 Encoder $Q^a_{k,R}(x)$ of RMQ

**Require:** Input $x \in \mathbb{R}^d$, Parameters $k$ and $\Delta'$

1. Sample $R$ as in (8) using public randomness
2. $x' = Rx$
3. **Output:** $Q^a_{k,R}(x) = (Q^a_k(x'(1)), \ldots, Q^a_k(x'(d)))^T$

The result below provides performance guarantees for $Q_b$. The key observation is that the output $Q_b(x, y)$ of the quantizer equals $\tilde{z} \varepsilon$ with $\tilde{z}$ found at the encoder, if $\varepsilon$ is set appropriately.

**Lemma 2:** Consider the Modulo Quantizer $Q_b$ described in Alg. 1 and 2 with parameter $\varepsilon$ set to satisfy

$$k \varepsilon \geq 2(\varepsilon + \Delta'). \tag{7}$$

Then, for every $x, y$ in $\mathbb{R}$ such that $|x - y| \leq \Delta'$, the output $Q_b(x, y)$ of MQ satisfies

$$\mathbb{E} [Q_b(x, y)] = x \quad \text{and} \quad |Q_b(x, y) - y| \leq \varepsilon, \quad \text{almost surely.}$$

In particular, we can set $\varepsilon = 2\Delta'/(k-2)$, to get $|Q_b(x, y) - x| \leq 2\Delta'/(k-2)$. Furthermore, the output of $Q_b$ can be described in log $k$ bits.

We close with a remark that the modulo operation used in our scheme is the simplest and easily implementable version of classic coset codes obtained using nested lattices used in distributed quantization (cf. [19], [37], [63]) and was used in [15] as well.

### B. Rotated Modulo Quantizer (RMQ)

We now describe Rotated Modulo Quantizer (RMQ). RMQ and the subsequent quantizers in this section will be used to quantize input vector $x \in \mathbb{R}^d$ with side information $y \in \mathbb{R}^d$, where $|x - y|_2 \leq \Delta$. RMQ first preprocesses the input $x$ and side information $y$ by randomly rotating them and then simply applies MQ for each coordinate. For rotation, we multiply both $x$ and $y$ with a matrix $R$ given by

$$R = \frac{1}{\sqrt{d}} \cdot HD, \tag{8}$$

where $H$ is the $d \times d$ Walsh-Hadamard Matrix (see [23]) and $D$ is a diagonal matrix with each diagonal entry generated uniformly from $\{-1, +1\}$. Note that we use public randomness to generate the same $D$ at both the encoder and the decoder. We formally describe the quantizer in Alg. 3 and 4.

**Remark 1:** We remark that the vector $R(x - y)$ has zero mean subgaussian coordinates with a variance factor of $\Delta^2/d$. This implies that for all coordinates $i \in [d]$, we have

$$P \left( |R(x - y)(i)| \geq \Delta' \right) \leq 2 e^{-\frac{\Delta'^2 d}{2 \Delta^2}}$$

(see, for instance, [12, Theorem 2.8]). This observation allows us to use $\Delta' \approx \Delta / \sqrt{d}$ for MQ applied to each coordinate.

### Algorithm 4 Decoder $Q^a_{k,R}(w, y)$ of RMQ

**Require:** Input $w \in \{0, \ldots, k-1\}^d$, $y \in \mathbb{R}^d$

1. Get $R$ from public randomness.
2. $y' = Ry$
3. **Output:** $Q^a_{k,R}(w, y) = R^{-1} \sum_{i \in [d]} Q^a_k(w(i), y'(i)) \epsilon_i$ using parameters $k$, $\varepsilon$, and $\Delta'$ for $Q^a_k$ of Alg. 2.

**Lemma 3:** Fix $\Delta \geq 0$. Let $Q_{k,R}$ be RMQ described in Alg. 3 and 4. Then, for $k \geq 4$, $\delta \in (0, \Delta)$, $\Delta' = \frac{\log k + \log(1/\delta)}{\log k}$.

6We assume that $d$ is a power of 2. If it isn’t, we can pad the vector by zeros to make it a power of 2; even in the worst-case, this only doubles the required bits.

7In practice, this can be implemented by using the same seed for pseudo-random number generator at encoder and decoder.

8We denote by $(e_1, \ldots, e_d)$ the standard basis of $\mathbb{R}^d$.

9In the proof, we provide a general bound which holds for all $k$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[ \frac{\sqrt{6(\Delta^2/d) \log(\Delta/\delta)}}{d} \] and the parameter \( \varepsilon \) of MQ set to \( \varepsilon = 2\Delta/(k-2) \), we have for \( X = Y = \mathbb{R}^d \) that
\[
\alpha(Q_{m, R}; \Delta) \leq \frac{24 \Delta^2}{(k-2)^2} \log \frac{\Delta}{\delta} + 154 \delta^2 \quad \text{and} \quad \beta(Q_{m, R}; \Delta) \leq 154 \delta^2.
\]

Furthermore, the output of quantizer \( Q_{m, R} \) can be described in \( d\log k \) bits.

**Remark 2:** The choice of \( \Delta' \) in the first statement of the Lemma 3 is based on Remark 1. We note that \( \delta \) is a parameter to control the bias incurred by our quantizer. By setting \( \Delta' = \Delta \) we can get an unbiased quantizer, but it only recovers the performance obtained by simply using MQ for each coordinate, an algorithm considered in [15] as well.

### C. Subsampled RMQ: A Wyner-Ziv Quantizer for \( \mathbb{R}^d \)

Our final quantizer is a modification of RMQ of previous section where we make the precision less than \( r \) bits by randomly sampling a subset of coordinates. Specifically, note that \( Q_{m, R}^d(x) \) sends \( d \) binary strings of \( \log k \) bits each. We reduce the resolution by sending only a random subset \( S \) of these strings. This subset is sampled using shared randomness and is available to the decoder, too. Note that \( Q_{n, R}^d \) applies \( Q_{m}^d \) to these strings separately; now, we use \( Q_{m}^d \) to decode the entries in \( S \) alone. The final estimate is scaled and centered appropriately to get an unbiased estimate. We describe the overall quantizer in Alg. 5 and 6.

**Algorithm 5** Encoder \( Q_{k, R}^d(x) \) of Subsampled RMQ

**Require:** Input \( x \in \mathbb{R} \), Parameters \( k, \Delta' \), and \( \mu 

1: Sample \( S \subset [d] \) u.a.r. from all subsets of \([d]\) of cardinality \( \mu d \) and sample \( R \) as in (8) using public randomness
2: **Output:** \( Q_{k, R}^d(x) = \{Q_{k}^d(Rz(i)) : i \in S\} \) using parameters \( k, \varepsilon \), and \( \Delta' \) for \( Q_{k}^d \) of Alg. 1

**Algorithm 6** Decoder \( Q_{k, R}^d(w, y) \) of Subsampled RMQ

**Require:** Input \( w \in \{0, \ldots, k-1\}^{\mu d}, y \in \mathbb{R} \)

1: Get \( S \) and \( R \) from public randomness
2: Compute \( \tilde{x} = (Q_{k}^d(w(i), R_y(i)), i \in S) \) using parameters \( k, \varepsilon \), and \( \Delta' \) for \( Q_{k}^d \) of Alg. 2
3: \( \tilde{x}_R = \frac{1}{\mu} \sum_{i \in S} (\tilde{x}(i) - R_y(i)) e_i + R_y 
4: **Output:** \( Q_{k, R}^d(w, y) = R^{-1}\tilde{x}_R 

**Remark 3:** In step 3 of Algorithm 6, the scaling of \( 1/\mu \) ensures that output of subsampled RMQ is an unbiased estimate of output of RMQ. We remark that, typically, when implementing random sampling, we set the unsampled components to 0. However, to get \( \Delta \) dependent bounds on MSE, we set the unsampled coordinates to the corresponding coordinate of side information and center our estimate appropriately to only have small bias.

The result below relates the performance of our final quantizer \( Q_{k, R} \) to that of \( Q_{m, R} \), which was already analysed in the previous section.

**Lemma 4:** Fix \( \Delta > 0 \). Let \( Q_{k, R} \) and \( Q_{m, R} \) be the quantizers described in Alg. 5 and 6 and Alg. 3 and 4, respectively. Then, for \( \mu d \in [d] \), we have for \( X = Y = \mathbb{R}^d \) that
\[
\alpha(Q_{k, R}; \Delta) \leq \frac{2\alpha(Q_{m, R}; \Delta)}{\mu} + \frac{2\Delta^2}{\mu} \quad \text{and} \quad \beta(Q_{k, R}; \Delta) = \beta(Q_{m, R}; \Delta).
\]

Furthermore, the output of quantizer \( Q_{k, R} \) can be described in \( \mu d \log k \) bits.

We are now equipped to prove our first main result. Our protocol \( \pi_k \) uses \( Q_{k, R} \) for each client as described in Section II and forms the estimate \( \hat{x} \) as in (4). We set the parameters needed for \( Q_{k, R} \) in Alg. 5 and 6 as follows: For client \( i \), we set the parameters of MQ as
\[
\delta = \frac{\Delta_i}{\sqrt{n}}, \quad \log k = \left\lceil \log(2 + \sqrt{12\ln n}) \right\rceil, \\
\Delta' = \sqrt{6(\Delta_i^2/d) \log(\Delta_i/\delta)}, \quad \varepsilon = \frac{2\Delta'/(k-2)}{2}, \quad (9)
\]
and set the parameter \( \mu \) as
\[
\mu d = \left\lceil \frac{r}{\log k} \right\rceil. \quad (10)
\]

We characterize the resulting error performance in the next result.

**Theorem 1:** For a \( n \geq 2 \), a fixed \( \Delta = (\Delta_1, \ldots, \Delta_n) \), and \( d \geq r \geq 2 \left\lceil \log(2 + \sqrt{12\ln n}) \right\rceil \), the protocol \( \pi_k \) with parameters as set in (9) and (10) is an \( r \)-bit protocol which satisfies
\[
\text{MSE}(\pi_k, x, y) \leq (79 \left\lceil \log(2 + \sqrt{12\ln n}) \right\rceil + 154) \left( \sum_{i=1}^{n} \frac{\Delta_i^2}{n} + \frac{2d}{nr} \right),
\]
for all \( x, y \) satisfying (2).

**Proof:** Denoting by \( Q_i \) the quantizer \( Q_{k, R} \) with parameters set for user \( i \), by Lemmas 1 and 4, we get
\[
\mathbb{E} \left[ \|\hat{x} - \tilde{x}\|^2 \right] \leq \sum_{i=1}^{n} \frac{\alpha(Q_i; \Delta_i)}{n^2} + \sum_{i=1}^{n} \frac{\beta(Q_i; \Delta_i)}{n} \leq \frac{1}{\mu n^2} \sum_{i=1}^{n} \left( \alpha(Q_{m, R, i}; \Delta_i) + \Delta_i^2 \right) + \sum_{i=1}^{n} \frac{\beta(Q_{m, R, i}; \Delta_i)}{n},
\]
where \( Q_{m, R, i} \) denotes RMQ with parameters set for user \( i \). Further, since \( k \geq 4 \) holds when \( n \geq 2 \) for our choice of parameters, by using Lemma 3 and substituting \( \delta^2 = \Delta_i^2/n \), we get
\[
\alpha(Q_{m, R, i}; \Delta_i) \leq \frac{12\Delta_i^2 \ln n}{(k-2)^2} + \frac{154\Delta_i^2}{n}, \quad \beta(Q_{m, R, i}; \Delta_i) \leq \frac{154\Delta_i^2}{n},
\]
which with the previous bound gives
\[
\mathbb{E} \left[ \| \hat{x} - \bar{x} \|^2 \right] 
\leq \frac{1}{\mu d} \left( \frac{12 \ln n}{(k-2)^2} + \frac{154}{n} + 1 + 154 \mu \right) \sum_{i=1}^{n} d \Delta_i^2 \frac{n^2}{n^2},
\]
where in the final bound we used our choice of \( k \) and the fact that \( [r/\log k] \geq r/2 \) if \( r \geq 2k \).

Remark 4: We note that by using MQ for each coordinate without rotating (or even with rotation using \( R \) as above) and with \( \Delta' = \Delta_i \) yields MSE less than
\[
O \left( \sum_{i=1}^{n} \Delta_i^2 \cdot \frac{d \log d}{n r} \right),
\]
for \( r \leq d \). Thus, our approach above allows us to remove the \( \log d \) factor at the cost of a (milder for large \( d \)) \( \log \log n \) factor. Thus, as can be seen from the lower bound presented in Theorem 2 below, our Wyner-Ziv estimator \( \pi_k \) is nearly optimal. Finally, \( Q_{\bar{y}} \) can be efficiently implemented as both the encoding and decoding procedures have nearly-linear time complexity\(^{10}\) of \( O(d \log d) \).

D. Lower Bound

We now prove a lower bound on the MSE incurred by any SMP protocol using \( r \) bits per client. The proof relies on the strong data processing inequality in [16] and is similar in structure to the lower bound for distributed mean estimation without side-information in [56].

Theorem 2: Fix \( \Delta = (\Delta_1, \ldots, \Delta_n) \). There exists a universal constant \( c < 1 \) such that for any \( r \)-bit SMP protocol \( \pi \), with \( r \leq c d \), there exists input \( (x, y) \in \mathbb{R}^{2d} \) satisfying (2) and such that
\[
\text{MSE}(\pi, x, y) \geq c \min_{i \in [d]} \Delta_i^2 d \frac{n}{nr}.
\]

IV. DISTRIBUTED MEAN ESTIMATION FOR UNKNOWN \( \Delta \)

Finally, we present our Wyner-Ziv estimator for the unknown \( \Delta \) setting. We first, in Section IV-A, describe the idea of correlated sampling from [22], which will serve as an essential building block for all our quantizers in this section. We then build towards our final quantizer, described in IV-D, by first describing its simpler versions in Section IV-B and IV-C. Once again, we restrict to the low-precision setting of \( r \leq d \).

A. The Correlated Sampling Idea

Suppose we have two numbers \( x \) and \( y \) lying in \([0, 1]\). A 1-bit unbiased estimator for \( x \) is the random variable \( \mathbb{1}_{\{U \leq x\}} \), where \( U \) is a uniform random variable in \([0, 1]\). The variance of such an estimator is \( x - x^2 \). We consider a variant of this estimator given by:
\[
\hat{X} = \mathbb{1}_{\{U \leq x\}} - \mathbb{1}_{\{U \leq y\}} + y, \tag{11}
\]
where, like before, \( U \) is a uniform random variable in \([0, 1]\). Such an estimator still uses only 1-bit of information related to \( x \). It is easy to check that this estimator unbiased estimator of \( x \), namely \( \mathbb{E}[\hat{X}] = x \). The variance of this estimator is given by
\[
\text{Var}(\hat{X}) = \mathbb{E}[\hat{X}^2] = |x - y| - (x - y)^2,
\]
which is lower than that of the former quantizer when \( x \) is close to \( y \). We build-on this basic primitive to obtain a quantizer with MSE bounded above by a \( \Delta \)-dependent expression, without requiring the knowledge of \( \Delta \).

B. Distance Adaptive Quantizer (DAQ)

DAQ and subsequent quantizers in this Section will be described for input \( x \) and side information \( y \) lying in \( \mathbb{R}^d \). The first component of our quantizer, DAQ, which uses (11) and incorporates the correlated sampling idea discussed earlier. Both the encoder and the decoder of DAQ use the same \( d \) uniform random variables \( \{U(i)\}_{i=1}^{d} \) between \([-1, 1]\), which are generated using public randomness. At the encoder, each coordinate of vector \( x \) is encoded to the bit \( \mathbb{1}_{\{U(i) \leq x(i)\}} \). At the decoder, using the bits received from the encoder, side information \( y \), and the public randomness \( \{U(i)\}_{i=1}^{d} \), we first compute bits \( \mathbb{1}_{\{U(i) \leq x(i)\}} \) for each \( i \in [d] \). Then, the estimate of \( x \) is formed as follows:
\[
Q_0(x, y) = \sum_{i=1}^{d} \left( \mathbb{1}_{\{U(i) \leq x(i)\}} - \mathbb{1}_{\{U(i) \leq y(i)\}} \right) e_i + y.
\]
We formally describe the quantizer in Alg. 7 and 8.

Algorithm 7 Encoder \( Q_0^e(x) \) of DAQ

Require: Input \( x \in \mathbb{R}^d \)
1: Sample \( U(i) \sim Unif([-1, 1]), \forall i \in [d] \)
2: \( \hat{x} = \sum_{i=1}^{d} \mathbb{1}_{\{U(i) \leq x(i)\}} \cdot e_i \)
3: Output: \( Q_0^e(x) = \hat{x} \), where \( \hat{x} \) is viewed as binary vector of length \( d \)

Algorithm 8 Decoder \( Q_0^d(w, y) \) of DAQ

Require: Input \( w \in \{0, 1\}^d \), \( y \in \mathbb{R}^d \)
1: Get \( U(i), \forall i \in [d] \), using public randomness
2: Set \( \hat{y} = \sum_{i=1}^{d} \mathbb{1}_{\{U(i) \leq y(i)\}} \cdot e_i \)
3: Output: \( Q_0^d(w, y) = 2(w - \hat{y}) + y \), where \( w \) is viewed as a vector in \( \mathbb{R}^d \)

The next result characterizes the performance for DAQ.

Lemma 5: Let \( Q_0 \) denote DAQ described in Algorithms 7 and 8. Then, for \( \mathcal{X} = \mathcal{Y} = \mathcal{B} \) and every \( \Delta > 0 \), we have
\[
\alpha(Q_0; \Delta) \leq 2 \Delta \sqrt{d} \quad \text{and} \quad \beta(Q_0; \Delta) = 0.
\]
Furthermore, the output of quantizer \( Q_0 \) can be described in \( d \) bits.
C. Rotated Distance Adaptive Quantizer (RDAQ)

Next, we proceed as for the known $\Delta$ setting and add a preprocessing step of rotating $x$ and $y$ using random matrix $R$ of (8), which is sampled using shared randomness. We remark that here random rotation is used to exploit the subgaussianity of the rotated $x$ and $y$, whereas in RMQ of previous section it was used to exploit the subgaussianity of $x - y$. After this rotation step, we proceed with a quantizer similar to DAQ, but we quantize each coordinate at multiple “scales.” We describe this step in detail below.

1) Using Multiple Scales: In DAQ, we considered each coordinate $x$ to be anywhere between $[-1, 1]$ and used one uniform random variable for each coordinate. Now, we will use $h$ independent uniform random variables for each coordinate, each corresponding to a different scale $[-M_j, M_j]$, $j \in \{0, 1, 2, \ldots, h-1\}$. For convenience, we abbreviate $[h_0] := \{0, 1, 2, \ldots, h-1\}$.

Specifically, let $U(i, j)$ be distribution uniformly over $[-M_j, M_j]$, independently for different $i \in [d]$ and different $j \in [h_0]$. The values $M_j$s correspond to different scales and are set, along with $h$, as follows: For all $j \in [h_0],

$$M_j^2 := \frac{6}{d} \cdot e^{s_j}, \quad \log h := \left\lfloor \log(1 + \ln^*(d/6)) \right\rfloor,$$

(12)

where $e^{s_j}$ denotes the $j$th iteration of $e$ given by $e^0 := 1, e^1 := e, e^{s_j} := e^{e^{s_{j-1}}}$. All the $dh$ uniform random variables are generated using public randomness and are available to both the encoder and the decoder.

The intervals $[-M_j, M_j]$ are designed to minimize the MSE of our quantizer by tuning its “resolution” to the “scale” of the input, and while still ensuring unbiased estimates. This idea of using multiple intervals $[-M_j, M_j]$ for quantizing the randomly rotated vector is from [44], where it was used for the case with no side information.

For a detailed reasoning behind our choice of $M_j$s, we refer the reader to the derivation of the mean square error upper bound of RDAQ given in Appendix C-G and particularly Remark 10.

2) Multiscale DAQ: After rotation, we proceed as in DAQ, except that we use different scale $M_j$ for different coordinates. Ideally, for the $i$th coordinate, we would like to use $M_{z^*(i)}$, where $z^*(i)$ is the smallest index such that both $Rx(i)$ and $Ry(i)$ lie in $[-M_{z^*(i)}, M_{z^*(i)}]$. However, since $y$ is not available to the encoder, we simplify resort to sending the smallest value $z(i)$ which is the smallest index such that $Rx(i) \in [-M_z(i), M_z(i)]$ and apply the encoder of DAQ $h$ times to compress $x$ at all scales, i.e., we send $h$ bits $(1_{(U(i,j) \leq Rx(i))} \cdot j \in [h_0])$.

Thus, the overall number of bits used by RDAQ’s encoder is $d \cdot (h + \lceil \log h \rceil)$. At RDAQ’s decoder, using $z(i)$, we compute the smallest index $z^*(i)$ containing both $Rx(i)$ and $Ry(i)$. In effect, the decoder emulates the encoder for DAQ applied to $Ry$, but for scale $M_{z^*(i)}$. The encoding and decoding algorithm of RDAQ are described in Alg. 9 and 10, respectively.

Algorithm 9 Encoder $Q_{B,R}^b(x)$ at for RDAQ

Required: Input $x \in \mathcal{B}$
1: Sample $U(i, j) \sim \text{Unif}[-M_j, M_j], i \in [d], j \in [h_0]$, and sample $R$ as in (8) using public randomness.
2: $x_R = Rx$
3: for $i \in [d]$
\quad $z(i) = \min\{j \in [h_0] : |x_R(i)| \leq M_j\}$
4: for $j \in [h_0]$
\quad $\tilde{x}_j = \sum_{i=1}^d 1_{(U(i,j) \leq Rx(i))} e_i$
5: Output: $Q_{B,R}^b(x) = ((\tilde{x}_0, \ldots, \tilde{x}_{h-1}), z)$, where we view $\tilde{x}_j$s as binary vectors

Algorithm 10 Decoder $Q_{D,R}^d(x, y)$ for RDAQ

Required: Input $(w, z) \in \{0, 1\}^{d \times h} \times [h]_0$ and $y \in \mathcal{B}$
1: Get $U(i, j), i \in [d], j \in [h_0]$, and $R$ using public randomness.
2: $y_R = Ry$
3: for $i \in [d]$
\quad $z'(i) = \min\{j \in [h_0] : |y_R(i)| \leq M_j\}$
\quad $z^*(i) = \max(\{z(i), z'(i)\})$
4: $w' = \sum_{i=1}^d 2M_{z^*(i)} (w(i, z^*(i)) - 1_{(U(i,z^*(i)) \leq y_R)}) e_i$
5: $\tilde{x}_R = w' + Ry$
6: Output: $Q_{D,R}^d(x, y) = R^{-1}\tilde{x}_R.$

Then, the quantized output $Q_{B,R}$ corresponding to input vector $x$ and side-information $y$ is

$$Q_{B,R}(x, y) = R^{-1} \left[ \sum_{i=1}^d 2M_{z^*(i)} (1_{(U(i,z^*(i)) \leq Rx(i))} \right.$$

$$\left. \quad - 1_{(U(i,z^*(i)) \leq Ry(i))}) e_i + Ry \right].$$

We remark that since rotated coordinates $Rx(i)$ and $Ry(i)$ have subgaussian tails, with very high probability $M_{z^*(i)}$ will be much less than 1, which helps in reducing the overall MSE significantly. The performance of the algorithm is characterized below.

Lemma 6: Let $Q_{B,R}$ be RDAQ described in Alg. 9 and 10. Then, for $X = Y = \mathcal{B}$ and every $\Delta > 0$, we have

$$\alpha(Q_{B,R}; \Delta) \leq 16\sqrt{3}\Delta$$

and $\beta(Q_{B,R}; \Delta) = 0$.

Furthermore, the output of quantizer $Q$ can be described in $d(h + \log h)$ bits.

D. Subsampled RDAQ: A Universal Wyner-Ziv Quantizer for Unit Euclidean Ball

Finally, we bring down the precision of RDAQ to $r$, as before for the known $\Delta$ setting, by retaining the output of RDAQ for only coordinates $i \in S$, where $S$ is generated uniformly at random from all subsets of $[d]$ of cardinality $\mu d$ using public randomness. Specifically, we execute Alg. 9 and 10 with $S$ replacing $[d]$ and multiplying $w'$ in Step 4 of Alg. 10 by normalization factor of $d/|S|$. The output of the
resulting encoder is given by
\[ Q_{\pi}^{*}(x) = \{ Q_{\pi}^{*}(x)(i) : i \in S \}, \] (13)
where \( Q_{\pi}^{*}(x)(i) \) represents the encoded bits \((\bar{x}_0(i), \ldots, \bar{x}_{k-1}(i), z(i))\) for the \( i \)-th coordinate using RDAQ, and the output of the resulting decoder is given by
\[ Q_{\pi}^{*}(x, y) = R^{-1} \left[ \frac{1}{\mu} \sum_{i \in S} 2Mz^{*}(i) \left( I\{U(i,z^{*}(i)) \leq Rz(i)\} - I\{U(i,z^{*}(i)) \leq Rq(i)\} \right) e_i + Ry \right]. \] (14)

**Lemma 7:** Let \( Q_{\pi}^{*} \) be the quantizers described in (13) and (14) and \( Q_{\pi}^{*} \) be RDAQ described in Alg. 9 and 10. Then, for \( \mu \in \mathbb{R} \), \( \alpha \in \mathbb{R} \), and every \( \Delta > 0 \), we have
\[ \alpha(Q_{\pi}^{*}; \Delta) \leq \frac{\alpha(Q_{\pi}^{*}; \Delta)}{\mu} \quad \text{and} \quad \beta(Q_{\pi}^{*}; \Delta) = 0. \]

Furthermore, the output of quantizer \( Q_{\pi}^{*} \) can be described in \( \mu d(h + \log h) \) bits.

We are now equipped to prove our second main result. Our protocol \( \pi^{*} \) uses \( Q_{\pi}^{*} \) for each client as described in Section II and forms the estimate \( \hat{x} \) as in (4). Unlike for the known \( \Delta \) setting, we now use the same parameters for \( Q_{\pi}^{*} \) for all clients, given by
\[ \mu d = \left\lceil \frac{r}{h + \log h} \right\rceil. \] (15)

**Theorem 3:** For \( d \geq r \geq 2(h + \log h) \) and \( h \) given in (12), the \( r \)-bit protocol \( \pi^{*} \) with parameters as set in (15) satisfies
\[ \operatorname{MSE}(\pi^{*}, x, y) \leq \left( 128\sqrt{3} \left( 1 + \ln^*(d/6) \right) \right) \left( \sum_{i=1}^{n} \frac{\Delta_i}{n} \cdot \frac{d}{nr} \right), \]
for all \( x, y \) satisfying (2), for every \( \Delta = (\Delta_1, \ldots, \Delta_n) \).

**Proof:** Denote by \( \hat{x} \) the output of the protocol. Then, by Lemmas 1 and 7, we get
\[ \operatorname{E}[\|\hat{x} - \bar{x}\|^2] \leq \frac{1}{n^2 \mu} \sum_{i=1}^{n} \alpha(Q_{\pi}^{*}; \Delta_i) \leq 16\sqrt{\frac{2}{n^2 \mu}} \sum_{i=1}^{n} \Delta_i, \]
where the previous inequality is by Lemma 6. The proof is completed by using \( \mu \geq \frac{r}{2(d(h + \log h))} \geq \frac{r}{4dn} \), which follows from (15) and the assumption that \( r \geq 2(h + \log h) \).

The Wyner-Ziv estimator \( \pi^{*} \) is universal in \( \Delta \): it operates without the knowledge of the distance between the input and the side information and yet gets MSE depending on \( \Delta \). Moreover, it can be efficiently implemented as both the encoding and the decoding procedures have nearly linear time complexity of \( O(d \log d) \).

### V. The High-Precision Regime

#### A. RMQ in the High-Precision Regime

For the known \( \Delta \) setting, our quantizer RMQ described in Alg. 4 and 5 remains valid even for \( r > d \). We will assume \( r = md \) for integer \( m \geq 2 \). For each client \( i \), we set
\[ \delta = \frac{\Delta_i}{n^2(2^{r/d} - 2)}, \quad \log k = \frac{r}{d} \]
\[ \Delta' = \sqrt{6(\Delta_i^2/d) \ln \delta_i}/\varepsilon = \frac{2\Delta'}{k - 2}. \] (16)

The performance of protocol \( \pi^{*} \) using RMQ with parameters set as in (16) for each client can be characterized as follows.

**Theorem 4:** For a fixed \( \Delta = (\Delta_1, \ldots, \Delta_n) \) and \( r = md \) for integer \( m \geq 2 \), the protocol \( \pi^{*} \) with parameters set as in (16) satisfies
\[ \operatorname{MSE}(\pi^{*}, x, y) \leq \left( 12n \ln n + \frac{24r}{d} + 154/n + 164 \right) \cdot \left( \sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{1}{n^2(2^{r/d} - 2)^2} \right), \]
for all \( x, y \) satisfying (2).

**Proof:** Denoting by \( Q_i \) the quantizer \( Q_{\pi, i} \) with parameters set for client \( i \), by Lemmas 1 and 3, we get
\[ \operatorname{E}[\|\hat{x} - \bar{x}\|^2] \leq \sum_{i=1}^{n} \alpha(Q_i; \Delta_i) + \frac{1}{n} \sum_{i=1}^{n} \beta(Q_i; \Delta_i) \]

Further, since \( k \geq 4 \) holds when \( r \geq 2d \) for our choice of parameters, by using Lemma 3 and substituting \( \delta^2 = \Delta_i^2/(n(2^{r/d} - 2)^2) \), we get
\[ \alpha(Q_i; \Delta_i) \leq \frac{12\Delta_i^2 \ln(n(2^{r/d} - 2)^2)}{(2^{r/d} - 2)^2} + \frac{154\Delta_i^2}{n(2^{r/d} - 2)^2}, \]
\[ \beta(Q_i; \Delta_i) \leq \frac{154\Delta_i^2}{n(2^{r/d} - 2)^2}. \]

which with the previous bound gives
\[ \operatorname{E}[\|\hat{x} - \bar{x}\|^2] \leq \left( 12n \ln n + \frac{24r}{d} + 154/n + 154 \right) \sum_{i=1}^{n} \frac{\Delta_i^2}{n^2(2^{r/d} - 2)^2}, \]
where use the inequality \( \ln x \leq x, \forall x \geq 0 \), to bound \( \ln(2^{r/d} - 2)^2/(2^{r/d} - 2)^2 \) by 1.

**Remark 5:** Similar to Remark 4, we note that using MQ for each coordinate without rotating (or even with rotation using \( R \) as above) with \( \Delta = \Delta_i \) yields MSE less than
\[ O \left( \sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d}{n^22^{r/d}} \right), \]
for \( r \geq d \). Thus our approach above allows us to remove the \( d \) factor at the cost of a (milder for large \( d \)) \( \log n + r/d \) factor.
B. Boosted RDAQ: RDAQ in the High-Precision Regime

Moving to the unknown \( \Delta \) setting, we describe an update to RDAQ described in Alg. 10 and 11 for the high-precision setting. For brevity, we denote by \( m := r/d \) the number of bits per dimension. A straightforward scheme to make use of the high precision is to independently implement the RDAQ quantizer approximately \( \lfloor m/\ln^* d \rfloor \) times and use the average of the quantized estimates as the final estimate. We will see that the MSE incurred by such an estimator is \( O(\Delta \ln^* d/m) \). We will show that this naive implementation can be significantly improved and an exponential decay in MSE with respect to \( m \) can be achieved.

We boost RDAQs performance as follows. Simply speaking, instead of sending the bits produced by multiple instances of the encoder of RDAQ, we send the “type” of each sequence. A similar idea appeared in [43] for the case without any side information. At the encoding stage of RDAQ given in Alg. 10 and 11, after random rotation and computing \( y \) in Steps 1 to 3 of Alg. 10, we repeat Step 4 \( N \) times with independent randomness each time and store only the total number of ones seen for each coordinate \( j \). Specifically, let \( U_i(j) \) be an independent uniform random variable in \([-M_j, M_j]\), for all \( i \in [d], j \in [h]_0 \), and \( t \in [N] \), which are generated using public randomness between the encoder and the decoder. Using this randomness, we compute \( \tilde{x}_{j,t} = \sum_{i=1}^{d} \mathbb{I}_{\{U_i(j) \leq x(R_{i})\}} e_i \) for each \( j \in [h]_0 \). Then, instead of storing \( \tilde{x}_{j,t} \) for each \( j \) and \( t \), we store the sum \( \sum_{t=1}^{N} \tilde{x}_{j,t} \) for each \( j \in [h]_0 \). Since each coordinate of the sum can be stored in \( \log(N+1) \) bits, the new encoder’s output can be stored in \( d(h \log(N+1) + \log h) \) bits. Thus, we can implement this scheme by using \( m = (h \log(N+1) + \log h) \) bits per dimension.

At the decoding stage, we rotate \( y \) and compute \( z^* \) in precisely the same manner as done in Steps 1 to 3 of the decoding Alg. 11 of RDAQ. Then, using the encoded input received, the side-information \( y \), the same random variables \( U_i(j) \) and random matrix \( R \) used by the encoder, the final estimate \( Q(x) \) is

\[
Q(x) = R^{-1} \left( \frac{1}{n} \cdot \sum_{i \in [d]} \sum_{t \in [N]} (B_{i,Rx} - B_{i,Ry}) e_i + R y \right),
\]

where \( B_{i,v} = \mathbb{I}_{\{U_i(z^*(i)) \leq v(i)\}} \) for \( v \in \mathbb{R}^d \).

The result below characterizes the performance of our quantizer Boosted RDAQ \( Q \).

**Lemma 8:** Let \( Q \) be Boosted RDAQ described above. Then, we have for \( \mathcal{X} = \mathcal{Y} = \mathbb{R}^d \) and every \( \Delta > 0 \), we have

\[
\alpha_u(Q; \Delta) \leq \frac{16\sqrt{3} \Delta}{N} \quad \text{and} \quad \beta_u(Q; \Delta) = 0.
\]

Furthermore, the output of the quantizer can be described in \( d(h \log(N+\log h)) \) bits.

Thus, when we have a total precision budget of \( r = dm \) bits using the Boosted RDAQ algorithm with number of repetitions \( N = 2\lceil (m-\log h)/h \rceil - 1 \), we get an exponential decay in MSE with respect to \( m \).

We consider the protocol \( \pi_u^* \) that uses the \( Q \) above for each client with \( M_j \) and \( h \) set as in (12), i.e., with

\[
N = 2\lceil (m-\log h)/h \rceil - 1, \quad M_j^2 = \frac{6e\gamma^*}{d}, \quad j \in [h]_0,
\]

\[
\log h = \lceil \log(1 + \ln^*(d/6)) \rceil.
\]

Therefore, by the previous lemma and Lemma 1, we get the following result.

**Theorem 5:** For \( r = dm \) with integer \( m \geq h + \log h \), the protocol \( \pi_u^* \) with parameters as set in (18) satisfies

\[
\text{MSE}(\pi_u^*, x, y) \leq \sum_{i \in [n]} \frac{\Delta_i}{n} \cdot \frac{64\sqrt{3}}{n^2 r^*(d(2+2\ln^*(6d)))},
\]

for all \( x, y \) satisfying (2), for every \( \Delta = (\Delta_1, \ldots, \Delta_n) \).

**Proof:** Denote by \( \hat{x} \) the output of the protocol. Then, by Lemmas 1 and Lemma 8, we get

\[
\mathbb{E} \left[ \|\hat{x} - x\|^2 \right] \leq \frac{1}{n^2} \sum_{i=1}^{n} \alpha(Q; \Delta_i)
\]

\[
\leq \frac{16\sqrt{3}}{n^2 N} \sum_{i=1}^{n} \Delta_i,
\]

where the previous inequality is by Lemma 8. The proof is completed by using

\[
N \geq \frac{2^{m/h}}{2^{1 + (\log h)/h}} \geq \frac{2^{m/h}}{4} \geq \frac{2^{m/(2 + 2\ln^*(6d))}}{4},
\]

where the first inequality follows from using \( |x| \geq x - 1 \) for the floor function in the value of \( N \) in (18), the second follows from the fact that \( \log x \leq x, \forall x \geq 0 \), and the third follows from \( |x| \leq x + 1 \) for the ceil function in the value of \( h \) in (18).

\]

VI. NUMERICAL EXPERIMENTS

We empirically demonstrate the performance of our proposed quantizers on the following mean estimation task.

Each client \( i \) has a \( d \)-dimensional vector \( x_i = \mu + U_i \), where \( \mu \in [0,1]^d \) is constant mean vector and \( U_i \) is a random vector whose each coordinate is a Uniform random variable in \([-\Delta', \Delta']/2\). The server has side information \( y_i \) corresponding to \( x_i \), where \( y_i = \mu + U_i^s \), and \( U_i^s \), too, is a random vector whose each coordinate is a Uniform random

\[
10^{-3} \quad 10^{-2} \quad 10^{-1} \quad 10^0 \quad 10^1 \quad 10^2 \quad 10^3 \quad 10^4 \quad 10^5
\]

\[
\Delta'
\]

Fig. 2. Comparison of RMQ, RDAQ, and RATQ at per coordinate precision of 6 bits and \( d = 512 \).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We see in Figures 2, 3, 4 and 5 that RMQ comfortably outperforms the other two quantizers at possible parameter choices. This is expected since the RMSE of RMQ is directly proportional to $\Delta'$, which is very small in our experiments.

For RDAQ and RATQ, we first normalize the vectors $\{x_i, y_i\}_{i \in [n]}$ using an bound $\sqrt{d}(1 + \Delta')/2$ on their $\ell_2$-norm. Then, we set $h = 4$ to compute the different scales $M_j$s in (18) for dimensions $d = 512, 1024$. In addition, we choose $N = 1$ and $N = 3$ for implementing 6 bit and 10 bit Boosted RDAQ, respectively. The final estimate is obtained by multiplying back the decoded output with $\sqrt{d}(1 + \Delta')/2$.

We use the following parameters for all the quantizers. For RMQ, we set $\epsilon = \frac{2\Delta'}{d}$ and $\frac{2\Delta'}{3d^{11}}$ for precision 6 bits and 10 bits, respectively. For RDAQ and RATQ, we first normalize the vectors $\{x_i, y_i\}_{i \in [n]}$ using an bound $\sqrt{d}(1 + \Delta')/2$ on their $\ell_2$-norm. Then, we set $h = 4$ to compute the different scales $M_j$s in (18) for dimensions $d = 512, 1024$. In addition, we choose $N = 1$ and $N = 3$ for implementing 6 bit and 10 bit Boosted RDAQ, respectively. The final estimate is obtained by multiplying back the decoded output with $\sqrt{d}(1 + \Delta')/2$.
will degrade much faster than RDAQ as the accuracy of side information degrades. Also, as seen from these figures, RDAQ is better than RATQ at lower values of $\Delta'$. However, for higher values of $\Delta'$, RATQ is better than RDAQ. This can be explained by the fact that, roughly, the RMSE of RDAQ is directly proportional to $\sqrt{\Delta'/2^{1/2}}$, whereas RMSE of RATQ is directly proportional to $1/2^r$, where $r$ is the per-coordinate precision.

In the other direction, we note that for all our protocols, there is a slight increase in RMSE for the same $\Delta'$ and bit precision as the dimension increases from 512 to 1024. This is because $\ell_2$ norm of the input and the $\ell_2$ distance between input and side information depend on the dimension for our example, and our MSE upper bounds for all the quantizers depend on either one or both of these quantities.

In Figure 6, we demonstrate the RMSE of these protocols for the different numbers of clients, namely, at $n = 5, 10, 20, 30, 40$ and 50. Specifically, we fix $\Delta' = 0.1$ and average the experiment over ten runs for statistical consistency. As can be seen in Figure 6 and as expected, the RMSE decreases with an increase in $n$ for all protocols.

Finally, we end with a remark on our choice of precision levels of 6 bits and 10 bits per dimension for this experiment. Notice that similar trends can be observed for precision levels lesser than dimension $d$. However, setting close to optimal parameters for these quantizers would have been much more tedious at precision levels lesser than the dimension. Since our experiment aimed to study the impact of side information on the accuracy of distributed mean estimation, we chose not to experiment with precision levels lesser than the dimension. The reason for not experimenting at 1 or 2 bits per dimension is that RDAQ is not operational below 6 bits per coordinate for the current dimension.

VII. CONCLUSION AND FUTURE WORK

In this paper, we described two different distributed mean estimation protocols that outperform previous protocols using side information. We also showed the application of our protocols to a distributed optimization problem. Our work shows that side information can be a valuable resource in many distributed learning scenarios. Exploring various side information sources and their usage needs to be further explored in distributed learning.

We also recommend experimenting with various distributed mean estimation protocols when they need to be used in a distributed learning application, since the best-performing one may vary with respect to the application, as seen in our experiments.

Finally, in Appendix A, we showed that our mean estimation protocols could improve the convergence of the parallel stochastic gradient descent algorithms under a bounded gradient assumption. Further exploring if our protocols can be used to improve the convergence of other sophisticated optimization algorithms, such as momentum-based algorithms, for instance, and under weaker assumptions on gradient bound are interesting future directions.

APPENDIX A

APPLICATION 1: COMMUNICATION CONSTRAINED DISTRIBUTED OPTIMIZATION

We consider the problem of minimizing an unknown convex function $f: \mathcal{X} \to \mathbb{R}$ over its domain $\mathcal{X} \subset \mathbb{R}^d$ using the set of $n$ clients who have access to independent noisy gradients of the function. In particular, the server runs an optimization algorithm, which is not directly given access to the function but can get $n$ different gradient estimates of the function at various points of its choice. This class of optimization algorithms includes various descent algorithms, which provide close to optimal convergence rate within the class and are appealing in practice due to their distributed nature.

Owing to our setup, the gradient estimates supplied by the $n$ clients must pass through $r$-bit quantizers, chosen from a fixed set of quantizers $Q_r$. The optimization algorithm $A$ only has access to the quantized outputs.

Our objective is to select quantizers $Q_{i,t}$, $\forall i \in [n], t \in [T]$, and an optimization algorithm $A$ to guarantee the minimum worst-case optimization error defined below. In our setting, we allow for adaptive gradient processing, whereby, the quantizer $Q_{i,t}$ selected in $t$th iteration may depend on all the previous quantized outputs. Specifically, denoting by $C_{i,t}$ the $i$th client’s quantized output at time $t$, which takes values in the output alphabet $\mathbb{R}^d$, the adaptive quantizer selection strategy $S := (S_1, \ldots, S_T)$ over $T$ iterations consists of mappings $S_i: \mathbb{R}^{d \times n \times (t-1)} \to Q^n_r$ that take $\{C_{i,t'}\}_{i \in [n], t' \in [t-1]}$ as input and outputs a tuple of $n$ quantizers $\{Q_{i,t}, l \in [n]\} \in Q^n_r$. We write $S_{Q, r, T}$ for the collection of all such quantizer selection strategies. The entire framework can be summarized as follows:

1. At iteration $t$, the first-order optimization algorithm $A$ makes a query for point $x_t$ to clients $C_1, \ldots, C_n$.
2. Upon receiving the point $x_t \in \mathcal{X}$, the client $C_i$, $i \in [n]$, outputs $\hat{g}_i(x_t)$, an unbiased estimate of $\nabla f(x_t)$.
3. The gradient estimate $\hat{g}_i(x_t)$ is passed through a quantizer $Q_{i,t} \in Q_r$ chosen based on strategy $S$, and the output $Y_{i,t}$ is observed by the first-order optimization algorithm $A$. The algorithm then uses all the messages $\{C_{i,t'}(x_t)\}_{i \in [n], t' \in [t]}$ to further update $x_t$ to $x_{t+1}$.

Denote by $C$ the collection of $n$ clients $(C_1, \ldots, C_n)$. Let $A_T$ be the set of all first-order optimization algorithms that make $T$ queries to $C$ and for the $t$th query $x_t$, get back the outputs $\{Y_{i,t}\}_{i \in [n]}$. We measure the performance of an optimization protocol $A$ and a quantizer selection strategy $S$ for a given function $f$ and clients $C_1, \ldots, C_n$, $i \in [n]$, using the metric $E(f, C, A, S)$ defined as

$$E(f, C, A, S) = \mathbb{E} \left[ f(\bar{x}_T) - \min_{x \in \mathcal{X}} f(x) \right],$$

where $\bar{x}_T := \frac{1}{T} \sum_{t \in [T]} x_t$ and the expectation is over the randomness in $\bar{x}_T$.

For a set of various function and client pairs above, denoted by $O$, the set of $r$-bit quantizers $Q_r$ and the number of quantizers $Q$.

13The set of $r$-bit quantizers $Q_r$ is used to model the communication constraints in a distributed setting.
iterations \(T\), we define the minimax optimization error as

\[
\mathcal{E}^*(\mathcal{X}, \mathcal{O}, T, \mathcal{Q}_r) = \inf_{A \in \mathcal{A}_T} \inf_{S \in \mathcal{S}_{\mathcal{O}, \mathcal{T}}} \sup_{(f, \mathcal{C}, \mathcal{A}, S)} \mathcal{E}(f, \mathcal{C}, \mathcal{A}, S).
\]

We now define the class of functions and state the assumptions related to the clients accessible to the algorithm \(A\).

**Convex and Smooth Function Family:** Throughout, we restrict ourselves to convex and \(L\)-smooth functions over \(\mathcal{X} \subseteq \mathbb{R}^d\), i.e., functions satisfying, \(\forall x \in [0, 1], \forall y, x, y \in \mathbb{R}^d\),

\[
f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \tag{19}
\]

\[
\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \tag{20}
\]

where \(\nabla f(x) \in \mathbb{R}^d\) denotes the gradient of \(f\) at input \(x\).

**Stochastic Gradients:** We assume that the update \(\hat{g}_i(x)\) by client \(\mathcal{C}_i, 1 \leq i \leq n\), when a point \(x \in \mathcal{X}\) is queried satisfies the following conditions:

\[
E[\hat{g}_i(x) | x] = \nabla f(x), \quad \text{(unbiased estimates)} \tag{21}
\]

\[
\|\hat{g}_i(x) - \nabla f(x)\|_2 \leq \sigma^2, \quad \text{(maximum deviation bound)} \tag{22}
\]

\[
\|\hat{g}_i(x)\|_2 \leq B^2, \quad \text{(a.s. bounded estimate)} \tag{23}
\]

**Assumption (21)** is standard in stochastic optimization literature (c.f. [13], [45], [46]). However, it is enough to assume a bound on the variance of stochastic gradients instead of (22) to prove convergence guarantees for smooth stochastic optimization without any communication constraints. The stronger assumption made here is to aid a much tighter analysis under communication constraints. In Section A-E, we provide a scheme which can operate under the standard variance bound. Denote by \(\mathcal{O}_{sc}\) the set of tuples of function and \(n\) clients, \((f, \mathcal{C}),\) satisfying (19), (20), (21), (22) and (23).

### A. Lower Bound

The following bound will serve as a basic benchmark for our problem. Let \(D > 0\) and \(\mathcal{X}_2(D) := \{\mathcal{X} \subseteq \mathbb{R}^d : \max_{x,y \in \mathcal{X}} \|x - y\|_2 \leq D\}\) be the collection of subsets of \(\mathbb{R}^d\) whose \(l_2\) diameter is at most \(D\).

**Theorem 6:** There exists an absolute constant \(0 \leq c_0 \leq 1\) such that for \(r \leq d\) and \(T \geq d/(6nr)\),

\[
\sup_{\mathcal{X} \in \mathcal{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{sc}, T, \mathcal{Q}_r) \geq \frac{c_0 D \sigma}{\sqrt{n} T} \cdot \sqrt{\frac{d}{r}}.
\]

### B. A General Convergence Bound

We present a general convergence bound based on a non-adaptive channel strategy. In particular, we fix same quantization process in every iteration, and the quantized outputs \(\{C_{i,t}\}_{i \in [n]}\) are passed through a mapping \(14\) \(\mathcal{M} : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^d\) in order to update the query.

We use PSGD as the first-order optimization algorithm; the overall optimization procedure is described in Algorithm 11. PSGD proceeds as SGD, with the additional projection step where it projects the updates back to domain \(\mathcal{X}\) using the map \(\Gamma_{\mathcal{X}}(y) := \min_{x \in \mathcal{X}} \|x - y\|, \forall y \in \mathbb{R}^d\).

Algorithm 11: PSGD Using Clients \(\mathcal{C}\)

1. for \(t = 0\) to \(T - 1\) do
2. \(x_{t+1} = \Gamma_{\mathcal{X}}(x_t - \eta_t \mathcal{M}(C_{1,t}, \ldots, C_{n,t}))\)
3. Output \(\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t\)

The convergence rate of Algorithm 11 is controlled by the worst-case \(L_2\)-norm \(\alpha' (\mathcal{M})\) and the worst-case bias \(\beta' (\mathcal{M})\) defined as

\[
\alpha' (\mathcal{M}) := \sup_{\{\forall x, i \in [n], \hat{g}_i \in \mathbb{R}^d, \|\hat{g}_i - \nabla f(x)\|_2 \leq \sigma^2\}} \sqrt{E[\|\mathcal{M}(C^n) - \nabla f(x)\|^2]}, \tag{24}
\]

\[
\beta' (\mathcal{M}) := \sup_{\{\forall x, i \in [n], \hat{g}_i \in \mathbb{R}^d, \|\hat{g}_i - \nabla f(x)\|_2 \leq \sigma^2\}} \|E[(\mathcal{M}(C^n) - \nabla f(x))\|}, \tag{25}
\]

where \(C^n = (C_1, \ldots, C_n)\) is the communication received at the server. Using a slight modification of the standard proof of convergence for PSGD in [13, Theorem 6.3], we can derive the following lemma.

**Lemma 9:** For any mapping \(\mathcal{M}\) and set of quantizers \(\{Q_i\}_{i \in [n]}\) defined above, the output \(\bar{x}_T\) of optimization algorithm given in Algorithm 11 satisfies

\[
\sup_{(f, \mathcal{C}, \mathcal{A}, S)} \mathcal{E}(f, \mathcal{C}, \mathcal{A}, S) \leq \frac{\sqrt{2\alpha' (\mathcal{M})D}}{\sqrt{T}} + \beta' (\mathcal{M}) \left( D + \frac{DB}{\alpha' (\mathcal{M}) \sqrt{2T}} \right) + \frac{LD^2}{2T},
\]

with the learning rate \(\eta_t = \frac{1}{L + \alpha' (\mathcal{M}) \sqrt{2T}}\), \(\forall t \in [T]\).

### C. Baseline Scheme: Parallel SGD

We begin by presenting the convergence result for the baseline scheme in our setup: the Parallel SGD algorithm. In Parallel SGD, all clients compress their stochastic gradient estimates to \(r\) bits using an efficient quantizer for the Euclidean ball and send it to the server, which then takes the average of the quantized gradients for the projected gradient descent step. We choose subsampled RATQ [44] for this efficient quantizer. We denote by \(\hat{Q}_{\text{ratq}}\) the subsampled version of RATQ using \(r\) bits, which is described in [44, Section 3.5]. After receiving the quantized outputs \(C_{i,t} = \hat{Q}_{\text{ratq}}(\hat{g}_i(x_t))\) \(\forall i \in [n]\); from all the \(n\) clients, the server takes the mapping \(\mathcal{M}\) to be the average of these outputs, i.e.,

\[
\mathcal{M}(C_t) = \frac{1}{n} \sum_{i=1}^{n} \hat{Q}_{\text{ratq}}(\hat{g}_i(x_t)). \tag{26}
\]

**Theorem 7:** Let \(S\) be the quantizer selection strategy which fixes the quantizer to be \(\hat{Q}_{\text{ratq}}\) for all clients at all iterations. Let \(A\) be the optimization algorithm described in Algorithm 11 where \(\mathcal{M}\) as described in (26) is used to make the PSGD step after the \(t\)-th query and the learning rate \(\eta_t = \frac{1}{L + \alpha' (\mathcal{M}) \sqrt{2T}}\), where \(\alpha' (\mathcal{M}) = c_0 \sqrt{\frac{\sigma^2}{n} + \frac{c_2dB^2 \log \log d}{n}}\) for some positive universal constant \(c_0\). Then, for positive universal constants \(c_1\) and \(c_2\) and \(r\) such that \(d \geq r \geq c_1 \log \log d\), we have

\[
\mathcal{E}(f, \mathcal{C}, \mathcal{A}, S) \leq \frac{c_2 D}{\sqrt{n} T} \sqrt{\frac{\sigma^2 + \frac{c_2dB^2 \log \log d}{r}}{r}} + \frac{LD^2}{2T}.
\]
We note that the term $\frac{d B^2 \log \log d}{r}$ illustrates the slowdown in convergence due to quantization error. This is nearly the best rate which can be achieved when one uses $r$-bit quantizers without any side information.\footnote{Similar convergence bounds (up to $\log \log d$ factor) for parallel SGD can be achieved by using subsampled version of rotated quantizer in [56] or the subsampled version of uniform quantizer after preprocessing due to Kashin’s representation (cf. [29], [40]).} Note that in cases in which $B$ is large relative to $\sigma^2$, the slowdown due to this term can be significant, and the algorithm maybe far away from our lower bound in Theorem 6.

**Algorithm 12 WZ-SGD Algorithm**

1: for Clients $i \in [n]$ do \hspace{0.5cm} \triangleright Setting quantizers
2: \hspace{0.5cm} if $i \in C_1$ then $Q_e = Q_u$
3: \hspace{0.5cm} else $Q_e = Q_{WZ,i}$
4: Initialize $x_0 \in \mathcal{X}$
5: for $t = 0$ to $T - 1$ do
6: \hspace{0.5cm} for Server do
7: \hspace{2cm} Broadcast $x_t$ to clients
8: \hspace{0.5cm} for Clients $i \in [n]$ do \hspace{0.5cm} \triangleright Encoding
9: \hspace{2cm} Compute $\hat{g}_i(x_t)$
10: \hspace{2cm} Send $Q^2_1(\hat{g}_i(x_t))$ to server
11: \hspace{0.5cm} for Server do \hspace{0.5cm} \triangleright Decoding
12: \hspace{2cm} for $i \in C_1$ do
13: \hspace{3cm} $Q^2_1(\hat{g}_i(x_t)) = Q^2_1(Q^1_1(\hat{g}_i(x_t)))$
14: \hspace{2cm} $Y_t = \frac{2}{\ell} \sum_{i \in C_1} Q^1_1(\hat{g}_i(x_t))$ \hspace{0.5cm} \triangleright Side information
15: \hspace{2cm} for $i \in C_2$ do
16: \hspace{3cm} $Q^1_1(\hat{g}_i(x_t), Y_t) = Q^2_1(Q^1_1(\hat{g}_i(x_t)), Y_t)$
17: \hspace{2cm} $x_{t+1} = \Gamma x_t - \eta_t \cdot \frac{2}{\ell} \sum_{i \in C_1} Q^1_1(\hat{g}_i(x_t), Y_t) Y_t$
18: \hspace{0.5cm} At Server Output: $\hat{x}_T = \frac{1}{T} \sum_{t \in [T]} x_t$

**D. WZ-SGD: An Almost Optimal Algorithm for Distributed Optimization**

We now present our main algorithm: WZ-SGD. WZ-SGD uses our first Wyner-Ziv estimator (see Section III-C) based on subsampled RMQ as a subroutine to form much more accurate gradient estimates compared to those formed in ParallelISGD. As a result of this, WZ-SGD significantly improves over the convergence rate of Theorem 7 and relegates the dependence of convergence rate on $B$ to only second order terms.

At each iteration $t$, WZ-SGD uses the clients in $C_1$ to form the side information estimate $Z_t$ at the server and then uses the clients in $C_2$ to estimate the gradient for performing the descent step, where\footnote{For simplicity, we assume that $n/2$ and $d/r_1$ are integers such that $d/r_1$ divides $n/2$.} $C_1 := \{C_{1,1}, \ldots, C_{1,n/2}\}$, $C_2 := \mathcal{C} \setminus C_1$.

1) The Side Information Estimate $Y_t$: The side information is formed as follows. Under the $r$-bit communication constraint, we divide the coordinates into blocks of dimension $r_1$, where $r_1 := \lceil r / \log r_1 \rceil$, and $\log r_1$ denotes the precision bits used by clients to represent each coordinate in the assigned block. This way we have $d/r_1$ blocks. We also equally partition the set $C_1$ into $d/r_1$ groups. Further, we assign every block of $r_1$ coordinates to every other group of $n/2$ $d/r_1$ clients.

To quantize the coordinates within any block, the group of clients assigned to that block will use a coordinate-wise uniform quantizer (CUQ). CUQ is an unbiased, uniform quantizer that has appeared recently in many works on gradient quantization. We denote by $Q_a : [-B, B] \rightarrow \{-B + 2B \cdot (i - 1)/(\ell_1 - 1) : i \in [\ell_1]\}$ the $\ell_1$-level CUQ quantizer. For a scalar input $x \in [-B, B]$,

$$Q_a(x) = \begin{cases} 
\frac{x(\ell_1 - 1)}{2B} \ell_1, & \text{w.p.} \quad \frac{x(\ell_1 - 1)}{2B} \ell_1 - x \\
\frac{x(\ell_1 - 1)}{2B} \ell_1, & \text{w.p.} \quad \frac{x(\ell_1 - 1)}{2B} \ell_1 - x 
\end{cases} \quad (27)
$$

Each client uses an $\ell_1$-level CUQ to quantize the associated block of coordinates separately. Thus, the overall communication by each client is $r_1 \cdot \log \ell_1 = r$ and satisfies the communication constraint.

For each block, we then form the side information by taking the average of the quantized outputs from all its associated clients. Denote by $Y_t$ the side information formed at the server by using the clients in $C_1$ at iteration $t$. Then, from the description of our scheme, for all coordinates $i \in \{r_1(j - 1) + 1, \ldots, r_1j\}$ and for all $j \in [d/r_1]$ we have

$$Y_t(i) = \frac{2d}{nr_1} \sum_{k \in S_j} Q_a(\hat{g}_k(x_t)(i)),$$

where $S_j$ denotes the set of $\frac{n/2}{r_1}$ clients assigned to form the side information for the coordinates $\{r_1(j - 1) + 1, \ldots, r_1j\}$, i.e.,

$$S_j = \{C_{1,n/2r_1}(j-1)+1, \ldots, C_{1,n/2r_1}(j)\} \quad (28)$$

We remark that to decode each quantized gradient estimate sent by clients in $C_2$, we will use $Y_t$ as side information. However, $Y_t$ will not be used as is but a version which is rotated\footnote{For decoding each quantized gradient sent by clients in $C_2$, $Y_t$ will be rotated using independent and identical versions of matrix $R$.} using a random matrix (8) will be used.\footnotetext{\hspace{1cm} 2) The Wyner-Ziv Gradient Estimate $Q_{WZ}$: We use the clients in $C_2$ to form the actual gradient estimate. The clients encode the stochastic gradients using a subsampled RMQ quantizer (see Section III-C for details). Therefore, for stochastic gradient $\hat{g}_j(x_t)$, the output encoded by client $C_1$ using subsampled RMQ is described as follows:

$$Q^e_{WZ,j}(\hat{g}_j(x_t)) = \{Q^e_{WZ}(R_j \hat{g}_j(x_t)(i)) : i \in D_j\}.$$

At the server, the communication for all $C_1 \subset C_2$ is decoded as follows:

$$Q_{WZ,j}(\hat{g}_j(x_t), Y_t) = R_j^{-1} \left( \frac{d}{r_2} \sum_{i \in D_j} (\hat{g}_j - R_j Y_t(i)) e_i + R_j Y_t \right)$$

where $r_2 = r / \log r_2$, $\hat{g}_j(i) = Q_{\theta}(R_j \hat{g}_j(x_t)(i), R_j Y_t(i))$, and the modulo quantizer $Q_{\theta}$ uses $\ell_2$ bits of precision. Finally, the server averages over all the quantized gradient estimates of clients in $C_2$ to get (see, line 17 in Algorithm 12)

$$\mathcal{M}(Q_{WZ,n/2+1}, \ldots, Q_{WZ,n}) = \frac{2}{n} \sum_{j=n/2+1}^{n} Q_{WZ,j}(\hat{g}_j(x_t), Y_t) \quad (29)$$}
Next, we present the convergence rate of the proposed $\hat{WZ}$-SGD algorithm for communication constrained distributed optimization.

**Theorem 8:** Let $\mathcal{S}$ be the communication protocol which uses the CUQ quantizer for clients $\mathcal{C}_1$ and the subsampled RMQ quantizer for clients in $\mathcal{C}_2$. Let $\mathcal{A}$ be the optimization algorithm described in Algorithm 12 with the learning rate

$$\eta_t = \frac{1}{L + \alpha'(M)}$$

where $\alpha'(M) = c_0 e^\frac{\alpha}{D} \cdot \frac{1}{nT}$

for some positive universal constant $c_0$. Then, for positive universal constants $c_1, c_2$, and $c_3$ and $r, n$ such that $d \geq r \geq c_1 \max \{\log \log(\frac{B}{\sigma} nT), \log(E/\sigma^2)\}$ and $nr \geq c_2 d^2 (E/\sigma)$, we have

$$\mathcal{E}(f, \mathcal{C}, \mathcal{A}, \mathcal{S}) \leq c_3 D\sqrt{nT} \cdot \frac{d\log \log \left(\frac{nT}{r}\right)}{r} + \frac{LD^2}{2T}.$$  

**Remark 6:** The condition on $nr$ is needed to remove any $B$ dependence from the MSE upper bound. Thus, in the setting where the number of clients $n$ is large, we match the lower bound in Theorem 6 up to a $\log \log nT$ factor.

**Algorithm 13 $\hat{WZ}$-SGD Algorithm**

1. for $\text{Clients } i \in [n] \text{ do} \quad \triangleright \text{Setting quantizers}$
2.  
3.  
4.  
5.  
6.  
7.  
8.  
9.  
10.  
11.  
12.  
13.  
14.  
15.  
16.  
17.  
18.  

The other assumptions (21) and (23) about the estimated gradients still hold.\(^\text{18}\) We show how the dependence of $B$ in the naive scheme, presented in Theorem 7, can be reduced using subsampled RDAQ.

At every iteration, the client indexed by $\mathcal{C}_1$ uses subsampled RATQ to compress their gradient estimates. The side information is then formed by taking sample average of the decoded estimates, similar to (28) (see line 14, in Algorithm 13).

On the other hand, the clients in $\mathcal{C}_2$ use the subsampled RDAQ quantizer $Q_{\hat{WZ}, a}$ from section IV-D. Note that the subsampled RDAQ decoder (14) uses the side information constructed by $\mathcal{C}_1$. Finally, the server takes the sample average of the decoded values estimated by $\mathcal{C}_2$ (see, line 17 in Algorithm 13) to form the mapping $\mathcal{M}$.

**Theorem 9:** Let $\mathcal{S}$ be the communication protocol which uses the subsampled RATQ quantizer for clients $\mathcal{C}_1$ and the subsampled RDAQ quantizer for clients in $\mathcal{C}_2$. Let $\mathcal{A}$ be the optimization algorithm described in Algorithm 13 with the learning rate

$$\eta_t = \frac{1}{L + \alpha'(M)\sqrt{nT}}$$

with $\alpha'(M) = \sqrt{\frac{2\sigma^2}{nT}} + \frac{2\sigma^2}{nT}$ and $h = 1 + \log^2(d/6)$. Further, suppose that the gradient estimated by all the clients satisfy the assumptions (21), (30) and (23). Then, for $d \geq r \geq \max \{h + \log h, 3 + \log(1 + \log h(d/3))\}$, we have

$$\mathcal{E}(f, \mathcal{C}, \mathcal{A}, \mathcal{S}) \leq \frac{2D}{\sqrt{nT}} \left(\frac{2\sigma^2}{nT} + \rho(B, \sigma, r, n) + \frac{LD^2}{2T}\right).$$

**Remark 7:** We remark that under the relaxed assumption of mean-square bounded deviation in (30), for $nr \geq (B^2/\sigma^2) d \log(1 + \log^4(d/3))$, the slowdown in the convergence rate is illustrated by $\rho \approx \frac{16\sqrt{3}d}{nT}$, and the universal scheme surpasses the performance of parallel SGD presented in Section A-C.

We end this section by pointing out the limitations of a natural scheme for distributed optimization.

**Remark 8 (Limitations of Centering Based Scheme):** We note that our framework allows for quantization schemes which were previously quantized gradients are used for gradient compression at the current iteration. For instance, we can use average of the compressed gradients at the previous iteration to center the current compression. That is, the server broadcasts the average to all the clients and the clients only need to compress the difference between the current stochastic gradient and this communicated average.

If the query points $x_{t-1}$ and $x_t$ do not deviate by much, then such type of compression schemes which are centered around the average of previous quantized gradients may turn out to be very efficient. Also, note that the typical learning rate for smooth optimization is $O(\sqrt{T/r})$, which means that the difference between the points $x_t$ and $x_{t-1}$ is not very large. Moreover, the smoothness assumption (20) allows to control the deviation between the true gradients at successive iterations in terms of the points queried at the two iterations. All this

\(^{18}\)Note that the lower bound in Theorem 6 under the almost sure assumption (22) holds for the relaxed mean-squared assumption (30) too.

---

**E. $\hat{WZ}$-SGD: A Universal Wyner-Ziv Algorithm for Distributed Optimization**

We now relax the almost sure (22) assumption on the gradients estimated by clients and present an *universal* algorithm $\hat{WZ}$-SGD, where the compression at the clients doesn’t need the learning rate of $\sigma$ and only the server needs to learn the rate to which $\sigma$ for Algorithm 11. Specifically, we assume that for all clients $i \in [n]$,

$$\mathbb{E}[\|\hat{g}_i(x) - \nabla f(x)\|^2] \leq \sigma^2. \quad \text{(m.s. deviation bound)} \quad (30)$$
hints at the fact that such a scheme where each client uses optimal quantizers for quantizing the difference vector without any side-information may turn out to be optimal. But note that for a very large value of smoothness constant, \( L \geq \sigma \sqrt{T}/D \), even with small deviation between successive query points, the deviation between the gradients will be large. This would in turn lead to variance of the quantized gradients having a dependence on the maximum gradient norm \( \sqrt{\|B^2 - \sigma^2\|} \), which would in turn lead to the leading term, in terms of \( n \) and \( T \), in convergence rate depending on \( \sqrt{B^2 - \sigma^2} \).

**APPENDIX B**

**APPLICATION 2: THE GAUSSIAN WYNER-ZIV PROBLEM**

Consider the random vectors \( X, Y \), where the coordinates \( \{X(i), Y(i)\}_{i=1}^{d} \) form an i.i.d. sequence. Furthermore, for all \( i \in [d] \), let

\[ X(i) = Y(i) + Z(i), \]

where \( Y(i) \) and \( Z(i) \) are independent and zero-mean Gaussian random variables with variances \( \sigma^2_Y \) and \( \sigma^2_Z \), respectively. The encoder has access to the sequence \( X = \{X(i)\}_{i=1}^{d} \), which it quantizes and sends to the decoder. The decoder, on the other hand, has access to \( Y \) (note that encoder does not have access to \( Y \)) and can use it to decode \( X \). A pair \((R, D)\) of non-negative numbers is an achievable rate-distortion pair if we can find a quantizer \( Q_d \) of precision \( dR \) and with mean square error \( E[\|Q_d(X, Y) - X\|^2] \leq dD \). For \( D \geq 0 \), denote by \( R(D) \) the infimum over all \( R \) such that \((R, D)\) constitute an achievable rate-distortion pair for all \( d \) sufficiently large. From\(^{19}\) [62], \( R(D) \) can be characterized as follow:

\[
R(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma^2_Y}{D} & \text{if } D \leq \sigma^2_Y \\ 0 & \text{if } D > \sigma^2_Y \end{cases}
\]

Several constructions that involve computational heavy methods such as error correcting codes and lattice encoding attain the rate-distortion function, asymptotically for large \( d \). In this section, we show that modulo quantizer with parameters set appropriately attains a rate very close to the rate-distortion function \( R(D) \). Moreover, we show that this rate can be achieved for arbitrary \( Y \) and \( Z \), as long as \( Z \) is a zero mean subgaussian random variable with variance factor \( \sigma^2_Z \). Our proposed quantizer \( Q_d(X, Y) \) uses the modulo quantizer to quantize \( X(i) \) with side information \( Y(i) \) at the decoder and the parameter \( k, \Delta \) set as follows:

\[
\delta = \sqrt{D/308}, \quad \log k = \left\lceil \log \left( 2 + \sqrt{\frac{24\sigma^2_Y}{D} \ln(\frac{308}{\delta})} \right) \right\rceil \]

\[
\Delta' = \sqrt{\delta \sigma^2_Z / \delta}, \quad \varepsilon = 2\Delta'/(k - 2),
\]

\[
\Delta = \delta \sigma^2_Y
\]

**Theorem 10:** Consider random vectors \( X, Y \) in \( \mathbb{R}^d \) with \( X(i) = Y(i) + Z(i) \) and \( Z(i) \) independent of \( Y(i) \) being a centered subgaussian random variable with variance factor of \( \sigma^2_z \), for all coordinates \( i \in \{1, \ldots, d\} \). Then, for \( D \leq (\sigma^2_Z/308) \), the quantizer \( Q_d(X, Y) \) described above has MSE less than \( dD \) and has rate \( R \) satisfying

\[
R \leq \frac{1}{2} \log \frac{\sigma^2_Y}{D} + O \left( \log \log \frac{\sigma^2_Y}{D} \right).
\]

**APPENDIX C**

**PROOFS**

**A. Proof of Lemma 1**

For the estimator \( \hat{x} \) in (4), with \( \hat{x} = Q_1(x_i, y_i) \), we have

\[
\mathbb{E} \left[ \left\| \frac{1}{n} \sum_{i \in [n]} Q_1(x_i, y_i) - \frac{1}{n} \sum_{i \in [n]} x_i \right\|^2 \right]
\]

\[
= \frac{1}{n^2} \sum_{i \in [n]} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\|^2 \right]
\]

\[
+ \frac{1}{n^2} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i, Q_1(x_j, y_j) - x_j\| \right]
\]

\[
= \frac{1}{n^2} \sum_{i \in [n]} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\|^2 \right]
\]

\[
+ \frac{1}{n^2} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i, Q_1(x_j, y_j) - x_j\| \right]
\]

\[
= \frac{1}{n^2} \sum_{i \in [n]} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\|^2 \right]
\]

\[
+ \left( \frac{1}{n} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\| \right] \right)^2
\]

\[
- \frac{1}{n^2} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\| \right] \cdot \left( \frac{1}{n} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\| \right] \right)
\]

\[
= \frac{1}{n^2} \sum_{i \in [n]} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\|^2 \right]
\]

\[
+ \frac{(n - 1)}{n^2} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\| \right] \cdot \left( \frac{1}{n} \sum_{i \neq j} \mathbb{E} \left[ \|Q_1(x_i, y_i) - x_i\| \right] \right)
\]

where the second identity uses the independence of \( Q_1(x_i, y_i) \) for different \( i \) and the final step uses Jensen’s inequality. The result follows by bound each term using the fact that \( x \) and \( y \) satisfy (2) and the definitions of \( \alpha(Q_1, \Delta_i) \) and \( \beta(Q_1, \Delta_i) \), for \( i \in [n] \).

**B. Proof of Lemma 2**

As mentioned in (5), the integer \( \tilde{z} \) found in Alg. 1 satisfies \( \mathbb{E} [\tilde{z}x] = x \) and \( |x - \tilde{z}x| < \varepsilon \). Therefore, it suffices to show that the output of the quantizer satisfies \( Q_{B}(x, y) = \tilde{z}x \).

To see that \( Q_{B}(x, y) = \tilde{z}x \), denote the lattice used in decoding Alg. 2 as \( \mathbb{Z}_{w, z} := \{kz + w \in \mathbb{Z} \} \). The decoding algorithm finds the point in \( \mathbb{Z}_{w, z} \) that is closest to \( y \). Note that \( w = \tilde{z} \mod k \), whereby \( \tilde{z}x \) is a point in this lattice. Further, for any other point \( \lambda \neq \tilde{z}x \) in the lattice, we must have

\[
|\lambda - \tilde{z}x| \geq k\varepsilon,
\]
and so, by triangular inequality, that
\[ |\lambda - y| \geq |\lambda - \tilde{z}e| - |\tilde{z}e - y| \geq k\varepsilon - |\tilde{z}e - y|. \]
Thus, \( \tilde{z}e \) is closer to \( y \) than \( \lambda \) if
\[ k\varepsilon > 2|\tilde{z}e - y|. \]

Next, by using (5) once again, we have
\[ |\tilde{z}e - y| \leq |\tilde{z}e - x| + |x - y| < \varepsilon + \Delta', \]
which by condition (7) in the lemma implies that (32) holds. It follows that \( |\lambda - y| > |\tilde{z}e - y| \) for every \( \lambda \in \mathbb{Z}^{w,\varepsilon} \), which shows that \( Q_R(x, y) = \tilde{z}e \) and completes the proof. \( \square \)

C. Proof of Lemma 3

Recall from Remark 1 that for the random matrix \( R \) given in (8), for every vector \( z \in \mathbb{R}^d \), the random variables \( Rz(i), i \in [d] \), are sub-Gaussian with variance parameter \( \|z\|^2/d \). Furthermore, we need the following bound for “truncated moments” of sub-Gaussian random variables. To prove this bound, we will use the following elementary fact.

Fact 1: For any nonnegative random variable \( U \) and \( x \geq 0 \),
\[ \mathbb{E}[U \mathbb{I}_{\{U > x\}}] = x\mathbb{P}(U > x) + \int_x^\infty \mathbb{P}(U > u)\,du. \]

Proof: To see the fact above, note that for a nonnegative random variable \( W [17, \text{Lemma 2.2.13}] \), we have
\[ \mathbb{E}[W] = \int_0^\infty \mathbb{P}(W > u)\,du. \]

Substituting \( W = U \mathbb{I}_{\{U > x\}} \), we get
\[ \mathbb{E}[U \mathbb{I}_{\{U > x\}}] = \int_0^x \mathbb{P}(U > u)\,du + \int_x^\infty \mathbb{P}(U > u)\,du. \]

Note that
\[ \mathbb{P}(U \mathbb{I}_{\{U > x\}} > u) = \begin{cases} \mathbb{P}(U > x) & x \geq u \\ \mathbb{P}(U > u) & x < u \end{cases}. \]

Substituting this expression for probability in the integral above, we get
\[ \mathbb{E}[U \mathbb{I}_{\{U > x\}}] = \int_0^x \mathbb{P}(U > x)\,du + \int_x^\infty \mathbb{P}(U > u)\,du = x\mathbb{P}(U > x) + \int_x^\infty \mathbb{P}(U > u)\,du. \]

Lemma 10: For a sub-Gaussian random \( Z \) with variance factor \( \sigma^2 \) and every \( t \geq 0 \), we have
\[ \mathbb{E}[Z^2 \mathbb{I}_{\{|Z| > t\}}] \leq 2(2t^2 + t^2)e^{-t^2/2\sigma^2}. \]

Proof: Upon substituting \( U = Z^2 \) and \( x = t^2 \) in Fact 1, along with the fact that \( Z \) is sub-Gaussian with variance parameter \( \sigma^2 \), we get
\[ \mathbb{E}[Z^2 \mathbb{I}_{\{Z^2 > t^2\}}] = t^2\mathbb{P}(Z^2 > t^2) + \int_{t^2}^\infty \mathbb{P}(Z^2 > u)\,du \leq 2t^2e^{-t^2/2\sigma^2} + 2\int_{t^2}^\infty e^{-u/2\sigma^2}\,du \leq 2(t^2 + 2\sigma^2)e^{-t^2/2\sigma^2}, \]
which completes the proof.

We now handle the MSE \( \alpha(Q) \) and bias \( \beta(Q) \) separately.

1) Bound for MSE \( \alpha(Q) \): Denote by \( Q_R(x, y) \) the final quantized value of the quantizer RMQ. For convenience, we abbreviate
\[ \hat{x}_R := R Q_M, R(x, y). \]

Observe that \( \hat{x}_R = \sum_{i \in [d]} Q_R(Rx(i), Ry(i))e_i \), where \( Q_R \) is the MQ of Alg. 1 and 2 with parameters \( k \geq 1 \) and \( \Delta' \) set as in the statement of the lemma. Since \( R \) is a unitary transform, we have
\[ \mathbb{E}\left[\|Q_R(x, y) - x\|_2^2\right] = \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] = \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2\right] = \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \leq \Delta'\}}\right] + \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{I}_{\{|R(x-y)(i)| > \Delta'\}}\right]. \]

We consider each error term on the right-side above separately. We can view the first term as the error corresponding to MQ, when the input lies in its “acceptance range.” Specifically, under the event \( \{|R(x-y)(i)| \leq \Delta'\} \), we get by Lemma 2 that
\[ |\hat{x}_R(i) - Rx(i)| \leq \varepsilon = \frac{2\Delta'}{k - 2}, \quad \text{almost surely,} \]
whence
\[ \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \leq \Delta'\}}\right] \leq 4d \varepsilon^2 \Delta'^2/2\Delta^2. \]

The second term on the right-side of (33) corresponds to the error due to “overflow” and is handled using concentration bounds for the rotated vectors. Specifically, we get
\[ \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{I}_{\{|R(x-y)(i)| > \Delta'\}}\right] \leq 2k^2 \varepsilon^2 \sum_{i=1}^d P(|R(x-y)(i)| \geq \Delta') \]
\[ \leq 2k^2 \varepsilon^2 \sum_{i=1}^d \left[\mathbb{E}\left[(\hat{x}_R(i) - Ry(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \geq \Delta'\}}\right] + \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \geq \Delta'\}}\right]\right] \]
\[ \leq 2k^2 \varepsilon^2 \sum_{i=1}^d \left[\mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \geq \Delta'\}}\right]\right] + 2d \sum_{i=1}^d \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{I}_{\{|R(x-y)(i)| \geq \Delta'\}}\right] \]
\[ \leq 4dk^2 \varepsilon^2 e^{-d\Delta'^2/2\Delta^2}. \]
which completes the proof.

\[ R \hat{\alpha}_R(x) \]

where the second inequality follows upon noting that from the description decoder of MQ in Alg. 2 that \(|x_R(i) - Rg(i)| \leq \varepsilon k\) almost surely for each \(i \leq |d|\); the third inequality uses the fact that \(R(x-y)(i)\) is sub-Gaussian with variance parameter \(\|x-y\|^2_2/d \leq \Delta^2/2d\); and fourth inequality is by Lemma 10.

Upon combining (33), (34), and (35), and substituting \(\varepsilon \leq 2\Delta/(k-2)\) and \(\Delta^2 = 6(\Delta^2/d) \log \Delta/d\), we obtain

\[
\mathbb{E} [\|Q_{R,H}(x, y) - x\|^2_2] \\
\leq d \varepsilon^2 + 4d\varepsilon^2 e^{-d\Delta^2/22\varepsilon^2} + 4(2\Delta^2 + d\Delta^2) e^{-d\Delta^2/22\varepsilon^2} \tag{36}
\]

where we used \((1 + 3 \ln u)/u \leq 3/e^2/3\) and \((\ln u)/u \leq 1/e\) for every \(u > 0\). We conclude by noting that for \(k \geq 4\),

\[
\left ( \frac{96}{e} \left ( \frac{k}{k-2} \right )^2 + \frac{24}{e^2/3} \right ) \leq 154.
\]

2) **Bias \(\beta(Q):** The calculation for the bias is similar to that we used to bound the second term on the right-side of (33). Using the notation \(\hat{x}_R\) introduced above, we have

\[
\|\mathbb{E} [Q_{R,H}] - x\|_2 \\
= \|\mathbb{E} [R^{-1}(\hat{x}_R - Rx)]\|_2 \\
= \|\mathbb{E} [R^{-1}(\hat{x}_R - Rx)]\|_2 \\
= \|\mathbb{E} [R^{-1} (\hat{x}_R - Rx)]\|_2 \\
= \|\mathbb{E} [\hat{x}_R - Rx]\|_2,
\]

where the second identity holds since \(R\) is a unitary matrix.

Further, since \(Q_{R,H}(x,y)\) is an unbiased estimate of \(x\) when \(|x-y| \leq \Delta^2\) (see Lemma 2), by (34) and (35) we obtain

\[
\|\mathbb{E} [\hat{x}_R - Rx]\|_2 \\
\leq \sum_{i=1}^{d} \mathbb{E} \left [ (\hat{x}_R(i) - Rx(i)) 1_{|R(x-y)(i)| \geq \Delta} \right ]^2 \\
\leq \sum_{i=1}^{d} \mathbb{E} \left [ (\hat{x}_R(i) - Rx(i))^2 1_{|R(x-y)(i)| \geq \Delta} \right ] \\
\leq 154 \delta^2,
\]

which completes the proof.

**D. Proof of Lemma 4**

1) **Mean Square Error \(\alpha(Q_{S,R}):** From the description of Algorithms 5 and 6, we know that the quantized output of subsampled RMQ \(Q_{WZ}\) for an input \(x\) is

\[
Q_{WZ}(x) = R^{-1} \hat{x}_R, \text{ where} \text{ } \hat{x}_R = \frac{1}{\mu} \sum_{i \in [d]} Q_n(Rx(i), Ry(i)) 1_{\{i \in S\}} \epsilon_i + Ry,
\]

and \(Q_n(Rx(i), Ry(i))\) denotes the quantized output of the modulo quantizer for an input \(Rx(i)\) and side-information \(Ry(i)\). Use the shorthand \(Q(Rx(i))\) for \(Q_n(Rx(i), Ry(i))\), we have

\[
\mathbb{E} [\|Q_{WZ}(x) - x\|^2_2] \\
= \sum_{i \in [d]} \mathbb{E} \left [ \left ( \frac{1}{\mu} (Q(Rx(i)) - Ry(i)) 1_{\{i \in S\}} - (Rx(i) - Ry(i)) \right )^2 \right ] \\
\leq 2 \sum_{i \in [d]} \mathbb{E} \left [ \left ( \frac{1}{\mu} (Q(Rx(i)) - Rx(i))^2 1_{\{i \in S\}} \right ] \\
+ 2 \sum_{i \in [d]} \mathbb{E} \left [ \left ( \frac{1}{\mu} (Rx(i) - Ry(i)) 1_{\{i \in S\}} \right ) \\
\leq 2 \sum_{i \in [d]} \mathbb{E} \left [ (Q(Rx(i)) - Rx(i))^2 \right ] \\
+ 2 \sum_{i \in [d]} \mathbb{E} \left [ (Rx(i) - Ry(i))^2 \right ] \cdot \mathbb{E} \left [ \left ( \frac{1}{\mu} 1_{\{i \in S\}} - 1 \right )^2 \right ] \\
= \frac{2\alpha(Q_{S,H})}{\mu} + \frac{2\Delta^2}{\mu},
\]

where we used the fact that \(R\) is a unitary matrix in the first identity, the inequality: \((a+b)^2 \leq 2(a^2+b^2)\) in the first inequality, the independence of \(S\) and \(R\) in the second identity, and once gain used the fact that \(R\) is unitary in the final step.

2) **Bias \(\beta(Q_{S,R}):** This follows upon noting that the conditional expectation (over \(S\)) of the output of subsampled RMQ given \(R\) is the vector \(R^{-1} \sum_{i \in [d]} Q_n(Rx(i), Ry(i)) \epsilon_i\), which, in turn, is equivalent in distribution to the output of RMQ. \(\square\)

**E. Proof of Theorem 2**

We denote \(\Delta_{\min} = \min_{i \in [d]} \Delta_i\) and set \(y_i\)s to be 0. Let \(x_1, \ldots, x_n\) be an iid sequence with common distribution such
that for all $j \in [d]$ we have

$$x_1(j) = \begin{cases} \frac{\Delta_{\min}}{\sqrt{d}} & \text{w.p. } \frac{1+\alpha(j)\delta}{2} \\ -\frac{\Delta_{\min}}{\sqrt{d}} & \text{w.p. } \frac{1-\alpha(j)\delta}{2}, \end{cases}$$

where $\alpha \in \{-1, 1\}^d$ is generated uniformly at random. We have the following Lemma for such $x_1$'s, which provides a lower bound for the MSE of any estimator of the mean of the distribution of $x_1$.

**Lemma 11:** For $x_1, \ldots, x_n$ generated as above and any estimator $\hat{x}$ of the mean formed using only r-bit quantized version of $x_1$, we have

$$\mathbb{E}\left[\left\|\hat{x} - \frac{\delta \Delta_{\min}}{\sqrt{d}} \alpha\right\|^2\right] \geq c' \cdot \frac{d \Delta_{\min}^2}{2nr},$$

where $c' < 1$ is a universal constant.

Proof of Lemma 11 follows from either [16, Proposition 2] or [2, Theorem 11].

The proof of Theorem 2 is completed by using this claim. Specifically, using $2a^2 + 2b^2 \geq (a+b)^2$, we have

$$2\mathbb{E}\left[\left\|\hat{x} - \bar{x}\right\|^2\right] + 2\mathbb{E}\left[\left\|\bar{x} - \frac{\delta \Delta_{\min}}{\sqrt{d}} \alpha\right\|^2\right] \geq \mathbb{E}\left[\left\|\hat{x} - \frac{\delta \Delta_{\min}}{\sqrt{d}} \alpha\right\|^2\right],$$

which, along with the observation that

$$\mathbb{E}\left[\left\|\bar{x} - \frac{\delta \Delta_{\min}}{\sqrt{d}} \alpha\right\|^2\right] \leq \frac{\Delta_{\min}^2}{n},$$

gives

$$\mathbb{E}\left[\left\|\hat{x} - \bar{x}\right\|^2\right] \geq \frac{c' \delta \Delta_{\min}^2}{2nr} - \frac{\Delta_{\min}^2}{n} \geq \frac{c' \delta \Delta_{\min}^2}{4nr},$$

when $(d/r) \geq 4/c'$. The proof is completed by setting $c = c'/4$. \hfill \Box

**Remark 9:** Since the lower bound in [2] holds for sequentially interactive protocols, if we allow interactive protocols for mean estimation where client $i$ gets to see the messages transmitted by the clients $j$ in $[i-1]$, and can design its quantizers based on these previous messages, even then the lower bound above will hold.

### F. Proof of Lemma 5

We will prove a general result which will not only prove Lemma 5 but will also be useful in the proof of Lemma 6. Consider $x$ and $y$ in $\mathbb{R}^d$ such that each coordinate of both $x$ and $y$ lies in $[-M, M]$. Also, consider the following generalization of DAQ:

$$Q_0(x, y) = \sum_{i=1}^{d} 2M \left(1_{\{U_i \leq x(i)\}} - 1_{\{U_i \leq y(i)\}}\right) e_i + y,$$

where $\{U_i\}_{i \in [d]}$ are iid uniform random variables in $[-M, M]$. We will show that

$$\mathbb{E}\left[Q_0(x, y)\right] = x$$

and

$$\mathbb{E}\left[\left\|Q_0(x, y) - x\right\|^2\right] \leq 2M \left\|x - y\right\|_1,$$

which upon setting $M = 1$ proves Lemma 5.

Towards proving (38), note that from the estimate formed by $Q_0$, it is easy to see that $\mathbb{E}\left[Q_0(x, y)\right] = x$. The MSE can be bounded as follows:

$$\mathbb{E}\left[\left\|Q_0(x, y) - x\right\|^2\right]$$

$$= \sum_{i=1}^{d} \mathbb{E}\left[(2M \left(1_{\{U_i \leq x(i)\}} - 1_{\{U_i \leq y(i)\}}\right) - (x(i) - y(i)))^2\right]$$

$$= \sum_{i=1}^{d} 4M^2 \frac{|x(i) - y(i)|}{2M} - \left\|x - y\right\|^2_2$$

$$= 2M \left\|x - y\right\|_1 - \left\|x - y\right\|^2_2,$$

where we used the observations that $2M \left(1_{\{U_i \leq x(i)\}} - 1_{\{U_i \leq y(i)\}}\right)$ is an unbiased estimate of $(x(i) - y(i))$ and that $(1_{\{U_i \leq x(i)\}} - 1_{\{U_i \leq y(i)\}})^2$ equals one if and only if exactly one of the indicators is one, which in turn happens with probability $\frac{|x(i) - y(i)|}{2M}$. \hfill \Box

### G. Proof of Lemma 6

1) **Worst-Case Bias $\beta(Q_0, R; \Delta)$:** Since the final interval $[-M_{h-1}, M_{h-1}]$ contains $[-1, 1]$, we can see that $\mathbb{E}\left[Q_0(x, y)\right] = x$.

2) **Worst-Case MSE $\alpha(Q_0, R; \Delta)$:** We denote by $B^x_{ij}$ and $B^y_{ij}$ the bits

$$B^x_{ij} = 1_{\{U(i, j) \leq Rx(i)\}} \quad \text{and} \quad B^y_{ij} = 1_{\{U(i, j) \leq Ry(i)\}}.$$

Then, the final quantized value of the quantizer RDAQ can be expressed as

$$Q_0(R; X) = R^{-1} \hat{x}_R \text{ where, with } z^*(i) \text{ denoting the smallest } M_j \text{ such that the interval } [-M_j, M_j] \text{ contains } Rx(i) \text{ and } Ry(i) \text{ and } |h|_0 = \{0, \ldots, h-1\},$$

$$\hat{x}_R := \sum_{i \in [1, \ldots, d]} \left( \sum_{j \in [h]} 2M_j \cdot \left(B^x_{ij} - B^y_{ij}\right) + Ry(i) \right) 1_{\{z^*(i)=j\}} e_i.$$

Since $R$ is a unitary transform, we get

$$\mathbb{E}\left[\left\|Q_0(x) - x\right\|^2\right]$$

$$= \mathbb{E}\left[\left\|RQ_0(x) - Rx\right\|^2\right]$$

$$= \mathbb{E}\left[\left\|\hat{x}_R - Rx\right\|^2\right]$$

$$= \sum_{i \in [d]} \mathbb{E}\left[\left\|\hat{x}_R(i) - Rx(i)\right\|^2\right]$$

$$= \sum_{i \in [d]} \left( \sum_{j \in [h]} 2M_j \cdot \left(B^x_{ij} - B^y_{ij}\right) \right).$$

20Note that the side information $y_i$'s are all set to 0.
where the last identity uses $\mathbb{I}_{\{z^*(i)=j_1\}} \mathbb{I}_{\{z^*(i)=j_2\}} = 0$ for all $j_1 \neq j_2$, to cancel the cross-terms in the expansion of $(\bar{x}(i) - Rx(i))^2$. Conditioning on $R$ and using the independence of $\mathbb{I}_{\{z^*(i)=j\}}$ from the randomness used in MQ, we get

$$
\mathbb{E} \left[ \|Q_{\mathcal{R}}(x) - x\|^2 \right] = \sum_{i \in [d]} \sum_{j \in [h_0]} \mathbb{E} \left[ (2M_j (B_{ij}^x - B_{ij}^y)) + Rg(i) - Rx(i))^2 \mathbb{I}_{\{z^*(i)=j\}} \right],
$$

$$
\leq \sum_{i \in [d]} \sum_{j \in [h_0]} \mathbb{E} \left[ 2M_j |Rx(i) - Rg(i)| \mathbb{I}_{\{z^*(i)=j\}} \right] 
+ \sum_{i \in [d]} \sum_{j \in [h-1]} \mathbb{E} \left[ 2M_j |Rx(i) - Rg(i)| \mathbb{I}_{\{z^*(i)=j\}} \right],
$$

(39)

where the first inequality follows from (38) in the proof of Lemma 5.

Next, noting that $\mathbb{I}_{\{z^*(i)=j\}} \leq \mathbb{I}_{\{\{RX(i) \geq M_j, -1\} \cup \{\{|RY(i)| \geq M_j, -1\}\}}$ almost surely, an application of the Cauchy-Schwarz inequality yields

$$
\mathbb{E} \left[ 2M_j |Rx(i) - Rg(i)| \mathbb{I}_{\{z^*(i)=j\}} \right] 
\leq 2M_j \mathbb{E} \left[ (Rx(i) - Rg(i))^2 \right]^{1/2} 
\cdot \mathbb{E} \left[ (\mathbb{I}_{\{|RX(i)| \geq M_j-1\}} + \mathbb{I}_{\{|RY(i)| \geq M_j-1\}})^2 \right]^{1/2}
\leq 2M_j \mathbb{E} \left[ (Rx(i) - Rg(i))^2 \right]^{1/2}
\cdot (2P(|Rx(i)| \geq M_j-1) + 2P(|Ry(i)| \geq M_j-1))^{1/2}
\leq 2M_j \mathbb{E} \left[ (Rx(i) - Rg(i))^2 \right]^{1/2} \left( 8e^{-dM^2/2} \right)^{1/2},
$$

(40)

where the second inequality uses $(a+b)^2 \leq 2a^2 + 2b^2$ and the third uses subgaussianity of $Rx(i)$ and $Ry(i)$.

Substituting the upper bound in (40) for the second term in the RHS of (39) and using $\mathbb{E} [X] \leq \mathbb{E} [X^2]^{1/2}$ for the first term, we get

$$
\mathbb{E} \left[ \|Q_{\mathcal{R}}(x) - x\|^2 \right] \leq \sum_{i \in [d]} \mathbb{E} \left[ |Rx(i) - Rg(i)| \right]^{1/2} \cdot \left( 2M_0 + \sum_{j \in [h-1]} 2M_j \cdot \left( 8e^{-dM^2/2} \right)^{1/2} \right)
$$

$$
\leq \sqrt{d} \cdot \mathbb{E} \left[ \|Rx - Ry\|^2 \right]^{1/2} \cdot \left( 2M_0 + \sum_{j \in [h-1]} 2M_j \cdot \left( 8e^{-dM^2/2} \right)^{1/2} \right)
$$

$$
= \sqrt{d} \cdot \|x - y\|^2 \cdot \left( 2\sqrt{\frac{2}{d}} + 2\sqrt{\frac{6e^*}{d}} \cdot \left( 8e^{-1.5e^*(j-1)} \right) \right)
$$

$$
= 8\sqrt{3} \cdot \sqrt{\|x - y\|^2} \left( 1 + \sum_{j \in [h-1]} e^{-0.5 \ e^*(j-1)} \right)
$$

$$
\leq 16\sqrt{3} \cdot \sqrt{\|x - y\|^2},
$$

where the second inequality uses the fact that $\sum_j \|a\|_1 \leq \sqrt{d} \|a\|_2$, the first and second identities follow from the fact that $R$ is unitary transform and substituting for $M_j$'s, the final inequality follows from the bound of 1 for $\sum_{j=1}^\infty e^{-0.5 \ e^*(j-1)}$, which, in turn, can seen as follows

$$
e^{-0.5 \ e^*(j-1)} = e^{-0.5} + e^{-0.5e} + e^{-0.5e^2} + \sum_{j=3}^\infty e^{-0.5e^{(j)}}
\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^2} + \sum_{j=3}^\infty e^{-0.5e^{(j)}}
\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^2} + \frac{1}{e^{e^2} - 1}
\leq 1.
$$

Remark 10: (41) sheds light on our choice of $M_j$'s. Specifically, $M_j$'s were chosen to jointly minimize the RHS in (41) and the number of dynamic ranges.

H. Proof of Lemma 7

1) Worst-Case Bias $\beta(Q_{z,u}; \Delta)$: It is straightforward to see that $\mathbb{E} [Q_{z,u}(x)] = x$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
2) Worst-Case MSE $\alpha(Q_{\mathcal{Z},u}; \Delta)$: We denote by $B_{ij}^x$ and $B_{ij}^y$ the bits

$$B_{ij}^x = 1_{\{U(i,j) \leq Rx(i)\}} \quad \text{and} \quad B_{ij}^y = 1_{\{U(i,j) \leq Ry(i)\}}.$$ 

Then, the quantized output can be stated as follows: noting that $Q_{\mathcal{Z},u}(x) = R^{-1} \hat{x}_R$ where, with $z^*(i)$ denoting the smallest $M_j$ such that the interval $[-M_j, M_j]$ contains $Rx(i)$ and $Ry(i)$,

$$\hat{x}_R := \sum_{i \in \{1, \ldots, d\}, j \in \{0, \ldots, h-1\}} 2M_j \cdot (B_{ij}^x - B_{ij}^y) 1_{\{z^*(i) = j\}} 1_{\{i \in S\}} : e_i + Ry,$$

Since $R$ is a unitary transform, the mean square error between $Q_{\mathcal{Z},u}(x)$ and $x$ can be bounded as in the proof of Lemma 6 as follows:

$$E[\|Q_{\mathcal{Z},u}(x) - x\|^2] = E[\|\hat{x}_R - Rx\|^2] = E[\|\hat{x}_R - Rx\|^2] + Ry(i)^2 1_{\{z^*(i) = j\}}$$

$$= \sum_{i \in [d]} \sum_{j \in [h]} E\left[2M_j \cdot (B_{ij}^x - B_{ij}^y) 1_{\{i \in S\}} + Ry(i) - Rx(i))^2 1_{\{z^*(i) = j\}} \right]$$

$$\leq \sum_{i \in [d]} \sum_{j \in [h]} E\left[2M_j \cdot |Rx(i) - Ry(i)| 1_{\{z^*(i) = j\}} \right],$$

where the inequality follows from similar calculations in the proof of Lemma 5. The rest of the analysis proceeds as that in the proof of Lemma 6. 

\[I. \ Proof \ of \ Theorem 6\]

Note that affine functions are 0-smooth and admitted in the class of $L$ smooth functions. We use affine functions as difficult functions and follow the general recipe of [3], which in turn builds on [1, Section 4.5] and [2], to show the lower bounds for convex, Lipschitz optimization under communication constraints. The difficult functions we construct are the same as in many existing lower bounds for convex functions such as [5]. We consider the domain $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \leq D/(2\sqrt{d})\}$, and consider the following class of functions on $\mathcal{X}$: For $v \in \{-1, 1\}^d$, let

$$f_v(x) := \frac{2\sigma^2 \delta^2}{d} \sum_{i=1}^d x(i) - v(i)D \frac{2}{2\sqrt{d}}, \quad \forall x \in \mathcal{X},$$

and $x^*_v$ be its minimizer. Note that the gradient $g_v(x)$ of $f_v$ at $x \in \mathcal{X}$ is equal to $-2\sigma^2 \delta^2 / \sqrt{d}$, i.e., constant $\forall x$. For each $f_v$ in (42), consider a sequence of $n$ clients $C$ that output $d$-dimensional gradient vectors $\{\hat{g}_i(x)\}_{i \in [n]}$, each of whose coordinates takes value $-\sigma / \sqrt{d} \text{ or } \sigma / \sqrt{d}$ independently with probabilities $(1 + 2\delta v(i))/2$ and $(1 - 2\delta v(i))/2$, respectively. The parameter $\delta > 0$ is to be chosen later. Note that the above client construction satisfies the set of assumptions in (21), (22) and (23).

Draw $V \sim \text{Unif}\{-1, 1\}^d$. With respect to the associated random function $f_V$, each client $C_i$ chooses a quantizer $Q_{i,t}$ to generate output $Q_{i,t}(\hat{g}_i(x_i))$. Denote by $Q^{nT} := \{Q_{i,1}, \ldots, Q_{i,n}\}_{i \in [n]}$ the vector of quantized outputs observed at the server. The following lower bound can be established by using results from [3, Lemma 3, 4]:

$$E[f_V(\bar{x}_T) - f_V(x^*_V)] \geq \frac{D\delta}{3} \left[1 - \frac{2}{d} \sum_{j=1}^d I(V(j) \land Q^{nT})\right].$$

(43)

It remains to bound the mutual-information term for which one can use the independence across the clients and derive the following data-processing inequality based on the other techniques from [3]:

$$\sum_{j=1}^d I(V(j) \land Q^{nT}) \leq 29nT\delta^2(d \land r),$$

where $\delta \in (0, 1/6)$. Combining this with (43) and setting $\delta = \sqrt{d/(232(d \land r)nT)}$, we finally get

$$E[f_V(\bar{x}_T) - f_V(x^*_V)] \geq \frac{1}{12\sqrt{58}} \frac{D\sigma}{\sqrt{n}12} \sqrt{\frac{d}{d \land r}},$$

where we need $T \geq d/(6n\sigma)$ in order to enforce $\delta \leq 1/6$. The proof is completed by noting that $E^*(\mathcal{X}, \mathcal{O}_{\mathcal{Z}}, T, Q, r) \geq E[f_V(\bar{x}_T) - f_V(x^*_V)].$

\[J. \ Proof \ of \ Lemma 9\]

Define $x^* = \text{argmin}_{x \in \mathcal{X}} f(x)$. We have that

$$E[f(x_{t+1}) - x^*] = E[f(x_{t+1}) - f(x_t)] + E[f(x_t) - x^*].$$

(44)

By smoothness,

$$E[f(x_{t+1}) - f(x_t)] \leq \nabla f(x_t)^	op E[x_{t+1} - x_t|x_t] + \frac{L}{2} E[\|x_{t+1} - x_t\|^2].$$

$$= -\eta_t \nabla f(x_t)^	op E[\mathcal{M}(C_{t}^n)|x_t] + \frac{L\eta_t^2}{2} E[\|\mathcal{M}(C_{t}^n)\|^2].$$

$$\leq -\eta_t \nabla f(x_t)^	op E[\mathcal{M}(C_{t}^n)|x_t] + \frac{\eta_t}{2} E[\|\mathcal{M}(C_{t}^n)\|^2].$$

$$= -\frac{\eta_t}{2} \|\nabla f(x_t)\|^2 + \frac{\eta_t}{2} E[\|\mathcal{M}(C_{t}^n) - \nabla f(x_t)\|^2].$$

which further using the definition of $\alpha'$ in (24) and the law of total expectation imply

$$E[f(x_{t+1}) - f(x_t)] \leq -\frac{\eta_t}{2} E[\|\nabla f(x_t)\|^2] + \frac{\eta_t}{2} \alpha'^2(\mathcal{M}).$$

(45)
By convexity,
\[
E \left[ f(x_t) - x^* \right] 
\leq E \left[ \nabla f(x_t)^T (x_t - x^*) \right] 
= E \left[ (\nabla f(x_t) - \mathcal{M}(C_{i}^{n}))^T (x_t - x^*) \right] 
+ E \left[ \mathcal{M}(C_{i}^{n})^T (x_t - x^*) \right] 
= E \left[ (\nabla f(x_t) - \mathcal{M}(C_{i}^{n}))^T (x_t - x^*) \right] 
+ \frac{1}{2\eta_t} E \left[ \|\mathcal{M}(C_{i}^{n})\|^2 \right] 
+ E \left[ \|x_t - x^*\|^2 - \|x_t - \eta \mathcal{M}(C_{i}^{n}) - x^*\|^2 \right] 
\leq E \left[ (\nabla f(x_t) - \mathcal{M}(C_{i}^{n}))^T (x_t - x^*) \right] 
+ \frac{1}{2\eta_t} E \left[ \|\mathcal{M}(C_{i}^{n})\|^2 \right] 
+ E \left[ \|x_t - x^*\|^2 - \|\Gamma_{X}(x_t - \eta \mathcal{M}(C_{i}^{n})) - x^*\|^2 \right] 
\leq \beta'(\mathcal{M}) \cdot D + \frac{\eta}{2} \alpha^2(\mathcal{M}) + \frac{\eta}{2} E \left[ \|\nabla f(x_t)\|^2 \right] 
+ \eta B \cdot \beta'(\mathcal{M}) + \frac{1}{2\eta_t} E \left[ \|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 \right], 
\tag{46}
\]
where the second inequality is due to a well-known property of the projection operator \( \Gamma_{X} \) (see, for instance, Lemma 3.1, [13]), third inequality follows from Cauchy-Schwarz inequality and using the definitions in (24) and (25). Plugging (45) and (46) in (44), we have
\[
E \left[ f(x_{t+1}) - x^* \right] 
\leq \beta'(\mathcal{M}) \cdot (D + \eta_t B) + \eta \cdot \alpha^2(\mathcal{M}) + \frac{1}{2\eta_t} E \left[ \|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 \right]. 
\]
Summing from \( t = 0 \) to \( T - 1 \), dividing by \( T \), using the assumption that the domain \( X \) has diameter at most \( D \), and setting \( \eta_t \) as provided, the proof is completed. This general convergence bound will be used in our upper bound proofs below.

K. Proof of Theorem 7

From [44, Theorem 3.7], we use the following result.

Lemma 12: Let \( Q_{\text{RATQ}} \) be the subsampled version of RATQ using \( r \geq 3 + \lceil \log(1 + \ln^2(d/3)) \rceil \) bits. Then for \( Y \) such that \( \|Y\|_2 \leq B^2 \), we have
\[
E \left[ Q_{\text{RATQ}}(Y) | Y \right] = Y 
\text{ and } 
E \left[ \|Q_{\text{RATQ}}(Y) - Y\|^2 \right] \leq \frac{dB^2}{3 + \log(1 + \ln^2(d/3))} - 1.
\]
Further, for \( t \in [T] \), we have \( \mathcal{M}(C_{i}^{n}) = \frac{1}{n} \sum_{i \in [n]} Q_{\text{RATQ}}(y_i(x_t)) \) as in (26). Thus we have,
\[
\alpha^2(\mathcal{M}) \leq E \left[ \left\| \frac{1}{n} \sum_{i \in [n]} Q_{\text{RATQ}}(y_i(x_t)) - \frac{1}{n} \sum_{i \in [n]} y_i(x_t) \right\|^2 \right] 
+ E \left[ \left\| \frac{1}{n} \sum_{i \in [n]} y_i(x_t) - \nabla f(x_t) \right\|^2 \right].
\]
Since \( \mathcal{M}(C_{i}^{n}) \) is an unbiased estimate, \( \beta'(\mathcal{M}) = 0 \). The proof is completed by bounding the two terms in the right-side above followed by using Lemma 9, which we do as follows. From Lemma 12, it follows that for \( t \in [T] \), we have
\[
E \left[ \left\| \frac{1}{n} \sum_{i \in [n]} Q_{\text{RATQ}}(y_i(x_t)) - \frac{1}{n} \sum_{i \in [n]} y_i(x_t) \right\|^2 \right] 
\leq \frac{dB^2}{n (3 + \log(1 + \ln^2(d/3)))}.
\]
From (30), we have
\[
E \left[ \left\| \frac{1}{n} \sum_{i \in [n]} y_i(x_t) - \nabla f(x_t) \right\|^2 \right] \leq \frac{\sigma^2}{n}.
\]

L. Proof of Theorem 8

1) Subgaussian and Subexponential Norms: For our analysis, it will be convenient to recall the definition of subgaussian \(^21\) and subexponential norms of a random variable.

Definition 1 [57]: A subgaussian norm of a subgaussian random variable \( X \), denoted \( \|X\|_{\psi_2} \), is defined as \( \|X\|_{\psi_2} := \inf \{ t > 0 : E [ e^{X^2/2} ] \leq 2 \} \). It follows that for a centered subgaussian random variable \( X \), \( \Pr(|X| \geq t) \leq 2 e^{-\frac{t^2}{12\|X\|_{\psi_2}}} \).

Definition 2 [57, Def. 2.7.51]: A subexponential norm of a subexponential random variable \( X \), denoted \( \|X\|_{\psi_1} \), is defined as \( \|X\|_{\psi_1} := \inf \{ t > 0 : \mathbb{E} [ e^{\frac{|X|^2}{2}} ] \leq 2 \} \). It follows that for a centered subexponential random variable \( X \), \( \Pr(|X| \geq t) \leq 2 e^{-\frac{t^2}{2\|X\|_{\psi_1}}} \).

2) Side Information Is Close to Gradient Estimates: We begin by noting that side-information \( Y \) \(^22\) is close to the stochastic gradient estimates computed by clients in \( C_2 \). Specifically, setting the parameters as \( \log \ell_1 = \lceil \log \frac{2B}{\sigma} + 1 \rceil \) and \( r_1 = \ell / \lceil \log \frac{2B}{\sigma} + 1 \rceil \) for clients in \( C_1 \), we get the following.

Lemma 13: For all \( x \in \mathbb{R}^d \), \( j \in C_2 \), \( i \in [d] \), and a universal constant \( c_3 > 0 \), we have
\[
\Pr(|R y_j(x)(i) - R Y(i)| \geq t) 
\leq 2 e^{-c_3 \min(\frac{t^2}{\ell^2}, \frac{t^2}{r_1^2})} + 2 e^{-c_3 \frac{t^2}{2r_1^2}},
\]
where \( R \) is a random Hadamard matrix (8) and for another universal constant \( c_4 > 0 \),
\[
\sigma^2 = c_4 8 \sigma^2 \frac{\log(2B/\sigma + 1)}{nT} \tag{47}
\]

Remark 11: In the analysis for RMQ presented in Section III-B, the difference between the coordinates of the rotated input and rotated side information had subgaussian tails. However, note that in Lemma 13, we can only prove a slightly weaker concentration result.

Towards proving Lemma 13, we begin by showing the following result which holds from the subgaussian properties of
\(^21\) \( \cdot \|X\|_{\psi_2} \) is indeed a norm.
\(^22\)For convenience, we drop the iteration subscript \( t \) in this Section.
uniform quantizer error and standard properties of subgaussian random variables.

**Lemma 14:** For all $x \in \mathbb{R}^d$ and $i \in [d]$ we have
\[ \|Y(i) - \nabla f(x)(i)\|_{\psi_2}^2 \leq \sigma^2. \]

**Proof:** We will prove the theorem for $Y(1)$ since the argument remains the same for all $Y(i)$'s. From the description of CUQ, we note that $Q_u(\hat{g}_j(x)(1))$ satisfies
\[ \|Q_u(\hat{g}_j(x)(1)) - \hat{g}_j(x)(1)\|_{\psi_2}^2 \leq \frac{4c_4B^2}{(\ell_1 - 1)^2}, \quad \forall j \in S_1, \]
for some universal constant $c_4 > 0$. Also, from (22), we have $\|\hat{g}_j(x)(1) - \nabla f(x)(1)\|_{\psi_2}^2 \leq c_4\sigma^2$ for the same constant $c_4$ above. Further, using the triangle inequality and the fact that $(a + b)^2 \leq 2a^2 + 2b^2$,
\[ \|Q_u(\hat{g}_j(x)(1)) - \nabla f(x)(i)\|_{\psi_2}^2 \leq c_4^2B^2 \left(\frac{1}{\ell_1 - 1}\right)^2 + 2c_4\sigma^2. \]

The proof is completed upon noting that the average of $N$ iid zero mean subgaussian random variables $\{X_i\}_{i \in [N]}$ has a subgaussian norm square equal to $\|X_i\|_{\psi_2}^2/N$ and the fact that we use $N = nr_1/(2d)$ samples to form $Y(1)$. \qed

**Remark 12:** In order to quantize a $d$-dimensional gradient to $r \leq d$ bits, the technique of uniform sampling has been used in recent papers on distributed optimization (cf. [44], [56]). However, notice that these works merely required the quantized gradient estimate to be close to the true gradient in mean square sense. In our case, in order to leverage our Wyner-Ziv compression algorithms, we need side-information to be close to the true gradient in much stronger sense. Therefore, we refrain from using uniform sampling and instead use the clients to quantize separate, smaller blocks of coordinates.

3) **Rotation of Side-Information Is Close to the Rotation of True Gradient:** Using standard properties of subgaussian random variables (see [57, Lemma 2.7.7 and Theorem 2.8.1]), we can show the following.

**Lemma 15:** For all $x \in \mathbb{R}^d$ and $i \in [d]$ we have for a universal constant $c_5 > 0$
\[ \Pr(|RY(i) - R\nabla f(x)(i)| \geq t) \leq 2e^{-c_5 \min(t/\sigma^2, t/\sqrt{a}/\sigma')} \cdot \frac{t}{\|v_1\|_{\psi_1}}. \]

**Proof:** The proof follows from combining two facts. First, note that for a sequence $\{X_i\}_{i \in [N]}$ of zero mean, iid subexponential random variables, we have from [57, Theorem 2.8.1]
\[ \Pr \left( \sum_{i=1}^{N} X_i \geq t \right) \leq 2e^{-c_5 \min(t/\sigma^2, t/\sqrt{a}/\sigma')} \cdot \frac{t}{\|v_1\|_{\psi_1}}, \]
for some universal constant $c_5 > 0$.

Also, note that the product of two subgaussian random variables $X$ and $Y$ is subexponential random with subexponential norm bounded as follows (see [57, Lemma 2.7.7]): $\|XY\|_{\psi_2} \leq \|X\|_{\psi_2}\|Y\|_{\psi_2}$. Notice that $|RY(i) - R\nabla f(x)(i)| = \sum_{i \in [d]} U(i)(i) V(i)$, where $U(i)$, $V(i)$ are zero mean, iid, subgaussian random variables with subgaussian norms as $1/\sqrt{\ell}$ and $\sigma'$ (cf. Lemma 14).

Finally, proceeding in the same manner as in [44, Lemma 5.8], we can show that the coordinates of the rotated stochastic gradient are close to the coordinates of the rotated true gradient.

**Lemma 16:** For all $x \in \mathbb{R}^d$ and the universal constant $c_4 > 0$ as in Lemma 14, we have
\[ \|R\hat{g}_j(x)(i) - R\nabla f(x)(i)\|_{\psi_2}^2 \leq c_4\sigma^2/d. \]

Thus, random rotation allows us to convert the $\ell_2$ norm bound in assumption (22) to a $\ell_\infty$ bound. We now choose $c_4 = \min(c_4, c_5)$. Using the inequality: $\max(a, b) \leq a + b$ and the property of subgaussian random variable in Definition 1, Lemma 13 follows from combining Lemmas 15 and 16. Next, we present a lemma similar to Lemma 10 towards evaluating bounds on $\alpha'({\mathcal{M}})$ and $\beta'({\mathcal{M}}).

**Lemma 17:** For a random variable $Z$ such that
\[ \Pr(|Z| \geq t) \leq 2e^{-\frac{c_3t^2}{\sigma^2} + 2e^{-\frac{c_3t\sqrt{\lambda}}{\sigma^2}} + 2e^{-\frac{c_3t^2}{\sigma^2}}}, \]
where $c_3 > 0$ is some universal constant, we have
\[ \mathbb{E}[Z^21_{|Z| > t}] \leq 2\left(\frac{\sigma^2}{c_3} + t^2\right) e^{-\frac{c_3t^2}{\sigma^2}} + 2\left(\frac{\sigma^2}{c_3} + t^2\right) e^{-\frac{c_3t\sqrt{\lambda}}{\sigma^2}} + 2\left(\frac{2\sigma^2}{c_3\sqrt{\lambda}} + 2\sigma t\right) e^{-\frac{c_3t\sqrt{\lambda}}{\sigma^2}}. \]

**Proof:** For any nonnegative random variable $U$, it can be seen
\[ \mathbb{E}[U1_{U > x}] = x \Pr(U > x) + \int_x^\infty \Pr(U > u) \, du. \]

Upon substituting $U = Z^2$ and $x = t^2$, along with the fact that $Z$ has the tail behaviour described above, we get
\[ \mathbb{E}[Z^21_{Z^2 > t^2}] = t^2 \Pr(Z^2 > t^2) + \int_{t^2}^\infty \Pr(Z^2 > u) \, du \leq 2t^2 \left(e^{-\frac{c_3u^2}{\sigma^2}} + e^{-\frac{c_3u\sqrt{\lambda}}{\sigma^2}} + e^{-\frac{c_3u^2}{\sigma^2}}\right) + 2\int_{t^2}^\infty e^{-\frac{c_3u}{\sigma^2}} \, du 
+ 2\int_{t^2}^\infty e^{-\frac{2c_3u}{\sigma^2}} \, du + 2\int_{t^2}^\infty e^{-\frac{c_3u\sqrt{\lambda}}{\sigma^2}} \, du \leq 2\left(\frac{\sigma^2}{c_3} + t^2\right) e^{-\frac{c_3t^2}{\sigma^2}} + 2\left(\frac{\sigma^2}{c_3} + t^2\right) e^{-\frac{c_3t\sqrt{\lambda}}{\sigma^2}} + 2\left(\frac{2\sigma^2}{c_3\sqrt{\lambda}} + 2\sigma t\right) e^{-\frac{c_3t\sqrt{\lambda}}{\sigma^2}}. \]

4) **Bounds on $\alpha'(\mathcal{M})$ and $\beta'(\mathcal{M})$:** Recall that $Q_{\mathcal{R}k,j}$ denotes the rotated modulo quantizer without any subsampling for client $j \in C_2$. From the description of RMQ in Algorithm 4, we have
\[ Q_{\mathcal{K},R_j,j}(\hat{g}_j(x), Y) = R_j^{-1}\left(\sum_{i \in [d]} Q_k(R_j\hat{g}_j(x)(i), R_jY(i)) \cdot e_i\right). \]
The key step of the proof is bounding MSE and bias of RMQ. Towards that, we have the following lemma.

Remark 13: The calculation for MSE and bias are different in the proof of Lemma 13 compared to those in Proof of Lemma 3. This is because of the weaker concentration results available in this case.

Lemma 18: Under the condition that \( nr \geq c_4 d^2 \log (B/\sigma) \), we have for all \( x \in \mathbb{R}^d, j \in C_2 \), and for some parameter \( \delta \in (0, \sqrt{d}/\ell_2) \) that

\[
\mathbb{E} \left[ \| Q_{n,R,j}(\hat{g}_j(x), Y) - \hat{g}_j(x) \|_2^2 \right] 
\leq 2802 \left( \frac{36 \sigma^2 c_3 (\ell_2 - 2)^2}{3 \delta^2} + 237 \delta^2, \right)
\leq 2 \mathbb{E} \left[ \| Q_{n,R,j}(\hat{g}_j(x), Y) - \hat{g}_j(x) \|_2^2 \right] \leq 237 \delta^2,
\]

where \( c_3 \) and \( c_4 \) are the universal constants same as in Lemma 13.

Proof: By considering events \( \{ |R_j(\hat{g}_j(x) - Y)(i)| \leq \Delta' \} \) and \( \{ |R_j(\hat{g}_j(x) - Y)(i)| \geq \Delta' \} \), and then using the facts for modulo quantizer, we have

\[
\mathbb{E} \left[ \| Q_{n,R,j}(\hat{g}_j(x), Y) - \hat{g}_j(x) \|_2^2 \right] 
\leq d \varepsilon^2 + \sum_{i=1}^{d} \mathbb{E} \left[ (Q_{n,R,j}(\hat{g}_j(x), Y) - \hat{g}_j(x))(i)^2 \right] 
\cdot \mathbb{1}_{\{ |R_j(\hat{g}_j(x) - Y)(i)| \geq \Delta' \}} 
\leq d \varepsilon^2 + 2 \varepsilon^2 \sum_{i=1}^{d} \mathbb{Pr}( |R_j(\hat{g}_j(x) - Y)(i)| \geq \Delta' ) 
+ 2 \sum_{i=1}^{d} \mathbb{E} \left[ (R_j(\hat{g}_j(x) - Y)(i))^2 \mathbb{1}_{\{ |R_j(\hat{g}_j(x) - Y)(i)| \geq \Delta' \}} \right] 
\leq d \varepsilon^2 + 2 \varepsilon^2 d \left( e^{-\frac{c_3 \Delta'^2}{\sigma^2}} + e^{-\frac{c_3 \sqrt{\Delta'} \varepsilon^2}{\sigma^2}} + e^{-\frac{c_3 \Delta'^2}{\sigma^2}} \right) 
+ 4 d \left( \frac{\sigma^2}{c_3} + \Delta'^2 \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
+ 4 d \left( \frac{\Delta'^2}{c_3} \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
+ 4 d \left( \frac{2 \sigma^2}{c_3} + \frac{2 \sigma'^2}{c_3 \sqrt{d}} + \Delta'^2 \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
\leq 4 \varepsilon^2 \eta \left( \frac{36 \sigma^2 c_3 (\ell_2 - 2)^2}{3 \delta^2} + 237 \delta^2 \right),
\]

For some parameter \( \delta \in (0, \sqrt{d}/\ell_2) \), we substitute the other parameters as \( \varepsilon = 2 \Delta'/(\ell_2 - 2) \) and \( \Delta'^2 = 9 \Delta^2 (\ln(\sqrt{d}/\delta))^2 \), where \( \Delta^2 = \max \{ \frac{\sigma^2}{c_3}, \frac{\sigma'^2}{c_3}, \frac{\Delta'^2}{\sigma^2} \} \). That gives

\[
\left( \frac{12 \ell_2}{\ell_2 - 2} \right)^2 \Delta^2 \left( \ln \frac{\sqrt{d} \Delta}{\delta} \right)^2 \left( \frac{2}{(\sqrt{d}/\delta)^3} \right) + \frac{1}{(\sqrt{d}/\delta)^4} \right)
\leq \left( \frac{12 \ell_2}{\ell_2 - 2} \right)^2 \Delta^2 \left( \ln \frac{\sqrt{d} \Delta}{\delta} \right)^2 \left( \frac{3}{(\sqrt{d}/\delta)^5} \right)
\]

where the first inequality uses the fact that \( \delta \in (0, \sqrt{d}/\Delta^2) \), the second inequality uses \( \ln x \leq 2 \sqrt{x}/e \), and the final inequality uses the assumption that \( n \geq 8 \). For the last three terms in (48), we have

\[
4 d \left( \frac{\sigma^2}{c_3} + \Delta'^2 \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} + 4 d \left( \frac{\sigma^2}{c_3} + \Delta'^2 \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
+ 4 d \left( \frac{\Delta'^2}{c_3 \sqrt{d}} \right) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
\leq 8 d (\Delta^2 + \Delta'^2) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
+ 4 d (2 \Delta^2 + 2 \Delta \Delta' + \Delta'^2) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
\leq 8 d (\Delta^2 + \Delta'^2) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} + 4 d (3 \Delta^2 + 2 \Delta'^2) e^{-\frac{c_3 \Delta'^2}{\sigma^2}} 
\leq 8 d (\Delta^2 + 9 \Delta^2 (\ln(\sqrt{d}/\delta))^2) \frac{1}{(\sqrt{d}/\delta)^3} 
+ 4 d (3 \Delta^2 + 18 \Delta^2 (\ln(\sqrt{d}/\delta))^2) \frac{1}{(\sqrt{d}/\delta)^3}
\]

For some parameter \( \delta \in (0, \sqrt{d}/\ell_2) \), we substitute the other parameters as \( \varepsilon = 2 \Delta'/(\ell_2 - 2) \) and \( \Delta'^2 = 9 \Delta^2 (\ln(\sqrt{d}/\delta))^2 \), where \( \Delta^2 = \max \{ \frac{\sigma^2}{c_3}, \frac{\sigma'^2}{c_3}, \frac{\Delta'^2}{\sigma^2} \} \). That gives

\[
\left( \frac{12 \ell_2}{\ell_2 - 2} \right)^2 \Delta^2 \left( \ln \frac{\sqrt{d} \Delta}{\delta} \right)^2 \left( \frac{2}{(\sqrt{d}/\delta)^3} \right) + \frac{1}{(\sqrt{d}/\delta)^4} \right)
\leq \left( \frac{12 \ell_2}{\ell_2 - 2} \right)^2 \Delta^2 \left( \ln \frac{\sqrt{d} \Delta}{\delta} \right)^2 \left( \frac{3}{(\sqrt{d}/\delta)^5} \right)
\]

where the first inequality is due to choice of \( \Delta \), the second is AM-GM inequality, the third one uses the fact: \( \delta \in (0, \sqrt{d}/\Delta) \), and the fourth one uses \( \ln x \leq 2 \sqrt{x}/e \). Substituting (49) and (50) in (48), we get

\[
\mathbb{E} \left[ \| Q_{n,R,j}(\hat{g}_j(x), Y) - \hat{g}_j(x) \|_2^2 \right] \leq 36 \Delta^2 \left( \frac{\ell_2 - 2}{\ell_2 - 2} \right)^2 \left( \frac{\sqrt{d} \Delta}{\delta} \right)^2 
+ 237 \delta^2.
\]

Finally, we note that whenever \( nr \geq c_4 d^2 \log (B/\sigma) \), \( \sigma'^2 \leq \sigma^2/d \) (see (47)). With our earlier choice of \( \Delta^2 = \max \{ \frac{\sigma^2}{c_3}, \frac{\sigma'^2}{c_3}, \frac{\Delta'^2}{\sigma^2} \} \), this further implies \( \Delta \leq \frac{\sigma^2}{c_3} \). Using this fact in the bound above establishes the MSE bound.
a) Bound for bias: Using the fact that for \( |R_j(\hat{g}_j(x) - Y(i))| \leq \Delta' \) the modulo quantizer gives an unbiased estimate and the Jensen’s inequality, we have
\[
\|E[Q_{\mathbf{R}}(R_{j}, Y)] - \hat{g}_j\|_2^2
\]
\[
= \sum_{i=1}^{d} E \left[ (Q_{\mathbf{R}}(R_{j}, Y) - \hat{g}_j)^2 \mathbb{I}_{(|R_j(\hat{g}_j - Y(i))| \leq \Delta')} \right]^2
\]
\[
\leq 237\delta^2.
\]
\( \Box \)

b) Completing the proof: We now calculate the MSE and bias for our Wyner-Ziv quantizer with subsampled RMQ. Note that the inequality (37) derived in Lemma 4 holds in this case. Therefore, we have for \( j \in C_2 \) that
\[
E[||Q_{\mathbf{R}}(\hat{g}_j(x), Y) - \hat{g}_j(x)||^2]
\]
\[
\leq \frac{2d}{r_2} E[||Q_{\mathbf{R}}(\hat{g}_j(x), Y) - \hat{g}_j(x)||^2]
\]
\[
+ \frac{2d}{r_2} E[||R\hat{g}_j(x) - RY||^2]
\]
\[
\leq \frac{2d}{r_2} \left( \frac{36\sigma^2}{c_3(\ell_2 - 2)^2} \left( \ln \frac{\sigma}{\sqrt{c_3}} \right)^2 + 237\delta^2 \right)
\]
\[
+ \frac{2d}{r_2} E[||R\hat{g}_j(x) - R\nabla f(x)||^2]
\]
\[
\leq \frac{2d}{r_2} \left( \frac{36\sigma^2}{c_3(\ell_2 - 2)^2} \left( \ln \frac{\sigma}{\sqrt{c_3}} \right)^2 + 237\delta^2 + \sigma^2 \right)
\]
\[
+ \frac{2d}{r_2} E[||Y - \nabla f(x)||^2],
\]
where the second last inequality uses bound from Lemma 13 and the fact
\[
E[||R\hat{g}_j(x) - RY||^2]
\]
\[
= E[||R\hat{g}_j(x) - R\nabla f(x)||^2] + E[||RY - R\nabla f(x)||^2],
\]
and the final inequality uses the fact that \( R \) is a unitary matrix and assumption (23).

Recall from Lemma 14 that the quantity \( (\nabla f(x)(i) - Y(i)) \) is subgaussian with variance parameter \( \sigma^2 \). Thus, \( E[(Y(i) - \nabla f(x)(i))^2] \leq c_5\sigma^2 \) for some universal constant \( c_5 > 0 \) (see, for instance, [57]). Again using the fact that \( d\sigma^2 < \sigma^2 \), whenever \( nr \geq c_4d^2 \log(B/\sigma) \), \( E[||Y - \nabla f(x)||^2] \leq \sigma^2 \).

Further, the bias remains unchanged compared to without subsampling case, i.e.,
\[
E[||Q_{\mathbf{R}}(\hat{g}_j(x), Y) - \hat{g}_j(x)||]
\]
\[
= E[||Q_{\mathbf{R}}(\hat{g}_j(x), Y) - \hat{g}_j(x)||] \leq \sqrt{237}\delta.
\]

At last, we set the following parameters for WZ-SGD:
\[
\log \ell_2 = \left[ c \log \log \left( \frac{B}{\sigma} \cdot nT \right) \right], \quad \delta = \frac{2\sigma}{nT} \cdot \frac{\sigma}{B}.
\]

Accordingly, we need to sample \( r_2 = r / \left[ c \log \log \left( \frac{B}{\sigma} \cdot nT \right) \right] \) coordinates at each client. Using the standard bounds for averaging of vectors in Lemma 1, respectively, we obtain
\[
\alpha^2(M_t) \leq c_1 \frac{\sigma^2 \log \left( \frac{r}{n} \cdot nT \right)}{n}, \quad \beta^2(M_t) \leq c_2 \frac{\sigma^2}{n^2T^2} \cdot \frac{\sigma^2}{B^2},
\]
for suitably chosen constants \( c_1, c_2 > 0 \) and \( M_t \) as defined in (29). The proof is completed by using the bounds on \( \alpha' \) and \( \beta' \) with Lemma 9.

\( \Box \)

M. Proof of Theorem 9

We proceed in a way similar to the proof of Theorem 7. Towards that, we first use the mean square assumption in (30) to write
\[
E\left[ \left\| \frac{2}{n} \sum_{i \in C_2} \hat{g}_i(x_i) - \nabla f(x_i) \right\|_2^2 \right] \leq \frac{2\sigma^2}{n}.
\]

Then, it only remains to bound the term \( E\left[ \left\| \frac{2}{n} \sum_{i \in C_2} Q_{\text{R-MQ}}(\hat{g}_i(x_i), Y_i) - \frac{2}{n} \sum_{i \in C_2} \hat{g}_i(x_i) \right\|_2^2 \right] \), where \( Y_i = \frac{2}{n} \sum_{i \in C_2} Q_{\text{R-MQ}}(\hat{g}_i(x_i)) \) is the side-information. For that, we follow the proof of Lemma 7 in Section C-H.

Fix any arbitrary client \( i \in C_2 \). Conditioning on its gradient estimate \( \hat{g}_i(x_i) \), the available side information at server \( Y_i = y_i \), we have using the proof of Lemma 7 that
\[
E\left[ \left\| Q_{\text{R-MQ}}(\hat{g}_i(x_i), Y_i) - \hat{g}_i(x_i) \right\|_2^2 \right] \leq 16\sqrt{3dB} \left\| \hat{g}_i(x_i) - y_i \right\| \frac{r}{n^2T^2} \left\| \frac{T}{n^2} \right\| - 1.
\]

By the law of total expectations, we also have
\[
E\left[ \left\| Q_{\text{R-MQ}}(\hat{g}_i(x_i), Y_i) - \hat{g}_i(x_i) \right\|_2^2 \right] \leq 16\sqrt{3dB} E\left[ \left\| \hat{g}_i(x_i) - Y_i \right\| \right] \frac{r}{n^2T^2} \left\| \frac{T}{n^2} \right\| - 1.
\]

The proof is completed by noting that
\[
E\left[ \left\| \hat{g}_i(x_i) - Y_i \right\|_2^2 \right] \leq E\left[ \left\| \hat{g}_i(x_i) - Y_i \right\|_2^2 \right]
\]
\[
= E\left[ \left\| \hat{g}_i(x_i) - \nabla f(x_i) \right\|_2^2 \right] + E\left[ \left\| \nabla f(x_i) - Y_i \right\|_2^2 \right]
\]
\[
\leq \sigma^2 + \frac{2\sigma^2}{n} + \frac{2dB^2}{n \left( \frac{\sqrt{dT} \log(1 + \ln(n))}{\sqrt{d}} \right)^2} - 1,
\]
where the first line is using Jensen’s inequality, the only identity is due to the unbiased property of subsampled RATQ (c.f. Lemma 12), and the last line is due to (30) and applying the value of \( \alpha(M_t) \) for \( C_1 \) in Theorem 7.\( \Box \)
N. Proof of Theorem 10

The proof of this Theorem is similar to that of Lemma 3. We denote by $Q(X(i), Y(i))$ the output of the modulo quantizer with side information $Y(i)$ and parameters $k, \Delta'$ set as in (31). Then, we have

$$
E \left[ ||Q_d(X, Y) - X||^2 \right] \\
\leq \sum_{i=1}^{d} E \left[ (Q(X(i), Y(i)) - X(i))^2 \right] \\
\leq \sum_{i=1}^{d} E \left[ (Q(X(i), Y(i)) - X(i))^2 1_{\{|X(i)-Y(i)| \leq \Delta'\}} \right] \\
+ \sum_{i=1}^{d} E \left[ (Q(X(i), Y(i)) - X(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right].
$$

(51)

We bound the first term on the right-side in a similar manner as the bound in (34). Specifically, under the event $\{|X(i)-Y(i)| \leq \Delta'\}$, we get by Lemma 2 that

$$
|Y(i) - X(i)| \leq \epsilon = \frac{2\Delta'}{k-2}, \quad \text{almost surely,}
$$

whereby

$$
\sum_{i=1}^{d} E \left[ (Y(i) - X(i))^2 1_{\{|X(i)-Y(i)| \leq \Delta'\}} \right] \leq d \epsilon^2.
$$

(52)

For the second term in the RHS note that $X(i) - Y(i)$ is subgaussian with variance factor $\sigma^2$. Therefore, by proceeding in a similar manner as the derivation of (35) we get

$$
\sum_{i=1}^{d} E \left[ (Q(X(i), Y(i)) - X(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right] \\
\leq 2 \sum_{i=1}^{d} E \left[ (Q(X(i), Y(i)) - Y(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right] \\
+ 2 \sum_{i=1}^{d} E \left[ (Y(i) - X(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right] \\
\leq 2k^2 \epsilon^2 \sum_{i=1}^{d} P(|X(i) - Y(i)| \geq \Delta') \\
+ 2 \sum_{i=1}^{d} E \left[ (X(i) - Y(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right] \\
\leq 4d \epsilon^2 e^{-d\Delta'^2 / 2\sigma^2} \\
+ 2 \sum_{i=1}^{d} E \left[ (X(i) - Y(i))^2 1_{\{|X(i)-Y(i)| \geq \Delta'\}} \right] \\
\leq 4d \epsilon^2 e^{-\Delta'^2 / 2\sigma^2} + 2(2\epsilon^2 + d\Delta'^2) e^{-\Delta'^2 / 2\sigma^2},
$$

(53)

where the second inequality follows upon noting from the description decoder of MQ in Alg. 2 that $|Q(X(i), Y(i)) - Y(i)| \leq \epsilon k$ almost surely for each $i \in [d]$; the third inequality uses the fact that $X(i) - Y(i)$ is sub-Gaussian with variance parameter $\sigma^2$; the fourth inequality is by Lemma 10.

Upon bounding the two terms on the right-side of (51) from above using (52), (54), we obtain

$$
E \left[ ||Q_d(X, Y) - X||^2 \right] \\
\leq d \epsilon^2 + 4d \epsilon^2 e^{-\Delta'^2 / 2\sigma^2} + 4(2\epsilon^2 + d\Delta'^2) e^{-\Delta'^2 / 2\sigma^2}.
$$

Note that the RHS in the upper bound above is precisely the same as in (36) with $\sigma^2$ replacing $\Delta'^2 / d$. Therefore proceeding in the same manner as in (36), we get

$$
E \left[ ||Q_d(X, Y) - X||^2 \right] \leq 24 \frac{\sigma^2}{(k-2)^2} \ln \frac{\sigma^2}{\delta} + 154 \delta^2.
$$

Substituting the value of $k$ and $\delta$ completes the proof. \( \Box \)

O. Proof of Lemma 8

For $Q(x)$ as in (17), we have

$$
Q(x) = \sum_{i=1}^{N} q_i / N,
$$

where $q_i$ for all $i \in \{1, \ldots, N\}$ is an unbiased estimate of $x$ and equals in distribution the output of the RDAQ quantizer for an input $x$ and side information $y$. Moreover, $q_i$s are mutually independent conditioned on $R$. Therefore,

$$
E \left[ ||Q(x) - x||^2 \right] = E \left[ \left\| \sum_{i=1}^{N} \frac{q_i}{N} - x \right\|^2 \right] \\
= E \left[ \left\| \sum_{i=1}^{N} \frac{q_i - x}{R} \right\|^2 \right] \\
= E \left[ \sum_{i=1}^{N} \frac{1}{N^2} E \left[ ||q_i - x||^2 \right] \right] \\
\leq 16 \sqrt{2} \frac{\Delta}{N},
$$

where the third identity follows from the conditional independence of $q_i$s after conditioning on $R$ and the fact that $q_i$ is an unbiased estimate of $x$. The final inequality follows from the fact that $q_i$ equals in distribution the output of the RDAQ quantizer and then using Lemma 6. \( \Box \)

REFERENCES

[1] J. Acharya, C. L. Canonne, P. Mayekar, and H. Tyagi, “Information-constrained optimization: Can adaptive processing of gradients help?” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 7126–7138.

[2] J. Acharya, C. L. Canonne, Z. Sun, and H. Tyagi, “Unified lower bounds for interactive high-dimensional estimation under information constraints,” 2020, arXiv:2010.06562.

[3] J. Acharya, C. L. Canonne, P. Mayekar, and H. Tyagi, “Information-constrained optimization: Can adaptive processing of gradients help?” 2021, arXiv:2104.00979.

[4] J. Acharya, C. De Sa, D. J. Foster, and K. Sridharan, “Distributed learning with sublinear communication,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 40–50.

[5] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, “Information-theoretic lower bounds on the Oracle complexity of stochastic convex optimization,” IEEE Trans. Inf. Theory, vol. 58, no. 5, pp. 3235–3249, May 2012.

[6] N. Ailon and B. Chazelle, “Approximate nearest neighbors and the fast Johnson-lindenstrauss transform,” in Proc. 38th Annul. ACM Symp. Theory Comput., May 2006, pp. 557–563.
S. Boucheron, G. Lugosi, and P. Massart, “Concentration Inequalities: A Nonasymptotic Theory of Independence.” Oxford, U.K.: Oxford Univ. Press, 2013.

S. Bubeck, “Convex optimization: Algorithms and complexity,” Found. Trends® Mach. Learn., vol. 8, no. 3-4, pp. 313–357, 2015.

W.-N. Chen, P. Kairouz, and A. Ozgur, “Breaking the communication-privacy-accuracy trilemma,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 3312–3324.

P. Davies, V. Gurunathan, N. Moshrefi, S. Ashkboos, and D. Alistarh, “New bounds for distributed mean estimation and variance reduction,” 2020, arXiv:2002.09268.

J. C. Duchi, M. I. Jordan, M. J. Wainwright, and Y. Zhang, “Optimality guarantees for distributed statistical estimation,” 2014, arXiv:1405.0782.

R. Durrett, Probability: Theory and Examples. Cambridge, MA: Cambridge Univ. Press, 2019.

F. Fagnani, I. Tabrizian, I. Markov, D. Alistarh, D. Roy, and A. Ramezani-Kebrya, “Adaptive gradient quantization for data-parallel SGD,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 3174–3185.

G. D. Forney, “Coset codes. I. Introduction and geometrical classification,” IEEE Trans. Inf. Theory, vol. IT-34, no. 5, pp. 1123–1151, Sep. 1988.

V. Gandikota, D. Kane, R. K. Maity, and A. Mazumdar, “vqSGD: Vector quantized stochastic gradient descent,” in Proc. 24th Int. Conf. Artif. Intell. Statist., in Proceedings of Machine Learning Research, 2021, pp. 2197–2205.

A. Ghosh, R. K. Maity, and A. Mazumdar, “Distributed Newton can communicate less and resist Byzantine workers,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 18028–18038.

T. Holenstein, “Parallel repetition: Simplification and the no-signaling case,” Theory Comput., vol. 5, no. 8, pp. 141–172, 2009.

K. J. Horadam, Hadamard Matrices and Their Applications. Princeton, NJ, USA: Princeton Univ. Press, 2012.

Z. Huang, W. Yilei, and K. Yi, “Optimal sparsity-sensitive bounds for distributed optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 6371–6381.

K. J. Horadam, Hadamard Matrices and Their Applications. Princeton, NJ, USA: Princeton Univ. Press, 2012.

D. J. Junghunwala, A. Gadhikar, J. Goshi, and Y. C. Eldar, “Adaptive quantization of model updates for communication-efficient federated learning,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 3110–3114.

D. Junghunwala, A. Mallick, A. H. Gadhikar, S. Kadhe, and G. Joshi, “Leveraging spatial and temporal correlations in sparsified mean estimation,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 14280–14292.

P. Kairouz et al., “Advances and open problems in federated learning,” 2019, arXiv:1912.04977.

B. Kashin, “Section of some finite-dimensional sets and classes of smooth functions (in Russian) izv, Acad. Nauk SSSR, vol. 41, pp. 334–351, Jan. 1977.

J. Konecný and P. Richtárik, “Randomized distributed mean estimation: Accuracy vs. communication,” Frontiers Appl. Math. Statist., vol. 4, p. 62, Dec. 2018.

J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in Proc. NIPS Workshop Private Multi-Party Mach. Learn., 2016.

S. B. Korada and R. L. Urbanke, “Polar codes are optimal for lossy source coding,” IEEE Trans. Inf. Theory, vol. 56, no. 4, pp. 1751–1768, Apr. 2010.

K. Liang and Y. Wu, “Improved communication efficiency for distributed mean estimation with side information,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2021, pp. 3185–3190.

K. Liang, H. Zhong, H. Chen, and Y. Wu, “Wyner-Ziv gradient compression for federated learning,” 2021, arXiv:2111.08277.

C.-Y. Lin, V. Kostina, and B. Hassibi, “Differentially quantized gradient descent,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2021, pp. 1200–1205.

C. Ling, S. Gao, and J.-C. Belfiore, “Wyner-Ziv coding based on multidimensional nested lattices,” IEEE Trans. Commun., vol. 60, no. 5, pp. 1328–1335, May 2012.

L. Liu, “Polar codes and polar lattices for efficient communication and source coding,” Ph.D. thesis, Dept. Elect. Electron. Eng., Imperial College London, London, U.K., 2016.

L. Liu, J. Shi, and C. Ling, “Polar lattices for lossy compression,” 2015, arXiv:1501.05683.

Y. Lu and C. De Sa, “Moniqa: Module quantized communication in decentralized SGD,” 2020, arXiv:2002.11787.

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

P. Mayekar, S. K. Jha, and H. Tyagi, “Wyner-Ziv compression is (almost) optimal for distributed optimization,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2022, pp. 578–583.

P. Mayekar, A. T. Suresh, and H. Tyagi, “Wyner-Ziv estimators: Efficient distributed mean estimation with side-information,” in Proc. Int. Conf. Intell. Statist., 2021, pp. 3502–3510.

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

P. Mayekar and H. Tyagi, “RATQ: A universal fixed-length quantizer for stochastic optimization,” IEEE Trans. Inf. Theory, vol. 67, no. 5, pp. 3130–3154, May 2021.

A. Nemirovsky, (1995). Information-Based Complexity of Convex Programming. [Online]. Available: http://www2.isye.gatech.edu/~nemirovsky/EMACO.pdf

A. Nemirovsky and D. B. Yudin, Problem Complexity and Method Efficiency in Optimization. Hoboken, NJ, USA: Wiley, 1983.

Y. Oohama, “Gaussian multiterminal source coding,” IEEE Trans. Inf. Theory, vol. 43, no. 6, pp. 1912–1923, Mar. 1997.

S. S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” IEEE Trans. Inf. Theory, vol. 49, no. 3, pp. 626–643, Mar. 2003.

A. Ramezani-Kebrya, F. Fagnani, I. Markov, V. Aksenov, D. Alistarh, and D. M. Roy, “NUQSGD: Provably communication-efficient data-parallel SGD via unisonform quantization,” 2019, arXiv:1908.06077.

A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, “FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization,” in Proc. Int. Conf. Intell. Statist., 2020, pp. 2021–2031.

M. Safaryan, E. Shulgin, and P. Richtárik, “Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor,” 2020, arXiv:2002.08958.

R. Saha, S. Rini, M. Rao, and A. Goldsmith, “Decentralized optimization over noisy, rate-constrained networks: How we agree by talking about how we disagree,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 5055–5059.

F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, “1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs,” in Proc. Interspeech, Sep. 2014, pp. 1058–1062.

S. U. Stich, J.-B. Cordonnier, and M. Jaggi, “Sparsified SGD with memory,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 4452–4463.

A. T. Suresh, Z. Sun, J. Ro, and F. Yu, “Correlated quantization for distributed mean estimation and optimization,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 20856–20876.

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., vol. 70, Aug. 2017, pp. 3329–3337.

R. Vershynin, High-Dimensional Probability: An Introduction With Applications in Data Science, vol. 47, Cambridge, U.K.: Cambridge Univ. Press, 2018.

T. Vogels, S. P. Karimireddy, and M. Jaggi, “PowerSGD: Practical low-rank gradient compression for distributed optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 14269–14278.
[59] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, “ATOMO: Communication-efficient learning via atomic sparsification,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9850–9861.

[60] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 1306–1316.

[61] W. Wen et al., “TernGrad: Ternary gradients to reduce communication in distributed deep learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1509–1519.

[62] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” IEEE Trans. Inf. Theory, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.

[63] R. Zamir, S. Shamai, and U. Erez, “Nested linear/lattice codes for structured multiterminal binning,” IEEE Trans. Inf. Theory, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

Prathamesh Mayekar received the B.E. degree in electronics and telecommunications engineering from the University of Mumbai, the M.Tech. degree in industrial engineering and operation research from the Indian Institute of Technology Bombay, and the Ph.D. degree in electrical communication engineering from the Indian Institute of Science, under the supervision by Prof. Himanshu Tyagi. He is currently an independent Researcher broadly working on problems at the intersection of information theory and optimization. He was a recipient of the 2018 Jack Keil Wolf ISIT Student Paper Award.

Shubham Jha (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the Indian Institute of Information Technology, Guwahati, India, in 2018. He is currently pursuing the Ph.D. degree with the Robert Bosch Center for Cyber-Physical Systems, Indian Institute of Science, Bengaluru, India. His research interests include the intersection of information theory, distributed optimization, and wireless communication systems.

Ananda Theertha Suresh received the Ph.D. degree from the University of California San Diego, under supervision of Prof. Alon Orlitsky. He is currently a Research Scientist with Google Research, New York. His research interests include the theoretical and algorithmic aspects of machine learning, information theory, and privacy, and statistics.

Himanshu Tyagi (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees in electrical engineering from the Indian Institute of Technology Delhi, New Delhi, India, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA. He is currently an Associate Professor with the Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru, India. His research interests include information theory, statistics, cryptography, machine learning, and blockchains.