# Wyner-Ziv Estimators: Efficient Distributed Mean Estimation with Side Information

Prathamesh Mayekar
Indian Institute of Science
prathamesh@iisc.ac.in

Ananda Theertha Suresh
Google Research
theertha@google.com

Himanshu Tyagi
Indian Institute of Science
htyagi@iisc.ac.in

**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.

# Contents

# 1 Introduction

## 1.1 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

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{1}$$

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 Bottou (2010), Kairouz et al. (2019), Konečný et al. (2016), Alistarh et al. (2017), Ramezani-Kebrya et al. (2019), Gandikota et al. (2019), Basu et al. (2019), Seide et al. (2014), Wang et al. (2018), Stich et al. (2018), Wen et al. (2017), Wangni et al. (2018), Lu and De Sa (2020), Vogels et al. (2019), Acharya et al. (2019)). 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 Suresh et al. (2017), Konečný and Richtárik (2018), Chen et al. (2020), Huang et al. (2019), Mayekar and Tyagi (2020b), Safaryan et al. (2020), Albasyoni et al. (2020), 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. 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. Similarly, distributed mean estimation with side information can be used for variance reduction in other problems such as power iteration or parallel SGD (*cf.* Davies et al. (2020)).

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 sever and can, therefore, be used for estimating the mean at the server, but is not available to the clients while quantizing the updates $\{x_i\}_{i=1}^n$.

## 1.2 The model

Consider the input $\mathbf{x} := (x_1, \ldots, x_n)$ and the side information $\mathbf{y} := (y_1, \ldots, y_n)$. 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 $\pi = (\pi_1, ..., \pi_n)$ where the communication $C_i = \pi_i(x_i, U) \in \{0,1\}^r$

of client[1] $i$, $i \in [n]$, can only depend on its local observation $x_i$ and public randomness $U$. Note that the clients are not aware of side information $\mathbf{y}$, 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 $C^n := (C_1, C_2, ..., C_n)$, the server uses the transcript $(C^n, U)$ of the protocol and the side information $\mathbf{y}$ to form the estimate of the sample mean[2] $\hat{\bar{x}} = \hat{\bar{x}}(C^n, U, \mathbf{y})$; see Figure 1 for a depiction of our setting. We call such a $\pi$ an *$r$-bit SMP protocol* with input $(\mathbf{x}, \mathbf{y})$ and output $\hat{\bar{x}}$.

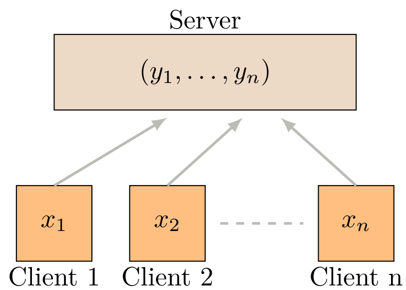


Figure 1: Problem setting of mean estimation with side information

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

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

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

$$\|x_i - y_i\|_2 \le \Delta_i, \quad \forall\, i \in [n]. \tag{2}$$

Denoting $\boldsymbol{\Delta} := (\Delta_1, \dots, \Delta_n)$, we are interested in the performance of our protocols for the following two settings:

1. **The *known* $\boldsymbol{\Delta}$ setting**, where $\Delta_i$ is known to client $i$ and the server;
2. **The *unknown* $\boldsymbol{\Delta}$ 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 $\boldsymbol{\Delta}$ setting, the quantizers of different clients can be chosen using the knowledge of $\boldsymbol{\Delta}$; in the unknown $\boldsymbol{\Delta}$ setting, they must be fixed irrespective of $\boldsymbol{\Delta}$.

In another direction, we distinguish the *low-precision* setting of $r \le d$ 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.

As a benchmark, we recall the result for distributed mean estimation with no side-information from Suresh et al. (2017). When all $x_i$s lie in the Euclidean ball of radius 1, Suresh et al. (2017) showed that the minmax MSE in the no side-information case is

$$\Theta\left(\frac{d}{nr}\right). \tag{3}$$

[1] $[n] := \{1, \dots, n\}$.

[2] While side information $y_i$ is associated with client $i$, we do not enforce this association in our general formulation at this point.

## 1.3 Our contributions

Drawing on ideas from distributed quantization problem in information theory (*cf*. Wyner and Ziv (1976)), specifically the Wyner-Ziv problem, we present *Wyner-Ziv estimators* for distributed mean estimation. In the known $\mathbf{\Delta}$ setting, for a fixed $\mathbf{\Delta}$, and the low-precision setting of $r \le d$, we propose an *r-bit SMP protocol* $\pi^*_{\mathtt{k}}$ which satisfies[3]

$$\mathcal{E}(\pi^*_{\mathtt{k}}, \mathbf{x}, \mathbf{y}) = O\left(\sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d \log\log n}{nr}\right),$$

for all $\mathbf{x}$ and $\mathbf{y}$ 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 $\sum_{i=1}^{n} \frac{\Delta_i^2 \log\log n}{n} \le 1$. 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 3.3.

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 Ailon and Chazelle (2006)) 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 $x_i$, 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.

Moreover, we show that this protocol $\pi^*_{\mathtt{k}}$ 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 $r \le d$, we can find $\mathbf{x}$ and $\mathbf{y}$ satisfying (2) such that

$$\mathcal{E}(\pi, \mathbf{x}, \mathbf{y}) = \Omega\left(\min_{i \in \{1,\ldots,n\}} \Delta_i^2 \cdot \frac{d}{nr}\right).$$

In the unknown $\mathbf{\Delta}$ setting, we propose a protocol $\pi^*_{\mathtt{u}}$ which adapts to the unknown distance $\Delta_i$ between $x_i$ and $y_i$ and, remarkably, provides MSE guarantees dependent on $\mathbf{\Delta}$. Specifically, for the low-precision setting of $r \le d$, the protocol satisfies[4]

$$\mathcal{E}(\pi^*_u, \mathbf{x}, \mathbf{y}) = O\left(\sum_{i=1}^{n} \frac{\Delta_i}{n} \cdot \frac{d \ln^* d}{nr}\right),$$

for all $\mathbf{x}$ and $\mathbf{y}$ in the unit Euclidean ball $\mathcal{B} := \{x \in \mathbb{R}^d : \|x\|_2 \le 1\}$ and satisfying (2). Thus, we improve upon the optimal estimator for the no side information counterpart (3) in the regime $\sum_{i=1}^{n} \frac{\Delta_i \ln^* d}{n} \le 1$. 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 Holenstein (2009) which enables to derive a $\mathbf{\Delta}$ dependent MSE bound.

Furthermore, both our quantizers can be extended to the high-precision regime of $r > d$. The quantizer for the known $\mathbf{\Delta}$ setting directly extends by using $r/d$ bits per dimension. The MSE of

[3] We denote by $\log(\cdot)$ logarithm to the base 2 and by $\ln(\cdot)$ logarithm to the base $e$.

[4] We denote by $\ln^*(a)$ the minimum number of iterated logarithms to the base $e$ that must be applied to $a$ to make it less than 1.

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 Davies et al. (2020) for the high-precision regime. The quantizer for the unknown $\boldsymbol{\Delta}$ setting can be extended by sending the "type" of the communication vector, following an idea proposed in Mayekar and Tyagi (2020a). The MSE of the SMP protocol using this quantizer for all the clients falls as $2^{-r/d \ln^* d}$ as opposed to $d/r$ that can be obtained using naive extensions of our quantizer.

Finally, in a different direction, we revisit the classic Gaussian rate-distortion problem (*cf.* Oohama (1997)) 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 $\boldsymbol{\Delta}$ 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 are not computationally efficient. The second reason is that we do not require the distribution to be exactly Gaussian and subgaussianity suffices.

## 1.4 Prior work

The known $\boldsymbol{\Delta}$ setting described above was first considered in Davies et al. (2020). The scheme of Davies et al. (2020) 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 upto a multiplicative factor of $\log \log n$ in the low-precision regime of $r \leq d$ . In comparison, the scheme of Davies et al. (2020) 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 Davies et al. (2020). We remark that the unknown $\boldsymbol{\Delta}$ 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 (Wyner and Ziv, 1976). Practical codes for settings where the observations are generated from known distributions have been constructed using channel codes; see, for instance, Korada and Urbanke (2010); Ling et al. (2012); Liu and Ling (2015); Pradhan and Ramchandran (2003); Zamir et al. (2002). 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 Davies et al. (2020) is similar to lattice schemes in Ling et al. (2012); Liu and Ling (2015); Zamir et al. (2002).

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 Duchi et al. (2014), Suresh et al. (2017) shows a lower bound on MSE of $\Omega\left(\dfrac{d}{nr}\right)$, when all $x_i$s lie in the Euclidean ball of radius one. Suresh et al. (2017) 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 Mayekar and Tyagi (2020b) to a much tighter $\log\log^* d$ factor. For a precision constraint of $r = \Theta(d)$, the variable-length quantizers proposed in Suresh et al. (2017), Alistarh et al. (2017), Ramezani-Kebrya et al. (2019) as well as the fixed-length quantizers in Mayekar and Tyagi (2020a), Gandikota et al. (2019) are order-wise optimal.

Our results for the low-precision regime in known $\mathbf{\Delta}$ setting are provided in Section 3 and in the unknown $\mathbf{\Delta}$ setting are provided in Section 4. In Section 5, we extend our results to the high-precision regime. In Section 6, 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 Section 7. Before presenting these results, we review some preliminaries in the next section.

# 2 Preliminaries and the structure of our protocols

While our lower bound for the known $\mathbf{\Delta}$ setting holds for an arbitrary SMP protocol, both the protocols we propose in this paper, for the known $\mathbf{\Delta}$ and the unknown $\mathbf{\Delta}$ settings, have a common structure. We use $r$-bit quantizers to form estimates of $x_i$s at the server and then compute the sample mean of the estimates of $x_i$s. 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 $\mathcal{Y} \subset \mathbb{R}^d$ consists of randomized mappings[5] $(Q^{\mathsf{e}}, Q^{\mathsf{d}})$ with the encoder mapping $Q^{\mathsf{e}} : \mathcal{X} \to \{0,1\}^r$ used by the client to quantize and the decoder mapping $Q^{\mathsf{d}} : \{0,1\}^r \times \mathcal{Y} \to \mathcal{X}$ used by the server to aggregate quantized vectors. The overall quantizer $Q$ is given by the composition mapping $Q(x,y) = Q^{\mathsf{d}}((Q^{\mathsf{e}}(x), y)$.

In our protocols, for input $\mathbf{x}$ and side information $\mathbf{y}$, client $i$ uses the encoder $Q_i^{\mathsf{e}}$ for the $r$-bit quantizer $Q_i$ to send $Q_i^{\mathsf{e}}(x_i)$. The server uses $Q_i^{\mathsf{e}}(x_i)$ 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

$$\hat{\bar{x}} := \frac{1}{n}\sum_{i=1}^{n} \hat{x}_i. \tag{4}$$

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

$$\begin{aligned}\alpha(Q;\Delta) &:= \sup_{x\in\mathcal{X}, y\in\mathcal{Y}: \|x-y\|_2\le\Delta} \mathbb{E}\left[\|Q(x,y)-x\|_2^2\right],\\ \beta(Q;\Delta) &:= \sup_{x\in\mathcal{X}, y\in\mathcal{Y}: \|x-y\|_2\le\Delta} \|\mathbb{E}\left[Q(x,y)-x\right]\|_2^2,\end{aligned}$$

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 $\|x-y\|_2 \le \Delta$.

The result below will be very handy for our analysis.

[5] We can use public randomness $U$ for randomizing.

**Lemma 2.1.** *For* $\mathbf{x} \in \mathcal{X}^n$ *and* $\mathbf{y} \in \mathcal{Y}^n$ *satisfying* (2) *and* $r$*-bit quantizers* $Q_i$*,* $i \in [n]$*, using independent randomness for different* $i \in [n]$*, the estimate* $\hat{\bar{x}}$ *in* (4) *and the sample mean* $\bar{x}$ *in* (1) *satisfy*

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

# 3 Distributed mean estimation with known $\boldsymbol{\Delta}$

In this section, we present our Wyner-Ziv estimator for the known $\boldsymbol{\Delta}$ setting. As described in Section 2, we use the 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 3.3. In Sections 3.1 and 3.2, we provide the basic building blocks of our final quantizer. Further, in Section 3.4, 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 $r \leq d$.

## 3.1 Modulo Quantizer (MQ)

The first subroutine used by our larger quantizer is the *Modulo Quantizer* (MQ). MQ is a one dimensional distributed 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 $\mathbb{Z}_\varepsilon = \{\varepsilon z : z \in \mathbb{Z}\}$. For a given input $x$, the encoder $Q^{\mathtt{e}}_{\mathtt{M}}$ 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 $\tilde{z}$ satisfies

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

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

Upon receiving this $w$, the decoder $Q^{\mathtt{d}}$ looks at the set $\mathbb{Z}_{w,\varepsilon} = \{(zk + w)\cdot\varepsilon : z \in \mathbb{Z}\}$ and decodes the point closest to $y$, which we denote by $Q_{\mathtt{M}}(x, y)$. 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. 3 we look for the closest point to $y$ in the lattice $Z_{w,\varepsilon} := \{(zk + w)\cdot\varepsilon : z \in \mathbb{Z}\}$, the output $Q_{\mathtt{M}}(x, y)$ satisfies

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

We summarize MQ in Alg. 2 and 3.

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

**Lemma 3.1.** *Consider the Modulo Quantizer* $Q_{\mathtt{M}}$ *described in Alg. 2 and 3 with parameter* $\varepsilon$ *set to satisfy*

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

**Require:** Input $x \in \mathbb{R}$, Parameters $k$, $\Delta'$, and $\varepsilon$
1: Compute $z_u = \lceil x/\varepsilon \rceil$, $z_l = \lfloor x/\varepsilon \rfloor$
2: Generate $\tilde{z} = \begin{cases} z_u, & w.p.\ x/\varepsilon - z_l \\ z_l, & w.p.\ z_u - x/\varepsilon \end{cases}$
3: **Output:** $Q_{\mathtt{M}}^{\mathtt{e}}(x) = \tilde{z} \bmod k$

Algorithm 2: Encoder $Q_{\mathtt{M}}^{\mathtt{e}}(x)$ of MQ

**Require:** Input $w \in \{0, \ldots, k-1\}$, $y \in \mathbb{R}$
1: Compute $\hat{z} = \arg\min\{|(zk + w) \cdot \varepsilon - y| \colon z \in \mathbb{Z}\}$
2: **Output:** $Q_{\mathtt{M}}^{\mathtt{d}}(w, y) = (\hat{z}k + w)\varepsilon$

Algorithm 3: Decoder $Q_{\mathtt{M}}^{\mathtt{d}}(w, y)$ of MQ

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

$$\mathbb{E}\left[Q_{\mathtt{M}}(x, y)\right] = x \quad \textit{and}$$
$$|Q_{\mathtt{M}}(x, y) - x| \leq \varepsilon, \quad \textit{almost surely.}$$

*In particular, we can set $\varepsilon = 2\Delta'/(k-2)$, to get $|Q_{\mathtt{M}}(x, y) - x| \leq 2\Delta'/(k-2)$. Furthermore, the output of $Q_{\mathtt{M}}$ 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*. Forney (1988); Liu (2016); Zamir et al. (2002)) and was used in Davies et al. (2020) as well.

## 3.2 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 Horadam (2012))[6] and $D$ is a diagonal matrix with each diagonal entry generated uniformly from $\{-1, +1\}$. Note that we use public randomness[7] to generate the same $D$ at both the encoder and the decoder. We formally describe the quantizer in[8] Alg. 4 and 5.

[6]We 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.

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

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

*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

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

(see, for instance, (Boucheron et al., 2013, Theorem 2.8)). This observation allows us to use $\Delta' \approx \Delta/\sqrt{d}$ for MQ applied to each coordinate.

**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:** $Q^{\mathtt{e}}_{\mathtt{M},R}(x) = [Q^{\mathtt{e}}_{\mathtt{M}}(x'(1)), \ldots, Q^{\mathtt{e}}_{\mathtt{M}}(x'(d)]^T$ using parameters $k$, $\varepsilon$, and $\Delta'$ for $Q^{\mathtt{e}}_{\mathtt{M}}$ of Alg. 2

Algorithm 4: Encoder $Q^{\mathtt{e}}_{\mathtt{M},R}(x)$ of RMQ

**Require:** Input $w \in \{0, \ldots, k-1\}^d$, $y \in \mathbb{R}^d$,
Parameters $k$ and $\Delta'$
1: Get $R$ from public randomness.
2: $y' = Ry$
3: **Output:** $Q^{\mathtt{d}}_{\mathtt{M},R}(w, y) = R^{-1} \displaystyle\sum_{i \in [d]} Q^{\mathtt{d}}_{\mathtt{M}}(w(i), y'(i)) e_i$
using parameters $k$, $\varepsilon$, and $\Delta'$ for $Q^{\mathtt{d}}_{\mathtt{M}}$ of Alg. 3,

Algorithm 5: Decoder $Q^{\mathtt{d}}_{\mathtt{M},R}(w, y)$ of RMQ

**Lemma 3.2.** *Fix $\Delta \geq 0$. Let $Q_{\mathtt{M},R}$ be RMQ described in Alg. 4 and 5. Then, for*[9] *$k \geq 4$, $\delta \in (0, \Delta)$, $\Delta' = \sqrt{6(\Delta^2/d)\ln(\Delta/\delta)}$ and the parameter $\varepsilon$ of MQ set to $\varepsilon = 2\Delta'/(k-2)$, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ that*

$$\alpha(Q_{\mathtt{M},R}; \Delta) \leq \frac{24\,\Delta^2}{(k-2)^2} \ln\frac{\Delta}{\delta} + 154\,\delta^2 \quad \textit{and}$$
$$\beta(Q_{\mathtt{M},R}; \Delta) \leq 154\,\delta^2.$$

*Furthermore, the output of quantizer $Q_{\mathtt{M},R}$ can be described in $d \log k$ bits.*

*Remark* 2. The choice of $\Delta'$ in the first statement of the Lemma 3.2 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 Davies et al. (2020) as well.

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

### 3.3 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 $Q^{\mathtt{e}}_{\mathtt{M},R}(x)$ 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 $Q^{\mathtt{d}}_{\mathtt{M},R}$ applies $Q^{\mathtt{d}}_{\mathtt{M}}$ to these strings separately; now, we use $Q^{\mathtt{d}}_{\mathtt{M}}$ to decode the entries in $S$ alone. We describe the overall quantizer in Alg. 6 and 7.

**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:** $Q^{\mathtt{e}}_{\mathtt{WZ}}(x) = \{Q^{\mathtt{e}}_{\mathtt{M}}(Rx(i)) : i \in S\}$ using parameters $k$, $\varepsilon$, and $\Delta'$ for $Q^{\mathtt{e}}_{\mathtt{M}}$ of Alg. 2

Algorithm 6: Encoder $Q^{\mathtt{e}}_{\mathtt{WZ}}(x)$ 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 $\tilde{x} = (Q^{\mathtt{d}}_{\mathtt{M}}(w(i), Ry(i)), i \in S)$ using parameters $k$, $\varepsilon$, and $\Delta'$ for $Q^{\mathtt{d}}_{\mathtt{M}}$ of Alg. 3
3: $\hat{x}_R = \frac{1}{\mu} \sum_{i \in S} (\tilde{x}(i) - Ry(i))\, e_i + Ry$
4: **Output:** $Q^{\mathtt{d}}_{\mathtt{WZ}}(w, y) = R^{-1}\hat{x}_R$

Algorithm 7: Decoder $Q^{\mathtt{d}}_{\mathtt{WZ}}(w, y)$ of subsampled RMQ

*Remark* 3. 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 $Q_{\mathtt{WZ}}$ to that of $Q_{\mathtt{M},R}$, which was already analysed in the previous section.

**Lemma 3.3.** *Fix $\Delta > 0$. Let $Q_{\mathtt{WZ}}$ and $Q_{\mathtt{M},R}$ be the quantizers described in Alg. 6 and 7 and Alg. 4 and 5, respectively. Then, for $\mu d \in [d]$, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ that*

$$\begin{aligned}\alpha(Q_{\mathtt{WZ}}; \Delta) &\leq \frac{\alpha(Q_{\mathtt{M},R}; \Delta)}{\mu} + \frac{\Delta^2}{\mu} \quad \textit{and} \\ \beta(Q_{\mathtt{WZ}}; \Delta) &= \beta(Q_{\mathtt{M},R}; \Delta).\end{aligned}$$

*Furthermore, the output of quantizer $Q_{\mathtt{WZ}}$ can be described in $\mu d \log k$ bits.*

We are now equipped to prove our first main result. Our protocol $\pi^*_{\mathtt{k}}$ uses $Q_{\mathtt{WZ}}$ for each client as described in Section 2 and forms the estimate $\hat{\bar{x}}$ as in (4). We set the parameters needed for $Q_{\mathtt{WZ}}$ in Alg. 6 and 7 as follows: For client $i$, we set the parameters of MQ as

$$\delta = \frac{\Delta_i}{\sqrt{n}}, \quad \log k = \left\lceil \log(2 + \sqrt{12 \ln n}) \right\rceil, \quad \Delta' = \sqrt{6(\Delta_i^2/d) \ln(\Delta_i/\delta)}, \quad \varepsilon = 2\Delta'/(k-2), \tag{9}$$

and set the parameter $\mu$ as

$$\mu d = \left\lfloor \frac{r}{\log k} \right\rfloor . \tag{10}$$

We characterize the resulting error performance in the next result.

**Theorem 3.4.** *For a $n \geq 2$, a fixed $\mathbf{\Delta} = (\Delta_1, ..., \Delta_n)$, 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*

$$\mathcal{E}(\pi_k^*, \mathbf{x}, \mathbf{y}) \leq (79 \lceil \log(2 + \sqrt{12 \ln n}) \rceil + 26) \left( \sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d}{nr} \right),$$

*for all $\mathbf{x}, \mathbf{y}$ satisfying (2).*

*Proof.* Denoting by $Q_i$ the quantizer $Q_{\mathtt{WZ}}$ with parameters set for user $i$, by Lemmas 2.1 and 3.3, we get

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\leq \sum_{i=1}^{n} \frac{\alpha(Q_i; \Delta_i)}{n^2} + \sum_{i=1}^{n} \frac{\beta(Q_i; \Delta_i)}{n} \\
&\leq \frac{1}{\mu n^2} \sum_{i=1}^{n} (\alpha(Q_{\mathtt{M},R,i}; \Delta_i) + \Delta_i^2) + \sum_{i=1}^{n} \frac{\beta(Q_{\mathtt{M},R,i}; \Delta_i)}{n},
\end{aligned}$$

where $Q_{\mathtt{M},R,i}$ 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 3.2 and substituting $\delta^2 = \Delta_i^2/n$, we get

$$\begin{aligned}
\alpha(Q_{\mathtt{M},R,i}; \Delta_i) &\leq \frac{12\Delta_i^2 \ln n}{(k-2)^2} + \frac{154\Delta_i^2}{n}, \\
\beta(Q_{\mathtt{M},R,i}; \Delta_i) &\leq \frac{154\Delta_i^2}{n},
\end{aligned}$$

which with the previous bound gives

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\leq \frac{1}{\mu d} \left( \frac{12 \ln n}{(k-2)^2} + \frac{154}{n} + 1 + 154\mu \right) \sum_{i=1}^{n} \frac{d\Delta_i^2}{n^2} \\
&\leq \frac{79 \lceil \log(2 + \sqrt{12 \ln n}) \rceil + 26}{r} \sum_{i=1}^{n} \frac{d\Delta_i^2}{n^2},
\end{aligned}$$

where in the final bound we used our choice of $k$, the assumption that $n \geq 2$ (which implies that $d \geq r \geq 6$), and the fact that $\lceil r/\log k \rceil \geq r/2$ if $r \geq 2 \log k$. $\square$

*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

$$O\left( \sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d \log d}{nr} \right),$$

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 3.5 below, our Wyner-Ziv estimator $\pi_k^*$ is nearly optimal. Finally, $Q_{\mathtt{WZ}}$ can be efficiently implemented as both the encoding and decoding procedures have nearly-linear time complexity[10] of $O(d \log d)$.

### 3.4 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 Duchi et al. (2014) and is similar in structure to the lower bound for distributed mean estimation without side-information in Suresh et al. (2017).

**Theorem 3.5.** *Fix $\mathbf{\Delta} = (\Delta_1, \ldots, \Delta_n)$. There exists a universal constant $c < 1$ such that for any $r$-bit SMP protocol $\pi$, with $r \le cd$, there exists input $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2d}$ satisfying (2) and such that*

$$\mathcal{E}(\pi, \mathbf{x}, \mathbf{y}) \ge c \min_{i \in [d]} \Delta_i^2 \cdot \frac{d}{nr}.$$

## 4 Distributed mean estimation for unknown $\mathbf{\Delta}$

Finally, we present our Wyner-Ziv estimator for the unknown $\mathbf{\Delta}$ setting. We first, in Section 4.1, describe the idea of correlated sampling from Holenstein (2009), which will serve as an essential building block for all our quantizers in this section. We then build towards our final quantizer, described in 4.4, by first describing its simpler versions in Section 4.2 and 4.3. Once again, we restrict to the low-precision setting of $r \le d$.

### 4.1 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 $\mathbb{1}_{\{U \le x\}}$, 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:

$$\hat{X} = \mathbb{1}_{\{U \le x\}} - \mathbb{1}_{\{U \le y\}} + y, \tag{11}$$

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 $\mathbb{E}\left[\hat{X}\right] = x$. The variance of this estimator is given by

$$\mathtt{Var}(\hat{X}) = \mathbb{E}\left[(\hat{X} - x)^2\right] = |x - y| - (x - y)^2,$$

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 $\mathbf{\Delta}$-dependent expression, without requiring the knowledge of $\mathbf{\Delta}$.

[10] The most expensive operation at both the encoder and decoder of this estimator is the Hadamard matrix multiplication operation, which requires $d \log d$ real operations.

## 4.2 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 $\mathbb{1}_{\{U(i)\le x(i)\}}$. 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 $\mathbb{1}_{\{U(i)\le y(i)\}}$ for each $i \in [d]$. Then, the estimate of $x$ is formed as follows:

$$Q_{\mathtt{D}}(x, y) = \sum_{i=1}^{d} \left(\mathbb{1}_{\{U(i)\le x(i)\}} - \mathbb{1}_{\{U(i)\le y(i)\}}\right) e_i + y.$$

We formally describe the quantizer in Alg. 8 and 9.

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

Algorithm 8: Encoder $Q_{\mathtt{D}}^{\mathtt{e}}(x)$ 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 $\tilde{y} = \sum_{i=1}^d \mathbb{1}_{\{U(i)\le y(i)\}} \cdot e_i$
3: **Output:** $Q_{\mathtt{D}}^{\mathtt{d}}(w, y) = 2(w - \tilde{y}) + y$, where $w$ is viewed as a vector in $\mathbb{R}^d$

Algorithm 9: Decoder $Q_{\mathtt{D}}^{\mathtt{d}}(w, y)$ of DAQ

The next result characterizes the performance for DAQ.

**Lemma 4.1.** *Let $Q_{\mathtt{D}}$ denote DAQ described in Algorithms 8 and 9. Then, for $\mathcal{X} = \mathcal{Y} = \mathcal{B}$ and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{D}}; \Delta) \le 2\Delta\sqrt{d} \quad \textit{and} \quad \beta(Q_{\mathtt{D}}; \Delta) = 0.$$

*Furthermore, the output of quantizer $Q_{\mathtt{D}}$ can be described in $d$ bits.*

## 4.3 Rotated Distance Adaptive Quantizer (RDAQ)

Next, we proceed as for the known $\boldsymbol{\Delta}$ 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.

**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 distributed 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 $j \in [h]_0$,

$$M_j^2 := \frac{6}{d} \cdot e^{*j}, \quad \log h := \lceil \log(1 + \ln^*(d/6)) \rceil, \tag{12}$$

where $e^{*j}$ denotes the $j$th iteration of $e$ given by $e^{*0} := 1, \quad e^{*1} := e, \quad e^{*j} := e^{e^{*(j-1)}}$. 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 Mayekar and Tyagi (2020b), where it was used for the case with no side information.

**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 simply 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 $(\mathbb{1}_{\{U(i,j) \leq Rx(i)\}}, j \in [h]_0)$.

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 decoder for DAQ applied to $Ry$, but for scale $M_{z^*(i)}$. The encoding and decoding algorithm of RDAQ are described in Alg. 10 and 11, respectively.

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

Algorithm 10: Encoder $Q^{\mathtt{e}}_{\mathtt{D},R}(x)$ at for RDAQ

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

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

**Require:** Input $(w, z) \in \{0,1\}^{d\times h} \times [h]_0^d$ 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** $i \in [d]$ **do**

$$z'(i) = \min\{j \in \{[h]_0\} : |y_R(i)| \le M_j\}$$
$$z^*(i) = \max\{z(i), z'(i)\}$$

4: $w' = \sum_{i=1}^{d} 2M_{z^*(i)} \left(w(i, z^*(i)) - \mathbb{1}_{\{U(i,z^*(i)) \le y_R\}}\right)$

5: $\hat{x}_R = w' + Ry$

6: **Output:** $Q^{\mathtt{d}}_{\mathtt{D},R}(w, y) = R^{-1}\hat{x}_R$.

Algorithm 11: Decoder $Q^{\mathtt{d}}_{\mathtt{D},R}(x)$ for RDAQ

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 4.2.** *Let $Q_{\mathtt{D},R}$ be RDAQ described in Alg. 10 and 11. Then, for $\mathcal{X} = \mathcal{Y} = \mathcal{B}$ and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{D},R}; \Delta) \le 16\sqrt{3}\Delta \quad \textit{and} \quad \beta(Q_{\mathtt{D},R}; \Delta) = 0.$$

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

## 4.4 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 $\boldsymbol{\Delta}$ 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. 10 and 11 with $S$ replacing $[d]$ and multiplying $w'$ in Step 4 of Alg. 11 by normalization factor of $d/|S|$. The output of the resulting encoder is given by

$$Q^{\mathtt{e}}_{\mathtt{WZ},u}(x) = \{Q^{\mathtt{e}}_{\mathtt{D},R}(x)(i) : i \in S\}, \tag{13}$$

where $Q^{\mathtt{e}}_{\mathtt{D},R}(x)(i)$ represents the encoded bits $([\tilde{x}_0(i), \ldots, \tilde{x}_{h-1}(i)], z(i))$ for the $i$th coordinate using RDAQ, and the output of the resulting decoder is given by

$$Q_{\mathtt{WZ},u}(x, y) = R^{-1}\left[\frac{1}{\mu}\sum_{i\in S} 2M_{z^*(i)}\left(\mathbb{1}_{\{U(i,z^*(i)) \le Rx(i)\}} - \mathbb{1}_{\{U(i,z^*(i)) \le Ry(i)\}}\right) + Ry\right]. \tag{14}$$

**Lemma 4.3.** *Let $Q_{\mathtt{WZ},u}$ be the quantizers described in (13) and (14) and $Q_{\mathtt{D},R}$ be RDAQ described in Alg. 10 and 11. Then, for $\mu d \in [d]$, $\mathcal{X} = \mathcal{Y} = \mathcal{B}$, and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{WZ},u}; \Delta) \le \frac{\alpha(Q_{\mathtt{D},R}; \Delta)}{\mu} \quad \textit{and} \quad \beta(Q_{\mathtt{WZ},u}; \Delta) = 0.$$

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

We are now equipped to prove our second main result. Our protocol $\pi^*_{\mathtt{u}}$ uses $Q_{\mathtt{WZ},u}$ for each client as described in Section 2 and forms the estimate $\hat{\bar{x}}$ as in (4). Unlike for the known $\mathbf{\Delta}$ setting, we now use the same parameters for $Q_{\mathtt{WZ},u}$ for all clients, given by

$$\mu d = \left\lfloor \frac{r}{h + \log h} \right\rfloor . \tag{15}$$

**Theorem 4.4.** *For $d \geq r \geq 2(h + \log h)$ and $h$ given in (12), the $r$-bit protocol $\pi^*_u$ with parameters as set in (15) satisfies*

$$\mathcal{E}(\pi^*_u, \mathbf{x}, \mathbf{y}) \leq (128\sqrt{3}\,(1 + \ln^*(d/6))) \left( \sum_{i\in[n]} \frac{\Delta_i}{n} \cdot \frac{d}{nr} \right),$$

*for all $\mathbf{x}, \mathbf{y}$ satisfying (2), for every $\mathbf{\Delta} = (\Delta_1, ..., \Delta_n)$.*

*Proof.* Denote by $\hat{\bar{x}}$ the output of the protocol. Then, by Lemmas 2.1 and Lemma 4.3, we get

$$\begin{aligned}\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\leq \frac{1}{n^2\mu} \sum_{i=1}^{n} \alpha(Q_{\mathtt{D},R}; \Delta_i) \\ &\leq \frac{16\sqrt{3}}{n^2\mu} \sum_{i=1}^{n} \Delta_i,\end{aligned}$$

where the previous inequality is by Lemma 4.2. The proof is completed by using $\mu \geq \frac{r}{2d(h+\log h)} \geq \frac{r}{4dh}$, which follows from (15) and the assumption that $r \geq 2(h + \log h)$. ☐

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

# 5 The high-precision regime

## 5.1 RMQ in the high-precision regime.

For the known $\mathbf{\Delta}$ 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

$$\delta = \frac{\Delta_i}{n^{\frac{1}{2}}(2^{r/d} - 2)}, \quad \log k = \frac{r}{d}, \quad \Delta' = \sqrt{6(\Delta_i^2/d)\ln \Delta_i/\delta}, \quad \varepsilon = \frac{2\Delta'}{k-2}. \tag{11}$$

The performance of protocol $\pi^*_k$ using RMQ with parameters set as in (11) for each client can be characterized as follows.

**Theorem 5.1.** *For a fixed $\mathbf{\Delta} = (\Delta_1, ..., \Delta_n)$ and $r = md$ for integer $m \geq 2$, the protocol $\pi^*_k$ with parameters set as in (11) satisfies*

$$\mathcal{E}(\pi^*_k, \mathbf{x}, \mathbf{y}) = \left(12 \ln n + \frac{24r}{d} + 154/n + 166\right) \left( \sum_{i\in[n]} \frac{\Delta_i^2}{n} \cdot \frac{1}{n(2^{r/d} - 2)^2} \right),$$

*for all* $\mathbf{x}, \mathbf{y}$ *satisfying* (2).

*Proof.* Denoting by $Q_i$ the quantizer $Q_{\mathtt{M},R}$ with parameters set for client $i$, by Lemmas 2.1 and 3.2, we get

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

Further, since $k \geq 4$ holds when $r \geq 2d$ for our choice of parameters, by using Lemma 3.2 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

$$\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] \leq \left(12 \ln n + \frac{24r}{d} + \frac{154}{n} + 154\right) \sum_{i=1}^{n} \frac{\Delta_i^2}{n^2(2^{r/d} - 2)^2},$$

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.

$\square$

*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

$$O\left(\sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d}{n 2^{2r/d}}\right),$$

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.

### 5.2 Boosted RDAQ: RDAQ in the high-precision regime.

Moving to the unknown $\boldsymbol{\Delta}$ 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 straight-forward 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 Mayekar and Tyagi (2020a) for the case without any side information. At the encoding stage of RDAG given in Alg. 10 and 11, after random rotation and computing $z$ 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 $i$ and scale $j$. Specifically, let $U_t(i,j)$ 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 $\tilde{x}_{j,t} = \sum_{i=1}^{d} \mathbb{1}_{\{U_t(i,j) \le x_R(i)\}} e_i$ for all $j \in [h]_0$. 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 $\log N$ bits, the new encoder's output can be stored in $d(h \log N + \log h)$. Thus, we can implement this scheme by using $m = (h \log N + \log h)$ 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 $U_t(i,j)$ and random matrix $R$ used by the encoder, the final estimate $Q(x)$ is

$$Q(x) = R^{-1} \left( \frac{1}{N} \cdot \sum_{i \in [d]} \sum_{t \in [N]} \left( B^t_{i,Rx} - B^t_{i,Ry} \right) e_i + Ry \right), \tag{12}$$

where $B^t_{i,v} = \mathbb{1}_{\{U_n(i,z^*(i)) \le v(i)\}}$ for $v$ in $\mathbb{R}^d$.

The result below characterizes the performance of our quantizer Boosted RDAQ $Q$.

**Lemma 5.2.** *Let $Q$ be Boosted RDAQ described above. Then, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and every $\Delta > 0$, we have*

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

*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 $N = 2^{\lfloor (m - \log h)/h \rfloor}$, 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

$$N = 2^{\lfloor (m - \log h)/h \rfloor}, \; M_j^2 = \frac{6e^{*j}}{d}, j \in [h]_0, \quad \log h = \lceil \log(1 + \ln^*(d/6)) \rceil. \tag{13}$$

Therefore, by the previous lemma and Lemma 2.1, we get the following result.

**Theorem 5.3.** *For $r = dm$ with integer $m \ge h + \log h$, the protocol $\pi_u^*$ with parameters as set in (13) satisfies*

$$\mathcal{E}(\pi_u^*, \mathbf{x}, \mathbf{y}) = \sum_{i \in [n]} \frac{\Delta_i}{n} \cdot \frac{64\sqrt{3}}{n 2^{r/(d(2 + 2\ln^*(d/6)))}},$$

*for all $\mathbf{x}, \mathbf{y}$ satisfying (2), for every $\boldsymbol{\Delta} = (\Delta_1, ..., \Delta_n)$.*

*Proof.* Denote by $\hat{\bar{x}}$ the output of the protocol. Then, by Lemmas 2.1 and Lemma 5.2, we get

$$\begin{aligned} \mathbb{E}\left[ \|\hat{\bar{x}} - \bar{x}\|_2^2 \right] &\le \frac{1}{n^2} \sum_{i=1}^{n} \alpha(Q; \Delta_i) \\ &\le \frac{16\sqrt{3}}{n^2 N} \sum_{i=1}^{n} \Delta_i, \end{aligned}$$

where the previous inequality is by Lemma 5.2. The proof is completed by using

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

where the first inequality follows from using $\lfloor x \rfloor \geq x - 1$ for the floor function in the value of $N$ in (13), the second follows from the fact that $\log x \leq x, \forall x \geq 0$, and the third follows from $\lceil x \rceil \leq x + 1$ for the ceil function in the value of $h$ in (13). □

# 6 The Gaussian Wyner-Ziv problem

Consider the random vectors $X$ and $Y$, 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 $\sigma_y^2$ and $\sigma_z^2$, 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 $\mathbb{E}\left[\|Q_d(X,Y) - X\|_2^2\right] \leq dD$. 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[11] Wyner and Ziv (1976), $R(D)$ can be characterized as follow:

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma_z^2}{D} & \text{if} \quad D \leq \sigma_z^2, \\ 0 & \text{if} \quad D > \sigma_z^2. \end{cases}$$

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 will 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 $\sigma_z^2$. 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:

$$\delta = \sqrt{D/308}, \quad \log k = \left\lceil \log\left(2 + \sqrt{\frac{24\sigma_z^2}{D}\ln\frac{308\sigma_z^2}{D}}\right)\right\rceil$$
$$\Delta' = \sqrt{6(\sigma_z^2)\ln(\sigma_z/\delta)}, \quad \varepsilon = 2\Delta'/(k-2), \tag{14}$$

**Theorem 6.1.** *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 $\sigma_z^2$, for all coordinates $i \in \{1, \ldots, d\}$. Then, for $D \leq (\sigma_z^2/308)$, the quantizer $Q_d(X, Y)$ described above has MSE less than $dD$ and has rate $R$ satisfying*

$$R \leq \frac{1}{2}\log\frac{\sigma_z^2}{D} + O\left(\log\log\frac{\sigma_z^2}{D}\right).$$

[11] The model considered in Wyner and Ziv (1976) and perhaps the more popular Wyner-Ziv model is $Y = X + Z$. Nevertheless, through MMSE rescaling this model can be converted to $X = Y' + Z'$ (see, for instance, Liu (2016)).

# 7 Proofs of results

## 7.1 Proof of Lemma 2.1

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

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{n}\cdot\sum_{i\in[n]} Q_i(x_i,y_i) - \frac{1}{n}\cdot\sum_{i\in[n]} x_i\right\|_2^2\right] \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{1}{n^2}\cdot\sum_{i\neq j}\mathbb{E}\left[\langle Q_i(x_i,y_i)-x_i, Q_j(x_j,y_j)-x_j\rangle\right] \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{1}{n^2}\cdot\sum_{i\neq j}\langle\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i, \mathbb{E}\left[Q_j(x_j,y_j)\right]-x_j\rangle \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \left(\frac{1}{n}\cdot\sum_{i}\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2\right)^2 \\
&\qquad\qquad - \frac{1}{n^2}\cdot\sum_i\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2^2 \\
&\leq \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{(n-1)}{n^2}\cdot\sum_i\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2^2,
\end{aligned}$$

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

## 7.2 Proof of Lemma 3.1

As mentioned in (5), the integer $\tilde{z}$ found in Alg. 2 satisfies $\mathbb{E}[\tilde{z}\varepsilon] = x$ and $|x - \tilde{z}\varepsilon| < \varepsilon$. Therefore, it suffices to show that the output of the quantizer satisfies $Q_{\mathtt{M}}(x, y) = \tilde{z}\varepsilon$.

To see that $Q_{\mathtt{M}}(x, y) = \tilde{z}\varepsilon$, denote the lattice used in decoding Alg. 3 as $\mathbb{Z}_{w,\varepsilon} := \{(zk+w)\cdot\varepsilon : z \in \mathbb{Z}\}$. The decoding algorithm finds the point in $\mathbb{Z}_{w,\varepsilon}$ that is closest to $y$. Note that $w = \tilde{z} \mod k$, whereby $\tilde{z}\varepsilon$ is a point in this lattice. Further, for any other point $\lambda \neq \tilde{z}\varepsilon$ in the lattice, we must have

$$|\lambda - \tilde{z}\varepsilon| \geq k\varepsilon,$$

and so, by triangular inequality, that

$$|\lambda - y| \geq |\lambda - \tilde{z}\varepsilon| - |\tilde{z}\varepsilon - y| \geq k\varepsilon - |\tilde{z}\varepsilon - y|.$$

Thus, $\tilde{z}\varepsilon$ is closer to $y$ than $\lambda$ if

$$k\varepsilon > 2|\tilde{z}\varepsilon - y|. \tag{15}$$

Next, by using (5) once again, we have

$$|\tilde{z}\varepsilon - y| \leq |\tilde{z}\varepsilon - x| + |x - y| < \varepsilon + \Delta',$$

which by condition (7) in the lemma implies that (15) holds. It follows that $|\lambda - y| > |\tilde{z}\varepsilon - y|$ for every $\lambda \in \mathbb{Z}_{w,\varepsilon}$, which shows that $Q_{\mathtt{M}}(x, y) = \tilde{z}\varepsilon$ and completes the proof. $\square$

### 7.3 Proof of Lemma 3.2

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 $\|z\|_2^2/d$. Furthermore, we need the following bound for "truncated moments" of sub-Gaussian random variables.

**Lemma 7.1.** *For a sub-Gaussian random $Z$ with variance factor $\sigma^2$ and every $t \geq 0$, we have*

$$\mathbb{E}\left[Z^2 \mathbb{1}_{\{|Z|>t\}}\right] \leq 2(2\sigma^2 + t^2)e^{-t^2/2\sigma^2}.$$

*Proof.* Note that for any nonnegative random variable $U$, it can be verified that

$$\mathbb{E}\left[U \mathbb{1}_{\{U>x\}}\right] = xP(U > x) + \int_x^\infty P(U > u)\, du.$$

Upon substituting $U = Z^2$ and $x = t^2$, along with the fact that $Z$ is sub-Gaussian with variance parameter $\sigma^2$, we get

$$\begin{aligned}
\mathbb{E}\left[Z^2 \mathbb{1}_{\{Z^2>t^2\}}\right] &= t^2 P(Z^2 > t^2) + \int_{t^2}^\infty P(Z^2 > u)\, du \\
&\leq 2t^2 e^{-t^2/2\sigma^2} + 2\int_{t^2}^\infty e^{-u/2\sigma^2}\, du \\
&\leq 2(t^2 + 2\sigma^2)e^{-t^2/2\sigma^2},
\end{aligned}$$

which completes the proof. $\square$

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

**Bound for MSE $\alpha(Q)$:** Denote by $Q_{\mathtt{M},R}(x, y)$ the final quantized value of the quantizer RMQ. For convenience, we abbreviate

$$\hat{x}_R := R\, Q_{\mathtt{M},R}(x, y).$$

Observe that $\hat{x}_R = \sum_{i\in[d]} Q_{\mathtt{M}}(Rx(i), Ry(i))e_i$, where $Q_{\mathtt{M}}$ is the MQ of Alg. 2 and 3 with parameters $k \geq$ and $\Delta'$ set as in the statement of the lemma. Since $R$ is a unitary transform, we have

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{M},R}(x, y) - x\|_2^2\right] &= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\leq\Delta'\}}\right] \\
&\qquad + \sum_{i=1}^d \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq\Delta'\}}\right] \qquad (16)
\end{aligned}$$

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 $\{|R\,(x-y)\,(i)| \le \Delta'\}$, we get by Lemma 3.1 that

$$|\hat{x}_R(i) - Rx(i)| \le \varepsilon = \frac{2\Delta'}{k-2}, \quad \text{almost surely},$$

whereby

$$\sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{|R(x-y)(i)| \le \Delta'}\right] \le d\,\varepsilon^2. \tag{17}$$

The second term on the right-side of (16) corresponds to the error due to "overflow" and is handled using concentration bounds for the rotated vectors. Specifically, we get

$$\begin{aligned}
&\sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)| \ge \Delta'\}}\right] \\
&\le 2\sum_{i=1}^{d} \left[\mathbb{E}\left[(\hat{x}_R(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)| \ge \Delta'\}}\right] + \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)| \ge \Delta'\}}\right]\right] \\
&\le 2k^2\varepsilon^2 \sum_{i=1}^{d} P(|R\,(x-y)\,(i)| \ge \Delta') + 2\sum_{i=1}^{d} \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)| \ge \Delta'\}}\right] \\
&\le 4dk^2\varepsilon^2 e^{-d\Delta'^2/2\Delta^2} + 2\sum_{i=1}^{d} \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)| \ge \Delta'\}}\right] \\
&\le 4dk^2\varepsilon^2 e^{-d\Delta'^2/2\Delta^2} + 4(2\Delta^2 + d\Delta'^2) e^{-\frac{d\Delta'^2}{2\Delta^2}},
\end{aligned} \tag{18}$$

where the second inequality follows upon noting that from the description decoder of MQ in Alg. 3 that $|\hat{x}_R(i) - Ry(i)| \le \varepsilon k$ almost surely for each $i \in [d]$; the third inequality uses the fact that $R(x-y)(i)$ is sub-Gaussian with variance parameter $\|x-y\|_2^2/d \le \Delta^2/d$; and fourth inequality is by Lemma 7.1.

Upon combining (16), (17), and (18), and substituting $\varepsilon = 2\Delta'/(k-2)$ and ${\Delta'}^2 = 6(\Delta^2/d)\log \Delta/\delta$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{M},R}(x,y) - x\|_2^2\right] &\le d\,\varepsilon^2 + 4dk^2\varepsilon^2 e^{-\frac{d\Delta'^2}{2\Delta^2}} + 4(2\Delta^2 + d\Delta'^2) e^{-\frac{d\Delta'^2}{2\Delta^2}} \\
&= 24\,\frac{\Delta^2}{(k-2)^2} \ln\frac{\Delta}{\delta} + 96\delta^2 \left(\frac{k}{k-2}\right)^2 \cdot \frac{\ln(\Delta/\delta)}{(\Delta/\delta)} + 8\delta^2 \cdot \frac{1 + 3\ln(\Delta/\delta)}{(\Delta/\delta)} \\
&\le 24\,\frac{\Delta^2}{(k-2)^2} \ln\frac{\Delta}{\delta} + \left(\frac{96}{e}\left(\frac{k}{k-2}\right)^2 + \frac{24}{e^{2/3}}\right) \cdot \delta^2,
\end{aligned} \tag{19}$$

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

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

**Bias $\beta(Q)$:** The calculation for the bias is similar to that we used to bound the second term on the right-side of (16). Using the notation $\hat{x}_R$ introduced above, we have

$$\begin{aligned}
&\|\mathbb{E}\left[Q_{\mathtt{M},R}\right] - x\|_2 \\
&= \|\mathbb{E}\left[R^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|R\mathbb{E}\left[R^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|\mathbb{E}\left[RR^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|\mathbb{E}\left[\hat{x}_R - Rx\right]\|_2,
\end{aligned}$$

where the second identity holds since $R$ is a unitary matrix.

Further, since $Q_{\mathtt{M}}(x, y)$ is an unbiased estimate of $x$ when $|x - y| \le \Delta'$ (see Lemma 3.1), by (17) and (18) we obtain

$$\begin{aligned}
\|\mathbb{E}\left[\hat{x}_R - Rx\right]\|_2^2 &\le \sum_{i=1}^{d} \mathbb{E}\left[\left(\hat{x}_R(i) - Rx(i)\right)\mathbb{1}_{|R(x-y)_i| \ge \Delta')}\right]^2 \\
&\le \sum_{i=1}^{d} \mathbb{E}\left[\left(\hat{x}_R(i) - Rx(i)\right)^2 \mathbb{1}_{|R(x-y)(i)| \ge \Delta')}\right] \\
&\le 154\,\delta^2,
\end{aligned}$$

which completes the proof. □

## 7.4 Proof of Lemma 3.3

**Mean Square Error $\alpha(Q_{S,R})$:** From the description of Algorithms 6 and 7, we know that the quantized output of subsampled RMQ $Q_{\mathtt{WZ}}$ for an input $x$ is

$$\begin{aligned}
&Q_{\mathtt{WZ}}(x) = R^{-1}\hat{x}_R, \text{ where} \\
&\hat{x}_R = \frac{1}{\mu}\sum_{i\in[d]}\left(Q_{\mathtt{M}}(Rx(i), Ry(i)) - Ry(i)\right)\mathbb{1}_{\{i\in S\}}\, e_i + Ry,
\end{aligned}$$

and $Q_{\mathtt{M}}(Rx(i), Ry(i))$ 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 $Q_{\mathtt{M}}(Rx(i), Ry(i))$, we have

$$\begin{aligned}
&\mathbb{E}\left[\|Q_{\mathtt{WZ}}(x) - x\|_2^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[\left(\frac{1}{\mu}\left(Q(Rx(i)) - Ry(i)\right)\mathbb{1}_{\{i\in S\}} - \left(Rx(i) - Ry(i)\right)\right)^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[\frac{1}{\mu^2}Q(Rx(i)) - Rx(i)^2\mathbb{1}_{\{i\in S\}}\right] \\
&\qquad + \sum_{i\in[d]} \mathbb{E}\left[\left(\frac{1}{\mu}\left(Rx(i) - Ry(i)\right)\mathbb{1}_{\{i\in S\}} - \left(Rx(i) - Ry(i)\right)\right)^2\right] \\
&= \sum_{i\in[d]} \frac{1}{\mu}\mathbb{E}\left[\left(Q(Rx(i)) - Rx(i)\right)^2\right] + \sum_{i\in[d]} \mathbb{E}\left[\left(Rx(i) - Ry(i)\right)^2\right] \cdot \mathbb{E}\left[\left(\frac{1}{\mu}\mathbb{1}_{\{i\in S\}} - 1\right)^2\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in[d]} \frac{1}{\mu}\mathbb{E}\left[(Q(Rx(i)) - Rx(i))^2\right] + \sum_{i\in[d]} \mathbb{E}\left[(Rx(i) - Ry(i))^2\right] \cdot \frac{1-\mu}{\mu} \\
&\leq \frac{\alpha(Q_{\mathtt{M},R})}{\mu} + \frac{\Delta^2}{\mu},
\end{aligned}
$$

where we used the independence of $S$ and $R$ in the third identity and used the fact that $R$ is unitary in the final step.

**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 $R^{-1}\sum_{i\in[d]} Q_{\mathtt{M}}(Rx(i), Ry(i))e_i$, which, in turn, is equivalent in distribution to the output of RMQ. $\square$

## 7.5 Proof of Theorem 3.5

We denote $\Delta_{min} = \min_{i\in[d]} \Delta_i$ and set $y_i$s to be 0. Let $x_1, ..., x_n$ 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 $x_i$s, which provides a lower bound for the MSE of any estimator of the mean of the distribution of $x_i$s.

**Lemma 7.2.** *For $x_1, ..., x_n$ generated as above and any estimator $\hat{\bar{x}}$ of the mean formed using only $r$-bit quantized version of $x_i$s, we have*[12]

$$
\mathbb{E}\left[\left\|\hat{\bar{x}} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\right\|_2^2\right] \geq c' \cdot \frac{d\Delta_{min}^2}{nr},
$$

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

Proof of Lemma 7.2 follows from either (Duchi et al., 2014, Proposition 2) or (Acharya et al., 2020, Theorem 11).

The proof of Theorem 3.5 is completed by using this claim. Specifically, using $2a^2+2b^2 \geq (a+b)^2$, we have

$$
2\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] + 2\mathbb{E}\left[\|\bar{x} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right] \geq \mathbb{E}\left[\|\hat{\bar{x}} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right],
$$

which, along with the observation that

$$
\mathbb{E}\left[\|\bar{x} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right] \leq \frac{\Delta_{min}^2}{n},
$$

gives

$$
\begin{aligned}
\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\geq \frac{c' d\Delta_{min}^2}{2nr} - \frac{\Delta_{min}^2}{n} \\
&\geq \frac{c'\Delta_{min}^2 d}{4nr},
\end{aligned}
$$

[12] Note that the side information $y_i$s are all set to 0.

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

*Remark* 6. Since the lower bound in Acharya et al. (2020) 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.

## 7.6 Proof of Lemma 4.1

We will prove a general result which will not only prove Lemma 4.1 but will also be useful in the proof of Lemma 4.2. 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:

$$Q_{\mathtt{D}}(x, y) = \sum_{i=1}^{d} 2M \left(\mathbb{1}_{\{U(i) \leq x(i)\}} - \mathbb{1}_{\{U(i) \leq y(i)\}}\right) e_i + y,$$

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

$$\mathbb{E}\left[Q_{\mathtt{D}}(x, y)\right] = x \quad \text{and} \quad \mathbb{E}\left[\|Q_{\mathtt{D}}(x, y) - x\|_2^2\right] \leq 2M\|x - y\|_1, \tag{20}$$

which upon setting $M = 1$ proves Lemma 4.1.

Towards proving (20), note that from the estimate formed by $Q_{\mathtt{D}}$, it is easy to see that $\mathbb{E}\left[Q_{\mathtt{D}}(x, y)\right] = x$. The MSE can be bounded as follows:

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D}}(x, y) - x\|_2^2\right] &= \sum_{i=1}^{d} \mathbb{E}\left[\left(2M\left(\mathbb{1}_{\{U_i \leq x(i)\}} - \mathbb{1}_{\{U_i \leq y(i)\}}\right) - (x(i) - y(i))\right)^2\right] \\
&= \sum_{i=1}^{d} 4M^2 \frac{|x(i) - y(i)|}{2M} - \|x - y\|_2^2 \\
&= 2M\|x - y\|_1 - \|x - y\|_2^2,
\end{aligned}$$

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

## 7.7 Proof of Lemma 4.2

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

**Worst-case MSE $\alpha(Q_{\mathtt{D},R}; \Delta)$:** We denote by $B_{ij}^x$ and $B_{ij}^y$ the bits

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

Then, the final quantized value of the quantizer RDAQ can be expressed as $Q_{\mathtt{D},R}(X) = R^{-1}\hat{x}_R$ where, with $z^*(i)$ denoting the smallest $M_j$ such that the interval $[-M_j, M_j]$ contains $Rx(i)$ and $Ry(i)$ and $[h]_0 = \{0, \ldots, h-1\}$,

$$\hat{x}_R := \sum_{i \in \{1,\ldots,d\}} \left( \sum_{j \in [h]_0} 2M_j \cdot \left(B_{ij}^x - B_{ij}^y\right) + Ry(i) \right) \mathbb{1}_{\{z^*(i)=j\}} e_i.$$

Since $R$ is a unitary transform, we get

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x) - x\|_2^2\right] &= \mathbb{E}\left[\|RQ_{\mathtt{D},R}(x) - Rx\|_2^2\right] \\
&= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i \in [d]} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i \in [d]} \mathbb{E}\left[\left( \sum_{j \in [h]_0} (2M_j \cdot \left(B_{ij}^x - B_{ij}^y\right) + Ry(i) - Rx(i)) \mathbb{1}_{\{z^*(i)=j\}} \right)^2\right] \\
&= \sum_{i \in [d]} \sum_{j \in [h]_0} \mathbb{E}\left[\left(2M_j \left(B_{ij}^x - B_{ij}^y\right) + Ry(i) - Rx(i)\right)^2 \mathbb{1}_{\{z^*(i)=j\}},\right]
\end{aligned}$$

where the last identity uses $\mathbb{1}_{\{z^*(i)=j_1\}}\mathbb{1}_{\{z^*(i)=j_2\}} = 0$ for all $j_1 \neq j_2$, to cancel the cross-terms in the expansion of $(\hat{x}_R(i) - Rx(i))^2$. Conditioning on $R$ and using the independence of $\mathbb{1}_{\{z^*(i)=j\}}$ from the randomness used in MQ, we get

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x) - x\|_2^2\right] &= \sum_{i \in [d]} \sum_{j \in [h]_0} \mathbb{E}\left[\mathbb{E}\left[\left(2M_j \left(B_{ij}^x - B_{ij}^y\right) + Ry(i) - Rx(i)\right)^2 \mid R\right] \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&\leq \sum_{i \in [d]} \sum_{j \in [h]_0} \mathbb{E}\left[2M_j |Rx(i) - Ry(i)| \mathbb{1}_{\{z^*(i)=j\}}\right], \\
&\leq \sum_{i \in [d]} \mathbb{E}\left[2M_0 |Rx(i) - Ry(i)| \mathbb{1}_{\{z^*(i)=0\}}\right] \\
&\quad + \sum_{i \in [d]} \sum_{j \in [h-1]} \mathbb{E}\left[2M_j |Rx(i) - Ry(i)| \mathbb{1}_{\{z^*(i)=j\}}\right], \\
&\leq \sum_{i \in [d]} \mathbb{E}\left[2M_0 |Rx(i) - Ry(i)|\right] \\
&\quad + \sum_{i \in [d]} \sum_{j \in [h-1]} \mathbb{E}\left[2M_j |Rx(i) - Ry(i)| \mathbb{1}_{\{z^*(i)=j\}}\right],
\end{aligned} \tag{21}$$

where the first inequality follows from (20) in the proof of Lemma 4.1.

Next, noting that

$$\mathbb{1}_{\{z^*(i)=j\}} \leq \mathbb{1}_{\{|RX(i)| \geq M_{j-1}\}} + \mathbb{1}_{\{|RY(i)| \geq M_{j-1}\}} \quad \text{almost surely},$$

an application of the Cauchy-Schwarz inequality yields

$$
\begin{aligned}
&\mathbb{E}\left[2M_j|Rx(i)-Ry(i)|\mathbb{1}_{\{z^*(i)=j\}}\right]\\
&\le 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}\mathbb{E}\left[(\mathbb{1}_{\{|RX(i)|\ge M_{j-1}\}}+\mathbb{1}_{\{|RY(i)|\ge M_{j-1}\}})^2\right]^{1/2}\\
&\le 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}\left(2P(|Rx(i)|\ge M_{j-1})+2P(|Ry(i)|\ge M_{j-1})\right)^{1/2}\\
&\le 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}\left(8e^{\frac{-dM_{j-1}^2}{2}}\right)^{1/2}, \qquad (22)
\end{aligned}
$$

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

Substituting the upper bound in (22) for the second term in the RHS of (21) and using $\mathbb{E}[X] \le \mathbb{E}\left[X^2\right]^{1/2}$ for the first term, we get

$$
\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x)-x\|_2^2\right] &\le \sum_{i\in[d]}\mathbb{E}\left[|Rx(i)-Ry(i)|^2\right]^{1/2}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right)\\
&\le \sqrt{d\cdot\mathbb{E}\left[\|Rx-Ry\|_2^2\right]}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right)\\
&= \sqrt{d\cdot\|x-y\|_2^2}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right)\\
&= \sqrt{d\cdot\|x-y\|_2^2}\left(2\sqrt{\frac{6}{d}}+\sum_{j\in[h-1]}2\sqrt{\frac{6e^{*j}}{d}}\cdot\left(8e^{-1.5e^{*(j-1)}}\right)\right)\\
&= 8\sqrt{3}\cdot\sqrt{\|x-y\|_2^2}\left(1+\sum_{j\in[h-1]}e^{-0.5e^{*}(j-1)}\right)\\
&\le 16\sqrt{3}\cdot\sqrt{\|x-y\|_2^2},
\end{aligned}
$$

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

$$
\begin{aligned}
e^{-0.5e^{*}(j-1)} &= e^{-0.5}+e^{-0.5e}+e^{-0.5e^e}+\sum_{j=3}^{\infty}e^{-0.5e^{*(j)}}\\
&\le e^{-0.5}+e^{-0.5e}+e^{-0.5e^e}+\sum_{j=3}^{\infty}e^{-0.5je^e}\\
&\le e^{-0.5}+e^{-0.5e}+e^{-0.5e^e}+\frac{1}{e^{e^e}-1}\\
&\le 1.
\end{aligned}
$$

□

## 7.8 Proof of Lemma 4.3

**Worst-case bias $\beta(Q_{\texttt{WZ},u};\Delta)$:** It is straightforward to see that $\mathbb{E}\left[Q_{\texttt{WZ},u}(x)\right] = x$.

**Worst-case MSE $\alpha(Q_{\texttt{WZ},u};\Delta)$:** We denote by $B^x_{ij}$ and $B^y_{ij}$ the bits

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

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

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

Since $R$ is a unitary transform, the mean square error between $Q_{\texttt{WZ},u}(x)$ and $x$ can be bounded as in the proof of Lemma 4.2 as follows:

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\texttt{WZ},u}(x) - x\|_2^2\right] &= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\left(2M_j\left(B^x_{ij} - B^y_{ij}\right)\mathbb{1}_{\{i\in S\}} + Ry(i) - Rx(i)\right)^2 \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&= \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\mathbb{E}\left[\left(2M_j\left(B^x_{ij} - B^y_{ij}\right)\mathbb{1}_{\{i\in S\}} + Ry(i) - Rx(i)\right)^2 \mid R\right] \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&\leq \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\frac{2M_j}{\mu} \cdot |Rx(i) - Ry(i)| \cdot \mathbb{1}_{\{z^*(i)=j\}}\right],
\end{aligned}$$

where the inequality follows from similar calculations in the proof of Lemma 4.1. The rest of the analysis proceeds as that in the proof of Lemma 4.2. □

## 7.9 Proof of Lemma 5.2

For $Q(x)$ as in (12), we have

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

where $q_i$ for all $i \in \{1, \dots N\}$ 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,

$$\begin{aligned}
\mathbb{E}\left[\|Q(x) - x\|_2^2\right] &= \mathbb{E}\left[\|\sum_{i=1}^{N} \frac{q_i}{N} - x\|_2^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\|\sum_{i=1}^{N} \frac{q_i}{N} - x\|_2^2 | R\right]\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} \frac{1}{N^2}\mathbb{E}\left[\|q_i - x\|_2^2 | R\right]\right] \\
&\le 16\sqrt{3}\,\frac{\Delta}{N},
\end{aligned}$$

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 4.2. □

## 7.10 Proof of Theorem 6.1

The proof of this Theorem is similar to that of Lemma 3.2. 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 (14). Then, we have

$$\begin{aligned}
\mathbb{E}\left[\|Q_d(X, Y) - X\|^2\right] &\le \sum_{i=1}^{d} \mathbb{E}\left[(Q(X(i), Y(i)) - X(i))^2\right] \\
&\le \sum_{i=1}^{d} \mathbb{E}\left[(Q(X(i), Y(i)) - X(i))^2 \mathbb{1}_{\{|(X(i)-Y(i))| \le \Delta'\}}\right] \\
&\qquad + \sum_{i=1}^{d} \mathbb{E}\left[(Q(X(i), Y(i)) - X(i))^2 \mathbb{1}_{\{|(X(i)-Y(i))| \ge \Delta'\}}\right]. \qquad (23)
\end{aligned}$$

We bound the first term on the right-side in a similar manner as the bound in (17). Specifically, under the event $\{|X(i) - Y(i)| \le \Delta'\}$, we get by Lemma 3.1 that

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

whereby

$$\sum_{i=1}^{d} \mathbb{E}\left[(Y(i) - X(i))^2 \mathbb{1}_{\{|X(i)-Y(i)\}| \le \Delta'}\right] \le d\,\varepsilon^2. \qquad (24)$$

For the second term in the RHS note that $X(i) - Y(i)$ is subgaussian with variance factor $\sigma_z^2$. Therefore, by proceeding in a similar manner as the derivation of (18) we get

$$\begin{aligned}
&\sum_{i=1}^{d} \mathbb{E}\left[(Q(X(i), Y(i)) - X(i))^2 \mathbb{1}_{\{|X(i)-Y(i)| \geq \Delta'\}}\right] \\
&\leq 2 \sum_{i=1}^{d} \left[\mathbb{E}\left[(Q(X(i), Y(i)) - Y(i))^2 \mathbb{1}_{\{|X(i)-Y(i)| \geq \Delta'\}}\right] + \mathbb{E}\left[(Y(i) - X(i))^2 \mathbb{1}_{\{|X(i)-Y(i)| \geq \Delta'\}}\right]\right] \\
&\leq 2k^2\varepsilon^2 \sum_{i=1}^{d} P(|X(i) - Y(i)| \geq \Delta') + 2 \sum_{i=1}^{d} \mathbb{E}\left[(X(i) - Y(i))^2 \mathbb{1}_{\{|X(i)-Y(i)| \geq \Delta'\}}\right] \\
&\leq 4dk^2\varepsilon^2 e^{-d\Delta'^2/2\sigma_z^2} + 2 \sum_{i=1}^{d} \mathbb{E}\left[(X(i) - Y(i))^2 \mathbb{1}_{\{|X(i)-Y(i)| \geq \Delta'\}}\right] \\
&\leq 4dk^2\varepsilon^2 e^{-\Delta'^2/2\sigma_z^2} + 4(2\sigma_z^2 + d\Delta'^2) e^{-\frac{\Delta'^2}{2\sigma_z^2}}, \qquad (25)
\end{aligned}$$

where the second inequality follows upon noting from the description decoder of MQ in Alg. 3 that $|Q(X(i), Y(i)) - Y(i)| \leq \varepsilon k$ almost surely for each $i \in [d]$; the third inequality uses the fact that $X(i) - Y(i)$ is sub-Gaussian with variance parameter $\sigma_z^2$; and the fourth inequality is by Lemma 7.1.

Upon bounding the two terms on the right-side of (23) from above using (24), (25), we obtain

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

Note that the RHS in the upper bound above is precisely the same as in (19) with $\sigma_z^2$ replacing $\Delta^2/d$.Therefore proceeding in the same manner as in (19), we get

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

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

□

# Acknowledgement

Prathamesh Mayekar would like to thank Shubham Jha for discussions on the Gaussian Wyner-Ziv problem.

Prathamesh Mayekar is supported by a Ph.D. fellowship from Wipro Limited. This work is supported, in part, by a research grant from Robert Bosch Center for Cyber Physical Systems (RBCCPS), Indian Institute of Science, Bangalore.

### References

Acharya, J., Canonne, C. L., and Tyagi, H. (2020). General lower bounds for interactive high-dimensional estimation under information constraints. *arXiv preprint arXiv:2010.06562*.

Acharya, J., De Sa, C., Foster, D. J., and Sridharan, K. (2019). Distributed Learning with Sublinear Communication. *arXiv:1902.11259*.

Ailon, N. and Chazelle, B. (2006). Approximate nearest neighbors and the fast johnson-lindenstrauss transform. *Proceedings of the ACM symposium on Theory of computing (STOC'06)*, pages 557–563.

Albasyoni, A., Safaryan, M., Condat, L., and Richtárik, P. (2020). Optimal gradient compression for distributed and federated learning. *arXiv preprint arXiv:2010.03246*.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. (2017). QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, pages 1709–1720.

Basu, D., Data, D., Karakus, C., and Diggavi, S. (2019). Qsparse-local-SGD: Distributed SGD with Quantization, Sparsification, and Local Computations. *Advances in Neural Information Processing Systems*.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence.*Oxford university press.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357.

Chen, W.-N., Kairouz, P., and Özgür, A. (2020). Breaking the communication-privacy-accuracy trilemma. *arXiv preprint arXiv:2007.11707*.

Davies, P., Gurunathan, V., Moshrefi, N., Ashkboos, S., and Alistarh, D. (2020). Distributed variance reduction with optimal communication. *arXiv e-prints*, pages arXiv–2002.

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

Forney, G. D. (1988). Coset codes. i. introduction and geometrical classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151.

Gandikota, V., Kane, D., Maity, R. K., and Mazumdar, A. (2019). vqsgd: Vector quantized stochastic gradient descent. *arXiv preprint arXiv:1911.07971*.

Holenstein, T. (2009). Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(8):141–172.

Horadam, K. J. (2012). *Hadamard matrices and their applications.*Princeton university press.

Huang, Z., Yilei, W., Yi, K., et al. (2019). Optimal sparsity-sensitive bounds for distributed mean estimation. *Advances in Neural Information Processing Systems*, pages 6371–6381.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2019). Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

Kashin, B. (1977). Section of some finite-dimensional sets and classes of smooth functions (in russian) izv. *Acad. Nauk. SSSR*, 41:334–351.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Konečnỳ, J. and Richtárik, P. (2018). Randomized distributed mean estimation: Accuracy vs. communication. *Frontiers in Applied Mathematics and Statistics*, 4:62.

Korada, S. B. and Urbanke, R. L. (2010). Polar codes are optimal for lossy source coding. *IEEE Transactions on Information Theory*, 56(4):1751–1768.

Ling, C., Gao, S., and Belfiore, J. (2012). Wyner-ziv coding based on multidimensional nested lattices. *IEEE Transactions on Communications*, 60(5):1328–1335.

Liu, L. (2016). Polar codes and polar lattices for efficient communication and source quantization. *Ph.D. Thesis*.

Liu, L. and Ling, C. (2015). Polar lattices are good for lossy compression. *CoRR*, abs/1501.05683.

Lu, Y. and De Sa, C. (2020). Moniqua: Modulo quantized communication in decentralized sgd. *arXiv preprint arXiv:2002.11787*.

Lyubarskii, Y. and Vershynin, R. (2010). Uncertainty principles and vector quantization. *IEEE Transactions on Information Theory*, 56(7):3491–3501.

Mayekar, P. and Tyagi, H. (2020a). Limits on gradient compression for stochastic optimization. *Proceedings of the IEEE International Symposium of Information Theory (ISIT' 20)*.

Mayekar, P. and Tyagi, H. (2020b). RATQ: A universal fixed-length quantizer for stochastic optimization. *https://arxiv.org/abs/1908.08200*.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609.

Oohama, Y. (1997). Gaussian multiterminal source coding. *IEEE Transactions on Information Theory*, 43(6):1912–1923.

Pradhan, S. S. and Ramchandran, K. (2003). Distributed source coding using syndromes (discus): design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643.

Ramezani-Kebrya, A., Faghri, F., and Roy, D. M. (2019). Nuqsgd: Improved communication efficiency for data-parallel sgd via nonuniform quantization. *arXiv preprint arXiv:1908.06077*.

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

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. (2014). 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. *Fifteenth Annual Conference of the International Speech Communication Association*.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. (2018). Sparsified sgd with memory. *Advances in Neural Information Processing Systems 31*.

Strassen, V. et al. (1965). The existence of probability measures with given marginals. *The Annals of Mathematical Statistics*, 36(2):423–439.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. (2017). Distributed mean estimation with limited communication. *Proceedings of the International Conference on Machine Learning (ICML' 17)*, 70:3329–3337.

Vogels, T., Karimireddy, S. P., and Jaggi, M. (2019). Powersgd: Practical low-rank gradient compression for distributed optimization.

Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. (2018). Atomo: Communication-efficient learning via atomic sparsification. *Advances in Neural Information Processing Systems*, pages 9850–9861.

Wangni, J., Wang, J., Liu, J., and Zhang, T. (2018). Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. (2017). TernGrad: Ternary gradients to reduce communication in distributed deep learning. *Advances in Neural Information Processing Systems*, pages 1509–1519.

Wyner, A. and Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on information Theory*, 22(1):1–10.

Yu, B. (1997). Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer.

Zamir, R., Shamai, S., and Erez, U. (2002). Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory*, 48(6):1250–1276.