DoStoVoQ: Doubly Stochastic Voronoi Vector Quantization SGD for Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The growing size of models and datasets have made distributed implementation of stochastic gradient descent (SGD) an active field of research. However the high bandwidth cost of communicating gradient updates between nodes remains a bottleneck; lossy compression is a way to alleviate this problem. We propose a new *unbiased* Vector Quantizer (VQ), named StoVoQ, to perform gradient quantization. This approach relies on introducing randomness within the quantization process, that is based on the use of unitarily invariant random codebooks and on a straightforward bias compensation method. The distortion of StoVoQ significantly improves upon existing quantization algorithms. Next, we explain how to combine this quantization scheme within a Federated Learning framework for complex high-dimensional model (dimension $> 10^6$), introducing DoStoVoQ. We provide theoretical guarantees on the quadratic error and (absence of) bias of the compressor, that allow to leverage strong theoretical results of convergence, e.g., with heterogeneous workers or variance reduction. Finally, we show that training on convex and non-convex deep learning problems, our method leads to significant reduction of bandwidth use while preserving model accuracy.

1 Introduction

In this paper, we consider the Federated Learning framework, in which a potentially large number K of *workers* cooperate to solve the following problem:

$$\min_{\theta \in \mathbb{R}^D} \sum_{k=1}^{K} f_k(\theta), \tag{1}$$

where each function $f_k : \mathbb{R}^D \to \mathbb{R}$ represents the empirical risk on worker $k \in [K]$ (where $[K] = \{1, \dots, K\}$) and D is the ambient dimension of our problem. Each worker potentially holds a fraction of the data, and can share information with a central server, which progressively aggregates and updates the model accordingly [20, 19].

Stochastic gradient algorithms [32] are particularly well suited in the *large scale learning* setting [6, 7]. The methods can easily be adapted to the distributed (and more generally federated) learning framework; see [19] and the references therein. For synchronous distributed Stochastic Gradient Descent, at every iteration, given the current parameter θ_t, each worker computes an unbiased estimate $g_{k,t+1}(\theta_t)$ of the gradient of the local loss function f_k. The central server then aggregates those oracles and performs the update.

Communicating the gradients from the local workers to the central server is often a major bottleneck. The drastic increase both in the number of parameters and of workers over the last years, has made this problem even more acute. Alleviating the communication cost is one of the crucial challenges of

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

federated learning [19, Sec. 3.5]. A central idea to tackle this issue is *communication compression*, which consists in applying a lossy compression to the parameters or gradients to be transmitted. Since compression alters the message transmitted, the number of iterations required to reach a given accuracy may increase, therefore compression is of interest in situations where the communication gains are large relative to the increase of communication rounds. The design of new compression schemes (see among others [34, 2, 4, 5, 39]) and the adaptation of the learning algorithms to this setting (see e.g. [37, 1, 40, 38, 41, 25, 30, 13, 12, 23] and the references therein) are an extremely active field of research.

Our main contribution is to introduce a novel **unbiased vector quantization** procedure allowing to reach **high-compression rate**, with a **small computational** overhead. More precisely, our contributions are as follow: first, we introduce `StoVoQ`, a vector quantization algorithm based on unitarily invariant random codebooks to automatically obtain **directionally unbiased** gradient oracles, and introduce a scalar **correction function**, that makes compression operator **unbiased** for a very modest computational cost. We further provide theoretical guarantees on the distortion of the compressor. In summary, `StoVoQ` algorithm is based on the following points, that are developed in Section 2.

1. **Vector quantization** The input vector $x \in \mathbb{R}^d$ is mapped onto its nearest neighbor in a codebook $\mathcal{C}_M = \{c_i\}_{i=1}^M$.
2. **Random codebook.** A **new codebook** is sampled every time a new quantization operation is performed. The proposed approach is different from classical random VQ which typically uses a random codebook, but which is sampled once and then kept fixed.
3. **Bias removal.** By relying on unitarily invariant distribution for the codewords generation, the quantized value of each vector $x \in \mathbb{R}^d$ is **directionnally unbiased**. The bias only depends on the number and distributions of the random of codewords and on $\|x\|$. This key property allows to derive a simple way to remove the quantization bias.

Then, we describe how to use `StoVoQ` within the FL framework: this yields the algorithm `DoStoVoQ`. We prove that this process satisfies a strong assumption on the compression process, that allows to automatically derive fast convergence rates. In Section 3, we describe `DoStoVoQ`, i.e., how we solve the optimization problem (1) in dimension D.

4. **Splitting and renormalizing gradients.** First, we split each gradient to compress into *buckets* $(x_i)_{i=1,\dots,L}$ of dimension $\mathbb{R}^d$, to use `StoVoQ` for each bucket.
5. **Synchronisation of random sequences of codebooks.** We ensure that those codebooks are independent, at each step and between each machine, by generating a new codebook each time. To avoid any subsequent communication cost, we synchronously generate the codebooks on the central and local servers, by initially sharing random seeds.

Remark that point 1 was also used in Dai et al. [8]. Points 2 to 3 and 5 are novel ideas that have not been leveraged in the FL framework. Finally, we demonstrate the effectiveness of random codebook quantization for gradient compression by extensive experiments in Section 4 on standard benchmarks like ImageNet or CIFAR10.

2 `StoVoQ` algorithm

Several compression operators [39, 31, 10, 4, 8, 41, 42] have been introduced recently as bandwidth reduction for distributed learning became a major challenge. In this section, we first discuss the importance of unbiasedness of compression operators in Subsection 2.1. We then present the `StoVoQ` compression scheme in Subsection 2.2. Finally, we compare `StoVoQ` to competing approaches, both theoretically and empirically on a small scale example with a high compression rate.

2.1 Unbiased gradient estimate to mitigate high compression rates

We here discuss an important property to mitigate high compression rates in FL settings. A *compression operator* Comp is a (random) mapping on $\mathbb{R}^d$. Consider the following assumption:

A1 (**Unbiased Compression with relatively bounded variance**). *A compression operator* Comp *is unbiased if for any* $x \in \mathbb{R}^d$, $\mathbb{E}[\mathrm{Comp}(x)] = x$. *It is said to have a* ω*-bounded relative variance, for some* $\omega > 0$, *if it satisfies, for all* $x \in \mathbb{R}^d$, $\mathbb{E}[\|\mathrm{Comp}(x) - x\|^2] \leq \omega \|x\|^2$.

The most classical compressors, especially Q-SGD and Rand-H satisfy A 1 with different ω, see Subsection 2.3 and Table 1. On the other hand, some compression operators are biased, i.e., $\mathbb{E}[\mathrm{Comp}(x)] \neq x$ for some $x \in \mathbb{R}$. Those operators are often deterministic, as is the case for Top-H compressor. The most classical assumption for biased operators, is the following contractive property along the direction of descent [37, 5, 12]:

A 2 (Biased Compression with contraction). *For $\delta > 0$, a compression operator is said to be $1/(1+\delta)$-contractive if for any $x \in \mathbb{R}^d$, we have $\mathbb{E}[\|\mathrm{Comp}(x) - x\|] \leq (1 - 1/(1+\delta))\|x\|$.*

Constants ω and δ from these two assumptions are both positive, and become larger as the compression rate increases. Alternative assumptions for the biased case have been introduced in [5].

Impact of unbiasedness on the compression of a single vector.[1] To understand the interaction between the number of workers K and the compression error, a simple situation is the case in which the workers use *independent and identically distributed compression operators* $(\mathrm{Comp}_k)_{k=1}^K$ to compress the *same vector* $x \in \mathbb{R}^d$. The central node aggregates $\{\mathrm{Comp}_k(x)\}_{k=1}^K$ into $K^{-1}\sum_{k=1}^K \mathrm{Comp}_k(x)$. A bias-variance decomposition of the quadratic error gives:

$$\mathbb{E}[\|K^{-1}\textstyle\sum_{k=1}^K \mathrm{Comp}_k(x) - x\|^2] = \|\mathbb{E}[\mathrm{Comp}_1(x)] - x\|^2 + K^{-1}\|\mathbb{E}[\mathrm{Comp}_1(x)] - x\|^2].$$

The variance of the aggregated vector is reduced by a factor K^{-1} when averaging the messages send by the K workers, while the bias is independent of K. For example, if we use an unbiased compressor satisfying A 1, we get

$$\mathbb{E}\left[K^{-1}\textstyle\sum_{k=1}^K \mathrm{Comp}_k(x)\right] = x, \qquad \mathbb{E}\left[\left\|x - K^{-1}\textstyle\sum_{k=1}^K \mathrm{Comp}_k(x)\right\|^2\right] \leq (\omega/K)\|x\|^2, \quad (2)$$

while for a deterministic biased compressor, we obtain that $K^{-1}\sum_{k=1}^K \mathrm{Comp}_k(x) = \mathrm{Comp}_1(x)$ has the same error as any of the individual compressed vector. We therefore pay particular attention to obtaining an unbiased compressor in the following.

2.2 StoVoQ definitions and main properties.

Algorithm 1: StoVoQ with distribution p

Input : $x \in \mathbb{R}^d$, p, M, P, seed s
Output: Codeword index $\mathbf{i}_c$, value $\mathbf{i}_r$

Sample $\mathcal{C}_M \sim p$ with seed s ; /* generate codebook with distribution p */
$c = \mathrm{VQ}(x, \mathcal{C}_M^p)$; /* perform Voronoi quant. */
$\mathbf{i}_c =$ index of c; /* get index of codeword */
$r = r_M^p(\|x\|)$; /* find radial bias in table */
$\mathbf{i}_r = \mathrm{SQ}(r^{-1})$; /* quantize r on P bits */

The basic idea behind VQ is to quantize a vector rather than each of its coordinates. A Vector Quantizer is a mapping $\mathrm{VQ}(\cdot, \mathcal{C}_M) : \mathbb{R}^d \to \mathcal{C}_M$ which maps $x \in \mathbb{R}^d$ to an element of a codebook $\mathcal{C}_M$, which is a finite subset of $\mathbb{R}^d$ with M elements. The code of StoVoQ is provided in Algorithm 1, and its crucial steps are described hereafter: we introduce the notion of **(a)** Voronoi quantization scheme before describing more precisely **(b)** random codebooks, **(c)** whose distributions are invariant by unitary transforms. Then, **(d)** a method to obtain an unbiased Voronoi scheme is presented and finally **(e)** its asymptotic properties (as $M \to \infty$) are given.

(a) Voronoi Quantization. Voronoi quantization [26, 28], aims at selecting the closest codeword from $\mathcal{C}_M$, i.e.:

$$\mathrm{VQ}(x, \mathcal{C}_M) \triangleq \mathrm{argmin}_{c \in \mathcal{C}_M} \|x - c\| . \quad (3)$$

Unfortunately, for any given $\mathcal{C}_M$, the Voronoi quantizer is not *unbiased*: indeed it is deterministic and $\mathrm{VQ}(x, \mathcal{C}_M) \neq x$ if $x \notin \mathcal{C}_M$. A classical approach to construct a bias-free VQ is to use the optimal "dual" VQ (or Delaunay quantization) [27], but this approach is numerically expensive (see Subsection 2.3). To mitigate the bias, we rather use random codebooks.

(b) Random Codebook. A key ingredient of StoVoQ is the use of a random codebook within the quantizer. We assume $\mathcal{C}_M = [C_1, \ldots, C_M]$ where *the codewords* $\{C_i\}_{i=1}^M$ are i.i.d. random vectors distributed according to p, the codeword distribution pdf. We denote $\mathcal{C}_M \sim p$ and use boldface to stress that $\mathcal{C}_M$ is random. When quantizing a sequence of vectors $\{x_t\}_{t=0}^\infty \subset \mathbb{R}^d$ we sample for each $t \in \mathbb{N}$ a **new codebook** $\mathcal{C}_{M,t} \sim p$, compute $\mathrm{VQ}(x, \mathcal{C}_{M,t})$ and transmit the index of the corresponding codeword $i_{c,t} \in [M]$. The codebook $\mathcal{C}_{M,t}$ is **not transmitted**: the transmitter and the receiver use the **same seeds** so that the same codebooks $\mathcal{C}_{M,t}$ can be reconstructed on both sides.

[1] The impact of unbiasedness for obtaining optimal convergence complexities in FL is discussed in Section 3.

(c) Unitary invariant Codewords. Denote by $\mathrm{U}(d) = \{U, U^*U = \mathrm{I}\}$ the set of unitary transforms over $\mathbb{R}^d$. We assume in the sequel that the codeword distribution p is unitary invariant, meaning that:

A3. *The distribution of the codewords p is invariant under the unitary group, i.e. for all $U \in \mathrm{U}(d)$, and any $x \in \mathbb{R}^d$, $p(Ux) = p(x)$.*

Examples of such distributions include isotropic Gaussian distributions ($p = \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$, $\sigma^2 > 0$) and the uniform distribution on the Sphere (which is specifically discussed in Appendix D.1). Under A 3, there exists a non-negative function p_{rad} on $\mathbb{R}_+$ such that, for all $x \in \mathbb{R}^d$, $p(x) = p_{\mathrm{rad}}(\|x\|)$.

(d) The quantization bias is radial. Under A 3, we have the following crucial unitary invariance property. For $A \subset \mathbb{R}^d$, and $U \in \mathrm{U}(d)$, we write $UA = \{Ux, x \in A\}$.

Lemma 1. *Assume A 3. For any nonnegative measurable function f, any $U \in \mathrm{U}(d)$ and $x \in \mathbb{R}^d$,* $\mathbb{E}_{\mathcal{C}_M \sim p}[f(\mathrm{VQ}(Ux, \mathcal{C}_M))] = \mathbb{E}_{\mathcal{C}_M \sim p}[f(U\,\mathrm{VQ}(x, \mathcal{C}_M))]$.

The proof is postponed to Appendix A.3. Taking $f(x) = x$, the previous result implies that for any $x \in \mathbb{R}^d$ and $U \in \mathrm{U}(d)$, it holds that $\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(Ux, \mathcal{C}_M)] = U\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(x, \mathcal{C}_M)]$. A direct consequence of the elementary Lemma 3 is that the quantization error is radial:

Theorem 1 (Quantization bias)**.** *Assume A 3. Then, for all $M \in \mathbb{N}$, there exists a function $r_M^p : \mathbb{R}_+ \mapsto \mathbb{R}_+$ such that for all $x \in \mathbb{R}^d$, $\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(x, \mathcal{C}_M)] = r_M^p(\|x\|)x$.*

The proof is postponed to Appendix A.4.

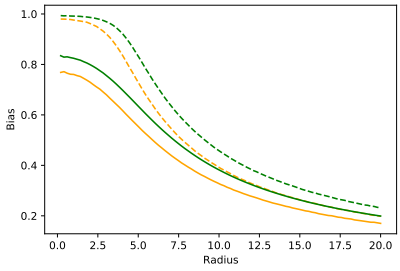


Figure 1: function r_M^p for $d = 4$ (dashed) and $d = 16$ (solid), $p = \mathcal{N}(0, \mathrm{I}_d)$ and $M = 2^{10}$(orange), and $M = 2^{13}$(green).

In words, the expectation of the quantized vector $\mathrm{VQ}(x, \mathcal{C}_M)$ is *colinear* to the vector x, i.e., $\mathrm{VQ}(x, \mathcal{C}_M)$ is **directionally unbiased**. Moreover, this radial bias only depends on $\|x\|$, M and the distribution p. This function is intractable, but it is straightforward to pre-compute it using Monte-Carlo method. We display r_M^p for $p = \mathcal{N}(0, \mathrm{I}_d)$ in Figure 1. Consequently, we can remove the bias of $\mathrm{VQ}(x, \mathcal{C}_M)$ by re-scaling the corresponding codeword by $1/r_M^p(\|x\|)$.

We now analyze the quantization distortion for a given $x \in \mathbb{R}^d$ vector. We need to strengthen the assumption about the distribution of the codewords. Consider the following assumption

A 4. *(1) there exists $\epsilon > 0$ such that $\int r^{2+\epsilon} p_{\mathrm{rad}}(r)\mathrm{d}r < \infty$ (2) for some $\delta > 0$, $m_\delta = \inf_{r \leq \delta} p_{\mathrm{rad}}(r) > 0$, and (3) p_{rad} is unimodal, i.e. the super level sets $\{r \in \mathbb{R}_+, p_{\mathrm{rad}}(r) \geq t\}$, for $t \geq 0$ are convex subsets of $\mathbb{R}_+$.*

A 4 is obviously satisfied if we take $p = \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for any $\sigma^2 > 0$.

Theorem 2. *Assume A 3-A 4. Define $C_d = \pi^{-1}\Gamma(1 + 2/d)\Gamma(1 + d/2)^{2/d}$. Then, for every $x \in \mathbb{R}^d$,*

$$\lim_{M \to \infty} M^{2/d} \mathbb{E}_{\mathcal{C}_M \sim p}[\| \mathrm{VQ}(x, \mathcal{C}_M) - x\|^2] = C_d p_{\mathrm{rad}}^{-2/d}(\|x\|)\,.$$

The proof is postponed to Appendix C.1. Note that $C_d \cong_{d \to \infty} d/(2\pi\mathrm{e})$ hence C_d grows only linearly with the dimension d. We can now exploit this result to control the radial bias as a function of $\|x\|$. Since $|r_M^p(\|x\|) - 1| \leq \|x\|^{-1}\{\mathbb{E}_{\mathcal{C}_M \sim p}[\| \mathrm{VQ}(x, \mathcal{C}_M) - x\|^2]\}^{1/2}$, Theorem 2 shows that

$$\limsup_{M \to \infty} M^{1/d} |r_M^p(\|x\|) - 1| \leq C_d^{1/2} p_{\mathrm{rad}}^{-1/d}(\|x\|)/\|x\|\,.$$

In other words, for any $x \in \mathbb{R}^d$, the radial bias $r_M^p(\|x\|)$ approaches 1 as $M \to \infty$ with a rate $O(M^{-1/d})$. We use an a scalar quantizer SQ to transmit $1/r_M^p(\|x\|)$. Because the range of values taken by $1/r_M^p(\|x\|)$ is limited, a small number of bits P is sufficient (we typically use $P = 3$ bits). The total number of transmitted bits is $\log_2(M) + \log_2(P)$. We use a random unbiased scalar quantizer (see e.g. [8, Eq. (2)]), a random mapping for $\mathbb{R} \to \mathcal{S}_P$ an ordered subset of $\mathbb{R}$ with P elements. A scalar quantizer is said to be unbiased if $\mathbb{E}[\mathrm{SQ}(r)] = r$ for all $r \in \mathbb{R}$. Assuming that SQ is independent of $\mathcal{C}_M$, we get for all $x \in \mathbb{R}^d$, $\mathbb{E}[\mathrm{SQ}(1/r_M^p(\|x\|))]\mathbb{E}_{\mathcal{C}_M \sim p}[\mathrm{VQ}(x, \mathcal{C}_M)] = x$. To save space, we present the details of the scalar quantization (based on nonuniform random dither) methods is presented in Appendix B.1.

(e) Random vs. Optimal codebooks: We finally motivate the choice of random codebooks and describe how to choose the codevector distribution p. For a given pdf q of the input the *(quadratic) distortion* is defined as:

$$\mathrm{Dist}(q, \mathcal{C}_M) = \int_{\mathbb{R}^d} \|x - \mathrm{VQ}(x, \mathcal{C}_M)\|^2 \, q(x) \, \mathrm{d}x = \mathbb{E}_{X\sim q}[\|X - \mathrm{VQ}(X, \mathcal{C}_M)\|^2] \,. \tag{4}$$

We stress that in this case the expectation is taken w.r.t. the input distribution q, the codebook being deterministic in (4). A *Voronoi optimal codebook* $\mathcal{C}_M^{q,*}$ is a minimizer of the distortion over the set of codebooks: $\mathrm{Dist}(q, \mathcal{C}_M^{q,*}) = \min_{|\mathcal{C}_M|=M} \mathrm{Dist}(q, \mathcal{C}_M)$. Zador's theorem [14] gives the distortion of the Voronoi optimal codebook in the limit of $M \to \infty$; see Appendix C.1 for a precise statement. Denote for $\beta \in \mathbb{R}_+$ and a function f on $\mathbb{R}^d$, $\|f\|_\beta = (\int |f(x)|^\beta \mathrm{d}x)^{1/\beta}$. It is known that if $\|q\|_{d/(d+2)} < \infty$, then as $M \to \infty$, $\mathrm{Dist}(q, \mathcal{C}_M) \approxeq M^{-2/d} J_d \|q\|_{d/(d+2)}$, and J_d is a universal constant J_d satisfying $J_d \approxeq_{d\to\infty} d/2\pi\mathrm{e}$ (see Appendix C.2 for the exact constant).

Using Theorem 2, we can quantify the loss between random codebook distributed according to p and the Voronoi optimal codebook for a given input distribution q when $M \to \infty$. Define

$$\mathrm{C}(q, p, d) = \int_{\mathbb{R}^d} p(x)^{-2/d} q(x) \mathrm{d}x \,. \tag{5}$$

If $\|q\|_{d/(d+2)} < \infty$, using the Hölder inequality with negative exponents (see [16, p. 191] and Appendix C.3),it holds that $\mathrm{C}(q, p, d) \geq \|q\|_{d/(d+2)}$.

Theorem 3. *Assume that p satisfies A 3-A 4, $\|q\|_{d/(d+2)} < \infty$, $\int_{\mathbb{R}^d} \|x\|^{2+\delta} q(x) \mathrm{d}x < \infty$ for some $\delta > 0$, and $\mathrm{C}(q, p, d) < \infty$. Then,*

$$\lim_{M\to\infty} \mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{Dist}(q, \mathcal{C}_M)] / \mathrm{Dist}(q, \mathcal{C}_M^{q,*}) = C_d J_d^{-1} \, \mathrm{C}(q, p, d) \|q\|_{d/(d+2)}^{-1}. \tag{6}$$

with C_d defined in Theorem 2. Moreover, assume that input distribution q satisfies A 3-A 4, and set the codeword distribution $p_{q,d,*} = q^{d/(d+2)}(x) / \int q^{d/(d+2)}(x) \mathrm{d}x$. Then, $\mathrm{C}(q, p_{q,d,*}, d) = \|q\|_{d/(d+2)}$.

The proof is postponed to Appendix C.2. In words, under general assumptions, the distortion achieved by a random quantizer $\mathrm{VQ}(\cdot, \mathcal{C}_M)$, $\mathcal{C}_M \sim p$ is rate optimal (with rate $M^{-2/d}$). If in addition q is unitarily invariant and unimodal, then a random codebook distributed according to $p_{q,d,*}$ reaches the optimal distortion bound, up to universal constants (depending only on the dimension d). Moreover, as $d \to \infty$, then $C_d J_d^{-1} \approxeq_{d\to\infty} 1$ and the efficiency gap vanishes. As an illustration, assume that the input distribution is standard Gaussian $q = \mathcal{N}(0, \mathrm{I}_d)$ and set the codeword distribution to be $p_\alpha = \mathcal{N}(0, \alpha^2 \mathrm{I}_d)$ where $\alpha^2 \in \mathbb{R}_+^*$. If $\alpha^2 d > 2$, then $\mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha^2 \mathrm{I}_d), d) = 2\pi\alpha^2 \{\alpha^2 d/(\alpha^2 d - 2)\}^{d/2}$ and $\|\mathcal{N}(0, \mathrm{I}_d)\|^{(2+d)/2} = (2\pi)(1 + 2/d)^{1+2/d}$. The function $\alpha \to \mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha^2 \mathrm{I}_d), d)$ has a unique minimum at $\alpha_d^2 = 1 + 2/d$ for which $\mathrm{C}(\mathcal{N}(0, \mathrm{I}_d), \mathcal{N}(0, \alpha_d^2 \mathrm{I}_d), d) = \|\mathcal{N}(0, \mathrm{I}_d)\|^{(2+d)/2}$ showing that a random codebook sampled from $\mathcal{N}(0, \alpha_d^2 \mathrm{I}_d)$ is optimal. It is interesting to note that the variance of the codeword distribution should be $(1 + 2/d)$ larger than the variance of the input distribution $\mathcal{N}(0, \mathrm{I}_d)$.

2.3 Related works

We compare `StoVoQ` with competing (random) compressors; additional details are given App. A.1.

QSGD. Alistarh et al. [2] compresses each coordinate of the scaled vector $x/\|x\|$ on $s+1$ codewords. QSGD is a scalar quantizer which requires $\mathcal{O}(\sqrt{d}\log_2(d))$ bits in its highest compression setting ($s = 1$, only two possible levels for each coordinate). The vector norm is transmitted with full precision $\|x\|$ (16 or 32 bits). This is in general substantially higher than the number of bits used by VQ methods. In deep learning problems, it reduces the communication cost by a factor of 4 to 7 [2, Sec. 5].

Top-H/Rand H. Achieving higher compression rates is possible through *sparsification* operators, that only transmit a few coordinates. The most popular schemes are `Top-`H and `Rand-`H compressors, that respectively map the vector to either its H largest coordinates, or a random subset of cardinality H, rescaled by d/H to ensure unbiasedness. `Top-`H is a biased operator, and the performance of `Rand-`H are poor on deep learning tasks [5, Figures 4 and 5].

Table 1: Per iteration communication complexity of most frequently used algorithms in dimension d. Constants H and M respectively correspond to a number of coordinates to be transmitted and a number of codewords, they are chosen by the user.

	Uncomp.	Scalar Quantization				Vector Quantization				
	SGD	`Sign`	`QSGD` $_{s\geq 1}$	Top-H	Rand-H	Polytope [10]	`HSQ-span` [8]	`HSQ-greed` [8]	`StoVoQ`	`DoStoVoQ`
#bits	$32d$	d	$32+s\sqrt{d}\log_2(d)$	$32H$	$32H$	$\log_2(2d)$	$\log_2(M)$	$\log_2(M)$	$\log_2(M)$	$\log_2(M)$
Unbiased	-		✓		✓	✓	✓		✓	✓ (Th.4)
A.1 $(\omega+1)$	-	-	$\sqrt{d}/s$	-	d/H	d	d	-		$O(M^{-2/d})$ (Th.4)
A.2 $(\delta+1)$	-	-	-	d/H	-	-		$M/\sigma_{\min}(C)$	-	-

HyperSphere Quantization (HSQ). `HSQ` was introduced by Dai et al. [8]. Two versions are considered: (1) a - greedy- Voronoi VQ referred to as `HSQ-greed` in Table 1, which is biased, and for which the theoretical guarantee provided in the paper (in their Lemma 3 and Theorem 3, which corresponds to a variant of A 2 and the subsequent convergence rate) *worsens* as M increases, making it mostly vacuous; (2) an unbiased version VQ (`HSQ-span`), which uses a minimum-norm decomposition of $x \in \mathrm{Span}(\mathcal{C}_M)$ the linear subspace generated by the codewords - this version suffers from a large variance (see Table 7) and potentially an ill-conditioning. Moreover, the performance of `HSQ-span` does not improve with M.

`StoVoQ` builds on `HSQ-greed`, that achieves high compression factors (up to 60-100 to obtain close to SOTA performance on CIFAR10), while preserving a good flexibility w.r.t. the compression level. `StoVoQ` approach allows to remove its inherent bias and provide a much stronger convergence analysis: **our approach is the first vector quantization scheme to provably benefit from an increasing number of elements in the codebook** M (and obviously benefits from the number of workers K, as it is unbiased).

Dual Quantization and Cross-polytope. An approach to constructing unbiased VQ is to use the dual VQ, also referred to as Delaunay Quantization (DQ); see [27]. DQ is unbiased for any $x \in \mathrm{ConvHull}(\mathcal{C}_M)$, the convex hull of $\mathcal{C}_M$. DQ requires to compute the barycentric coordinates for $x \in \mathrm{ConvHull}(\mathcal{C}_M)$, that is to solve $(\lambda_1^x, \dots, \lambda_M^x) = \mathrm{argmin}_{\lambda_1,\dots,\lambda_M} \|x - \sum_{i=1}^M \lambda_i c_i\|^2$, under the constraints $\lambda_i \geq 0, \sum_{i=1}^M \lambda_i = 1$. The quantizer is obtained by drawing a codeword c_i with probability $[\lambda_1^x, \dots, \lambda_M^x]$. Computing the barycentric coordinates is in general very demanding unless $\mathcal{C}_M$ has a very simple structure (see Appendix B for details). The `Cross-Polytope` method Gandikota et al. [10] is a simple instance of DQ, with a codebook $\mathcal{C}_{2d}^{\mathrm{CP}}$ composed of the $2d$ canonical vectors $\left\{ \pm\sqrt{d}e_i = \pm(0,\dots,0,\sqrt{d},0\dots 0), i \in [d] \right\}$, that relies on the inclusion $\mathrm{B}_2(0;1) \subset \mathrm{B}_1(0;\sqrt{d}) = \mathrm{ConvHull}(\mathcal{C}_{2d}^{\mathrm{CP}})$. The barycentric decomposition can then easily be computed. Unfortunately, this method suffers from a large variance, as the quantization error $\|\mathrm{VQ}^{\mathrm{CP}}(x, \mathcal{C}_M) - x\|$ of *any* x is *lower bounded* by $\sqrt{d}-1$, which means the error has the same quadratic error than the `Rand-1` compressor.

Table 1 summarizes the number of bits required to exchange the compressed value of a vector $x \in \mathbb{R}^d$ for the compression methods considered in this Section, as well as the assumptions they satisfy.

Numerical comparisons: In Table 7, we compare the distortions achieved by the compression methods given in Table 1 for a communication budget of 16 bits for $d = 16$ and assuming that the input distribution is $q = \mathcal{N}(0, \mathrm{I}_d)$. The compression factor is 32 (assuming 32 bits floating point per coordinate). Such a compression rate is out of reach for QSGD, that requires, even for $s = 1$ at least $\sqrt{d}\log(d) + R$ bits, where R is the number of bits to encode the norm (32 in [2]). For `QSGD` we have quantized the norm (using an uniform quantizer) on 3 bits and obtained an averaged distortion of 36.10 (for $K = 1$) and 1.82 for ($K = 20$) - the total number of bits is 19-. We use $H = 2$ for `Top-`H and `Rand-`H and use a scalar quantizer with 8 bits. For `HSQ`, we use 6 bits for the norm, using the unbiased uniform quantizer given in [8] and a Voronoi optimal codebook for the uniform distribution on the unit-sphere with $M = 2^{10}$ codewords. For `StoVoQ` we use a random codebook with $M = 2^{13}$ codewords; the codewords are sampled from a $\mathcal{N}(0, (1 + 2/d)\,\mathrm{I}_d)$, and 3 bits are allocated for the scalar quantization of $1/r_M^p$ (the inverse of the radial bias). Finally, we average the result of 2 independent compressions for `Polytope` (following the replication technique described in [10]). We use $n = 10^4$ vectors, and report in Table 7 the distortion and sample variance. For `StoVoQ` with $K = 20$, the codebooks of the different workers are independent.

Table 2: Distortion for Gaussian inputs, for a fixed budget of 16 bits with $d = 16$.

Method	Sign [4]	Top-2	Rand-2	Polytope [10]	HSQ-span [8]	HSQ-greed [8]	StoVoQ
# Bits (obj =16)	16	2×8	2×8	$\log_2(2 \times 16) \times 2 + 6$	$\log_2(2^{10}) + 6$	$\log_2(2^{10}) + 6$	$\log_2(2^{13}) + 3$
Unbiased			✓	✓	✓		✓
$K = 1$	6.21 (0.02)	8.40 (0.04)	102.8 (0.9)	113.9 (0.6)	146.9 (0.6)	9.03 (0.04)	6.97 (0.02) :
$K = 20$	6.26 (0.02)	8.76 (0.04)	5.40 (0.04)	5.98 (0.03)	7.58 (0.04)	9.10 (0.04)	0.838 (0.005)

3 DoStoVoQ algorithm

We illustrate how the StoVoQ compression scheme can be implemented in FL. To avoid cumbersome technical details, we focus here on the Federated-SGD algorithm. At iteration $t+1$, each worker computes a stochastic gradient $g_{k,t+1}$ of the loss f_k at the current model θ_t, compresses it into $\hat{g}_{k,t+1} = \mathrm{Comp}(g_{k,t+1})$ and send it to the central server, that performs the update step $\theta_t = \theta_{t-1} - \gamma_t / K \sum_{k=1}^K \hat{g}_{k,t}$. The code of the resulting algorithm, DoStoVoQ-SGD, is given in Algorithm 2. At iteration $t+1$, the crucial steps are:

1. Worker $k \in [K]$ computes the norm $\|g_{k,t+1}\|$ of the $D \times 1$ gradient $g_{k,t+1}$ and then splits the scaled gradient $g_{k,t+1} \times \sqrt{D}/\|g_{k,t+1}\|$ into *L-buckets* of size d: $g_{k,t+1} \times \sqrt{D}/\|g_{k,t+1}\| = [b^1_{k,t+1}, \ldots, b^L_{k,t+1}]$. The norm $\|g_{k,t+1}\|$ is transmitted to the central node using a high-resolution scalar quantizer (or without quantization).
2. Each worker quantizes the buckets $\{b^1_{k,t+1}, \ldots, b^L_{k,t+1}\}$ using StoVoQ. **Independent** codebooks $\{\mathcal{C}_{M,k,t+1}\}_{k\in[K]}$ are used to ensure that the quantizers remain conditionally independent (see below for a precise statement). The double stochasticity (each worker uses random codebooks, which are independent between workers and across iterations) motivates the name DoStoVoQ. At iteration t, the same codebook is used for all buckets of worker k. Formally, for $\ell \in [L]$ we apply (in parallel) $\mathtt{StoVoQ}(b^\ell_{k,t+1}, p, M, P, s_{k,t+1})$, with a sequence of different seeds $(s_{k,t+1})_{k\in[K],t\geq 0}$. This sequence is shared between the workers and the central node at initialization.
3. The central node computes $(\hat{g}_{k,t+1})_{k\in K}$ from all messages received, performs the update on $(\theta_t)_{t\geq}$, and broadcasts θ_{t+1} to the workers.

These steps would similarly allow to incorporate StoVoQ within any of the advanced FL algorithms, and Theorem 4 is the crucial assumption to derive the convergence rates, as described in Section 2. Natural extensions to DoStoVoQ-Fed-Avg, DoStoVoQ-DIANA and DoStoVoQ-VR-DIANA are provided in Appendix D.2.

Bias and variance of the compressed gradient with K workers. Consider the two filtrations $(\mathcal{F}_t)_{t\geq 0}$ and $(\mathcal{G}_t)_{t\geq 0}$ defined recursively as follows $\mathcal{F}_0 = \sigma(\emptyset)$ and for $t \geq 0$, $\mathcal{G}_{t+1} = \mathcal{F}_t \vee \sigma(\{g_{k,t+1}, k \in [K]\})$ and $\mathcal{F}_{t+1} = \mathcal{G}_{t+1} \vee \sigma(\{\hat{g}_{k,t+1}, k \in [K]\})$. With these notations, for any $t \geq 0$, θ_t is $\mathcal{F}_t$-measurable.

Theorem 4. *At any iteration $t+1$ in DoStoVoQ, the K compressed stochastic gradients $(\hat{g}_{k,t+1})_{k\in[K]}$ are (i) independent conditionally to $\mathcal{G}_{t+1}$ (ii) conditionally unbiased, i.e., for all $k \in [K]$, we have $\mathbb{E}\left[\hat{g}_{k,t+1} \,|\, \mathcal{G}_{t+1}\right] = g_{k,t+1}$, (iii) satisfy the relatively bounded error condition of A 1, i.e. there exists a constant ω_M such that, for all $k \in [K]$: $\mathbb{E}\left[\|\hat{g}_{k,t+1} - g_{k,t+1}\|^2 \,\middle|\, \mathcal{G}_{t+1}\right] \leq \omega_M \|g_{k,t+1}\|^2$.*

Moreover, ω_M decreases with the number of codewords M and the P, as $\omega_M = O(M^{-2/d}) + O(2^{-P})$ [the dependence on p, d, and D is made explicit in the proof].

Algorithm 2: DoStoVoQ-SGD over T iterations

Input : T nb of steps, $(\gamma_t)_{t\geq 0}$ LR, θ_0, p, M, P ;
Output: $(\theta_t)_{t\geq 0}$

for $t = 1, \ldots, T$ **do**
  w_0 sends θ_{t-1} and different seeds $s_{k,t}$ to each w_k;
  for $k = 1, \ldots, K$ **do**
    Compute local gradient $g_{k,t}$ at θ_{t-1};
    Split $g_{k,t} \times \sqrt{D}/\|g_{k,t}\|$ on $[b^1_{k,t}, \ldots, b^L_{k,t}]$;
    for $\ell = 1, \ldots, L$ *(in parallel)* **do**
      $(\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell}) = \mathtt{StoVoQ}(b^\ell_{k,t}, p, M, P, s_{k,t})$
    end
    Send $(\|g_{k,t}\|, (\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell})_{\ell\in[L]})$ to w_0 ;
  end
  Reconstruct $(\hat{g}_{k,t})_{k\in K}$;
  Update: $\theta_t = \theta_{t-1} - \gamma_t \frac{1}{K} \sum_{k=1}^K \hat{g}_{k,t}$;
end

The first statement stems from the fact that each bucket is quantized using StoVoQ which is unbiased. The second statement is more challenging; proof is postponed to Appendix A.5. We stress that this result differs from Theorem 2, which corresponds to the distortion of a source with distribution q.

Convergence results. Theorem 4 proves that our compression method satisfies the assumptions needed to obtain fast convergence rate, for DoStoVoQ-SGD, and for its variants DoStoVoQ-(VR)-DIANA. Consider a Smooth and Strongly Convex (SSC) function $F = \sum_{k=1}^{K} f_k$, with condition number $\kappa > 1$. We measure the complexity of the algorithm by the number of iterations t required to obtain a model θ_t such that $\mathbb{E}[F(\theta_t)] - \min_{\mathbb{R}^D} F \leq \epsilon$. The result of VR-DIANA [17], which provides a complexity of $O_{\kappa\to\infty}\left(\kappa\left(1+\omega_M/K\right)\log(\epsilon^{-1})\right)$ [17, Corollary 2], applies to DoStoVoQ-VR-DIANA.

Convergence rates for DoStoVoQ-DIANA (without VR), and on non-convex optimization problems can be obtained from Horváth et al. [17, Corollary 1,3,4]. As in the strongly-convex case, complexities increase by a factor depending on $(1+\omega_M/K)$ w.r.t. uncompressed algorithm. Intuitively, *the impact on the optimization complexity of a high compression is mitigated by the number of workers*, which supports the use of independent and unbiased compressors when the number of workers is large and high compression factors are required.

Indeed, these complexities can be compared to: (1) the one of *uncompressed* variance reduced distributed methods [9] that achieve a complexity of $O_{\kappa\to\infty}\left(\kappa\log(\epsilon^{-1})\right)$ (in the SSC case); (2) the complexity for biased compression operators satisfying A 2, Beznosikov et al. [5, Theorem 13] that obtain $O_{\kappa\to\infty}(\kappa(1+\delta)\log(\epsilon^{-1}))$ for compressed GD (independently of the number of workers); (3) the complexities of compressed SGD methods with *error feedback* in [12][2], that also have no dependency on the number of workers. **Overall,the unbiased character is crucial to mitigate the variance increase resulting from high compression rates.**

4 Numerical experiments

4.1 Least Squares Regression (LSR)

We consider a least-squares problem with $n = 2^{14}$ samples, a bucket size $d = 16$, $D = 2^9$, and $K = 32$ workers; each worker has access to a subset $m = 2^{11}$ samples (picked with replacement) to introduce a dependency in the data used by the workers. For $i \in [n]$, we assume $X_i \sim \mathcal{N}(0, \mathrm{I}_D)$ and $Y_i \sim \mathcal{N}(X_i^\top \omega_*, 1)$ where $\omega_* \in \mathbb{R}^D$. We solve $\inf_{\omega\in\mathbb{R}^D} \sum_{i=1}^{n} \|Y_i - X_i^\top\omega\|^2$ via a gradient descent with step size $1/\alpha L$ where α is fine-tuned for each quantization method and $L \approx 2n$ is the smoothness constant. We use DoStoVoQ with $M = 2^{13}$ codewords sampled from $\mathcal{N}(0, (1+2/d)\,\mathrm{I}_d)$ for DoStoVoQ and $M = 2^{10}$ on the unit Sphere for HSQ s.t. the number of bits transmitted at each round by the worker is set to 16 (see Table 7). Figure 2 reports the excess-log of the train loss over $T = 10$ iterations, for a standard GD. DoStoVoQ outperforms HSQ-greed: indeed the linear convergence rate of distributed GD is faster for an unbiased compressor than for the biased approach.

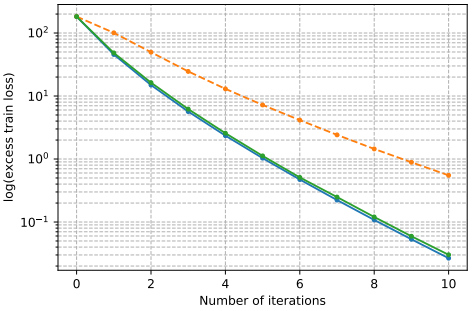


Figure 2: Comparison between GD (blue), HSQ-greed (orange) and DoStoVoQ (green), on a LSR problem in dimension $D = 2^9$.

4.2 Applications to Deep Neural Networks training

Setting. We now describe our experimental framework for training two standard models of Deep Neural Networks: a VGG-16 [35] and a ResNet-18 [15]. We follow the standard procedure of training those models both on CIFAR-10 and ImageNet; the hyper-parameters are fine-tuned to optimize the accuracy *without quantization*. We do not compress the affine constant part of the affine convolutional layers and batch normalization layers. We apply independent DoStoVoQ on

[2] authors provide complexities for 10 algorithms in Table 1, with Error Feedback and under A 2.

Table 3: Average accuracy over 5 experiments, after 100 epochs on CIFAR-10.

Algorithm	SGD	QSGD 2 bits	QSGD 4 bits	QSGD 8 bits	HSQ $d=16$	HSQ $d=8$	Dos. $d=16$	Dos. $d=8$
Raw bits per bucket	$32d$	$\sqrt{d}\log(d)$			$\log(d)$			
Effective Compression factor	1	~ 13	~ 8	~ 4	34	17	38	20
$K=1$ worker	91.9	91.7	92.1	91.9	92.0	92.0	92.0	92.1
$K=8$ worker	92.0	91.8	91.8	92.0	91.8	92.0	91.8	92.1

Table 4: Distortion for on a subset $\mathcal{G}$ of the gradients of a layer of CIFAR-10, for a fixed budget of 16 bits with $d=16$.

Method	Top-2	Rand-2	Polytope [10]	HSQ-span [8]	HSQ-greed [8]	DoStoVoQ
# Bits (obj =16)	2×8	2×8	$\log_2(2\times 16)\times 2+6$	$\log_2(2^{10})+6$	$\log_2(2^{10})+6$	$\log_2(2^{13})+3$
Unbiased		✓	✓	✓		✓
$K=1$	0.0022	0.025	0.028	0.034	0.0021	0.0026

batches of 32 buckets of size $d=16$ (i.e. we transmit a high-resolution norm for $D=32\cdot 16=512$ coefficients).

CIFAR-10. We use the implementation of HSQ [8]: the batch size is 256 for CIFAR-10, the total number of epochs is 100, the initial learning rate is 0.1, which is divided by 10 and 50 at epochs 51 and 71. We report the accuracy of DoStoVoQ, QSGD, and HSQ-greed in table 4. By design, the compression factor of Q-SGD for $d=16$ is 13, which is significantly less than HSQ or DoStoVoQ. Both HSQ and DoStoVoQ perform similarly and the accuracy gap between the two methods are under the sample variance (computed over 5 seed and about 0.2). In Table 4 we report the distortion of a random subset of gradients $\mathcal{G}=\{g_t, t\in[|\mathcal{G}|]\}$ (with $|\mathcal{G}|=10^2$, $d=16$, $D=2^5\times d$) obtained from a given layer of a VGG on CIFAR-10, i.e.: $|\mathcal{G}|^{-1}\sum_{g_t\in\mathcal{G}}\left\|K^{-1}\sum_{k=1}^{K}(g_{k,t}-\hat{g}_{k,t})\right\|^2$, where $(\hat{g}_{k,t})_{k\in[K]}$ correspond to k independent workers compressing their own gradient $g_{k,t}$. The choice of the layer does not affect significantly the results. Even with the actual gradient distribution, DoStoVoQ outperforms for a given compression factor each unbiased method. This is on pair with the observation that the gradients of a Deep Neural Network are approximately Gaussian distributed [3, 41, 4]. Additional experiments can be found in the Appendix.

ImageNet. For ImageNet, we use different bucket sizes, the standard batch size of 256, and only $K=1$ worker for energy savings (recall Imagenet training last about 1 day for a single worker on academic hardware). An initial learning rate of 0.1 is divided by 10 at epoch 30 and 60, while the model is trained for 90 epochs. A ResNet here obtains 69.9%, and with a compression factor of 8, the performance drops by 2.5%. Using $d=16$, we reach a compression factor of 38, while the Top-1 accuracy drops by only 4.8%: this is a substantially higher compression rate than the concurrent work QSGD on the ImageNet dataset.

Computational impact. In the case of deep Neural Networks, our training procedure requires neither a substantial modifications of standard pipelines, nor a modification of the hyper-parameters which allows to save computational resources. Green Algorithm ([22]) shows that this work generated around 15kg of CO2, and require 400 kWh. A typical experiment lasted few hours on CIFAR-10 and about 3 days on ImageNet, which is in the standard range for this type of prototypical codes. This work could have future impact on FL, to reduce their electrical consumption.

Broader impact. Federated learning enables multiple actors to build a common model without data sharing, hence respecting privacy. However classic FL methods consume an important amount of energy in transmitting information. Our method DoStoVoQ can be adapted to any FL framework while enabling important bandwidth savings. These savings highly counterbalance the computational impact of our experiments.

References

[1] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7564–7575. Curran Associates, Inc., 2018.

[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

[3] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5151–5159, 2018.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[5] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On Biased Compression for Distributed Learning. *arXiv:2002.12410 [cs, math, stat]*, February 2020. arXiv: 2002.12410.

[6] Léon Bottou. On-line learning and stochastic approximations. 1999. doi: 10.1017/CBO9780511569920.003.

[7] Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg, 2010. Physica-Verlag HD. ISBN 978-3-7908-2604-3. doi: 10.1007/978-3-7908-2604-3_16.

[8] Xinyan Dai, Xiao Yan, Kaiwen Zhou, Han Yang, Kelvin KW Ng, James Cheng, and Yu Fan. Hyper-sphere quantization: Communication-efficient sgd for federated learning. *arXiv preprint arXiv:1911.04655*, 2019.

[9] Aaron Defazio, Francis R Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.

[10] Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar. vqsgd: Vector quantized stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics*, pages 2197–2205. PMLR, 2021.

[11] Richard Gardner. The brunn-minkowski inequality. *Bulletin of the American Mathematical Society*, 39(3):355–405, 2002.

[12] Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly Converging Error Compensated SGD. *arXiv:2010.12292 [cs, math]*, October 2020. arXiv: 2010.12292.

[13] Eduard Gorbunov, Konstantin Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster non-convex distributed learning with compression. *arXiv preprint arXiv:2102.07845*, 2021.

[14] Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer, 2007.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Edwin Hewitt and Karl Stromberg. *Real and abstract analysis: a modern treatment of the theory of functions of a real variable*. Springer-Verlag, 2013.

[17] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic Distributed Learning with Gradient Quantization and Variance Reduction. *arXiv:1904.05115 [math]*, April 2019. arXiv: 1904.05115.

[18] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 315–323, Lake Tahoe, Nevada, December 2013. Curran Associates Inc.

[19] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *arXiv:1912.04977 [cs, stat]*, December 2019. arXiv: 1912.04977.

[20] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv:1610.02527 [cs]*, October 2016. arXiv: 1610.02527.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Loïc Lannelongue, Jason Grealey, and Michael Inouye. Green Algorithms: Quantifying the carbon emissions of computation. *arXiv:2007.07610 [cs]*, October 2020. arXiv: 2007.07610.

[23] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtarik. Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization. In *International Conference on Machine Learning*, pages 5895–5904. PMLR, November 2020. ISSN: 2640-3498.

[24] Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't Use Large Mini-Batches, Use Local SGD. *arXiv e-prints*, art. arXiv:1808.07217, August 2018.

[25] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed Learning with Compressed Gradient Differences. *arXiv:1901.09269 [cs, math, stat]*, June 2019. arXiv: 1901.09269.

[26] Gilles Pagès and Jacques Printems. Optimal quadratic quantization for numerics: the gaussian case. *Monte Carlo methods and applications*, 9(2):135–165, 2003.

[27] Gilles Pagès and Benedikt Wilbertz. Sharp rate for the dual quantization problem. In *Séminaire de Probabilités XLIX*, volume 2215 of *Lecture Notes in Math.*, pages 405–454. Springer, Cham, 2018.

[28] Gilles Pagès and Benedikt Wilbertz. Sharp rate for the dual quantization problem. In *Séminaire de Probabilités XLIX*, pages 405–454. Springer, 2018.

[29] Abhishek Panigrahi, Raghav Somani, Navin Goyal, and Praneeth Netrapalli. Non-gaussianity of stochastic gradient noise. *arXiv preprint arXiv:1910.09626*, 2019.

[30] Constantin Philippenko and Aymeric Dieuleveut. Artemis: tight convergence guarantees for bidirectional compression in Federated Learning. *arXiv:2006.14591 [cs, stat]*, November 2020. arXiv: 2006.14591.

[31] Ali Ramezani-Kebrya, Fartash Faghri, and Daniel M Roy. Nuqsgd: Improved communication efficiency for data-parallel sgd via nonuniform quantization. *arXiv preprint arXiv:1908.06077*, 2019.

[32] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *Annals of Mathematical Statistics*, 22(3):400–407, September 1951. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177729586. Number: 3 Publisher: Institute of Mathematical Statistics.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[34] F. Seide, H. Fu, Jasha Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. pages 1058–1062, January 2014.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Sebastian U. Stich. Local SGD Converges Fast and Communicates Little. *arXiv:1805.09767 [cs, math]*, May 2019. arXiv: 1805.09767.

[37] Sebastian U. Stich and Sai Praneeth Karimireddy. The Error-Feedback Framework: Better Rates for SGD with Delayed Gradients and Compressed Communication. *arXiv:1909.05350 [cs, math, stat]*, September 2019. arXiv: 1909.05350.

[38] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with Memory. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4447–4458. Curran Associates, Inc., 2018.

[39] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems*, 32:14259–14268, 2019.

[40] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient Sparsification for Communication-Efficient Distributed Optimization. *Advances in Neural Information Processing Systems*, 31:1299–1309, 2018.

[41] An Xu, Zhouyuan Huo, and Heng Huang. Optimal gradient quantization condition for communication-efficient distributed training. *arXiv preprint arXiv:2002.11082*, 2020.

[42] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. The zipml framework for training models with end-to-end low precision: The cans, the cannots, and a little bit of deep learning. *arXiv preprint arXiv:1611.05402*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 2 for quantization and Section 4 for associated experiments.
   (b) Did you describe the limitations of your work? [Yes] See broader impact and Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Detailed experiments carbon footprint can be find in Section 4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Also see Appendix in Supplemental Material.
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code available in Supplementary Material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In particular Table 7 presents standard deviations, and variances of NN model accuracies from Section 4 can be found in Appendix.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4 for further references.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] As mentioned in Section 4, code is partly inspired from [8].

(b) Did you mention the license of the assets? [Yes] Only open source and/or Academic assets are used.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Radial biases already computed available in Supplemental Material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Use of publicly available data (CIFAR10 [21] and Imagenet [33]).

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Contents

A Proofs

A.1 Classical compressors mentioned in the main text

For completeness, we here recall the formal definitions of the scalar compression operators mentioned in the main text. For $i \in [d]$, denote by e_i the i-th canonical vector. Let $H \in [d]$.

Definition 1 (`Sign`)**.** *For any $x \in \mathbb{R}^d$, $\mathtt{Sign}(x) := \sum_{i\in[d]} \operatorname{sign}(x_i)\, e_i$.*

Definition 2 (`Top-H`)**.** *For any $x \in \mathbb{R}^d$, $\mathtt{Top\text{-}H}(x) := \sum_{i\in T_H} x_i e_i$, where T_H is the set composed of the indices of the H largest (in absolute value) coordinates of x.*

Definition 3 (`Rand-H`)**.** *For any $x \in \mathbb{R}^d$, $\mathtt{Rand\text{-}H}(x) := \frac{d}{H}\sum_{i\in R_H} x_i e_i$, where R_H is the set composed of H random indices picket uniformly without replacement.*

Definition 4 (s-quantization operator)**.** *Let $s \geq 1$ and $p \geq 1$. Given $x \in \mathbb{R}^d$, the s-quantization operator $\mathcal{C}_s$ is defined by:*

$$\mathcal{C}_s(x) := \|x\|_p \times \sum_{i=1}^{d} \operatorname{sign}(x_i)\left\{s^{-1}\lfloor s|x_j|/\|x\|_p\rfloor + \mathbb{1}(\{U_i \leq s|x_j|/\|x\|_p - \lfloor s|x_j|/\|x\|_p\rfloor\})\right\} e_i\,.$$

where $\{U_i\}_{i=1}^d$ are d-independent uniform random variables on $[0,1]$.

The s-quantization scheme verifies A 1 with $\omega_s = \min(d/s^2, \sqrt{d}/s)$. Proof can be found in Alistarh et al. [2, see Appendix A.1].

A.2 Notations

For $u, v \in \mathbb{R}^d$, $\langle u, v\rangle = u^\top v$ denotes the standard scalar product. For $p \geq 1$ and $x \in \mathbb{R}^d$, $\|x\|_p = \left\{\sum_{i=1}^d |x_i|^p\right\}^{1/p}$. When $p = 2$, we sometimes drop the subscript, i.e. we write $\|x\|$ as a shorthand notation of $\|x\|_2$.

A function $\varphi : \mathbb{R}^d \to \mathbb{R}$ is said to be a *radial function* if and only if φ is invariant under unitary transforms, i.e. for all $x \in \mathbb{R}^d$ and $U \in \mathrm{U}(d)$, $\varphi(Ux) = \varphi(x)$.

We denote for $t > 0$ by $\Gamma(t) = \int_0^{+\infty} u^{t-1}\mathrm{e}^{-u}\mathrm{d}u$ the Gamma function. $\mathrm{Leb}\, d$ the Lebesgue measure on $\mathbb{R}^d$. $\mathrm{B}(x; r)$ is the (Euclidean) ball centered at $x \in \mathbb{R}^d$ with radius $r > 0$. We denote by $S_{d-1} = \{x \in \mathbb{R}^d, \|x\| = 1\}$ the unit-sphere and σ_{d-1} the uniform distribution ofn S_{d-1}.

A.3 Proof of Lemma 1

Note that, for any $U \in \mathrm{U}(d)$ and $x \in \mathbb{R}^d$,

$$\mathrm{VQ}(Ux, \mathcal{C}_M) = \operatorname{argmin}_{c\in\mathcal{C}_M} \|Ux - c\| = \operatorname{argmin}_{c\in\mathcal{C}_M} \|x - U^\top c\| = U\,\mathrm{VQ}(x, U^\top\mathcal{C}_M)\,, \quad (7)$$

where $U^\top\mathcal{C}_M = \{U^\top C_1, \ldots, U^\top C_n\}$. Using (7) and A 3, we get

$$\mathbb{E}_{\mathcal{C}_M\sim p}[g(\mathrm{VQ}(Ux, \mathcal{C}_M))] = \mathbb{E}_{\mathcal{C}_M\sim p}[g(U\,\mathrm{VQ}(x, U^\top\mathcal{C}_M))] = \mathbb{E}_{\mathcal{C}_M\sim p}[g(U\,\mathrm{VQ}(x, \mathcal{C}_M))]\,.$$

A.4 Proof of Theorem 1

We preface the proof of the Theorem by stating and proving two elementary lemmas.

Lemma 2. *Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a function such that $f(Ux) = Uf(x)$ for any $x \in \mathrm{S}_{d-1}$ and $U \in \mathrm{U}(d)$. Then, there exists $r \in \mathbb{R}$ such that $f(x) = rx$ for all $x \in \mathbb{R}^d$.*

Proof. For all $x \in \mathrm{S}_{d-1}$, define $g(x) = f(x) - \langle f(x), x\rangle x$. It is easily checked that for all $x \in \mathbb{R}^d$ and $U \in \mathrm{U}(d)$, $g(Ux) = Ug(x)$. Let U_x be the reflection symmetry with axis $\mathbb{R}x$: $U_x x = x$ and for any vector $y \in \mathbb{R}^d$ orthogonal to x, $U_x y = -y$. Since $g(x) = g(U_x x) = U_x g(x) = -g(x)$, we get that $g(x) = 0$ for all $x \in \mathrm{S}_{d-1}$. Finally, denote by $U_{x\to e_1}$ (where e_1 is the first canonical vector) any unitary transform satisfying $U_{x\to e_1} x = e_1$. We get

$$\langle f(x), x\rangle = \langle U_{x\to e_1}^\top f(U_{x\to e_1} x), x\rangle = \langle f(U_{x\to e_1} x), U_{x\to e_1} x\rangle = \langle f(e_1), e_1\rangle = r\,,$$

which concludes the proof. □

Lemma 3. *Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a function such that $f(Ux) = Uf(x)$ for any $x \in \mathbb{R}^d$ and $U \in \mathrm{U}(d)$. Then, there exists a function $r : \mathbb{R} \to \mathbb{R}$ such that $f(x) = r(\|x\|)x$. Moreover, $r(x) = \|f(\|x\|e_1)\|/\|x\|$.*

Proof. Let $\lambda > 0$, and define for $x \in \mathrm{S}_{d-1}$, $f_\lambda(x) = f(\lambda x)$. Lemma 2 shows that there exists $\rho(\lambda) \in \mathbb{R}$ such that, for all $x \in \mathrm{S}_{d-1}$, $f_\lambda(x) = f(\lambda x) = \rho(\lambda)x$. Hence for $x \in \mathbb{R}^d$, $f(x) = f_{\|x\|}(x/\|x\|) = \rho(\|x\|)x/\|x\|$. Hence $|\rho(\|x\|)| = \|f(x)\| = \|f(\|x\|U_{x/\|x\|\to e_1} x/\|x\|)\| = \|f(\|x\|e_1)\|$. The proof follows. □

Proof of Theorem 1. The proof follows from Lemmas 1 and 3. □

A.5 Proof of Theorem 4

We preface the proof by several technical lemmas. These lemmas establish important properties of random vector quantization that are of interest beyond the proof of the theorem.

Lemma 4. *Let $c_1, c_2, x \in \mathbb{R}^d$ and $\lambda \in (0,1)$. If $\|x - c_1\| \le \|x - c_2\|$ and $\|\lambda x - c_2\| \le \|\lambda x - c_1\|$, then $\|c_2\| \le \|c_1\|$.*

Proof. Indeed, we have both $\|x\|^2 - 2\langle x, c_1\rangle + \|c_1\|^2 \le \|x\|^2 - 2\langle x, c_2\rangle + \|c_2\|^2$ and $\lambda^2\|x\|^2 - 2\lambda\langle x, c_2\rangle + \|c_2\|^2 \le \lambda^2\|x\|^2 - 2\lambda\langle x, c_1\rangle + \|c_1\|^2$. Thus $-2\langle x, c_1\rangle \le -2\langle x, c_2\rangle + \|c_2\|^2 - \|c_1\|^2$ and $-2\langle x, c_2\rangle \le -2\langle x, c_1\rangle + \lambda^{-1}\|c_1\|^2 - \lambda^{-1}\|c_2\|^2$. Combining both inequalities, we get $(\lambda^{-1} - 1)\|c_2\|^2 \le (\lambda^{-1} - 1)\|c_1\|^2$ and as $\lambda^{-1} - 1 > 0$, we conclude $\|c_2\|^2 \le \|c_1\|^2$. □

We now make an additional assumption on the codeword distribution p.

A5. *The distribution p is* radially *homogeneous, i.e. p is unitarily invariant and for any $\beta \in (0,1]$ and $x \in \mathbb{R}^d$:*

$$\|\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \beta\mathcal{C}_M)]\| \le \|\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \mathcal{C}_M)]\|$$

In words, this means that contracting all codewords by a factor $\beta \in (0,1]$ reduces the norm of the expectation of the nearest neighbor of any x. This condition is slightly more restrictive than A 3. It is satisfied by the standard Gaussian distribution. Under this assumption, we have the following Lemma.

Lemma 5. *Assume A 3-A 5 and consider the function $r_M^p : \mathbb{R}_+ \to \mathbb{R}$ defined in Theorem 1. For any $M \in \mathbb{N}^*$ the function $\rho \to r_M^p(\rho)$ is non-increasing on $\mathbb{R}_+$. If in addition A 4 is satisfied, then for any $\rho \in \mathbb{R}_+^*$, $r_M^p(\rho) \le 1$.*

Proof. Let $x \in \mathbb{R}^d$ and $\lambda > 1$. By definition, we have:

$$r_M^p(\lambda\|x\|)\lambda x = \mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(\lambda x, \mathcal{C}_M)] = \lambda\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \lambda^{-1}\mathcal{C}_M)], \tag{8}$$

where we have used that a.s., $\mathrm{VQ}(\lambda x, \mathcal{C}_M) = \lambda\,\mathrm{VQ}(x, \lambda^{-1}\mathcal{C}_M)$. On the other hand:

$$\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \mathcal{C}_M)] = r_M^p(\|x\|)x. \tag{9}$$

Combining both equations under A 2, we get:

$$\begin{aligned}\|r_M^p(\lambda\|x\|)\lambda x\|^2 &\overset{eq.\ (8)}{=} \|\lambda\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \lambda^{-1}\mathcal{C}_M)]\|^2\\ &\overset{A\ 5}{\le} \|\lambda\mathbb{E}_{\mathcal{C}_M\sim p}[\mathrm{VQ}(x, \mathcal{C}_M)]\|^2\\ &\overset{eq.\ (9)}{=} \|\lambda r_M^p(\|x\|)x\|^2.\end{aligned}$$

Overall, we obtain that $(r_M^p(\lambda\|x\|))^2 \leq (r_M^p(\|x\|))^2$, thus that r_M^p is non-increasing.

We now consider the second statement. Note first that

$$\mathbb{E}_{\mathcal{C}_M\sim p}\left[\max_{i\in[M]}\|C_i\|\right] \leq \mathbb{E}_{\mathcal{C}_M\sim p}\left[\sum_{i=1}^{M}\|C_i\|^2\right] = M\mathbb{E}_{C_1\sim p}[\|C_1\|^2]\,. \tag{10}$$

Since $\|\operatorname{VQ}(x,\mathcal{C}_M)\| \leq \max_{i\in[M]}\|C_i\|$, for all $x\in\mathbb{R}^d$ it holds that

$$\|\mathbb{E}_{\mathcal{C}_M\sim p}[\operatorname{VQ}(x,\mathcal{C}_M)]\| \leq M^{1/2}(\mathbb{E}_{C_1\sim p}[\|C_1\|^2])^{1/2}\,.$$

Hence, for all $x\in\mathbb{R}^d$ such that $\|x\| \geq M^{1/2}(\mathbb{E}_{C_1\sim p}[\|C_1\|^2])^{1/2}$, $\|\mathbb{E}_{\mathcal{C}_M\sim p}[\operatorname{VQ}(x,\mathcal{C}_M)]\| \leq \|x\|$. For all $\lambda\in(0,1)$, using A 5, we get

$$r_M^p(\lambda\|x\|)\lambda\|x\| = \|\mathbb{E}_{\mathcal{C}_M\sim p}[\operatorname{VQ}(\lambda x,\mathcal{C}_M)]\| = \lambda\|\mathbb{E}_{\mathcal{C}_M\sim p}[\operatorname{VQ}(x,\lambda^{-1}\mathcal{C}_M)]\| \leq \lambda\|x\|\,,$$

which concludes the proof. □

Lemma 6. *Assume A 3-A 4. Then, for any $M\in\mathbb{N}^*$, $\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2]$ is (a radial function) which is non-decreasing, i.e. for any $x\in\mathbb{R}^d$ and $\lambda\in[0,1]$, $\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(\lambda x,\mathcal{C}_M)-\lambda x\|^2] \leq \mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2]$.*

Proof. By Lemma 1, for any $U\in\mathrm{U}(d)$ and $x\in\mathbb{R}^d$, we get

$$\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(Ux,\mathcal{C}_M)-Ux\|^2] = \mathbb{E}_{\mathcal{C}_M\sim p}[\|U\operatorname{VQ}(x,\mathcal{C}_M)-Ux\|^2] = \mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2]$$

showing that this function is radial. We write, for any $x\in\mathbb{R}^d$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2] &= 2\int_{t=0}^{\infty} t\mathbb{P}_{\mathcal{C}_M\sim p}\left(\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2 > t\right)\mathrm{d}t \\
&= 2\int_{t=0}^{\infty} t\mathbb{P}_{\mathcal{C}_M\sim p}\left(\min_{i\in[M]}\|C_i-x\|^2 > t\right)\mathrm{d}t \\
&= 2\int_{t=0}^{\infty} t\left(1-\mathbb{P}_{C_1\sim p}\left(\|C_1-x\|^2\leq t\right)\right)^M\mathrm{d}t \\
&= 2\int_{t=0}^{\infty} t\left(1-\mathbb{P}_{C_1\sim p}\left(\mathrm{B}_2(x;\sqrt{t})\right)\right)^M\mathrm{d}t\,.
\end{aligned}$$

By Anderson's theorem [11], we have that $\mathbb{P}_{C_1\sim p}\left(\mathrm{B}_2(x;\sqrt{t})\right)$ is (a radial function) which is non-increasing, i.e. for any $x\in\mathbb{R}^d$ and $\lambda\in[0,1]$, $\mathbb{P}_{C_1\sim p}\left(\mathrm{B}_2(\lambda x;\sqrt{t})\right) \geq \mathbb{P}_{C_1\sim p}\left(\mathrm{B}_2(x;\sqrt{t})\right)$.

Consequently, the quadratic error $\mathbb{E}_{\mathcal{C}_M\sim p}[\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2]$ is non-decreasing radial function. □

Next, we provide a control on the second order moment of $\operatorname{VQ}(x,\mathcal{C}_M)$.

Lemma 7. *Assume A 3-A 4. Then, for any $M\in\mathbb{N}^*$, the function $x\to\mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\operatorname{VQ}(x,\mathcal{C}_M)\|^2\right] =: M_{d,p,M}(\|x\|)$ is radial and $r\to M_{d,p,M}(r)$ is non-decreasing. Moreover, for any $M_0\geq 1$ there exists a constant $C_{M_0,R,d,p}$ such that for all $M\geq M_0$, $M_{d,p,M}(R)\leq C_{M_0,R,d,p}$.*

Proof. The fact that $x\mapsto\mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\operatorname{VQ}(x,\mathcal{C}_M)\|^2\right]$ is radial is a consequence of Lemma 1. We can thus denote $M_{d,p,M}$ this function. Moreover, by Lemma 4 for any $x\in\mathbb{R}^d$, $\lambda\in(0,1)$ and **almost surely** any codebook $\mathcal{C}_M$, we have that $\|\operatorname{VQ}(\lambda x,\mathcal{C}_M)\|^2 \leq \|\operatorname{VQ}(x,\mathcal{C}_M)\|^2$. (We apply Lemma 4 with $c_1=\operatorname{VQ}(x,\mathcal{C}_M)$ and $c_2=\operatorname{VQ}(\lambda x,\mathcal{C}_M)$). Consequently, for any $\lambda\in(0,1)$, we have $\mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\operatorname{VQ}(\lambda x,\mathcal{C}_M)\|^2\right] \leq \mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\operatorname{VQ}(x,\mathcal{C}_M)\|^2\right]$, and $M_{d,p,M}$ is non-decreasing.

To prove the second statement, we use the following decomposition: for any $x\in\mathbb{R}^d$ such that $\|x\|=R$, and a.s. any $\mathcal{C}_M$:

$$\begin{aligned}
\|\operatorname{VQ}(x,\mathcal{C}_M)\|^2 &\leq 2\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2 + 2\|x\|^2 \\
&\leq 2\|\operatorname{VQ}(x,\mathcal{C}_M)-x\|^2 + 2R^2.
\end{aligned}$$

Taking the expectation, and using again Lemma 1, we get

$$M_{d,p,M}(R) \leq 2\mathbb{E}_{\mathcal{C}_M \sim p}[\| \operatorname{VQ}(Re_1, \mathcal{C}_M) - Re_1\|^2] + 2R^2 \,.$$

We finally prove that, for a given $x \in \mathbb{R}^d$, $\mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M) - x\|^2\right]$ is non-increasing with M. Indeed, writing $\mathcal{C}_M = [C_1, \ldots, C_M] = \mathcal{C}_{M-1} \cup \{C_M\}$ (which amounts to define a **coupling** between $\mathcal{C}_M$ and $\mathcal{C}_{M-1}$), we obtain

$$\begin{aligned}\|\operatorname{VQ}(x, \mathcal{C}_M) - x\|^2 &= \min\left(\|\operatorname{VQ}(x, \mathcal{C}_{M-1}) - x\|^2, \|C_M - x\|^2\right) \\ &\leq \|\operatorname{VQ}(x, \mathcal{C}_{M-1}) - x\|^2 \,.\end{aligned}$$

□

In the next Lemma and the rest of the proof, to avoid the cumbersome notation $(r_M^p)^{-1}$ we omit the dependency on p in r_M^p and simply write r_M^{-1}. We now show that the quadratic error is uniformly decreasing on $\mathrm{B}_2(0; R)$ at speed $M^{-2/d}$.

Lemma 8. *Assume A 3-A 4-A 5. Then, for any $R > 0$ and $M_0 \geq 1$ there exists a constant $C_{M_0,R} < \infty$ such that, for all $M \geq M_0$,*

$$\sup_{x \in \mathrm{B}_2(0;R)} \mathbb{E}_{\mathcal{C}_M \sim p}\left[\left\|r_M^{-1}(\|x\|)\operatorname{VQ}(x, \mathcal{C}_M) - x\right\|^2\right] \leq C_{M_0,R} M^{-2/d}.$$

Proof. Using Lemma 1, the function $x \to \mathbb{E}_{\mathcal{C}_M \sim p}\left[\left\|r_M^{-1}(\|x\|)\operatorname{VQ}(x, \mathcal{C}_M) - x\right\|^2\right]$ is radial. For $x \in \mathbb{R}^d$, we get that

$$\begin{aligned}&\mathbb{E}_{\mathcal{C}_M \sim p}\left[\left\|r_M^{-1}(\|x\|)\operatorname{VQ}(x, \mathcal{C}_M) - x\right\|^2\right] \\ &\leq 2\mathbb{E}_{\mathcal{C}_M \sim p}\left[\left\|(r_M^{-1}(\|x\|) - 1)\operatorname{VQ}(x, \mathcal{C}_M)\right\|^2\right] + 2\mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M) - x\|^2\right] \\ &\leq 2(r_M^{-1}(\|x\|) - 1)^2\mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M)\|^2\right] + 2\mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M) - x\|^2\right] \,. \qquad (11)\end{aligned}$$

Set $M_0 \geq 1$. First, by Lemma 7, for any $M \geq M_0$, we get

$$\sup_{x \in \mathrm{B}_2(0;R)} \mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M)\|^2\right] \leq C_{M_0,R,d,p} \qquad (12)$$

Second, by Lemma 5, for any $x \in \mathrm{B}_2(0; R)$, we have $0 \leq (r_M^{-1}(\|x\|) - 1) \leq (r_M^{-1}(R) - 1)$, and by the remark following Theorem 2, we get:

$$\limsup_{M \to \infty} M^{1/d}|r_M(R) - 1| \leq C_d^{1/2} p_{\mathrm{rad}}^{-1/d}(R)/R \,,$$

thus

$$(r_M^{-1}(R) - 1)^2 = |r_M(R) - 1|^2 / r_M^2(R) = O(M^{-2/d}) \,. \qquad (13)$$

Thirdly, Lemma 6 gives that

$$\sup_{x \in \mathrm{B}_2(0;R)} \mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(x, \mathcal{C}_M) - x\|^2\right] = \mathbb{E}_{\mathcal{C}_M \sim p}\left[\|\operatorname{VQ}(Re_1, \mathcal{C}_M) - Re_1\|^2\right],$$

Using Theorem 2, we obtain

$$\lim_{M \to \infty} M^{2/d}\mathbb{E}_{\mathcal{C}_M \sim p}[\| \operatorname{VQ}(Re_1, \mathcal{C}_M) - Re_1\|^2] = C_d p_{\mathrm{rad}}^{-2/d}(R) \,. \qquad (14)$$

Plugging (12)-(13) and (14) into (11), we obtain that

$$\sup_{x \in \mathrm{B}_2(0;R)} \mathbb{E}_{\mathcal{C}_M \sim p}\left[\left\|r_M^{-1}(\|x\|)\operatorname{VQ}(x, \mathcal{C}_M) - x\right\|^2\right] = O(M^{-2/d}) \,.$$

□

A random scalar quantizer is defined by a scalar output codebook, $\mathcal{O}_Q = [o_1, \dots, o_Q]$ assumed to be ordered $o_1 < \cdots < o_Q$ and an external randomization, which may be taken without loss of generality as $U \sim \mathrm{Unif}([0,1])$; see Appendix B.1 for a construction. Here Q is the number of codewords and the number of bits required to encode the output vectors (the scalar quantizer rate) is $\log_2(Q)$. A scalar quantizer is said to be *uniform* if for all $i \in [Q-1]$, $o_{i+1} - o_i = \delta$ for some $\delta > 0$. It is shown in Appendix B.1 that, for all $L > 0$, we may contruct a uniform random scalar quantizer with Q codewords, satisfying for all $x \in [0, L]$,

$$\mathbb{E}_{U\sim\mathrm{Unif}([0,1])}[\mathrm{SQ}(x, L, Q, U)] = x \tag{15}$$

$$\mathbb{E}_{U\sim\mathrm{Unif}([0,1])}[\{\mathrm{SQ}(x, L, Q, U) - x\}^2] \leq L^2/4(Q-1)^2\,. \tag{16}$$

We use this random scalar quantifier in the following proposition.

Proposition 1. *Assume A 3-A 4-A 5. Let $R > 0$ and $M_0 \geq 1$. For $Q \geq 2$, denote by $\omega_{M,Q}(R) = \max_{x\in \mathrm{B}_2(0;R)} \Omega_{M,Q}(x)$ where*

$$\Omega_{M,Q}(x) = \mathbb{E}_{\mathcal{C}_M\sim p, U\sim\mathrm{Unif}([0,1])}\left[\left\|\mathrm{SQ}(r_M^{-1}(\|x\|), r_M^{-1}(R), Q, U)\,\mathrm{VQ}(x, \mathcal{C}_M) - x\right\|^2\right]\,.$$

Then, there exists a constant $C_{M_0}(R)$, such that for all $M \geq M_0$,

$$\omega_{M,Q}(R) \leq C_{M_0}(R)\{M^{-2/d} + Q^{-2}\}.$$

Proof. It follows from Lemma 1 that the function $x \to \Omega_{M,Q}(x)$ is radial. If A is a scalar random variable, $\mathbf{B}$ is a random vector, A and $\mathbf{B}$ are independent, $\mathbb{E}[A^2] < \infty$, and $\mathbb{E}[\|\mathbf{B}\|^2] < \infty$, then

$$\mathbb{E}[\|A\mathbf{B} - \mathbb{E}[A]\mathbb{E}[\mathbf{B}]\|^2] = \mathbb{E}[(A - \mathbb{E}[A])^2]\mathbb{E}[\|\mathbf{B}\|^2] + \{\mathbb{E}[A]\}^2\mathbb{E}[\|\mathbf{B} - \mathbb{E}[\mathbf{B}]\|^2]\,.$$

Setting $A \leftarrow \mathrm{SQ}\left(r_M^{-1}(\|x\|)\right)$, $\mathbf{B} \leftarrow \mathrm{VQ}(x, \mathcal{C}_M)$, we get

$$\begin{aligned}\Omega_{M,Q}(x) &= \mathbb{E}_{\mathcal{C}_M\sim p}\left[\left\|r_M^{-1}(\|x\|)\,\mathrm{VQ}(x, \mathcal{C}_M) - x\right\|^2\right] \\ &+ \mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\mathrm{VQ}(x, \mathcal{C}_M)\|^2\right]\mathbb{E}_{U\sim\mathrm{Unif}([0,1])}\left[\left\{\mathrm{SQ}(r_M^{-1}(\|x\|), r_M^{-1}(R), Q, U) - r_M^{-1}(\|x\|)\right\}^2\right]\,.\end{aligned}$$

We now make the following observations:

1. The function $r_M^{-1}(\|x\|)$ is bounded on $\mathrm{B}_2(0;R)$ by $r_M^{-1}(R)$ (see Lemma 5). Hence,
$$\mathbb{E}_{U\sim\mathrm{Unif}([0,1])}\left[\left\{\mathrm{SQ}(r_M^{-1}(\|x\|), r_M^{-1}(R), Q, U) - r_M^{-1}(\|x\|)\right\}^2\right] \leq r_M^{-2}(R)/4(Q-1)^2.$$
2. The second order moment $\mathbb{E}_{\mathcal{C}_M\sim p}\left[\|\mathrm{VQ}(x, \mathcal{C}_M)\|^2\right]$ is upper bounded by a constant independent of M on $\mathrm{B}_2(0;R)$ by Lemma 7.
3. Using Lemma 8, we can upper bound the first term for any $M \geq M_0$
$$\mathbb{E}_{\mathcal{C}_M\sim p}\left[\left\|r_M^{-1}(\|x\|)\,\mathrm{VQ}(x, \mathcal{C}_M) - x\right\|^2\right] \leq C_{M_0,R}M^{-2/d}\,.$$

This concludes the proof. □

We now give the proof of the main result Theorem 4.

Proof of Theorem 4. We consider the process described in `DoStoVoQ`. For $t \geq 0$, $k \in [K]$, we define $(\hat{b}^\ell_{k,t+1})_{\ell\in[L]}$ such that $\hat{g}_{k,t+1} = \|g_{k,t+1}\|D^{-1/2}(\hat{b}^1_{k,t+1}, \dots, \hat{b}^L_{k,t+1})$, where for any $\ell \in [L]$ we have $\hat{b}^\ell_{k,t+1} = \texttt{StoVoQ}(b^\ell_{k,t+1}, p, M, P, s_{k,t+1})$.

1. Conditional independence property. We observe that for all $k \in [K]$, $g_{k,t+1}$ is $\mathcal{G}_{t+1}$-measurable. Moreover, the seeds $(s_{k,t+1})_{k\in[K],t\geq 0}$ are independent, and there exists a functional ϕ such that for all $k \in [K]$, $\hat{g}_{k,t+1} = \phi(g_{k,t+1}, s_{k,t+1})$. We conclude that the compressed stochastic gradients $(\hat{g}_{k,t+1})_{k\in[K]}$ are mutually independent conditionally to $\mathcal{G}_{t+1}$.

2. Unbiasedness. In the sequel, we fix $t \geq 0$, $k \in [K]$. Regarding the bias, we use the fact that for any $\ell \in [L]$, $b^\ell_{k,t+1}$ is $\mathcal{G}_{t+1}$-measurable. Moreover, as $\hat{b}^\ell_{k,t+1} = \texttt{StoVoQ}(b^\ell_{k,t+1}, p, M, P, s_{k,t+1})$, we have that $\mathbb{E}_{\mathcal{C}_M \sim p}[\hat{b}^\ell_{k,t+1} | \mathcal{G}_{t+1}] = b^\ell_{k,t+1}$, using the fact that `StoVoQ` is unbiased.

Consequently, $\mathbb{E}_{\mathcal{C}_M \sim p}\left[\|g_{k,t+1}\|/\sqrt{D} \times \left(\hat{b}^1_{k,t+1}, \ldots, \hat{b}^L_{k,t+1}\right)_{\ell \in [L]} \,\middle|\, \mathcal{G}_{t+1} \right] = g_{k,t+1}$.

3. Relative error bound. We write:

$$\mathbb{E}\left[\|\hat{g}_{k,t+1} - g_{k,t+1}\|^2 \,\middle|\, \mathcal{G}_{t+1} \right] = \frac{\|g_{k,t+1}\|^2}{D} \sum_{\ell \in [L]} \mathbb{E}\left[\|\hat{b}^\ell_{k,t+1} - b^\ell_{k,t+1}\|^2 \,\middle|\, \mathcal{G}_{t+1} \right].$$

Remark that $\sum_{\ell \in [L]} \|b^\ell_{k,t+1}\|^2 = D$. Consequently, for all $\ell \in [L]$, $b^\ell_{k,t+1} \in \mathrm{B}_2(0; \sqrt{D})$. Using Proposition 1, with $R = D$, we get:

$$\mathbb{E}\left[\|\hat{g}_{k,t+1} - g_{k,t+1}\|^2 \,\middle|\, \mathcal{G}_{t+1} \right] = \frac{\omega_{M,Q}(R)}{D} \|g_{k,t+1}\|^2$$

which concludes the proof.

□

B Scalar and vector Quantization

B.1 Unbiased random scalar quantization

A random scalar quantizer is a random map from the real line to a (scalar) codebook $\mathcal{O}_Q = \{o_1, \ldots, o_Q\} \subset \mathbb{R}$ where $Q \geq 2$. It is assumed that $-\infty < o_1 < \cdots < o_Q < \infty$. The resolution (or code rate) is $P = \log_2(Q)$ is the number of bits needed to uniquely specify a codeword. A scalar quantizer is said to be *uniform* if for all $i \in [Q-1]$, $o_{i+1} - o_i = \delta$, for some $\delta > 0$. Note that in such case $\delta = \{o_Q - o_1\}/(Q-1)$.

For $x \in \mathbb{R}$ and $u \in [0,1]$, consider a function $\mathrm{SQ}(x, \mathcal{O}_Q, u) \in \mathcal{O}_Q$. If $U \sim \mathrm{Unif}([0,1])$, then $\mathrm{SQ}(x, \mathcal{O}_Q, U)$ is a random scalar quantizer. A random scalar quantizer is said to be *unbiased* if for all $x \in [o_1, \ldots, o_Q]$, $\mathbb{E}_{U \sim \mathrm{Unif}([0,1])}[\mathrm{SQ}(x, \mathcal{O}_Q, U)] = x$.

A simple way to construct an unbiased scalar quantizer goes a follows. We first compute the index $j(x) \in [Q]$ such that $x \in \left[o_{j(x)}, o_{j(x)+1}\right)$. Note that $x = \lambda^*_{j(x)}(x) o_{j(x)} + (1 - \lambda^*_{j(x)}(x)) o_{j(x)+1}$ where

$$\lambda^*_{j(x)}(x) = (x - o_{j(x)})/(o_{j(x)+1} - o_{j(x)}) \in (0,1] \,.$$

For $u \in (0,1]$, we set

$$\mathrm{SQ}(x, \mathcal{O}_Q, u) = \mathbb{1}_{\{u \leq \lambda^*_{j(x)}(x)\}} o_{j(x)} + \mathbb{1}_{\{u > \lambda^*_{j(x)}(x)\}} o_{j(x)+1} \,.$$

Since $\mathbb{E}_{U \sim \mathrm{Unif}([0,1])}(U \leq \lambda_j(x)^*(x)) = \lambda^*_{j(x)}$ the unbiasedness follows. It is easily seen that the distortion of a scalar quantizer is directly related to the diameter of the quantizer.

Proposition 2. *For all $x \in [o_1, o_Q]$, it holds that*

$$\mathbb{E}_{U \sim \mathrm{Unif}([0,1])}[\{\mathrm{SQ}(x, \mathcal{O}_Q, U) - x\}^2] \leq (1/4) \sup_{i \in [Q-1]} \{o_{i+1} - o_i\}^2 \,.$$

If the scalar quantizer is uniform,

$$\mathbb{E}_{U \sim \mathrm{Unif}([0,1])}[\{\mathrm{SQ}(x, \mathcal{O}_Q, U) - x\}^2] \leq (1/4)(Q-1)^{-2} \{o_Q - o_1\}^2 \,.$$

Proof. For all $x \in [o_1, o_Q]$, we get

$$|\,\mathrm{SQ}(x, \mathcal{O}_Q, U) - x| \leq (1/2)\{o_{j(x)+1} - o_{j(x)}\}$$

The proof follows. □

Unbiased random scalar quantization is a special case of dual vector quantization, introduced in the next section.

B.2 Dual Vector Quantization

We introduce a new notion of vector quantization, called *dual quantization* (or *Delaunay quantization*). The principle of dual quantization is to map an $\mathbb{R}^d$-valued vector x onto a codebook $\mathcal{C}_M$ using a random splitting operator $\text{Dual-VQ}(x, \mathcal{C}_M, U)$ such that, for all $x \in \text{ConvHull}(\mathcal{C}_M)$,

$$\mathbb{E}_{U\sim\text{Unif}([0,1])}[\text{Dual-VQ}(x, \mathcal{C}_M, U)] = x\,. \tag{17}$$

We stress that in this case the unbiasedness is not due to the use of a random codebook but makes use of an external randomization. In practice, a dual quantizer procedure amounts to define a probability distribution of $\mathcal{C}_M$, with weights $(\lambda_1^*(x), \ldots, \lambda_M^*(x))$, $\lambda_i^*(x) \geq 0$, $\sum_{j=1}^M \lambda_j^*(x) = 1$. Set $\Lambda_0^*(x) = 0$ and for $i \in [M]$, $\Lambda_i^*(x) = \sum_{j=1}^i \lambda_j^*(x)$. Note that $\Lambda_M^*(x) = 1$. If $u \in \left(\Lambda_{j-1}^*(x), \Lambda_j^*(x)\right]$, $j \in [M]$, we set $\text{Dual-VQ}(x, \mathcal{C}_M, u) = c_j$. In such that, for all $x \in \text{ConvHull}(\mathcal{C}_M)$, we get

$$\mathbb{E}_{U\sim\text{Unif}([0,1])}[\text{Dual-VQ}(x, \mathcal{C}_M, U)] = \sum_{i=1}^M \lambda_i^*(x) c_i = x\,.$$

The distortion of a dual quantizer is therefore given, for $x \in \mathbb{R}^d$, by

$$\mathbb{E}_{U\sim\text{Unif}([0,1])}[\|\,\text{Dual-VQ}(x, \mathcal{C}_M, U) - x\|^2] = \sum_{i=1}^M \lambda_i^*(x)\|x - c_i\|^2\,. \tag{18}$$

For $x \in \text{ConvHull}(\mathcal{C}_M)$, the probability distribution $(\lambda_1^*(x), \ldots, \lambda_M^*(x))$ is obtained by solving the following convex optimization program:

$$(\lambda_1^*(x), \ldots, \lambda_M^*(x)) = \text{argmin}_{(\lambda_1,\ldots,\lambda_M)\in\mathcal{S}(x,\mathcal{C}_M)} \sum_{i=1}^M \lambda_i\|x - c_i\|^2\,, \tag{19}$$

where

$$\mathcal{S}(x, \mathcal{C}_M) = \left\{(\lambda_1, \ldots, \lambda_M) \in \mathbb{R}_+^M, \sum_{i=1}^M \lambda_i = 1, \sum_{i=1}^M \lambda_i c_i = x\right\}\,. \tag{20}$$

The support of $(\lambda_1^*(x), \ldots, \lambda_M^*(x))$ is $M + 1$ at most. For a distribution q on $\mathbb{R}^d$, we define

$$\text{Dual-Dist}(q, \mathcal{C}_M) = \int q(x)\left\{\sum_{i=1}^M \lambda_i^*(x)\|x - c_i\|^2\right\}\mathrm{d}x\,. \tag{21}$$

For a given input distribution q, an optimal codebook $\mathcal{C}_M^*$ of cardinality M satisfies $\text{Dual-Dist}(q, \mathcal{C}_M^*) \leq \text{Dual-Dist}(q, \mathcal{C}_M)$ for all $\mathcal{C}_M$ satisfying $|\mathcal{C}_M| = M$.

Theorem 5 (Rates, see [27])**.** Asymptoptic rate. *Assume that the pdf q is compactly supported on $\mathbb{R}^d$-valued.*

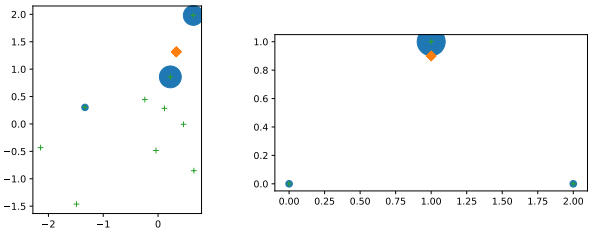
$$\lim_{M\to\infty} M^{\frac{2}{d}} \inf_{|\mathcal{C}_M|=M} \text{Dual-Dist}(q, \mathcal{C}_M) =: Q_2^D(q) = Q_2^D(\text{Unif}([0,1])^d)\big\|q\big\|_{\frac{d}{d+2}}\,. \tag{22}$$

B.3 HSQ methods - see Dai et al. [8]

In this Section, we provide a detailed review of the two methods proposed by Dai et al. [8]. In Appendix B.3.1, we first discuss `HSQ-Span` and explain why it cannot compete with approaches based on Voronoi quantization. In Appendix B.3.2, we discuss `HSQ-greed`.

B.3.1 `HSQ-Span`

The first method, `HSQ-Span`, is unbiased but suffers form a large variance. Indeed, it relies on decomposing the vector $x \in \mathbb{R}^d$ as a **linear** combination of the codewords in $\mathcal{C}_M$, assuming that $\text{Span}\{c_i, i \in [M]\} = \mathbb{R}^d$ (a codebook satisfying this property is said to be *full-rank*). Because typically $M \gg d$, there are infinitely many solutions to the linear problem $\sum_{i=1}^M \alpha_i c_i = x$, i.e. $\mathcal{A}(x, \mathcal{C}_M) = \{(\alpha_1, \ldots, \alpha_M) \in \mathbb{R}^M, \sum_{i=1}^M \alpha_i c_i = x\}$ is infinite. Note that contrary to the Dual quantization approach, we do not assume that $\alpha_i \geq 0$ for $i \in [M]$ or $\sum_{i\in[M]} \alpha_i = 1$. However, for



(a) $x \sim \mathcal{N}(0, \mathrm{I}_2)$ and $M = 10$. (b) $x \sim \mathcal{N}(0, \mathrm{I}_2)$ and $M = 3$.

Figure 3: `Delaunay quantization` for a vector x (orange diamond), for a given set of codewords (green +),and corresponding weights (area of the blue spheres). Remark that all but three points have a 0 probability of being picked, making the quadratic error much smaller than for `HSQ-span`.

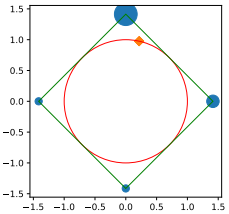


Figure 4: `Cross-Polytope` [10] is a particular case of `Delaunay quantization`. The codewords are the vertices of $\mathrm{B}_1(0; \sqrt{d})$. A vector x (orange diamond) lying on the unit Ball $\mathrm{B}_2(0; 1)$ (red circle) is decomposed with weights (area of the blue spheres) of codewords on the Ball of radius $\sqrt{d}$ (green).

any $i \in [M]$, we pick the codeword c_i with probability $|\alpha_i|/\|\alpha\|_1$, and encode x as $\mathrm{sign}(\alpha_i)\|\alpha\|_1 c_i$. In `HSQ-Span`, the **minimal norm** solution in $\mathcal{A}(x, \mathcal{C}_M)$ is chosen, i.e. solve

$$\alpha^*(x) := (\alpha_1^*(x), \dots, \alpha_M^*(x)) = \mathrm{argmin}_{(\alpha_1,\dots,\alpha_M)\in\mathcal{A}(x,\mathcal{C}_M)} \sum_{i=1}^{M} \alpha_i^2 \,, \tag{23}$$

The main advantage are that

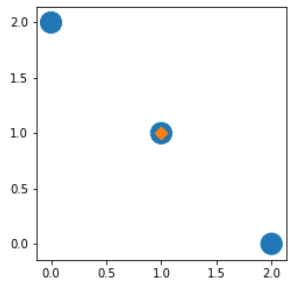
1. **Fast computation.** First, as $\alpha^*(x) = C^\dagger x$, where $C^\dagger$ is the Moore-Penrose inverse of the codewords matrix $C = [c_1, \dots, c_M]$, provided a fixed codebook $\mathcal{C}_M$, it is possible to compute only once $C^\dagger$ and to then obtain $\alpha^*(x)$ for any x by a simple matrix-vector product.
2. **Unbiased.** Second, this approach is unbiased. Its quadratic error thus linearly decays with the number of workers.

However (1) its variance is high and (2) does not decrease with M. Indeed, the minimal norm solution $\alpha^*(x)$ tends to put weight on **all** codewords. For example, we represent in Figure 6 the weights on each vector for 3 situations in dimension $d = 2$. Intuitively, the probability of of selecting c_i is not a decreasing function $\|x - c_i\|^2$ (see e.g., Figure 6b), which results in the large variance; even if there exists i such that $c_i = x$, there is a non vanishing probability of selecting $c_j \neq c_i$ s (Figure 6a). We illustrate the second point in Figure 5 which gives the evolution of the distortion for $d = 16$ w.r.t. M for $K \in \{1, 8\}$ workers. The error does not decrease.

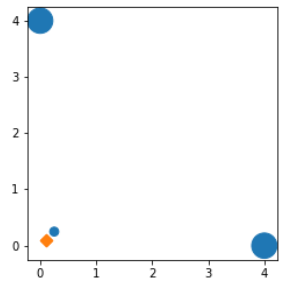
B.3.2 `HSQ-greed`

`HSQ-greed` is closed to `DoStoVoQ`: Dai et al. [8] still consider a full-rank codebook $\mathcal{C}_M$, and simply encode x by $\mathrm{VQ}(x, \mathcal{C}_M)$. We list here the main differences to our approach:

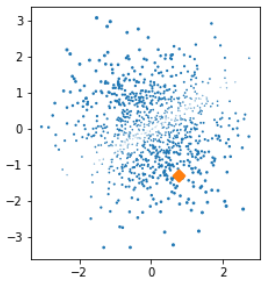
1. the same codebook is used during all iterations and on all workers. This makes it impossible or cumbersome to apply the convergence result developed in the federated learning literature, which require that the **compression on each workers are independent** (at least between iterations).
2. No assumption is made on the codebook distribution (apart from the fact that it is full-rank). The importance of unitary invariance is not mentioned. In practice, authors use an codebook generated by applying a k-means algorithm on a larger set of scaled Gaussian isotropic vectors. This pre-processing slightly improves the distribution of the codewords but is in practice of limited impact (see paragraph (e) in Subsection 2.2 in the main text.).



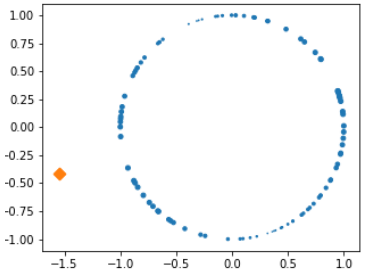
(a) $x_1 = (1, 1)$,
$\mathcal{C}_M = [(2, 0); (0, 2); (1, 1)]$.
$\alpha^*(x_1) \simeq (33\%, 33\%, 33\%)$.



(b) $x_2 = (0.1, 0.1)$,
$\mathcal{C}_M = [(4, 0); (0, 4); (1/4, 1/4)]$
$\alpha^*(x_2) \simeq (47\%, 47\%, 6\%)$.



(c) $x \sim \mathcal{N}(0, \mathrm{I}_2)$ and $M = 1000$.



(d) $x \sim \mathcal{N}(0, \mathrm{I}_2)$ and $M = 100$, $p = \mathcal{U}(\mathcal{S}_1(\mathbb{R}^2))$.

Figure 6: `HSQ-Span`: weights (size of the blue point) on each of the codewords of $\mathcal{C}_M$ when decomposing x (orange diamond) .

3. Codewords are chosen of norm 1. This means we also need to encode $\|x\|$ together with $\mathrm{VQ}(x, \mathcal{C}_M)$, which is typically done on using 6 bits per bucket.
4. The method is biased, so does not benefit from a large number of workers. No analysis of the quadratic error is provided.

Theoretical results. Dai et al. [8] present a convergence result for `HSQ-greed`, namely in Lemma 3 and the subsequent Theorem 3. Note however that the proof of this result is **not** provided in the paper[3]. Second, the guarantee provided is almost vacuous. Indeed, authors rely on an alternative assumption[4] on the **alignment** of the compressed value $\mathrm{VQ}(x, \mathcal{C}_M)$ with x:

Definition 5 (Compression with preserved alignement)**.** *There exists $\alpha > 0$ such that for all $x \in \mathbb{R}^d$, we have $\langle \mathrm{Comp}(x), x \rangle^2 \geq (1 - \alpha)\|x\|^2$.*

This assumption becomes stronger as $1 - \alpha$ increases. However, Lemma 3 indicates that $1 - \alpha \geq \sigma_{\min}(C)/M$, with $\sigma_{\min}$ the minimal

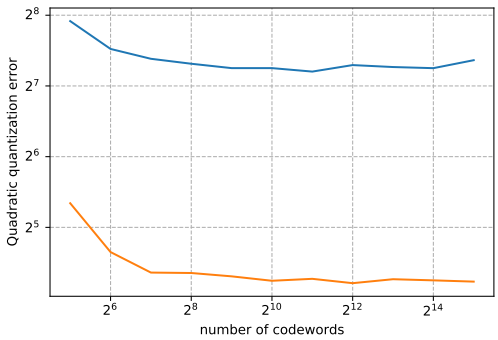


Figure 5: `HSQ-Span`: Distortion as a function of M (log-scale): $K = 1$ (blue) $K = 8$ (orange).

[3]The appendices of the paper were not available, neither on `https://arxiv.org/pdf/1911.04655.pdf`, nor on `https://paperswithcode.com/paper/hyper-sphere-quantization-communication`, on the 1st of June, 2021.

[4]See the work of Beznosikov et al. [5] for a discussion between the possible assumptions.

eigenvalue of the codebook matrix C. The bound guarantee thus worsens with M. A similar multiplicative factor $1/(1-\alpha) \propto M$ appears in their convergence rate Theorem 3. We note that without any assumption on the codebook distribution, it seems difficult to obtain any result, as the worst case codebook that satisfies the full-rank assumption could be arbitrarily bad (typically a unique codeword perturbed by a tiny amount of noise $\mathcal{C}_M = [c_1 + \eta\epsilon_i]_{i\in[M]}$, with η very small).

B.4 Alignment under A 3, for `StoVoQ` without debiasing function (new)

We can leverage our tight analysis of unitarily invariant distribution to obtain a result on the expected alignment between a vector $x \in \mathbb{R}^d$ and the output of `StoVoQ` (without bias removal) applied on x. More precisely, we have the following lemma:

Lemma 9. *Assume Lemma 1, then :*

$$\mathbb{E}_{\mathcal{C}_M\sim p}[\langle x, \mathrm{VQ}(x, \mathcal{C}_M)\rangle] \geq r_M^p(\|x\|)\|x\|^2.$$

We proved that on a ball of radius R, $r_M^p(\|x\|) \geq r_M^p(R)$ and that $r_M^p(R) = 1 - O(M^{-2d})$: in other words our guarantee **does** improve with M. This result is thus much stronger than the one of [8]. Note however that without debiasing, it is not possible to directly leverage the literature on federated learning: our result is only on the *expected* alignment and it would require novel proofs (relying on 1. expected alignment and 2. Bounded variance) to give a convergence result.

C Unitarily invariant random codebooks

We gather in this section the theoretical results on random codebooks distributed according to a unitarily invariant distribution. In Appendix C.1 we provide a proof of Theorem 2. In Appendix C.2, we show that random codebook are asymptotically optimal when the distribution of inputs is also unitarily invariant, provided that the codebook distribution is appropriately chosen. In Appendix C.3, we state an elementary lower bound. In Appendix C.4, we extend our results for spherical codebooks. **This extension is not mentionned in the main document**.

C.1 Proof of Theorem 2

Let $\{C_i\}_{i=1}^\infty$ be an i.i.d. sequence of random vector with pdf p w.r.t. the Lebesgue measure. For each $n \in \mathbb{N}$, denotes by $\mathcal{C}_M = \{C_1, \ldots, C_M\}$ the associated sequence of codebook. Define for $k > 0$ and $\lambda > 0$, the pdf of the Weibull distribution ($\mathrm{Weibull}(k, \lambda)$) with shape parameter k and scale parameter λ is given, for $x \geq 0$, by:

$$w_{k,\lambda}(x) = (k/\lambda)(x/\lambda)^{k-1}\mathrm{e}^{-(x/\lambda)^k}\,. \tag{24}$$

The survival function of $\mathrm{Weibull}(k, \lambda)$ is denoted by $\bar{W}_{k,\lambda}(t) = \mathrm{e}^{-(t/\lambda)^k}$. Denote by V_d the volume of the unit ball in $\mathbb{R}^d$,

$$V_d = \pi^{d/2}/\Gamma(d/2+1)\,. \tag{25}$$

We preface the proof by a lemma.

Lemma 10. *Assume A 3-A 4. Then, for every $x \in \mathbb{R}^d$ and $t \geq 0$,*

$$\lim_{M\to\infty} \mathbb{P}_{\mathcal{C}_M\sim p}(M^{1/d}\|\,\mathrm{VQ}(x, \mathcal{C}_M) - x\| > t) = \bar{W}_{d,(V_d p(x))^{-1/d}}(t)\,.$$

Proof. We get for $t \geq 0$,

$$\begin{aligned}\mathbb{P}_{\mathcal{C}_M\sim p}\big(M^{1/d}\min_{i=1:M}\|C_i - x\| \geq t\big) &= \Big(1 - \mathbb{P}_{C_1\sim p}(\|C_1 - x\| \leq tM^{-1/d})\Big)^M \\ &= \Big(1 - P\big(\mathrm{B}(x; tM^{-1/d})\big)\Big)^M\end{aligned}$$

where for $A \subset \mathbb{R}^d$ a Borel set, $P(A) = \int p(x)\mathbb{1}_A(x)\mathrm{d}x$. It follows from the Lebesgue differentiation theorem that

$$P\big(\mathrm{B}(x; t\,M^{-1/d})\big) \sim_{M\to\infty} p(x)\,\mathrm{Leb}_d\Big(\mathrm{B}(x; t\,M^{-1/d})\Big) = p(x)V_d t^d M^{-1}$$

where Leb_d denotes the Lebesgue measure. Hence, for any $t \geq 0$

$$\mathbb{P}_{\mathcal{C}_M \sim p}\big(M^{1/d} \min_{i=1:M} \|C_i - x\| \geq t\big) \longrightarrow_{M\to\infty} \bar{W}_{d,\{V_d p(x)\}^{-1/d}}(t) = \mathrm{e}^{-p(x)V_d t^d}\,.$$

□

Proof of Theorem 2. The proof relies on Lemma 10 and on the uniform integrability of the sequence $\big(M^{1/d}\min_{i=1:M}\|C_i - x\|\big)^2$, $M \geq 1$. Let $R > 0$.

$$\mathbb{E}_{\mathcal{C}_M\sim p}\left[M^{2/d}\min_{i\in[M]}\|C_i - x\|^2 \mathbb{1}_{\{M^{2/d}\min_{i=1:M}\|C_i-x\|^2 \geq R\}}\right] = \int_R^\infty \Big\{1 - P\big(\mathrm{B}(x; M^{-1/d}t^{1/2})\big)\Big\}^M \mathrm{d}t$$

By Anderson's inequality (see [11]) $P\big(\mathrm{B}(x; M^{-1/d}t^{1/2})\big) \geq P\big(\mathrm{B}(0; M^{-1/d}t^{1/2})\big)$ so that

$$\int_R^{+\infty}\Big\{1 - P\big(\mathrm{B}(x; M^{-1/d}t^{1/2})\big)\Big\}\,\mathrm{d}t \leq \int_R^{+\infty} \underbrace{\Big(1 - P\big(\mathrm{B}(0; M^{-1/d}t^{1/2})\big)\Big)^M}_{=:\phi_M(t)}\,\mathrm{d}t.$$

Let $\rho \in (0,1)$. Decompose $\phi_M(t) = A_M(t) + B_M(t)$ with

$$A_M(t) = \Big\{1 - P\big(\mathrm{B}(0; M^{-1/d}t^{1/2})\big)\Big\}^M \mathbb{1}_{\{M^{-1/d}t^{1/2} > t^{\rho/2}\}}$$
$$B_M(t) = \exp\left(-M\Big[\frac{P(\mathrm{B}(0; M^{-1/d}t^{1/2}))}{\mathrm{Leb}_d\big(\mathrm{B}(0; M^{-1/d}t^{1/2})\big)} V_d\, M^{-1/d} t^{d/2}\Big]\right)\mathbb{1}_{\{M^{-1/d}t^{1/2} \leq t^{\rho/2}\}}.$$

Note that

$$A_M(t) \leq 1 - P\big(\mathrm{B}(0; t^{\rho/2})\big).$$

Now let $\delta > 0$. We upper-bound $B_M(t)$ as follows

$$\begin{aligned} B_M(t) &\leq \exp\left(-\inf_{s\in(0,\delta]} \frac{P\big(\mathrm{B}(0;s)\big)}{\mathrm{Leb}_d\big(\mathrm{B}(0;s)\big)} V_d t^{d/2}\right) + \exp\left(-\inf_{s\in\big(\delta, t^{\rho/2}\big]} \frac{P\big(\mathrm{B}(0;s)\big)}{\mathrm{Leb}_d\big(\mathrm{B}(0;s)\big)} V_d t^{d/2}\right) \\ &\leq \exp\left(-\inf_{s\in(0,\delta]} \frac{P\left(\mathrm{B}(0;s)\right)}{\mathrm{Leb}_d\big(\mathrm{B}(0;s)\big)} V_d t^{d/2}\right) + \exp\Big(-P\left(\mathrm{B}(0;\delta)\right) t^{d(1-\rho)/2}\Big)\,. \end{aligned}$$

Let us denote $B_1(t)$ and $B_2(t)$ the two terms on the right hand side of the previous equation. Note that $B_i(t)$, $i = 1, 2$ do not depend on M. Now let us show that these functions are integrable.

$$\int_0^{+\infty} A_M(t)dt \leq \int_0^{+\infty}\Big\{1 - P(\mathrm{B}(0; t^{\rho/2}))\Big\}\,\mathrm{d}t = \int_0^{+\infty} \mathbb{P}(\|C_1\| > t^{\rho/2})\mathrm{d}t = \mathbb{E}[\|C_1\|^{2/\rho}] < +\infty\,.$$

For the next two terms we use the elementary inequality $\inf_{s\in(0,\delta]} \frac{P(\mathrm{B}(0;s))}{\lambda_d(\mathrm{B}(0;s))} \geq m_\delta$ so that

$$\int_0^{+\infty} B_1(t)\mathrm{d}t \leq \int_0^{+\infty} \mathrm{e}^{-V_d m_\delta t^{d/2}}\mathrm{d}t < +\infty$$

and

$$\int_0^{+\infty} B_2(t)\mathrm{d}t \leq \int_0^{+\infty} \mathrm{e}^{-m_\delta V_d t^{d(1-\rho)/2}} < +\infty.$$

Consequently, $\lim_{R\to+\infty}\sup_M \int_R^{+\infty}\phi_M(t)\mathrm{d}t = 0$ which ensures uniform integrability.

We conclude by using that the second moment of a $\mathrm{Weibull}(k,\lambda)$ is given by $\lambda^2\Gamma(1+2/k)$ and that $V_d = \pi^{d/2}/\Gamma(1+d/2)$. □

C.2 Proof of Theorem 3

The proof of Theorem 3 follows almost immediately from Theorem 2 using the classical Zador theorem, stated for completeness below (see [14] for a proof of Zador theorem, which has a long history).

Theorem 6 (Zador's Theorem)**.**

- *Assume that* $\int \|x\|^{r+\delta} p(x)\mathrm{d}x < \infty$*, for some* $\delta > 0$*. Then,*

$$\lim_{M\to\infty} M^{2/d}\,\mathrm{Dist}(p, \mathcal{C}_M^*) = Q_2(p) = Q_2([0,1]^d)\left(\int p^{d/(d+2)}(x)\mathrm{d}x\right)^{(d+2)/d}, \qquad (26)$$

  where $Q_2(p)$ *is the quantization coefficient of the distribution* p *and* $Q_2([0,1]^d)$ *that of the uniform distribution over ther unit hypercube,* $\mathrm{Unif}([0,1]^d)$*. If the distribution* p *is standard normal* $\mathcal{N}(0, \mathrm{I}_d)$*, then* $Q_2\big(\mathcal{N}(0,\mathrm{I}_d)\big) \sim_{d\to\infty} d$*.*

- *There exists a universal constant* $C_{d,r+\delta} \in (0,\infty)$ *such that, for any pdf* p *on* $\mathbb{R}^d$

$$\mathrm{Dist}(p, \mathcal{C}_M^*) \le C_{d,r\delta}\sigma_{r+\delta}^r M^{-r/d}$$

  where $\sigma_{r+\delta}(p) =: \inf_{a\in\mathbb{R}^d}\big(\int |x|^{r+\delta}|x|p(x)\mathrm{d}x\big)^{\frac{1}{r+\delta}}$.

C.3 An elementary lower-bound

The Hölder inequality with negative exponents (see [16, p. 191]) shows that for $0 < r < 1$ and $s \in \mathbb{R}$ such that $r^{-1} + s^{-1} = 1$ (hence $s < 0$), $\int p^{-2/d}(x)q(x)\mathrm{d}x \ge \big\{\int p^{-2s/d}(x)\mathrm{d}x\big\}^{1/s}\big\{\int q^r(x)\mathrm{d}x\big\}^{1/r}$. Setting $s = -d/2$ and $r = 2/(d+2)$, we get that $\mathrm{C}(q,p,d) \ge \|q\|_{d/(d+2)}$.

C.4 Asymptotic distortion of a random quantizer on the unit sphere S_{d-1}

We now consider random codebooks on the unit hypesphere $S_{d-1} = \{x \in \mathbb{R}^d, \|x\| = 1\}$. We compute the distortion of a codebook distributed uniformly on the unit-sphere as a function of the number of codewords M and of the ambient dimension d. Denote by σ_{d-1} the uniform distribution on S_{d-1}. Denote

$$\kappa_d = \Big(\frac{2\sqrt{\pi}\,\Gamma((d+1)/2)}{\Gamma(d/2)}\Big)^{1/(d-1)} \qquad (27)$$

Theorem 7. *Assume* $d \ge 2$ *and assume that the codewords* $\{C_n\}_{n\ge1}$ *are i.i.d. uniformly distributed on the unit hyper-sphere* S_{d-1} *of* $\mathbb{R}^d$*. For every* $x \in S_{d-1}$*, and* $t \ge 0$

$$\lim_{n\to\infty} \mathbb{P}_{\mathcal{C}_M\sim\sigma_{d-1}}(M^{1/(d-1)}\|\,\mathrm{VQ}(x,\mathcal{C}_M) - x\| \ge t) = \bar{W}_{d-1,\kappa_d}(t)\,, \quad .$$

Furthermore,

$$\lim_{M\to\infty} M^{2/(d-1)}\,\mathbb{E}_{\mathcal{C}_M\sim\sigma_{d-1}}\big[\|\,\mathrm{VQ}(x,\mathcal{C}_M) - x\|^2\big] = \kappa_d^2\,\Gamma\big(1 + 2/(d-1)\big).$$

Proof. Since the uniform distribution over S_{d-1} is unitary invariant, we get for all $x \in S_{d-1}$,

$$\mathbb{P}_{C_1\sim\sigma_{d-1}}\big(C_1 \in \mathrm{B}(x;r)\big) = \frac{\sigma_{d-1}\big(\mathrm{B}(x;r)\cap S_{d-1}\big)}{\sigma_{d-1}(S_{d-1})} \sim_{r\to0^+} \frac{\mathrm{Leb}_{d-1}(\mathrm{B}(0;r))}{\sigma_{d-1}(S_{d-1})}$$

and, using that $\sigma_{d-1}(S_{d-1}) = 2\pi^{d/2}/\Gamma(d/2)$ we get

$$\frac{\mathrm{Leb}_{d-1}(\mathrm{B}(0;r))}{\sigma_{d-1}(S_{d-1})} = \frac{V_{d-1}r^{d-1}}{\sigma_{d-1}(S_{d-1})} = \frac{\pi^{\frac{d-1}{2}}r^{d-1}}{\Gamma(\frac{d-1}{2}+1)}\frac{\Gamma(\frac{d}{2})}{2\pi^{\frac{d}{2}}} = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d+1}{2})}\frac{r^{d-1}}{2\sqrt{\pi}} = \big(r/\kappa_d\big)^{d-1}.$$

Taking $r = M^{-1/(d-1)}t$ yields that

$$\mathbb{P}_{\mathcal{C}_M\sim\sigma_{d-1}}\big(M^{-1/(d-1)}\|\,\mathrm{VQ}(x,\mathcal{C}_M) - x\| > t\big) = \Big(1 - \mathbb{P}_{C_1\sim\sigma_{d-1}}\big(C_1 \in \mathrm{B}(0; M^{-1/(d-1)}t)\big)\Big)^M$$
$$\longrightarrow_{M\to\infty} \mathrm{e}^{-(t/\kappa_d)^{d-1}}.$$

This completes the first part of the proof. For the second part of the proof, it is required to check the uniform integrability which follows from the fact that the above equivalence ($\sim$) also holds as a lower bound inequality. □

D Algorithmic extensions

D.1 Spherical codebooks

In this section, we describe a spherical version of StoVoQ and DoStoVoQ. Beyond the obvious change from the codeword distribution from Gaussian to uniform on the sphere, a key modification stems form the fact that each quantized vector has norm 1: the debiasing function does not depend on $\|x\|$, but only on the number of codewords M. Consequently, the bias correction does not need to be transmitted and can be directly performed on the central server.

On the other hand, the norm of each bucket has to be transmitted: the vector quantization is applied to the *shape*, i.e. the unitary vector $x/\|x\|$. We use a scalar quantizer for the norm, typically over 4-6 bits. For completeness, the codes of those two algorithms are given in Algorithms 3 and 4.

Algorithm 3: Spherical-StoVoQ

Input : $x \in \mathbb{R}^d$, d, M, P, seed s
Output: Codeword index $\mathbf{i}_c$, value $\mathbf{i}_r$

Sample $\mathcal{C}_M \sim \sigma_{d-1}$ with seed s ; /* sample codebook with uniform distribution σ_{d-1} on the sphere */
$c_l = \mathrm{VQ}(x/\|x\|, \mathcal{C}_M)$; /* quantize (select a codeword in spherical codebook $\mathcal{C}_M$) */
$\mathbf{i}_{c_l} \leftarrow$ index of c_l; /* get index of codeword */
$\mathbf{i}_r = \mathrm{SQ}(\|x\|)$; /* quantize r on P bits */

Algorithm 4: Spherical-DoStoVoQ over T iterations

Input : T nb of steps, $(\gamma_t)_{t\geq 0}$ LR, θ_0, d, M, P ;
Output: $(\theta_t)_{t\geq 0}$

for $t = 1, \ldots, T$ **do**
  w_0 sends θ_{t-1} and different seeds $s_{k,t}$ to each w_k;
  for $k = 1, \ldots, K$ **do**
    Compute local gradient $g_{k,t}$ at θ_{t-1};
    Split $g_{k,t}$ on $[b^1_{k,t}, \ldots, b^L_{k,t}]$;
    for $\ell = 1, \ldots, L$ *(in parallel)* **do**
      $(\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell}) = \text{Spherical} - \mathtt{StoVoQ}(b^\ell_{k,t}, p, M, P, s_{k,t})$
    end
    Send $(\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell})_{\ell\in[L]}$ to w_0 ;
  end
  Reconstruct $(\hat{g}_{k,t})_{k\in K}$;
  Update: $\theta_t = \theta_{t-1} - \gamma_t \frac{1}{K}\sum_{k=1}^K \hat{g}_{k,t}$;
end

D.2 Extension to DoStoVoQ-DIANA and DoStoVoQ-VR-DIANA

In this subsection, we provide the adaptations of the DoStoVoQ algorithm to algorithms designed to handle heterogeneous workers, and for which the best complexities are achieved, namely DIANA [25] and VR-DIANA [17]. Those algorithms are based on the fundamental idea: relying on control variates $(h_{k,t})_{k\in[K],t\geq 0}$, updated at each iteration, that converge (in the convex case), for each worker k, to $\nabla f_k(\theta^*)$. Instead of compressing $g_{k,t}$, the algorithm compresses the difference between the actual gradient and the control variate $g_{k,t} - h_{k,t}$. The impact of those control variates (often referred to as *memory*) is to mitigate the discrepancy between workers' gradients that stems from the heterogeneity of the data-distribution between different workers. As explained in Appendix E it is particularly relevant to reduce this discrepancy to maximize the impact of the multiple workers. The same idea can be incorporated within a variance reduced algorithm, we here focus on SVRG [18] (extension to SAGA [9] or other variants is straightforward). To incorporate variance reduction to the algorithm, we further assume that each f_k is a finite sum $\frac{1}{S}\sum_{s\in[S]} f_{k,s}$. Algorithms DoStoVoQ-DIANA and DoStoVoQ-DIANA-SVRG are provided in respectively Algorithms 5 and 6.

Algorithm 5: `DoStoVoQ-DIANA` over T iterations . Lines specific to the `Diana` approach are highlighted in **blue**

Input : T nb of steps, $(\gamma_t)_{t\geq 0}$ LR, θ_0, p, M, P, l.r. α ;
Output: $(\theta_t)_{t\geq 0}$

Set $h_{k,0} = 0$ **for all** $k \in [K]$ **(or alternatively** $h_{0,k} = \nabla f_k(\theta_0)$**)**;
for $t = 1, \ldots, T$ **do**
  w_0 sends θ_{t-1} and different seeds $s_{k,t}$ to each w_k;
  for $k = 1, \ldots, K$ **do**
    Compute local gradient $g_{k,t}$ at θ_{t-1};
    Set $\Delta_{k,t} = g_{k,t} - h_{k,t}$;
    Split $\Delta_{k,t} \times \sqrt{D}/\|\Delta_{k,t}\|$ on $[\delta_{k,t}^1, \ldots, \delta_{k,t}^L]$;
    for $\ell = 1, \ldots, L$ *(in parallel)* **do**
      $(\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell}) = \texttt{StoVoQ}(\delta_{k,t}^\ell, p, M, P, s_{k,t})$
    end
    Reconstruct $(\hat{\Delta}_{k,t})_{k\in K}$;
    Update memory: $h_{k,t+1} = h_{k,t} + \alpha \hat{\Delta}_{k,t}$;
    Send $(\|\Delta_{k,t}\|, (\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell})_{\ell\in[L]})$ to w_0 ;
  end
  On the central node w_0;
  Reconstruct $(\hat{\Delta}_{k,t})_{k\in K}$;
  Update: $\theta_t = \theta_{t-1} - \gamma_t(\bar{h}_t + \frac{1}{K}\sum_{k=1}^K \hat{\Delta}_{k,t})$;
  Update averaged memory : $\bar{h}_{t+1} \left(:= \frac{1}{K}\sum_{k\in[K]} h_{k,t}\right) = \bar{h}_t + \frac{\alpha}{K}\sum_{k\in[K]} \hat{\Delta}_{k,t}$;
end

Algorithm 6: `DoStoVoQ-DIANA-SVRG` over T iterations . Lines specific to the variance reduction approach are highlighted in **green**

Input : T nb of steps, $(\gamma_t)_{t\geq 0}$ LR, θ_0, p, M, P, l.r. α ;
Output: $(\theta_t)_{t\geq 0}$

Set $h_{k,0} = 0$ **for all** $k \in [K]$ **(or alternatively** $h_{0,k} = \nabla f_k(\theta_0)$**)**;
for $t = 1, \ldots, T$ **do**
  Sample $u_t \sim \mathcal{B}(S^{-1})$;
  w_0 sends θ_{t-1}, u_t and different seeds $s_{k,t}$ to each w_k;
  for $k = 1, \ldots, K$ **do**
    if $u_t = 1$ **then**
      Set $\eta_{k,s,t} = \theta_t$ **for all** $s \in [S]$**;**
    Sample $s_{k,t} \sim \mathrm{Unif}[S]$;
    Set $\mu_{t,k} = S^{-1}\sum_{s\in S} \nabla f_{k,s}(\eta_{k,s,t})$;
    Set $g_{k,t} = \nabla f_{k,s_{k,t}}(\theta_{t-1}) - \nabla f_{k,s_{k,t}}(\eta_{k,s_{k,t},t}) + \mu_{k,t}$;
    Set $\Delta_{k,t} = g_{k,t} - h_{k,t}$;
    Split $\Delta_{k,t} \times \sqrt{D}/\|\Delta_{k,t}\|$ on $[\delta_{k,t}^1, \ldots, \delta_{k,t}^L]$;
    for $\ell = 1, \ldots, L$ *(in parallel)* **do**
      $(\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell}) = \texttt{StoVoQ}(\delta_{k,t}^\ell, p, M, P, s_{k,t})$
    end
    Reconstruct $(\hat{\Delta}_{k,t})_{k\in K}$;
    Update memory: $h_{k,t+1} = h_{k,t} + \alpha \hat{\Delta}_{k,t}$;
    Send $(\|\Delta_{k,t}\|, (\mathbf{i}_c^{t,k,\ell}, \mathbf{i}_r^{t,k,\ell})_{\ell\in[L]})$ to w_0 ;
  end
  On the central node w_0;
  Reconstruct $(\hat{\Delta}_{k,t})_{k\in K}$;
  Update: $\theta_t = \theta_{t-1} - \gamma_t(\bar{h}_t + \frac{1}{K}\sum_{k=1}^K \hat{\Delta}_{k,t})$;
  Update averaged memory : $\bar{h}_{t+1} \left(:= \frac{1}{K}\sum_{k\in[K]} h_{k,t}\right) = \bar{h}_t + \frac{\alpha}{K}\sum_{k\in[K]} \hat{\Delta}_{k,t}$;
end

E Additional experiments

In this Section, we compare by Monte Carlo the distortions achieved by different compression schemes for 3 types of input x. For a given (random) compressor generically denoted $Q(\cdot)$ and $x \in \mathbb{R}^d$, we decompose $Q(x) = Q_{\parallel}(x) + Q_{\perp}(x)$, where $Q_{\parallel}(x) = \|x\|^{-2} x x^\top Q(x)$. With these notations, $Q_{\parallel}(x)$ and $Q_{\perp}(x)$ are the components of the quantization error which are colinear and orthogonal to x, respectively. By construction, $\|x - Q(x)\|^2 = \|x - Q_{\parallel}(x)\|^2 + \|Q_{\perp}(x)\|^2$. The distortion is computed for $K = 1$ and $K \in \{8, 20\}$ workers (depending on the experiments). We compare 10 compression schemes, corresponding to 7 algorithms (some with several variants): the signed algorithm (`Sign`) (see Definition 1), `Top-H` with $H = 2$ (see Definition 2), `Rand-H` with $H = 2$ (see Definition 3), `HSQ-Span` (see Appendix B.3.1) with $M = 2^{10}$ and a 6 bits scalar quantizer for the norm, `HSQ-greed` (see appendix B.3.2) with $M = 2^{10}$ and a 6 bits scalar quantizer for the norm, `Polytope` (see Appendix B.2) with and without quantization of the norm, three variants of `StoVoQ` with a Gaussian random codebook with $M = 2^{13}$ and $p = \mathcal{N}(0, (1 + 2/d)\, \mathrm{I}_d)$: `GRVQ` which is `StoVoQ` without the radial debiasing step, `StoVoQ` without quantization of r_M^p, and `StoVoQ` with an unbiased scalar quantization of $(r_M^p)^{-1}$ over $P = 3$ bits (strictly speaking, only this last column corresponds to the algorithm `StoVoQ`, the two previous versions have been added to assess the influence of the debiasing by $\{r_M^p\}^{-1}$ and the quantization of $\{r_M^p\}^{-1}$).

We compare those algorithms over three tasks:

1. **Task 1:** Compression of a random vector from a standard Gaussian input distribution in dimension $d = 16$. We compare $K = 1$ to $K = 20$. Results are given in Appendix E.1.
2. **Task 2:** Compression of "real" gradients, extracted from a training performed with a VGG16 on CIFAR10 with SGD, extracted at epoch 10, on which a pre-processing similar to `DoStoVoQ` is applied. The minibatch gradients on a given layer are divided into buckets of dimension $d = 16$. A normalisation is applied to sets of $L = 32$ buckets (the normalisation for the blocks of $d \times L = 512$ coefficients are scalar quantized with a high-resolution scalar quantizer and sent to the parameter server). Results are given in Appendix E.2.1. We compare the performance with 1 and 8 workers, when *all workers compress the same gradient*. The goal of this task is to assess the impact of the actual distribution of the normalised minibatch gradients values w.r.t. a Gaussian distribution.
3. **Task 3:** Compression of "real" gradients, with multiple workers, each worker compresses a different minibatch stochastic gradient, computed at the same parameter (as described in Subsection 4.2): this is the most practical setting, and we explain the resulting trade-offs, especially in terms of the distribution of stochastic gradient noise (see [29]) and the inhomogeneity between workers. We perform this task on (i) the same setting as for task 2, and (ii) the gradients from the LS experiment introduced in Subsection 4.1. Results are given in Appendix E.2.2.

E.1 Distortion for Gaussian input

Setup: We here compare all methods on a Gaussian input $x \sim \mathcal{N}(0, \mathrm{I}_d)$ for $d = 16$. Monte Carlo is performed over 10^4 repetitions. Standard deviation is negligible.

Observations. Results are provided in Table 5. We make the following observations:

1. We first observe that in the single worker case, `Sign`, `Top-2`, `HSQ-greed` and `StoVoQ-GRVQ` achieve a global error or respectively 6.4, 8.7, 9.1 and 6.8. (These errors are obtained by summing the radial and orthogonal numbers). `StoVoQ` achieves an error of 11 which is slightly higher, `Rand-2`, `Polytope`, `HSQ-span` suffer a much higher errors of 110, 121, 147. This confirms the theoretical predictions.
2. We observe in practice here the fundamental differences between biased / unbiased compression methods and also methods that ensure the independence of the compression on each individual worker: while all biased methods do not benefit from the multiplicity of workers, for unbiased and independent compression, the distortion for $K = 1$ is divided by K. Here, the quadratic errors, both radial and orthogonal, are reduced by a factor 20. Overall, over 20 workers, the error obtained by `StoVoQ`, with debiasing and scalar quantization is 0.5. This is by far the best method in terms of global distortion for 20 workers.

Table 5: **Task 1:** Distortion for Gaussian inputs

Method	Sign	Top-2	Rand-2	Polytope	
Variant					norm-quant.
$K = 1$	1.0 ‖ 5.4	4.8 ‖ 3.9	12 ‖ 98	5.8 ‖ 115	5.8 ‖ 115
$K = 20$	1.0 ‖ 5.4	4.7 ‖ 3.8	0.6 ‖ 4.8	0.3 ‖ 5.6	0.3 ‖ 5.6

Method	HSQ-span	HSQ-greed	StoVoQ		
Variant	norm-quant.	norm-quant.	GRVQ	Unbiased	Unbiased+quant.
$K = 1$	3.8 ‖ 143	1.3 ‖ 7.8	1.8 ‖ 5.0	0.5 ‖ 10.5	0.5 ‖ 10.5
$K = 20$	0.2 ‖ 7.0	1.3 ‖ 7.5	1.7 ‖ 0.25	0.03 ‖ 0.5	0.03 ‖ 0.5

3. StoVoQ-GRVQ **vs** StoVoQ-Unbiased. For StoVoQ, the application of debiasing increases the non-radial quantization distortion, by a factor of nearly 2 (from 5 to 10), while simultaneously reducing the radial distortion, form 2 to 0.5. This increase is unavoidable to obtain the unbiased character, that is necessary to reduce the error beyond 1. Indeed, it is important to remark the radial bias for StoVoQ-GVRQ and HSQ is not negligible (1.3 and 1.8 respectively): in fact , this radial distortion is also of order $M^{-2/d}$ thus using an even larger codebook would not reduce it significantly.

4. StoVoQ-Unbiased vs StoVoQ-Unbiased+Scalar-Quantization. We observe that the impact of the scalar quantization is negligible here, which indicates that the impact of the scalar quantization of the norm is limited.

5. HSQ vs StoVoQ-GRVQ: These two methods are somehow similar for a single worker: HSQ relies on a gain-shape decomposition with the a scalar quantization of the norm and a vector quantization of the normalized vector using spherical codebooks whereas GRVQ uses random Gaussian codebooks with a variance matched to the input variance. We observe that overall StoVoQ-GRVQ slightly outperforms HSQ for $K = 1$. This is in favor of using Gaussian codebooks.

E.2 Distortion for neural networks gradients

E.2.1 Task 2: Impact of the distribution

Setup. We compare the distortion for $K = 1$ and 8, on stochastic gradients sampled from the training of a VGG16, with SGD, at epoch 10. The gradients are partitioned into blocks of size 2^9; then each block is scaled and split into buckets of dimension $d = 16$. Those buckets are then compressed using each of the possible methods in dimension 16. The results presented are obtained using 1000 stochastic gradient .

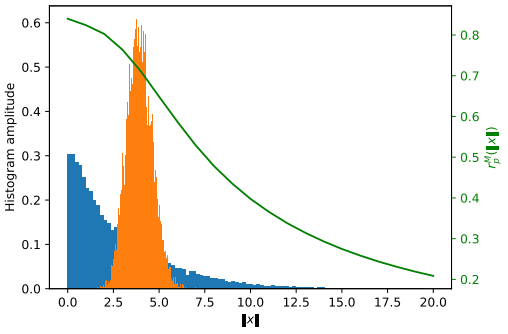


Figure 7: Histograms of the VGG16 gradient buckets (blue), of Gaussian vectors (orange), and the radial bias for the associated dimension $d = 16$ (green).

The main objective is to compare the impact of the distribution of the gradients on the distortion of the different compressors. For example, if the stochastic gradient noise is heavy tailed (leading equivalently to sparse gradients), methods relying on sparsification, e.g., *Top-2*, is expected to perform significantly better than Gaussian random codebook (recall that the optimality result Theorem 3 assumes that the distribution of the codewords matches the distribution of the inputs; the choice of a Gaussian distribution for the codewords implicitly assumes that the distribution of the gradients is approximately Gaussian).

For $K = 8$, we assume that each workers compresses the same gradient : we compute the error of $K^{-1} \sum_{k \in [K]} \hat{g}_{t,k} - g_t$, where $\hat{g}_{t,k}$ stands for the output of the k-th compressor on g_t.

Table 6: **Task 2:** empirical distortion from a sample of gradients sampled from a VGG-16 at epoch 10, (same gradients on each worker).

Method	Sign	Top-2	Rand-2	Polytope	
Variant					norm-quant.
$K=1$	0.46‖0.35	0.35‖0.16	0.86‖4.8	0.50‖7.0	0.50‖7.1
$K=8$	0.46‖0.35	0.35‖0.16	0.15‖0.61	0.07‖0.9	0.07‖0.9

Method	HSQ-span	HSQ-greed	StoVoQ		
Variant	norm-quant.	norm-quant.	GRVQ	Unbiased	Unbiased+quant.
$K=1$	0.24 ‖ 7.5	0.09 ‖ 0.5	0.05 ‖ 0.2	0.02 ‖ 0.36	0.02 ‖ 0.4
$K=8$	0.07 ‖ 0.9	0.09 ‖ 0.4	0.04 ‖ 0.03	0.002 ‖ 0.05	0.003 ‖ 0.05

As shown in Appendix A.5, the distortion of StoVoQ is a non-decreasing function of the norm of the vector to be compressed. In Figure 7, we represent simultaneously the histogram of the bucket norms, and the histogram of the norms of the Gaussian vectors used in **Task 1**. This suggests a departure from the Gaussian distribution for the stochastic gradient noise.

Observations Results are provided in Table 6. We highlight the following points.

1. Again, the unbiased version of StoVoQ achieves the best distortion.
2. Even though the distribution of the norms is very different from the norm of the Gaussian vectors (as illustrated in Figure 7), the distortion of StoVoQ is not severely impaired. Especially, the error for StoVoQ for a single worker is 0.4 vs 0.7 for Top-2, while for Gaussian inputs it was 11 vs 8.7 for Top-2.

E.2.2 Task 3: Signal-Noise ratio on the various gradients

Setup: We now consider that each worker computes and compresses a different stochastic gradient. More precisely, we collect samples of the stochastic gradients during an epoch: $[g_{t,1}^\top, \dots, g_{t,K}^\top]$, where $g_{t,k}$ is computed by the worker k on distinct minibatch of size b (all the gradients are $\{g_{t,k}\}_{k=1}^K$ are evaluated for the same value of the parameters). The compressed version is denoted $\{\hat{g}_{t,k}\}_{k=1}^K$.

In the homogeneous setting, for all $k \in [K]$, $g_{t,k}(\theta_t) = \nabla F(\theta_t) + \epsilon_{t,k}$, with $(\epsilon_{t,k})_{t,k}$ is the stochastic gradient noise $\mathbb{E}\left[\epsilon_{t,k} \,|\, \mathcal{F}_{t-1}\right] = 0$, where $\mathcal{F}_{t-1}$ collects the past observations.

The central node averages the quantized stochastic gradient sent by the workers: $\tilde{g}_t := K^{-1} \sum_{k=1}^K \hat{g}_{t,k}$. We report in Tables 4 and 8 the normalized averaged error defined as

$$T^{-1} \sum_{t \in [T]} \frac{\| \frac{1}{K} \sum_{k=1}^K \hat{g}_{t,k} - g_{t,k} \|^2}{\| \frac{1}{K} \sum_{k=1}^K g_{t,k} \|^2}. \tag{28}$$

We here discuss in which settings we expect the multiple workers to improve w.r.t. a single worker. More precisely, we show that the impact of enforcing unbiased independent compression for the different workers increases with the "dependence" of stochastic gradients. Consider the following two cases. **Example 1: (large noise, low correlation between stochastic gradients)** each worker computes a stochastic gradient that is nearly independent of the other workers. The error made is **not** reduced by the multiplicity of workers. **Example 2: (low or no noise, strong consensus between stochastic gradients)** if each worker computes the same gradient, we recover **task 2**. The variance reduction obtained by using multiple workers and independent compressors is proportional to the number of workers. More generally, this is true when the noise is small w.r.t. the gradient of the function.

This signal/noise ratio fundamentally impacts the performance of algorithms using compressions operators: in **example 2**, it is crucial to use unbiased and independent workers, while in **example 1**, it is more important to reduce the distortion for a single worker.

Table 7: **Task 3:** normalized distortion for a mini-batch of size 4096 of a VGG-16 at epoch 10.

Method	Sign	Top-2	Rand-2	Polytope	
Variant					norm-quant.
$K=1$	0.3‖0.2	0.5‖0.2	0.5‖6.2	0.2‖7.3	0.2‖7.3
$K=8$	0.3‖0.1	0.5‖0.1	0.09‖1.8	0.06‖2.0	0.06‖2.0

Method	HSQ-span	HSQ-greed	StoVoQ		
Variant	norm-quant.	norm-quant.	GRVQ	Unbiased	Unbiased+quant.
$K=1$	0.2 ‖ 8.5	0.09 ‖ 0.5	0.06 ‖ 0.2	0.02 ‖ 0.4	0.02 ‖ 0.4
$K=8$	0.09 ‖ 2.3	0.09 ‖ 0.4	0.1 ‖ 0.07	0.01 ‖ 0.1	0.01 ‖ 0.1

Many factors impact this "consensus" between workers: first of all the mini-batch size. The noise variance is inversely proportional to b: as b increases, each stochastic gradient becomes closer to $\nabla F(\theta)$. More generally, all variance reduction techniques tend to increase the "consensus". On the other hand, heterogeneity between workers increases the discrepancy between gradients (but memory techniques as in DoStoVoQ-DIANA mitigate this discrepancy). Finally, performing several steps [36], as in Local-SGD also has a similar impact of averaging the noise over several iterations, and thus increases the consensus.

We thus evaluate all algorithms on two tasks:

1. First, on gradients extracted from the LSR task: in this task, data is distributed on all workers, that compute a batch gradient. The gradients obviously depend on the workers (each worker has access to a different subset of the data), but because the workers are homogeneous, these gradients have a strong consensus. We give the results in Table 8. We observe that the reduction by a factor 8 is preserved when using $K = 8$. This explains why our method outperforms HSQ in Figure 2.
2. Second, on gradients from the VGG16 trained with SGD on CIFAR. On this task, the noise level is much higher and the consensus much weaker. This is expected in very high dimensional models and non convex objective (roughly speaking, the gradients on different workers nearly point in random descent directions). We thus do not see any strong effect of the number of workers on the distortion for $b \leq 512$. Increasing further the batch size, to $b = 4096$, we recover the gain of multiple workers. Results are given in Table 7. The distortion is twice smaller with StoVoQ-unbiased than with any other method. While a batch of 4096 is very high, very large batch were used in a successful training of CIFAR and IMAGENET in Lin et al. [24]. More generally, when communication cost is a major concern, increasing the batch size and the number of local iterations is natural, to increase the quality of updates transmitted.

Table 8: **Task 3:** normalized distortion for LSR (see Section 4).

Method	Sign	Top-2	Rand-2	Polytope	
Variant					norm-quant.
$K=1$	0.05‖0.3	0.4‖0.2	0.6‖6.3	0.3‖7.3	0.3‖7.3
$K=8$	0.04‖0.09	0.4‖0.08	0.1‖1.3	0.07‖1.4	0.07‖1.4

Method	HSQ-span	HSQ-greed	StoVoQ		
Variant	norm-quant.	norm-quant.	GRVQ	Unbiased	Unbiased+quant.
$K=1$	0.3 ‖ 9.4	0.09 ‖ 0.5	0.1 ‖ 0.3	0.03 ‖ 0.6	0.03 ‖ 0.6
$K=8$	0.08 ‖ 1.9	0.09 ‖ 0.1	0.1 ‖ 0.06	0.008 ‖ 0.1	0.008 ‖ 0.1