EDEN: Communication-Efficient and Robust Distributed Mean Estimation for Federated Learning

Shay Vargaftik [*1] **Ran Ben Basat** [*2] **Amit Portnoy** [*3]
Gal Mendelson [4] **Yaniv Ben-Itzhak** [1] **Michael Mitzenmacher** [5]

Abstract

Distributed Mean Estimation (DME) is a central building block in federated learning, where clients send local gradients to a parameter server for averaging and updating the model. Due to communication constraints, clients often use lossy compression techniques to compress the gradients, resulting in estimation inaccuracies.

DME is more challenging when clients have diverse network conditions, such as constrained communication budgets and packet losses. In such settings, DME techniques often incur a significant increase in the estimation error leading to degraded learning performance.

In this work, we propose a robust DME technique named EDEN that naturally handles heterogeneous communication budgets and packet losses. We derive appealing theoretical guarantees for EDEN and evaluate it empirically. Our results demonstrate that EDEN consistently improves over state-of-the-art DME techniques.

1. Introduction

In the Distributed Mean Estimation (DME) problem, each of n senders has a d-dimensional vector of real numbers. Each sender sends information over the network to a central receiver, who uses this information to estimate the mean of these vectors. This problem is a central building block in many federated learning scenarios, where at each training round, a parameter server averages clients' parameter updates (i.e., neural network gradients) and updates its model (McMahan et al., 2017). As neural network gradients are often large (e.g., can exceed a billion dimensions (Dean et al., 2012; Shoeybi et al., 2019; Huang et al., 2019)), transmission over the network is often a bottleneck, and thus applying lossy compression to the gradients can be essential to adhere to client communication constraints, reduce the training time, and allow better inclusion and scalability.

Typically, the desired design property is that the receiver's resulting estimate will be unbiased. That is, the receiver's derived estimate $\hat{x}$ for a sender's vector $x \in \mathbb{R}^d$ should satisfy $\mathbb{E}[\hat{x}] = x$. Unbiasedness is attractive because, under natural conditions (including independence of estimates), as it yields a Mean Squared Error (MSE) between the mean of the received estimates and the mean of the true vectors that decays linearly with respect to the number of clients (e.g., see Vargaftik et al. (2021)). Besides being a useful property for DME in isolation, in federated learning contexts this can remove the need for error feedback mechanisms that are commonly used to deal with biased estimates (Seide et al., 2014; Karimireddy et al., 2019), but are often not practical due to client participation patterns (Kairouz et al., 2021).

For unbiasedness, modern DME techniques employ randomized rounding techniques, commonly known as stochastic quantization (SQ), to map each vector coordinate to one of a limited number of possibilities, yielding a compressed form.

Some SQ-based techniques have known issues when used in DME. In particular, the resulting error is sensitive to the vector's distribution and the difference between the largest and smallest coordinates. This is specifically problematic in federated learning, where neural network gradients' coordinates can differ by orders of magnitude, rendering vanilla SQ inapplicable for accurate DME in many settings.

To address this limitation, recent works suggest the vector be *randomly rotated* prior to stochastic quantization (Suresh et al., 2017). That is, the clients and the parameter server draw rotation matrices according to some known distribution (e.g., uniform); the clients then send the quantization of the rotated vectors while the parameter server applies the inverse rotation on the estimated rotated vector. Intuitively, the coordinates of a vector rotated by uniform random rotation are identically distributed (albeit weakly dependent) and are closely concentrated around their mean, leading to a small

[*]Equal contribution
[1]VMware Research [2]University College London [3]Ben-Gurion University [4]Stanford University [5]Harvard University.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

expected difference between the coordinates that allows for an accurate quantization. For $x \in \mathbb{R}^d$, this approach achieves a Normalized MSE ($NMSE$)[1] of $O(\frac{\log d}{n})$ using $O(1)$ bits per coordinate per client (i.e., $O(nd)$ bits in total).

Another approach makes use of Kashin's representation (Lyubarskii & Vershynin, 2010; Caldas et al., 2018b; Safaryan et al., 2020). Roughly speaking, it allows representing a d-dimensional vector using larger vectors with $\lambda \cdot d$ coefficients for some $\lambda > 1$, where each coefficient is smaller. Applying stochastic quantization to the Kashin coefficients allows an $NMSE$ of $O(\frac{1}{n})$ using $O(\lambda)$ bits per coordinate. Compared with (Suresh et al., 2017), using Kashin's representation yields a lower $NMSE$ at the cost of increased computational complexity (Vargaftik et al., 2021).

Recent works propose algorithms that rely on clients' gradient similarity to improve guarantees. For example, (Davies et al., 2021) suggests an algorithm where if all clients' gradients have pairwise Euclidean distances of at most $a \in \mathbb{R}$, the resulting $NMSE$ is $O(a^2)$ using $O(1)$ bits per coordinate on average. This solution provides a good bound when gradients are similar (and thus a is small). However, it may be less efficient for federated learning, where clients often have different data distributions (and thus a may be large).

The recently introduced DRIVE (Vargaftik et al., 2021) is a state-of-the-art DME algorithm that uses a single bit per coordinate. Formally, DRIVE offers an $NMSE$ of $O\left(\frac{1}{n}\right)$ using $(1 + o(1))$ bits per coordinate and improves over existing DME techniques utilizing a similar communication budget both analytically and empirically. DRIVE's improvement stems from employing a deterministic quantization instead of a stochastic one after a random rotation, yielding an asymptotic $NMSE$ improvement. DRIVE still produces unbiased estimates by adequately scaling the gradients.

A communication budget of one bit per coordinate has been thoroughly studied (Seide et al., 2014; Wen et al., 2017; Bernstein et al., 2018; Karimireddy et al., 2019; Ben-Basat et al., 2021; Vargaftik et al., 2021), and used to accelerate distributed learning systems (Jiang et al., 2020; Bai et al., 2021). However, one bit per coordinate does not support many federated learning scenarios where clients have different communication budgets and network conditions. We expand on alternative compression approaches, which are not directly applicable to DME, in Appendix A.

In this work, we propose **E**fficient **DME** for diverse **N**etworks (EDEN) – a robust DME technique that supports heterogeneous communication budgets and packet loss rates. EDEN achieves an $NMSE$ of $O\left(\frac{1}{n}\right)$ using b bits per coordinate, for any constant b, including for $b < 1$, i.e., less than one bit per coordinate. An additional feature of EDEN is that it naturally handles packet loss without retransmission by replacing lost coordinates with 0 values. We extend our theoretical results to this setting for constant packet loss rates and empirically demonstrate this robustness.

EDEN achieves improved accuracy using a novel formalization of the quantization framework. While previous work defines the quantization via a set of quantization points, our solution requires choosing a set of *intervals* whose union covers the real interval. Then, each point is quantized to the center of mass of its interval and *not* to the closest quantization point, which is counter-intuitive. That is, our solution may quantize some points to quantization levels farther away from them than the closest. Nonetheless, such a method can reduce the entropy of the quantized vector, allowing for better $NMSE$ given a communication budget.

We implement and evaluate EDEN in PyTorch (Paszke et al., 2019) and TensorFlow (Abadi et al., 2015)[2] and show that EDEN can compress vectors with more than 67 million coordinates within 61 ms. Compared with state-of-the-art DME techniques, EDEN consistently provides better mean estimation, which translates to higher accuracy in various federated and distributed learning tasks and scenarios.

2. EDEN

We start with preliminaries, overview EDEN, and then describe the complete details and guarantees.

2.1. Preliminaries

We assume that each sender has access to randomness that is shared with the central receiver. This assumption is standard (e.g., Suresh et al. (2017); Ben-Basat et al. (2021); Vargaftik et al. (2021)) and can be implemented by having a shared seed for a PseudoRandom Number Generator (PRNG). Importantly, each sender uses a different seed and thus its shared randomness is independent of that of other senders.

Formally, we are interested in efficiently solving the DME problem. In this problem, we have a set of $n \in \mathbb{N}^+$ senders and a central receiver. Each sender $c \in \{1, \ldots, n\}$ has its own vector $x_c \in \mathbb{R}^d$, $x_c \neq 0$,[3] and sends a message to the receiver. The receiver then produces an estimate of the average of these sender vectors. In particular, we focus on the setting where each sender message yields an estimate of each sender's vector $\widehat{x}_c$, and the receiver computes the average of the $\widehat{x}_c$ as an estimate of the average of the x_c,

[1] The normalized MSE is the mean's estimate MSE normalized by the mean clients' gradient squared norms (§2.1).

[2] Our PyTorch and TensorFlow implementations are available as open source at https://github.com/amitport/EDEN-Distributed-Mean-Estimation.

[3] For ease of exposition, we hereafter assume that $x_c \neq 0$ for all c since this case can be handled with one additional bit. Further, in ML applications, zero gradients essentially never occur in practice.

with the goal of minimizing its $NMSE$ defined as,

$$NMSE \triangleq \frac{\mathbb{E}\left[\left\|\frac{1}{n}\sum_{c=1}^{n}\widehat{x}_c - \frac{1}{n}\sum_{c=1}^{n}x_c\right\|_2^2\right]}{\frac{1}{n}\cdot\sum_{c=1}^{n}\|x_c\|_2^2}.$$

In federated learning and other techniques based on stochastic gradient descent (SGD) and its variants (e.g., McMahan et al. (2017); Li et al. (2020); Karimireddy et al. (2020)), each round includes a mean estimation of the local vectors. Indeed, the $NMSE$ affects the convergence rate and often the final accuracy of the models. Further, the provable convex convergence rates for compressed SGD have a linear dependence on the $NMSE$ (Bubeck (2015), Theorem 6.3).

2.2. EDEN's Overview

Figure 1 depicts a high-level illustration of EDEN.

2.2.1. Senders.

To compress a vector, each sender employs three consecutive steps: rotation, quantization, and scaling.

Random Rotation. Each sender uses the shared randomness with the receiver to randomly rotate its vector and to do so *independently* from other senders. Rotation can be expressed by multiplying the vector by a *rotation matrix*. In particular, a rotation matrix $R \in \mathbb{R}^{d\times d}$ satisfies $R^TR = I$, which also implies that for any $x \in \mathbb{R}^d : \|Rx\|_2 = \|x\|_2$. For ease of notation, we use $\mathcal{R}(x)$ to denote Rx when R is selected uniformly at random; in §5 we present an efficient implementation. Similarly, $\mathcal{R}^{-1}(x)$ denotes the inverse rotation, i.e., $R^{-1}x = R^Tx$. For sender c and its vector $x_c \in \mathbb{R}^d$, we denote its rotated vector by $\mathcal{R}_c(x_c)$.

Deterministic Quantization. To encode a real-valued gradient using a finite number of bits, one must *quantize* it. To design the quantization, we leverage the fact that after randomly rotating a vector, all its coordinates are identically distributed. This distribution quickly converges to a normal distribution with the vector's dimension. Specifically, for a vector $x \in \mathbb{R}^d$, we have that as d tends to infinity, the distribution of each $\mathcal{R}(x)$'s coordinate tends to a normal distribution $\mathcal{N}(0, \frac{\|x\|_2^2}{d})$ (Vargaftik et al., 2021).

Leveraging this, we can calculate the best quantization to approximate the standard normal distribution[5] $\mathcal{N}(0,1)$ *offline*. Then, at run time, each sender multiplies its rotated vector by a factor of $\eta_x = \frac{\sqrt{d}}{\|x\|_2}$ and finds the best quantization for its own rotated coordinates' distribution. We now formalize the above, starting with defining a family of deterministic quantizations for the normal distribution.

Let $\mathcal{I}$ be a set of intervals with disjoint interiors such that $\cup_{I\in\mathcal{I}} I = \mathbb{R}$. We further require two properties:

1. $\mathcal{I}$ is *symmetric*; that is, $[a,a'] \in \mathcal{I} \Longrightarrow [-a',-a] \in \mathcal{I}$.
2. $[-a,a] \in \mathcal{I} \implies a < 1$.

For ease of exposition, we first consider finite sets $\mathcal{I}$; in §4.3 and the appendix, we relax this and allow certain infinite interval families. For example, two such partitions are $\{(-\infty,0],[0,\infty)\}$ and $\{(-\infty,-\frac{1}{2}],[-\frac{1}{2},\frac{1}{2}],[\frac{1}{2},\infty)\}$. (Note a, a' can be (minus or plus) infinity in our definition.)

Next, for each such interval $I = [a,a'] \in \mathcal{I}$, we denote its center of mass by $q_I = \mathbb{E}[z|z \in I]$ where $z \sim \mathcal{N}(0,1)$, i.e., $q_I = \frac{\int_a^{a'} t\cdot e^{-\frac{1}{2}t^2}dt}{\int_a^{a'} e^{-\frac{1}{2}t^2}dt}$. Also, for $z \in \mathbb{R}$, let $\mathcal{I}(z)$ denote the interval that encompasses z.[4] We then define the quantization operator $\mathcal{Q}_\mathcal{I}(z) = q_{\mathcal{I}(z)}$. When clear from context, we omit the subscript $\mathcal{I}$ and write $\mathcal{Q}$. That is, z is quantized to the center of mass of the interval in which it lies. This definition generalizes seamlessly to vector quantization, where for $y = (y[1],\ldots,y[d]) \in \mathbb{R}^d$ we denote $\mathcal{Q}(y) = \Big(\mathcal{Q}(y[1]),\ldots,\mathcal{Q}(y[d])\Big)$. Also, by the properties of $\mathcal{I}$, we obtain $\mathcal{Q}(-y) = -\mathcal{Q}(y)$ and $y[j]\cdot\mathcal{Q}(y[j]) \geq 0$ for all $j \in \{1,\ldots,n\}$ leading to $\langle y, \mathcal{Q}(y)\rangle \geq 0$ for all $y \in \mathbb{R}^d$.

For sender c and its rotated vector $\mathcal{R}_c(x_c)$, its quantized vector is $\mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c))$. That is, the sender multiplies its rotated vector by η_{x_c} before applying the quantization.

We note that it always holds that $\mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c)) \neq 0$. Namely, the quantization process, by design, cannot nullify a client's vector. This is because $\|\eta_{x_c}\cdot\mathcal{R}_c(x_c)\|_2^2 = d$, which means that the absolute value of at least one coordinate is at least 1. Thus, by the second property of $\mathcal{I}$, this coordinate cannot lie in an interval that maps it to 0. In turn, by property 1, it also implies $\langle\mathcal{R}_c(x_c), \mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c))\rangle > 0$.

In §3, we detail how to optimize $\mathcal{I}$ for different communication budgets and how to perform the quantization efficiently.

Scaling. After rotation and quantization, each sender c calculates a *scale* $S_c \in \mathbb{R}_+$ that is used by the receiver to scale the estimate. As we detail in §2.3, scaling is the key for removing the bias introduced by the quantization.

Finally, each sender c sends a representation of $\mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c))$ and S_c to the receiver. This can be done with $\lceil\log_2|\mathcal{I}|\rceil\cdot d + o(d)$ bits, i.e., using the log of the number of quantization values many bits per quantized value, and representing the scale using a sub-linear number of bits in the vector's dimension (in practice, we use a fixed number of bits, e.g., 64, to send the scale and ignore the rounding error).

2.2.2. Receiver

The receiver reconstructs each sender's c vector by first performing the inverse rotation, i.e., it uses the shared randomness to generate the same rotation matrix and computes $\mathcal{R}_c^{-1}(\mathcal{Q}_c(\eta_{x_c}\cdot\mathcal{R}_c(x_c)))$. Then, the result is scaled by S_c to obtain the estimate, i.e., $\widehat{x}_c = S_c\cdot\mathcal{R}_c^{-1}(\mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c)))$.

[4] If z is an endpoint of intervals, the one closer to zero is chosen.

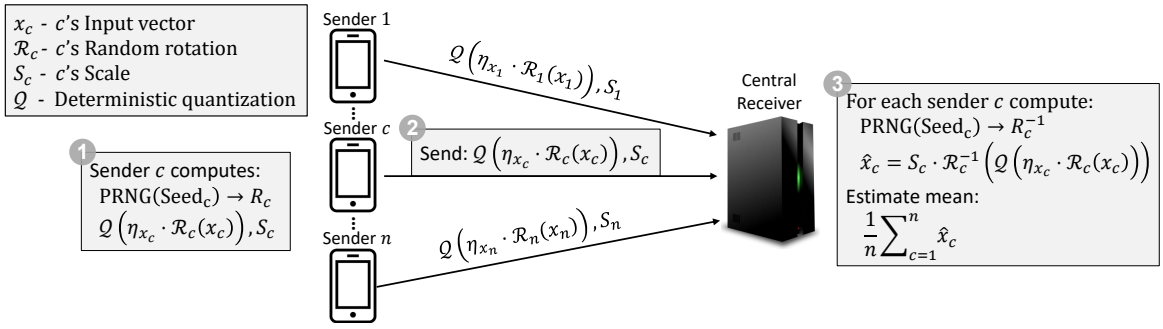


Figure 1. EDEN's compress and decompress methods.

Finally, the receiver averages the results from all senders and obtains the estimate of the mean, i.e., $\frac{1}{n}\sum_{c=1}^{n}\widehat{x}_c$.

EDEN also supports network packet losses without the need for retransmitting the lost coordinates (detailed in §4).

2.3. EDEN's Scale

An appealing property of EDEN we establish in this work is that each sender can efficiently calculate the scale S_c to make its estimate unbiased even though it uses a biased quantization technique. In particular, each sender c uses:

$$S_c = \frac{\|x_c\|_2^2}{\langle \mathcal{R}_c(x_c), \mathcal{Q}(\eta_{x_c}\cdot\mathcal{R}_c(x_c))\rangle}\ .$$

With this scale, we obtain the following formal guarantee whose proof appears in Appendix B.

Theorem 2.1. *For all $x\in\mathbb{R}^d$, using EDEN with the scale $S=\frac{\|x\|_2^2}{\langle\mathcal{R}(x),\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\rangle}$ results in $\mathbb{E}[\widehat{x}]=x$.*

Intuitively, this scale ensures that the reconstructed vector $\widehat{x}_c$ lies on a hyperplane tangent to the original vector's x_c point on the sphere. Since the rotation has no preferred direction, the expected value of the reconstructed vector produces precisely the original one. Specifically, the proof relies on this property by showing that for each rotation, there exists a matching rotation with the same bias but the opposite sign, and each such pair's average yields the original vector.

2.4. EDEN's *NMSE*

We start with the following definition. For estimation of a single vector x, we define the vector-*NMSE* (*vNMSE*) as

$$vNMSE \triangleq \frac{\mathbb{E}\left[\|x-\widehat{x}\|_2^2\right]}{\|x\|_2^2}\ .$$

For a sender c, we use $vNMSE(c)$ to denote its $vNMSE$. Now, since the estimates of the sender vectors are independently) and unbiased (according to Theorem 2.1), we obtain the following result whose proof appears in Appendix C.

Lemma 2.2. *Consider n senders. It holds that*

$$NMSE = \frac{\sum_{c=1}^{n} vNMSE(c)\cdot\|x_c\|_2^2}{n\cdot\sum_{c=1}^{n}\|x_c\|_2^2}\ .$$

Observe that for the special case where all senders use the same set $\mathcal{I}$, it holds that $NMSE=\frac{1}{n}\cdot vNMSE$ since $vNMSE(c)$ is the same for all senders.

Accordingly, we obtain a bound on the $NMSE$ by bounding each client's $vNMSE$ using the following theorem, whose proof appears in Appendix D. The proof relies on a novel mathematical framework that leverages the fact that the rotated vector's distribution is that of a vector of independent $\mathcal{N}(0,1)$ random variables $Z\in\mathbb{R}^d$ multiplied by the input vector's norm and divided by Z's norm (Vargaftik et al., 2021). Specifically, we define events that control quantities of interest (e.g., that the norm of Z is highly concentrated around its mean). We then show that these events hold with high probability and infer that our $vNMSE$ converges to the following function of the quantization error of a single $\mathcal{N}(0,1)$ variable.

Theorem 2.3. *Let $z\sim\mathcal{N}(0,1)$. For all $x\in\mathbb{R}^d$, with $S=\frac{\|x\|_2^2}{\langle\mathcal{R}(x),\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\rangle}$, EDEN satisfies:*

$$vNMSE \le \frac{1}{\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]} - 1 + O\left(\sqrt{\frac{\log d}{d}}\right).$$

Also, $\mathbb{E}\left[z\mathcal{Q}(z)\right]=\mathbb{E}\left[\mathcal{Q}(z)\cdot\left[\mathbb{E}\left[z\right]\middle|\mathcal{Q}(z)\right]\right]=\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]$ and thus $\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]=1-\mathbb{E}\left[(z-\mathcal{Q}(z))^2\right]$. This means that minimizing the $vNMSE$ bound is achieved by minimizing the quantization's MSE with respect to $z\sim\mathcal{N}(0,1)$.

Next, we obtain the following corollary.

Corollary 2.4. *For $d\to\infty$, the vNMSE upper bound in Theorem 2.3 approaches $\frac{1}{\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]}-1$, where $z\sim\mathcal{N}(0,1)$.*

Our proofs have focused on the upper bound, but Corollary 2.4 is tight; our proofs could be extended to a lower bound, and our experiments coincide with this claim.

We later give examples of these guarantees and how they relate to the number of bits used per coordinate in §3.

3. Optimal 2^b-Values Quantization

With b bits per coordinate, we can use 2^b quantization values. Since our goal is to minimize the MSE from a standard normal random variable to its quantization, we *precalculate* the optimal quantization for 2^b values using the known *Lloyd-Max Scalar Quantizer* (Lloyd, 1982; Max, 1960) for the normal distribution.[5]

For ease of exposition, for an integer bit budget $b \in \mathbb{N}^+$, we denote by $\mathcal{I}_b$ the optimal set of 2^b intervals and by $Q_{\mathcal{I}_b}$ the resulting quantization values. For example, the intervals and quantization values for $b = 1$ and $b = 2$ are:

$$\mathcal{I}_1 = \{(-\infty, 0], [0, \infty)\} \ , \quad Q_{\mathcal{I}_1} = \left\{\pm\sqrt{\frac{2}{\pi}}\right\} \approx \{\pm 0.79788\}.$$

$$\mathcal{I}_2 \approx \{(-\infty, -0.9816], [-0.9816, 0], [0, 0.9816], [0.9816, \infty)\},$$
$$Q_{\mathcal{I}_2} \approx \{\pm 0.45278, \pm 1.51042\}.$$

We note that when Q is built using the Lloyd-Max quantizer, the quantization can be efficiently computed by

$$\mathcal{Q}(x) = \text{argmin}_{y \in Q^d} \left\| \sqrt{d} \cdot \frac{x}{\|x\|_2} - y \right\|_2 .$$

Namely, for such quantization value, the center of mass of a scaled coordinate's interval is also the closest quantization value to that coordinate. For clarity, we now show how Corollary 2.4 applies for $Q_{\mathcal{I}_1}$ and $Q_{\mathcal{I}_2}$.

Example 1. For $Q_{\mathcal{I}_1}$ we obtain

$$\frac{1}{\mathbb{E}\left[\left(\mathcal{Q}_{\mathcal{I}_1}(z)\right)^2\right]} - 1 = \frac{1}{\sum_{I \in \mathcal{I}_1} q_I^2 \cdot \mathbb{P}(z \in I)} - 1 =$$
$$\frac{1}{\frac{1}{2}\left(\sqrt{\frac{2}{\pi}}\right)^2 + \frac{1}{2}\left(-\sqrt{\frac{2}{\pi}}\right)^2} - 1 = \frac{1}{\frac{2}{\pi}} - 1 \approx 0.571.$$

That is, as $d \to \infty$, the $vNMSE$ goes to approximately 0.571, which coincides with the corresponding result for DRIVE (Vargaftik et al., 2021). In fact, without coordinate losses (§4.2), using EDEN with $Q_{\mathcal{I}_1}$ is equivalent to using DRIVE since $S \cdot \mathcal{Q}_{\mathcal{I}_1}(\eta_x \cdot \mathcal{R}(x)) = \frac{\|x\|_2^2}{\|\mathcal{R}(x)\|_1} \cdot \text{sign}(\mathcal{R}(x))$.

Example 2. For $Q_{\mathcal{I}_2}$ we have that

$$\Pr\left[z \in [0, 0.9816]\right] = \frac{1}{\sqrt{2\pi}} \int_0^{0.9816} e^{-t^2/2} dt \approx 0.33685.$$

Therefore, we obtain:

$$\frac{1}{\mathbb{E}\left[\left(\mathcal{Q}_{\mathcal{I}_2}(z)\right)^2\right]} - 1 = \frac{1}{\sum_{I \in \mathcal{I}_2} q_I^2 \cdot \mathbb{P}(z \in I)} - 1 \approx$$
$$\frac{1}{2 \cdot 0.33685 \cdot (0.45278)^2 + 2 \cdot 0.16315 \cdot (1.51042)^2} - 1$$
$$\approx \frac{1}{0.88228} - 1 \approx 0.134,$$

which is an improvement of by more than a factor of 4 in comparison to Example 1.

In practice, we find that the empirical $vNMSE$ (and the resulting $NMSE$) match that of Corollary 2.4 in all our experiments, for any d that is larger than a few hundreds.

4. Handling Heterogenity and Loss

We next detail how EDEN operates with general bit budget constraints and lossy networks, and discuss its compatibility with variable-length encoding techniques.

4.1. Heterogeneous Sender Bit Budget

Often in federated learning, senders may have different resource constraints, particularly networking constraints (Nishio & Yonetani, 2019). Therefore, it is beneficial for an algorithm to allow senders to use different amounts of compression, tuned to their own available communication budget. Accordingly, we provide two generalizations that maintain the strong guarantees of Theorems 2.1 and 2.3 and allow EDEN's senders to adapt their bit budget per coordinate. Specifically, we allow each sender to use its own set of quantization values and to use a non-integer number of bits b per coordinate (in expectation).

Super-bit compression ($b \geq 1$). For a sender who wishes to use an integer b bits per coordinate, we simply use $Q_{\mathcal{I}_b}$. For non-integer $b > 1$, we propose the following generalization. We quantize each coordinate using $Q_{\mathcal{I}_{\lfloor b \rfloor + 1}}$ with probability $b - \lfloor b \rfloor$, and with $Q_{\mathcal{I}_{\lfloor b \rfloor}}$ with probability $1 - (b - \lfloor b \rfloor)$. This means that each client's quantization is a distribution over $Q_{\mathcal{I}_{\lfloor b \rfloor + 1}}$ and $Q_{\mathcal{I}_{\lfloor b \rfloor}}$. The choice of which coordinates to send using more bits are selected using shared randomness (to simulate independent weighted coin flips). In practice, this means the actual bit usage may slightly deviate from (but is concentrated around) its expected value of b. This approach avoids introducing additional overhead from needing to communicate this information explicitly (although b

[5] One can slightly lower the quantization error by optimizing the quantization values for the actual distribution of the rotated coordinates (which is a shifted Beta distribution (Vargaftik et al., 2021)). However, as mentioned there, this distribution approaches the normal distribution rapidly as d grows (e.g., the difference is negligible even for d of several hundred), and our focus is on federated learning where d is considerably larger (e.g., millions).

does need to be sent or otherwise agreed upon). We observe that Theorems 2.1 and 2.3 hold for this scenario (the proofs are in Appendix B and D respectively).

For example, for $b = 1.5$, each coordinate is quantized using $Q_{\mathcal{I}_1}$ or $Q_{\mathcal{I}_2}$ with equal probability. The resulting $vNMSE$ for this case is (for $d \to \infty$):

$$\frac{1}{\frac{1}{2}\cdot\mathbb{E}\left[\left(\mathcal{Q}_{\mathcal{I}_1}(z)\right)^2\right]+\frac{1}{2}\cdot\mathbb{E}\left[\left(\mathcal{Q}_{\mathcal{I}_2}(z)\right)^2\right]}-1 \approx \frac{1}{\frac{1}{2}\cdot\left(\frac{2}{\pi}\right)+\frac{1}{2}\cdot(0.88228)}-1\approx 0.317.$$

We note that the naive solution of simply dividing the vector into two halves and sending each separately (one half using EDEN with $Q_{\mathcal{I}_1}$ and the other with $Q_{\mathcal{I}_2}$) yields a higher $vNMSE$ for the reconstructed vector (i.e., $\frac{1}{2}\cdot 0.571+\frac{1}{2}\cdot 0.134 = 0.352$ instead of 0.317).

Sub-bit compression ($b < 1$). For sub-bit compression, we use $Q_{\mathcal{I}_1}$ with random sparsification. Formally, the random sparsification procedure has only a single parameter $p \in (0, 1]$. For a vector $x \in \mathbb{R}^d$, its random sparsification is $\frac{1}{p}\cdot m_{rs}\circ x$, where $m_{rs} \in \{0, 1\}^d$ is a uniformly random sample of size $\|m_{rs}\|_1 = d \cdot p$ and $\circ$ stands for the coordinate-wise product (i.e., m_{rs} is a random mask). For random sparsification, it holds that

$$vNMSE = \frac{\sum_{i=1}^{d}(1-p)x[i]^2+p(\frac{1}{p}-1)^2x[i]^2}{\|x\|_2^2} = \frac{1}{p}-1\,.$$

There are two ways to apply the random sparsification: before or after the rotation. In practice, we find that the resulting $vNMSE$ is similar in both cases and use the sparsification prior to rotation which is more efficient since it reduces the dimension of the compressed vector.

For example, for $b = 0.7$, we sparsify uniformly at random 30% of the coordinates (i.e., set $p = 0.7$), multiply the remaining coordinates by a factor of $\frac{1}{0.7}$ to preserve unbiasedness, and then compress them using EDEN with $Q_{\mathcal{I}_1}$ (i.e., a single bit per coordinate). In turn, the receiver decodes the compressed sparsified vector and then restores the original using the same random mask (i.e., generating the same random mask using the shared randomness).

We strengthen the above choice using the following formal result whose proof appears in Appendix E.

Lemma 4.1. *Consider two unbiased compression techniques $\mathcal{A}$ and $\mathcal{B}$ (i.e., $\forall x : \mathbb{E}[\mathcal{A}(x)] = \mathbb{E}[\mathcal{B}(x)] = x$) with independent randomness. Then,*

1. $\forall x : \frac{\mathbb{E}[\|x-\mathcal{A}(x)\|_2^2]}{\|x\|_2^2} \le A$ *and* $\frac{\mathbb{E}[\|x-\mathcal{B}(x)\|_2^2]}{\|x\|_2^2} \le B \implies \forall x : \frac{\mathbb{E}[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2]}{\|x\|_2^2} \le A + AB + B\,.$
2. $\forall x : \frac{\mathbb{E}[\|x-\mathcal{A}(x)\|_2^2]}{\|x\|_2^2} \ge A$ *and* $\frac{\mathbb{E}[\|x-\mathcal{B}(x)\|_2^2]}{\|x\|_2^2} \ge B \implies \forall x : \frac{\mathbb{E}[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2]}{\|x\|_2^2} \ge A + AB + B\,.$

Accordingly, we get that EDEN with sparsification has $vNMSE \le \frac{\pi}{2\cdot p}-1+O\left(\sqrt{\frac{\log d}{d\cdot p^2}}\right)$ and obtain the following.

Corollary 4.2. *EDEN's $vNMSE$ with constant $b \in (0, 1]$ bits per coordinate satisfies*

$$\lim_{d\to\infty} vNMSE = \frac{\pi}{2\cdot b}-1\,.$$

For example, using $b = 0.1$ yields a $vNMSE$ of ≈ 14.707. More generally, for any fixed bit budget $b > 0$ we have that $vNMSE = O(1)$ and thus we get $NMSE = O(\frac{1}{n})$.

4.2. Lossy Networks

Distributed and federated learning systems (e.g., Bai et al. (2021); Jiang et al. (2020)) typically assume reliable packet delivery, e.g., using TCP to retransmit lost packets or using RDMA or RoCEv2, which rely on a lossless fabric. However, it is useful to design algorithms that can cope with packet loss on standard IP networks. Indeed, a recent effort by Ye et al. (2021) extends SGD to support packet loss.

We model packet loss as a sparsification with parameter $p \in (0, 1]$, so a fraction p of the coordinates arrive at the receiver. Hence this modeling is similar to that of the sub-bit regime, but there are inherent differences: (1) *The sparsification may not be random*. Instead, we assume an oblivious adversary may pick any subset of packets to drop. That is, the adversary may choose to drop packets based only on the packet indices, without any knowledge about the content of the packets; (2) The sparsification is done after the rotation; (3) The quantization scheme is not restricted to using $\mathcal{I}_1$.

To support this scheme, no changes at the sender are required. For the receiver, before employing the inverse rotation and scaling, it simply treats any lost coordinates as 0 and multiplies the reconstructed rotated vector by $\frac{1}{p}$ (the receiver determines p by counting the number of received coordinates). This scheme preserves our theoretical guarantees due to the following Lemma, which is proven in Appendix F.

Lemma 4.3. *Let $x \in \mathbb{R}^d$ and let $m_{ds} \in \{0, 1\}^d$ be a* deterministic *mask. Denote $p = \frac{\|m_{ds}\|_1}{d}$ and let $\mathcal{R}_{ds}(x) = \frac{1}{p}\cdot m_{ds}\circ\mathcal{R}(x)$. Then, using EDEN with $\mathcal{R}_{ds}(x)$ instead of $\mathcal{R}(x)$ results in:*

1. $\mathbb{E}\left[\widehat{x}\right] = x\,.$
2. $vNMSE \le \frac{1}{p\cdot\mathbb{E}\left[(\mathcal{Q}(z))^2\right]}-1+O\left(\sqrt{\frac{\log d}{d\cdot p^2}}\right).$

Lemma 4.3 shows that when performing DME, we can settle for partial information from all the senders and can replace

missing information by 0s and rescale each vector accordingly. Of course, this increases the error, but Lemma 4.3 bounds this increase. This feature enables the use of lossy transport protocols such as UDP (e.g., as proposed by Ye et al. (2021)), or allows the receiver to avoid waiting for retransmissions of lost packets. Hence we can trade-off error and delay (latency) by allowing partial information. Another benefit is that lossy protocols often have reduced overheads (e.g., smaller headers) since they do not require maintaining state and reliable delivery.

4.3. Variable-Length Encoding

A standard approach to compressing vectors with a small number of possible values that are unequally distributed is using entropy encoding methods such as Huffman (Huffman, 1952) or arithmetic encoding (Pasco, 1976; Rissanen, 1976). Intuitively, the number of times each quantization value appears in $\mathcal{Q}(\eta_x \cdot \mathcal{R}(x))$ may not be $\frac{d}{|\mathcal{I}|}$. Indeed, for $\mathcal{I}_b$ with $b \geq 2$, the probability of the values is not equal.

Formally, denoting by $p_I = \mathbb{P}(z \in I)$ the probability that a normal variable $z \in \mathcal{N}(0,1)$ lies in the interval I, $H_{\mathcal{I}} \triangleq \sum_{I \in \mathcal{I}} -p_I \log_2(p_I)$ is the *entropy* of the distribution induced by $\mathcal{I}$. For example, as discussed in Example 2 (§3), for $\mathcal{I}_2$, the intervals $[-0.9816, 0], [0, 0.9816]$ have probability that is more than double than that of $(-\infty, 0.9816], [0.9816, \infty)$. The entropy of the distribution is $H_{\mathcal{I}_2} \approx 1.91$ bits. This suggests that, for a large enough dimension d, we should be able to compress the vector to at most $H_{\mathcal{I}_2} + \epsilon$ bits per coordinate for any constant ϵ.

Such an encoding may also allow us to use more quantization values for the same bit budget. For example, using the Lloyd-Max Scalar Quantizer (Lloyd, 1982; Max, 1960) with 9 quantization values, we get an entropy of ≈ 2.98 bits, which would allow us to use three bits per coordinate for large enough vectors. In particular, this would reduce the $vNMSE$ by nearly 20%, compared to $\mathcal{I}_3$, at the cost of additional computation.

Taking this a step further, it is possible to consider the resulting entropy when choosing the set of intervals. To optimize the quantization, we are looking for a set $\mathcal{I}$ that maximizes $\mathbb{E}\left[\left(\mathcal{Q}_{\mathcal{I}}(z)\right)^2\right]$ such that its entropy is bounded by b. This problem is called Entropy-Constrained Vector Quantization (ECVQ) (Chou et al., 1989). The algorithm proposed in Chou et al. (1989) has several tunable parameters that may affect the output of the algorithm. We implemented the algorithm and scanned a large variety of parameter values. For $b = 3$, for example, the best obtained $vNMSE$ is ≈ 0.02274, compared to an $vNMSE$ of ≈ 0.03572 for EDEN with $\mathcal{I}_3$ and without entropy encoding.

We propose a simpler approach that is more computationally efficient. Given a bandwidth constraint $b > 0$, let $\Delta_b > 0$ denote the smallest real number such that $H_{\mathcal{I}_{\Delta_b}} \leq b$ where $\mathcal{I}_{\Delta_b} = \left\{[\Delta_b \cdot (n - \frac{1}{2}), \Delta_b \cdot (n + \frac{1}{2})] \mid n \in \mathbb{Z}\right\}$.This choice of $\mathcal{I}$ respects the properties that are required by Theorems 2.1 and 2.3 for any fixed $\Delta_b \in (0, 2)$.

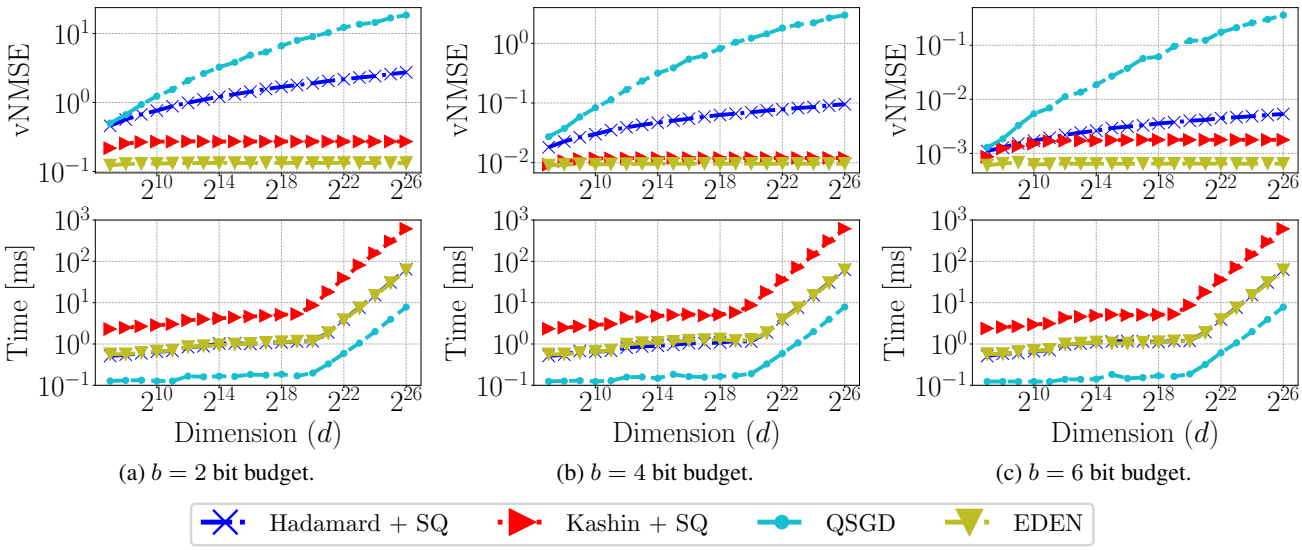
For example, using $b = 3$, we have $\Delta_3 \approx 0.5224$. Then, the resulting $vNMSE$ is ≈ 0.022741 (within 0.1% from the ECVQ solution), an improvement of $\approx 20\%$ over the Lloyd-Max Scalar Quantizer with 9 quantization values (which we can encode with $b = 3$ bits per coordinate by applying entropy encoding) and of $\approx 36\%$ over 8 quantization values (which is encodable with $b = 3$ without entropy encoding). A benefit of our approach is that it allows computing the quantization much faster as each $a \in \mathbb{R}$ is efficiently mapped into the interval $\left[\Delta_b \cdot \left(\left\lfloor \frac{a}{\Delta_b} \right\rceil - \frac{1}{2}\right), \Delta_b \cdot \left(\left\lfloor \frac{a}{\Delta_b} \right\rceil + \frac{1}{2}\right)\right]$, where $\lfloor \cdot \rceil$ rounds to the nearest integer (i.e., the interval's index is $n = \left\lfloor \frac{a}{\Delta_b} \right\rceil$).

Rate-distortion theory implies a lower bound on $\mathbb{E}\left[(z - \mathcal{Q}_{\mathcal{I}}(z))^2\right]$ for any quantization interval set $\mathcal{I}$ (Cover, 1999). Specifically, it implies that $\mathbb{E}\left[(z - \mathcal{Q}_{\mathcal{I}}(z))^2\right] \geq 4^{-b}$ for any $\mathcal{I}$ such that $H_{\mathcal{I}} \leq b$. Note that since $\mathbb{E}\left[(\mathcal{Q}_{\mathcal{I}}(z))^2\right] = 1 - \mathbb{E}\left[(z - \mathcal{Q}_{\mathcal{I}}(z))^2\right]$, we get that the lower bound on the $vNMSE$ attainable using any quantization is $\frac{1}{1-4^{-b}} - 1 = \frac{4^{-b}}{1-4^{-b}}$. Also, in Appendix G.1, we illustrate the different quantization values and their expected squared error (Figure 4) and show that our $\mathcal{I}_{\Delta_b}$ approach is close to the lower bound while allowing computationally efficient quantization. In practice, the entropy $H_{\mathcal{I}}$ may deviate from its expectation since the frequency of a quantization value, q_I, may not be exactly $p_I \cdot d$. However, due to concentration, this has minimal effect. We elaborate on this in Appendix G.2.

5. Evaluation

We evaluate EDEN using different federated and distributed learning tasks. We compare with a non-compressed baseline that uses 32-bit floating-point representation for each coordinate (Float32) and the following DME techniques: (1) Stochastic quantization (SQ) applied after the randomized Hadamard transform (Hadamard + SQ) (Suresh et al., 2017; Konečnỳ & Richtárik, 2018)[6]; (2) SQ applied over the vector's Kashin's representation (Kashin + SQ) (Lyubarskii & Vershynin, 2010; Caldas et al., 2018b; Safaryan et al., 2020); and (3) QSGD (Alistarh et al., 2017), which normalizes the input vector by its euclidean norm and separately sends its sign and its quantized (using SQ) absolute values.

[6] SQ (Barnes et al., 1951; Connolly et al., 2021) normalizes the vector into the range $[0, 2^b - 1]$ (using min-max normalization), adds uniform noise in $(-0.5, 0.5)$, and then rounds to the nearest integer. thus providing an unbiased estimate of each coordinate.



(a) $b = 2$ bit budget. (b) $b = 4$ bit budget. (c) $b = 6$ bit budget.

Figure 2. The *vNMSE* and compression time as a function of the dimension d for LogNormal(0,1) distribution.

We exclude methods that involve client-side memory since these can often work in conjunction with all tested methods (e.g., Karimireddy et al. (2019); Richtárik et al. (2021)) and are less applicable in cross-device federated scenarios (Kairouz et al., 2021). Also, we omit DRIVE (Vargaftik et al., 2021) since its performance is identical to that of EDEN with $\mathcal{I}_1$ and no packet losses.

Unless otherwise noted, we evaluate the algorithms without variable-length encoding which increases encoding time. We compare with SQGD + Elias Omega encoding (Alistarh et al., 2017) and optimized stochastic quantization + Huffman (Suresh et al., 2017) in Appendix G.1.

Similarly to Hadamard + SQ, Kashin + SQ, and DRIVE, instead of using a uniform random rotation (which requires $O(d^3)$ time and $O(d^2)$ space) to rotate the vector, we use the randomized Hadamard transform (a.k.a. *structured* random rotation (Suresh et al., 2017; Ailon & Chazelle, 2009)) that admits a fast, *in-place*, parallelizable, and GPU-friendly, $O(d \log d)$ time implementation (Fino & Algazi, 1976; Li et al., 2018; Ailon & Chazelle, 2009). As with these prior works, we find essentially negligible difference in our evaluation between using the Hadamard rotations and fully random rotations. We discuss this further and show supporting empirical measurements in Appendix H.

5.1. Implementation Optimization

In a natural implementation, EDEN has additional complexity when using $b \neq 1$ bits per coordinate. Indeed, for $b < 1$, we need to sample the sparsification mask, and for $b > 1$, we need to identify the interval each rotated coordinate lies in and take its center of mass. The latter can be efficiently done using a binary search - e.g., `torch.bucketize`, leading to an encoding complexity of $O(d \cdot b)$).

Instead, we implement EDEN using a fine-grained lookup table with a resulting encoding complexity of $O(d)$ (i.e., *independent of* b). That is, we map each value z to an integer $n_z = \left\lfloor \frac{z}{\gamma} \right\rceil$ for a suitably selected small value γ, and our lookup table maps n_z to the message the sender sends. Similarly, we have a receiver lookup table that maps n_z to an estimated value. The choice of γ provides a trade-off between space and accuracy. We note that, even with tables whose size is small compared to the encoded vector (e.g., 0.1%), the table's granularity is fine enough to get that the additional error is negligible (e.g., less than 0.01%) compared with the error of the algorithm. Further, it does *not* affect the unbiasedness of the algorithm, which is guaranteed by taking the correct scale (see Theorem 2.1). This approach allows us to encode and decode coordinates with minimal computation, especially when variable length encoding is not used.

5.2. *vNMSE*, *NMSE*, and Encoding Speed

We next evaluate the *vNMSE* and encoding speed of EDEN, comparing to three other DME techniques. In Figure 2, we provide representative results for a bit budget $b = 2, 4, 6$ for vectors that are drawn from a LogNormal(0,1) distribution. Each data point is averaged over 100 trials. EDEN offers the best *vNMSE* and is faster than Kashin + SQ, which offers the second lowest error. In line with theory, the *vNMSE* of Hadamard + SQ and the fast QSGD increases with the dimension. In all experiments, EDEN's encoding time accounts to less than 1% of the computation of the gradient. Appendix I.1 provides further experiments of *NMSE* and encode speeds, all indicating similar trends.

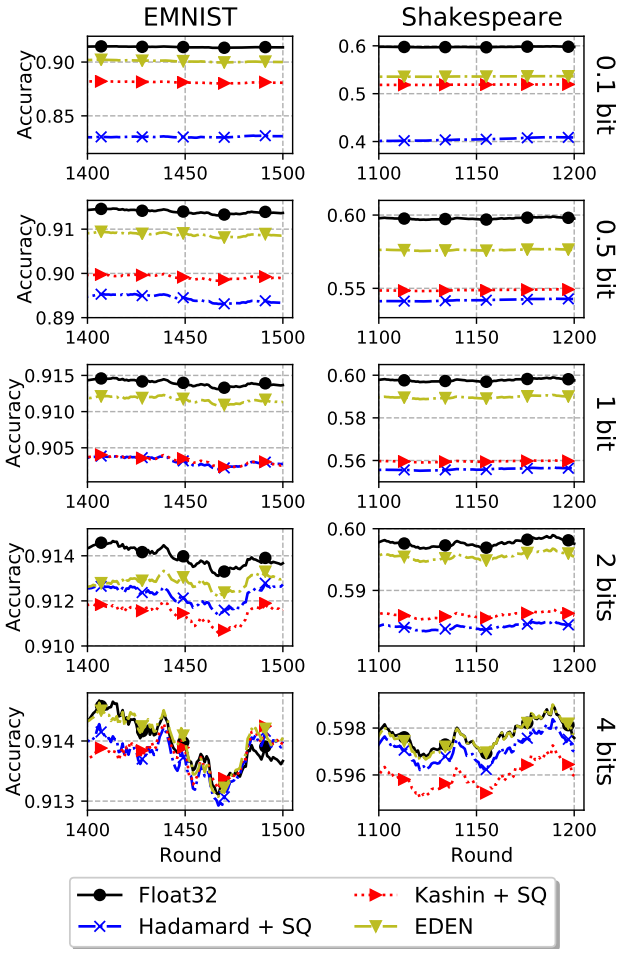


Figure 3. FedAvg over the EMNIST and Shakespeare tasks (columns) at various bit budgets (rows). We report training accuracy per round with a smoothing rolling mean window of 200 rounds. Sparsification is done using a random mask as described in §4.1. Plots are zoomed-in on the last 100 rounds (note the y-axis differences). A zoom-out version is included in Appendix I.2.

5.3. Federated Learning

We evaluate EDEN over the federated versions of the EMNIST (Cohen et al., 2017) image classification task and the Shakespeare (Shakespeare) next-word prediction task. We excluded QSGD, which was less competitive in these experiments. We run *FedAvg* (McMahan et al., 2017) with the Adam server optimizer (Kingma & Ba, 2015) and sample $n = 10$ clients per round. We re-use code, client partitioning, models, and hyperparameters from the federated learning benchmark of Reddi et al. (2021). Those are restated for convenience in Appendix I.2.

Figure 3 shows how EDEN compares with other compression schemes at various bit budgets. We notice that EDEN considerably outperforms other methods at the lower bit regimes. At 4 bits, all methods converge near the baseline, while EDEN still maintains a relative advantage.

5.4. Additional Evaluation

Due to space limits, we defer additional evaluation results to the Appendix. In particular, we provide experiments for variable-length encoding (Appendix G.1); structured rotation performance against the theory of uniform rotation (Appendix H); $NMSE$, $vNMSE$, and encoding speed (Appendix I.1); distributed logistic regression (Appendix I.3); comparison of sub-bit compression and network loss (Appendix I.4); distributed power iteration (Appendix I.5); homogeneous federated learning (Appendix I.6); and cross-device federated learning (Appendix I.7).

To summarize these experiments, we show EDEN outperforms its competitors in nearly all cases, offering a combination of speed, accuracy, overall compression, and robustness, and we believe that this will make it the best choice for many applications.

6. Conclusions

In this paper, we presented EDEN, a robust and accurate distributed mean estimation technique. EDEN suits various network scenarios, including packet losses and heterogeneous clients. Further, we proved strong accuracy guarantees for a wide range of usage scenarios, including using entropy encoding to compress quantized vectors further and working over lossy networks while maintaining high precision. Our evaluation results indicate that EDEN considerably outperforms all tested techniques in nearly all settings.

As future work, we propose to study how to combine EDEN with techniques that provide fast receiver decode procedures, e.g., using a single rotation for all senders to avoid inverse rotating individual vectors. Another direction is to combine EDEN with techniques such as secure aggregation and differential privacy. It is also interesting to explore if EDEN can be adapted to all-reduce techniques, which benefit large-scale distributed deployments where a parameter server might be a bottleneck. Finally, while EDEN naturally extends to linear schemes such as weighted mean, we propose to study how to incorporate non-linear aggregation functions, such as approximate geometric median, that may improve the training robustness (Pillutla et al., 2022).

Our source code is available at:
`https://github.com/amitport/EDEN-Distributed-Mean-Estimation`.

Acknowledgements

We thank the anonymous reviews and Moshe Gabel for their insightful feedback and suggestions. Michael Mitzenmacher was supported in part by NSF grants CCF-2101140, CNS-2107078, and DMS-2023528. Amit Portnoy was supported in part by the Cyber Security Research Center at Ben-Gurion University of the Negev.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Ailon, N. and Chazelle, B. The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

Alistarh, D.-A., Hoefler, T., Johansson, M., Konstantinov, N. H., Khirirat, S., and Renggli, C. The Convergence of Sparsified Gradient Methods. *Advances in Neural Information Processing Systems*, 31, 2018.

Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., and Schmidt, L. Practical and Optimal LSH for Angular Distance. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 1225–1233, 2015.

Bai, Y., Li, C., Zhou, Q., Yi, J., Gong, P., Yan, F., Chen, R., and Xu, Y. Gradient Compression Supercharged High-Performance Data Parallel DNN Training. In *ACM SOSP*, 2021.

Banner, R., Nahshan, Y., and Soudry, D. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *NeurIPS*, 2019.

Barnes, R., Cooke-Yarborough, E., and Thomas, D. An Electronic Digital Computor Using Cold Cathode Counting Tubes for Storage. *Electronic Engineering*, 1951.

Ben-Basat, R., Mitzenmacher, M., and Vargaftik, S. How to Send a Real Number Using a Single Bit (And Some Shared Randomness). In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pp. 25:1–25:20, 2021. URL https://doi.org/10.4230/LIPIcs.ICALP.2021.25.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signSGD: Compressed Optimisation for Non-Convex Problems. In *International Conference on Machine Learning*, pp. 560–569, 2018.

Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. On Biased Compression For Distributed Learning. *arXiv preprint arXiv:2002.12410*, 2020.

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

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A Benchmark for Federated Settings. *arXiv preprint arXiv:1812.01097*, 2018a.

Caldas, S., Konečný, J., McMahan, H. B., and Talwalkar, A. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *arXiv preprint arXiv:1812.07210*, 2018b.

Charikar, M., Chen, K., and Farach-Colton, M. Finding Frequent Items in Data Streams. In *International Colloquium on Automata, Languages, and Programming*, pp. 693–703. Springer, 2002.

Chmiel, B., Ben-Uri, L., Shkolnik, M., Hoffer, E., Banner, R., and Soudry, D. Neural Gradients are Near-Lognormal: Improved Quantized and Sparse Training. In *International Conference on Learning Representations*. OpenReview.net, 2021. URL https://openreview.net/forum?id=EoFNy62JGd.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. Structured Evolution with Compact Architectures for Scalable Policy Optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Choromanski, K. M., Rowland, M., and Weller, A. The Unreasonable Effectiveness of Structured Random Orthogonal Embeddings. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/bf8229696f7a3bb4700cfddef19fa23f-Paper.pdf.

Chou, P. A., Lookabaugh, T., and Gray, R. M. Entropy-Constrained Vector Quantization. *IEEE Transactions on acoustics, speech, and signal processing*, 37(1):31–42, 1989.

Chung, F. and Lu, L. Concentration Inequalities and Martingale Inequalities: a Survey. *Internet Mathematics*, 3 (1):79–127, 2006.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. EMNIST: Extending MNIST to Handwritten Letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Connolly, M. P., Higham, N. J., and Mary, T. Stochastic Rounding and Its Probabilistic Backward Error Analysis. *SIAM Journal on Scientific Computing*, 43(1): A566–A585, 2021. doi: 10.1137/20M1334796. URL https://doi.org/10.1137/20M1334796.

Cover, T. M. *Elements of Information Theory*. John Wiley & Sons, 1999.

Davies, P., Gurunanthan, V., Moshrefi, N., Ashkboos, S., and Alistarh, D. New Bounds For Distributed Mean Estimation and Variance Reduction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=t86MwoUCCNe.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M. a., Senior, A., Tucker, P., Yang, K., Le, Q., and Ng, A. Large Scale Distributed Deep Networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper/2012/file/6aca97005c68f1206823815f66102863-Paper.pdf.

Fei, J., Ho, C.-Y., Sahu, A. N., Canini, M., and Sapio, A. Efficient Sparse Collective Communication and its Application to Accelerate Distributed Deep Learning. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pp. 676–691, 2021.

Fino, B. J. and Algazi, V. R. Unified Matrix Treatment of the Fast Walsh-Hadamard Transform. *IEEE Transactions on Computers*, 25(11):1142–1146, 1976.

Gorbunov, E., Burlachenko, K. P., Li, Z., and Richtarik, P. MARINA: Faster Non-Convex Distributed Learning with Compression. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3788–3798. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/gorbunov21a.html.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Herendi, T., Siegl, T., and Tichy, R. F. Fast Gaussian Random Number Generation Using Linear Transformations. *Computing*, 59(2):163–181, 1997.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9:1735–1780, 1997.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., and Chen, z. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/093f65e080a295f8076b1c5722a46aa2-Paper.pdf.

Huffman, D. A. A Method for the Construction of Minimum-redundancy Codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

Ivkin, N., Rothchild, D., Ullah, E., Braverman, V., Stoica, I., and Arora, R. Communication-Efficient Distributed SGD With Sketching. *Advances in neural information processing systems*, 2019.

Jiang, Y., Zhu, Y., Lan, C., Yi, B., Cui, Y., and Guo, C. A Unified Architecture for Accelerating Distributed DNN Training in Heterogeneous GPU/CPU Clusters. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pp. 463–479. USENIX Association, November 2020. ISBN 978-1-939133-19-9. URL https://www.usenix.org/conference/osdi20/presentation/jiang.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14 (1–2):1–210, 2021. doi: 10.1561/2200000083. URL http://dx.doi.org/10.1561/2200000083.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error Feedback Fixes SignSGD and other Gradient Compression Schemes. In *International Conference on Machine Learning*, pp. 3252–3261, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic Controlled Averag-

ing for Federated Learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Kohavi, R. et al. Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *KDD*, volume 96, pp. 202–207, 1996.

Konečnỳ, J. and Richtárik, P. Randomized Distributed Mean Estimation: Accuracy vs. Communication. *Frontiers in Applied Mathematics and Statistics*, 4:62, 2018.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*, 2017.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Toronto*, 2009.

Laurent, B. and Massart, P. Adaptive Estimation of a Quadratic Functional by Model Selection. *Annals of Statistics*, pp. 1302–1338, 2000.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Cortes, C., and Burges, C. MNIST Handwritten Digit Database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the Intrinsic Dimension of Objective Landscapes. In *International Conference on Learning Representations*, 2018. Code available at: https://github.com/uber-research/intrinsic-dimension.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated Optimization in Heterogeneous Networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, B. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations*, 2018.

Lloyd, S. Least Squares Quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Lyubarskii, Y. and Vershynin, R. Uncertainty Principles and Vector Quantization. *IEEE Transactions on Information Theory*, 56(7):3491–3501, 2010.

Malinovskiy, G., Kovalev, D., Gasanov, E., Condat, L., and Richtárik, P. From Local SGD to Local Fixed-Point Methods for Federated Learning. In *International Conference on Machine Learning*, pp. 6692–6701, 2020. URL http://proceedings.mlr.press/v119/malinovskiy20a.html.

Max, J. Quantizing for Minimum Distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

Mitzenmacher, M. and Upfal, E. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge university press, 2017.

Nishio, T. and Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2019.

Pasco, R. C. *Source Coding Algorithms for Fast Data Compression*. PhD thesis, Stanford University CA, 1976.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8026–8037. 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

Platt, J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, January 1998. URL https://www.microsoft.com/en-us/research/publication/fast-training-of-support-vector-machines-using-sequential-minimal-optimization/.

Rader, C. M. A New Method of Generating Gaussian Random Variables by Computer. Technical report, Massachusetts Inst. of Tech Lexington Lincoln Lab, 1969.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Richtárik, P., Sokolov, I., and Fatkhullin, I. EF21: A New, Simpler, Theoretically Better, and Practically Faster Error Feedback. In *Advances in Neural Information Processing Systems*, 2021. URL https://papers.nips.cc/paper/2021/file/231141b34c82aa95e48810a9d1b33a79-Paper.pdf.

Rissanen, J. J. Generalized Kraft Inequality and Arithmetic Coding. *IBM Journal of research and development*, 20 (3):198–203, 1976.

Safaryan, M., Shulgin, E., and Richtárik, P. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA*, 2020.

Sapio, A., Canini, M., Ho, C.-Y., Nelson, J., Kalnis, P., Kim, C., Krishnamurthy, A., Moshref, M., Ports, D., and Richtarik, P. Scaling Distributed Machine Learning with In-Network Aggregation. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 785–808, 2021.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-Bit Stochastic Gradient Descent and Its Application to Data-Parallel Distributed Training of Speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Shakespeare, W. The Complete Works of William Shakespeare. https://www.gutenberg.org/ebooks/100.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified SGD with Memory. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/b440509a0106086a67bc2ea9df0a1dab-Paper.pdf.

Suresh, A. T., Felix, X. Y., Kumar, S., and McMahan, H. B. Distributed Mean Estimation With Limited Communication. In *International Conference on Machine Learning*, pp. 3329–3337. PMLR, 2017.

Thomas, D. B. Parallel Generation of Gaussian Random Numbers Using the Table-Hadamard Transform. In *2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines*, pp. 161–168. IEEE, 2013.

Vargaftik, S., Ben-Basat, R., Portnoy, A., Mendelson, G., Ben-Itzhak, Y., and Mitzenmacher, M. DRIVE: One-bit Distributed Mean Estimation. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 362–377. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/0397758f8990c1b41b81b43ac389ab9f-Paper.pdf.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A Field Guide to Federated Optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. In *Advances in neural information processing systems*, pp. 1509–1519, 2017.

Ye, H., Liang, L., and Li, G. Decentralized Federated Learning with Unreliable Communications. *arXiv preprint arXiv:2108.02397*, 2021.

Ye, X., Dai, P., Luo, J., Guo, X., Qi, Y., Yang, J., and Chen, Y. Accelerating CNN Training by Pruning Activation Gradients. In *European Conference on Computer Vision*, pp. 322–338. Springer, 2020.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. Orthogonal Random Features. *Advances in neural information processing systems*, 29: 1975–1983, 2016.

A. Alternative Compression Methods

In this paper, we focus on the DME problem, in which the participants do not keep state, and the estimate of each vector is desired to be unbiased for the $NMSE$ to decrease linearly with respect to the number of senders. We give a few examples of other approaches (i.e., works that do not directly address the DME problem).

Some works (e.g., Beznosikov et al. (2020)) investigate the convergence rate of Stochastic Gradient Decent (SGD) for biased compression (which are known to achieve lower error). Another approach to leverage the lower error of biased compression is using Error Feedback (EF). Namely, if the senders are persistent (the same devices are used over multiple rounds) and have the memory to store the error of their compressed gradient, they can use this information to compensate for the estimation error between rounds. Indeed, works such as Seide et al. (2014); Alistarh et al. (2018); Richtárik et al. (2021) show that EF-based approaches can greatly increase the accuracy of the learned models and ensure convergence of biased compressors such as Top-k (Stich et al., 2018) and SketchedSGD (Ivkin et al., 2019).

For a setting with persistent clients, recent works (e.g., Mishchenko et al. (2019); Gorbunov et al. (2021)) also suggest encoding the difference between the current gradient and the one from the previous round. Intuitively, when the mini-batch sizes are sufficiently large, the sampled gradients are less noisy, and encoding the differences allows faster convergence. This approach is orthogonal to EDEN which can encode the difference in such a setting.

For distributed cluster learning, some works aim at optimizing streaming aggregation (i.e., All-Reduce operations) via programmable hardware (Sapio et al., 2021) or taking advantage of data sparsity (Fei et al., 2021). These approaches are known to be orthogonal to (and can work in conjunction with) compression techniques (Vargaftik et al., 2021; Fei et al., 2021). For example, if the input is sparse (or is sparsified), one can use EDEN to encode only the non-zero coordinates.

Deep gradient compression (Lin et al., 2018) leverages redundancy in neural network gradients to reduce the number of transmitted bits. They leverage momentum correction, local gradient clipping, momentum factor masking, and warm-up training, and report compression ratios of 270x-600x.

For further overview we refer the reader to Konečný et al. (2017); Kairouz et al. (2021); Wang et al. (2021).

B. EDEN's Unbiasedness

For clarity, we restate the theorem.

Theorem 2.1. *For all $x \in \mathbb{R}^d$, using EDEN with the scale $S = \frac{\|x\|_2^2}{\langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle}$ results in $\mathbb{E}[\widehat{x}] = x$.*

Proof. Our proof follows similar lines to that of Vargaftik et al. (2021). Denote $x' = (\|x\|_2, 0, \ldots, 0)^T$ and let $R_{x \to x'} \in \mathbb{R}^{d \times d}$ be a rotation matrix such that $R_{x \to x'} \cdot x = x'$. Further, denote $R_x = R R_{x \to x'}^{-1}$. Using these definitions we have that,

$$\begin{aligned}\widehat{x} =& R_{x \to x'}^{-1} \cdot R_{x \to x'} \cdot \widehat{x} = S \cdot R_{x \to x'}^{-1} \cdot R_{x \to x'} \cdot R^{-1} \cdot \mathcal{Q}\left(\eta_x \cdot R \cdot x\right) \\ =& S \cdot R_{x \to x'}^{-1} \cdot R_x^{-1} \cdot \mathcal{Q}\left(\eta_x \cdot R_x \cdot R_{x \to x'} \cdot x\right) = S \cdot R_{x \to x'}^{-1} \cdot R_x^{-1} \cdot \mathcal{Q}\left(\eta_x \cdot R_x \cdot x'\right) .\end{aligned}$$

Let C_i be a vector containing the values of the i'th column of R_x. Then, $R_x \cdot x' = \|x\|_2 \cdot C_0$ and we obtain,

$$R_x^{-1} \cdot \mathcal{Q}\left(\eta_x \cdot R_x \cdot x'\right) = \left(\langle C_0, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle, \ldots, \langle C_{d-1}, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle\right)^T .$$

Now, observe that

$$\langle R \cdot x, \mathcal{Q}(\eta_x \cdot R \cdot x)\rangle = \langle R_x \cdot x', \mathcal{Q}(\eta_x \cdot R_x \cdot x')\rangle = \|x\|_2 \cdot \langle C_0, \mathcal{Q}(\eta_x \cdot \|x\|_2 \cdot C_0)\rangle \ .$$

This yields,

$$\widehat{x} = R_{x \to x'}^{-1} \cdot \|x\|_2 \cdot \left(1, \frac{\langle C_1, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle}, \ldots, \frac{\langle C_{d-1}, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x \cdot \|x\|_2 \cdot C_0\right)\rangle}\right)^T . \tag{1}$$

Now, consider an algorithm EDEN' that operates exactly as EDEN but, instead of directly using the sampled rotation matrix $R = R_x \cdot R_{x \to x'}^{-1}$ it calculates and uses the rotation matrix $R' = R_x \cdot I' \cdot R_{x \to x'}^{-1} = R_{x \to x'} \cdot R \cdot I' \cdot R_{x \to x'}^{-1}$ where I' is identical to the d-dimensional identity matrix with the exception that $I'[0,0] = -1$ instead of 1.

Since both $R_{x\to x'}$ and $I'\cdot R_{x\to x'}^{-1}$ are fixed rotation matrices, R' and R follow the same distribution.

Now, consider a run of both algorithms where $\widehat{x}$ is the reconstruction of EDEN for x with a sampled rotation R and $\widehat{x}'$ is the corresponding reconstruction of EDEN' for x with the rotation R'.

According to (1) it holds that: $\widehat{x}+\widehat{x}' = R_{x\to x'}^{-1}\cdot\|x\|_2\cdot(2,0,\ldots,0)^T = 2\cdot x$. This is because both runs are identical except that the first column of R_x and $R_x\cdot I'$ have opposite signs and thus for all $i\in\{1,2,\ldots,d-1\}$:

$$\frac{\langle C_i,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}{\langle C_0,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}+\frac{\langle C_i,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot -C_0\right)\rangle}{\langle -C_0,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot -C_0\right)\rangle} = \frac{\langle C_i,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}{\langle C_0,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}-\frac{\langle C_i,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}{\langle C_0,\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle} = 0.$$

Finally, it holds that $\mathbb{E}\left[\widehat{x}+\widehat{x}'\right]=2\cdot x$. Also, since R_x and $R_x\cdot I'$ follow the same distribution, due to the linearity of expectation, both algorithms have the same expected value. This yields $\mathbb{E}\left[\widehat{x}\right]=\mathbb{E}\left[\widehat{x}'\right]=x$, and concludes the proof. □

C. EDEN's *NMSE*

Lemma C.1. *Consider n senders. It holds that*

$$NMSE=\frac{\sum_{c=1}^{n} vNMSE(c)\cdot\|x_c\|_2^2}{n\cdot\sum_{c=1}^{n}\|x_c\|_2^2}\ .$$

Proof. It holds that,

$$\begin{aligned}MSE&=\mathbb{E}\left[\left\|\frac{1}{n}\cdot\sum_{c=1}^{n}x_c-\frac{1}{n}\cdot\sum_{c=1}^{n}\widehat{x_c}\right\|_2^2\right]=\frac{1}{n^2}\cdot\sum_{c,c'}\mathbb{E}\left[\langle x_c-\widehat{x_c},x_{c'}-\widehat{x_{c'}}\rangle\right]\\&=\frac{1}{n^2}\cdot\sum_{c}\mathbb{E}\left[\langle x_c-\widehat{x_c},x_c-\widehat{x_c}\rangle\right]+\frac{1}{n^2}\cdot\sum_{c\neq c'}\mathbb{E}\left[\langle x_c-\widehat{x_c},x_{c'}-\widehat{x_{c'}}\rangle\right]\\&=\frac{1}{n^2}\sum_{c}\mathbb{E}\left[\|x_c-\widehat{x_c}\|_2^2\right]=\frac{1}{n^2}\cdot\sum_{c}\|x_c\|_2^2\cdot\mathbb{E}\left[\frac{\|x_c-\widehat{x_c}\|_2^2}{\|x_c\|_2^2}\right]=\frac{1}{n^2}\cdot\sum_{c}\|x_c\|_2^2\cdot vNMSE(c)\ .\end{aligned}$$

Here, we used $\mathbb{E}\left[\langle x_c-\widehat{x_c},x_{c'}-\widehat{x_{c'}}\rangle\right]=0$. This holds since the estimates of the different clients are unbiased (by Theorem 2.1) and independent. Finally, dividing the result by $\frac{1}{n}\cdot\sum_c\|x_c\|_2^2$ yields the result. □

D. EDEN's *vNMSE*

We devide the proof into two parts. First, we prove the main result in §D.1. Then, for better readability, we defer auxiliary lemmas to §D.2.

D.1. Theorem proof

For clarity, we restate the theorem.

Theorem 2.3. *Let $z\sim\mathcal{N}(0,1)$. For all $x\in\mathbb{R}^d$, with $S=\frac{\|x\|_2^2}{\langle\mathcal{R}(x),\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\rangle}$, EDEN satisfies:*

$$vNMSE\le\frac{1}{\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]}-1+O\left(\sqrt{\frac{\log d}{d}}\right).$$

Proof. We begin with bounding the sum of squared errors (SSE).

The SSE in estimating $\mathcal{R}(x)$ using $S\cdot\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))$ equals that of estimating x using $\widehat{x}$. Therefore,

$$\begin{aligned}\|x-\widehat{x}\|_2^2&=\|\mathcal{R}(x-\widehat{x})\|_2^2=\|\mathcal{R}(x)-\mathcal{R}(\widehat{x})\|_2^2=\left\|\mathcal{R}(x)-\mathcal{R}(\mathcal{R}^{-1}\left(S\cdot\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\right))\right\|_2^2\\&=\|\mathcal{R}(x)-\mathcal{S}\cdot\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\|_2^2=\|\mathcal{R}(x)\|_2^2-2S\left\langle\mathcal{R}(x),\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\right\rangle+S^2\|\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\|_2^2\\&=\|x\|_2^2-2S\left\langle\mathcal{R}(x),\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\right\rangle+S^2\|\mathcal{Q}(\eta_x\cdot\mathcal{R}(x))\|_2^2\ .\end{aligned}$$

Using $S = \frac{\|x\|_2^2}{\langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle}$, the SSE becomes:

$$\begin{aligned}\|x - \widehat{x}\|_2^2 &= \|x\|_2^2 - 2S \langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle + S^2 \|\mathcal{Q}(\eta_x \cdot \mathcal{R}(x))\|_2^2 \\ &= \frac{\|x\|_2^4 \|\mathcal{Q}(\eta_x \cdot \mathcal{R}(x))\|_2^2}{\langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle^2} - \|x\|_2^2 .\end{aligned}$$

Thus, the resulting $vNMSE$ in this case is:

$$\mathbb{E}\left[\frac{\|x - \widehat{x}\|_2^2}{\|x\|_2^2}\right] = \mathbb{E}\left[\frac{\|x\|_2^2 \|\mathcal{Q}(\eta_x \cdot \mathcal{R}(x))\|_2^2}{\langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle^2}\right] - 1 .$$

Next, since $\mathcal{R}(x)$ is uniformly distributed on a sphere with a radius $\|x\|_2$, its distribution is given by $\|x\|_2 \cdot \frac{Z}{\|Z\|_2}$ where $Z = (z_1, \ldots, z_d)$ such that $\{z_i\}_{i=1}^d$ are i.i.d random variables and $z_i \sim \mathcal{N}(0,1)\ \forall i$. This yields

$$\frac{\|x\|_2^2 \|\mathcal{Q}(\eta_x \cdot \mathcal{R}(x))\|_2^2}{\langle \mathcal{R}(x), \mathcal{Q}(\eta_x \cdot \mathcal{R}(x)) \rangle^2} \overset{\mathrm{d}}{=} \frac{\|x\|_2^2 \left\|\mathcal{Q}(\sqrt{d} \cdot \frac{Z}{\|Z\|_2})\right\|_2^2}{\left\langle \|x\|_2 \cdot \frac{Z}{\|Z\|_2}, \mathcal{Q}(\sqrt{d} \cdot \frac{Z}{\|Z\|_2}) \right\rangle^2} = \frac{d \cdot \left\|\mathcal{Q}(\sqrt{d} \cdot \frac{Z}{\|Z\|_2})\right\|_2^2}{\left\langle \sqrt{d} \cdot \frac{Z}{\|Z\|_2}, \mathcal{Q}(\sqrt{d} \cdot \frac{Z}{\|Z\|_2}) \right\rangle^2} = \frac{d \cdot \left\|\mathcal{Q}(\tilde{Z})\right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2} ,$$

where $\overset{\mathrm{d}}{=}$ means equality in distribution and we denote $\tilde{Z} = \sqrt{d} \cdot \frac{Z}{\|Z\|_2}$. Thus, our goal is to upper-bound:

$$\mathbb{E}\left[\frac{d \cdot \left\|\mathcal{Q}(\tilde{Z})\right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2}\right] .$$

For some $0 < \alpha, \beta < \frac{1}{2}$, denote the events

$$A = \left\{ d \cdot (1-\alpha) \le \|Z\|_2^2 \le d \cdot (1+\alpha) \right\} , \quad B = \left\{ \left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle > \frac{\mathbb{E}\left[\left\|\mathcal{Q}(\tilde{Z})\right\|_2^2 \right]}{\sqrt{1+\beta}} \right\} .$$

Further denote

$$f(Z) \triangleq \frac{d \cdot \left\|\mathcal{Q}(\tilde{Z})\right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2} .$$

Then,

$$\begin{aligned} f(Z) \le& \mathbb{E}\left[f(Z) \cdot \mathbb{1}_{A \cap B}\right] + \sup_Z \left(f(Z)\right) \cdot \mathbb{P}(A^c \cup B^c) \\ \le& \mathbb{E}\left[f(Z) \cdot \mathbb{1}_{A \cap B}\right] + \sup_Z \left(f(Z)\right) \cdot \left(\mathbb{P}(A^c) + \mathbb{P}(A \cap B^c)\right) . \end{aligned}$$

Next, it holds that

$$\mathbb{E}\left[f(Z) \cdot \mathbb{1}_{A \cap B}\right] \le \mathbb{E}\left[\frac{d \cdot \left\|\mathcal{Q}(\tilde{Z})\right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2} \cdot \mathbb{1}_{A \cap B}\right] = \mathbb{E}\left[\frac{(1+\beta) \cdot d}{\mathbb{E}\left[\left\|\mathcal{Q}(\tilde{Z})\right\|_2^2 \right]} \cdot \mathbb{1}_{A \cap B}\right] \le \frac{1+\beta}{\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2 \right] - \frac{1}{2} \cdot \alpha \cdot M(\mathcal{I})} .$$

In the above, we used Lemma D.4 by which given that A holds, it holds that

$$\mathbb{E}\left[\left\|\mathcal{Q}(\tilde{Z})\right\|_2^2 \right] \ge \mathbb{E}\left[\left\|\mathcal{Q}(\frac{Z}{\sqrt{1+\alpha}})\right\|_2^2 \right] \ge d \cdot \left(\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2 \right] - (\sqrt{1+\alpha} - 1) \cdot M(\mathcal{I}) \right) ,$$

where $M(\mathcal{I}) > 0$ is a constant that depends on the quantization and we replaced $\sqrt{1+\alpha} - 1 \ge \frac{1}{2} \cdot \alpha$ for any $\alpha < \frac{1}{2}$.

Now, we use three Lemmas whose proofs appear in §D.2:

1. By Lemma D.1 it holds that $\sup_Z \left(f(Z)\right) \leq d^2$.
2. By Lemma D.2 it holds that $\mathbb{P}(A^c) \leq 2 \cdot e^{-\frac{\alpha^2}{8} \cdot d}$.
3. By Lemma D.3, for $\alpha = \beta \cdot \frac{0.005}{\sqrt{M(\mathcal{I})}}$ it holds that $\mathbb{P}(A \cap B^c) \leq e^{-\alpha^2 \cdot M(\mathcal{I}) \cdot d}$.

These yield

$$\mathbb{E}\left[f(Z)\right] \leq \frac{1 + \alpha \cdot 200 \cdot \sqrt{M(\mathcal{I})}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - \frac{1}{2} \cdot \alpha \cdot M(\mathcal{I})} + d^2 \cdot \left(2 \cdot e^{-\frac{\alpha^2}{8} \cdot d} + e^{-\alpha^2 \cdot M(\mathcal{I}) \cdot d}\right) \ ,$$

where we used $\beta = \alpha \cdot \frac{\sqrt{M(\mathcal{I})}}{0.005} = \alpha \cdot 200 \cdot \sqrt{M(\mathcal{I})}$. Next, for some constant $k > 0$ setting $\alpha = \sqrt{k \cdot \frac{\ln d}{d}}$ yields

$$\mathbb{E}\left[f(Z)\right] \leq \frac{1 + \sqrt{k \cdot \frac{\ln d}{d}} \cdot 200 \cdot \sqrt{M(\mathcal{I})}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - \frac{1}{2} \cdot \sqrt{k \cdot \frac{\ln d}{d}} \cdot M(\mathcal{I})} + 2 \cdot d^2 \cdot \left(e^{-\frac{k \cdot \ln d}{8}}\right) + d^2 \cdot \left(e^{-k \cdot M(\mathcal{I}) \cdot \ln d}\right) \ .$$

Let $k = 2.5 \cdot \max\left\{8, \frac{1}{M(\mathcal{I})}\right\}$. This yields

$$\mathbb{E}\left[f(Z)\right] \leq \frac{1 + \sqrt{\frac{2.5 \cdot \max\left\{8, \frac{1}{M(\mathcal{I})}\right\} \ln d}{d}} \cdot 200 \cdot \sqrt{M(\mathcal{I})}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - \frac{1}{2} \cdot \sqrt{\frac{2.5 \cdot \max\left\{8, \frac{1}{M(\mathcal{I})}\right\} \ln d}{d}} \cdot M(\mathcal{I})} + \frac{3}{\sqrt{d}} \ .$$

To simplify the asymptotics of the above, we use the lower bound $\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] \geq 0.1$ (from Lemma D.7). Thus, for sufficiently large d we find

$$\begin{aligned}
\mathbb{E}\left[f(Z)\right] &\leq \frac{1 + c_1 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}} + \frac{3}{\sqrt{d}} = \frac{1}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}} + \frac{c_1 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}} + \frac{3}{\sqrt{d}} \\
&= \frac{1 + \frac{c_2 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]} + \frac{1 + c_1 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}} + \frac{3}{\sqrt{d}} \\
&= \frac{1}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]} + \frac{c_2 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] \cdot \left(\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}\right)} + \frac{c_1 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big] - c_2 \cdot \sqrt{\frac{\ln d}{d}}} + \frac{3}{\sqrt{d}} \\
&\leq \frac{1}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]} + \frac{c_2 \cdot \sqrt{\frac{\ln d}{d}}}{\left(\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]\right)^2} + \frac{c_1 \cdot \sqrt{\frac{\ln d}{d}}}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]} + \frac{3}{\sqrt{d}} = \frac{1}{\mathbb{E}\Big[\left(\mathcal{Q}(z)\right)^2\Big]} + O\left(\sqrt{\frac{\ln d}{d}}\right) \ .
\end{aligned}$$

This concludes the proof. □

D.2. Lemmas proof

Lemma D.1. *It holds that*

$$\sup_Z \left(f(Z)\right) = \sup_Z \left(\frac{d \cdot \left\|\mathcal{Q}(\tilde{Z})\right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2}\right) \leq d^2 \ .$$

Proof. We can rewrite and obtain

$$\frac{d \cdot \left\| \mathcal{Q}(\tilde{Z}) \right\|_2^2}{\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle^2} = \frac{1}{\langle \frac{Z}{\|Z\|_2}, \frac{\mathcal{Q}(\tilde{Z})}{\left\|\mathcal{Q}(\tilde{Z})\right\|_2} \rangle^2} \leq \frac{1}{\left(\frac{1}{d}\right)^2} = d^2 \,.$$

We used the fact that the maximal absolute value entry in a unit vector is at least $\frac{1}{\sqrt{d}}$ and that the maximal entry in both vectors has the same: (1) sign, due the symmetry of the quantization; (2) index, since for any $a_1, a_2 \in \mathbb{R}$ it holds that $a_1 \geq a_2 \implies \mathcal{Q}(a_1) \geq \mathcal{Q}(a_2)$. □

Lemma D.2. $\mathbb{P}(A^c) \leq 2 \cdot e^{-\frac{\alpha^2}{8} \cdot d}$.

Proof. We use a result from Laurent & Massart (2000) (Lemma 1) which we restate here for clarity:

Let U be chosen according to a chi-squared distribution with D degrees of freedom. Then, for any $\lambda > 0$:

$$\mathbb{P}(U - D \geq 2\sqrt{D\lambda} + 2\lambda) \leq e^{-\lambda} \quad \textit{and} \quad \mathbb{P}(D - U \geq 2\sqrt{D\lambda}) \leq e^{-\lambda} \,.$$

First, observe the above lemma yields that for any $\lambda > 0$:

$$\mathbb{P}(|U - D| \geq 2\sqrt{D\lambda} + 2\lambda) \leq 2 \cdot e^{-\lambda} \,.$$

We want to bound $\mathbb{P}(A^c) = \mathbb{P}\left(\left| \|Z\|_2^2 - d \right| \geq \alpha \cdot d\right)$. First, observe that $\alpha \cdot d \geq 2 \cdot \sqrt{d \cdot \frac{\alpha^2 \cdot d}{8}} + 2 \cdot \frac{\alpha^2 \cdot d}{8}$. Next, since $\|Z\|_2^2$ is chi-squared we obtain,

$$\begin{aligned}\mathbb{P}(A^c) = \; & \mathbb{P}\left(\left| \|Z\|_2^2 - d \right| \geq \alpha \cdot d\right) \leq \\ & \mathbb{P}\left(\left| \|Z\|_2^2 - d \right| \geq 2 \cdot \sqrt{d \cdot \frac{(\alpha^2 \cdot d)}{8}} + 2 \cdot \frac{(\alpha^2 \cdot d)}{8}\right) \leq 2 \cdot e^{-\frac{(\alpha^2 \cdot d)}{8}} \,.\end{aligned}$$ □

Lemma D.3. $\mathbb{P}(A \cap B^c) \leq e^{-\alpha^2 \cdot M(\mathcal{I}) \cdot d}$ *where $M(\mathcal{I})$ is a constant that depends on $\mathcal{I}$ and* $\alpha = \beta \cdot \frac{0.005}{\sqrt{M(\mathcal{I})}}$.

Proof. Observe that we cannot use a concentration bound directly on $\left\langle \tilde{Z}, \mathcal{Q}(\tilde{Z}) \right\rangle$ since its entries are not independent (i.e., they are normalized by $\|Z\|_2$). Instead, we rely on event A and use that

$$\mathbb{P}(A \cap B^c) \leq \mathbb{P}\left(\left\langle \frac{Z}{\sqrt{1+\alpha}}, \mathcal{Q}\left(\frac{Z}{\sqrt{1+\alpha}}\right)\right\rangle \leq \frac{1}{\sqrt{1+\beta}} \cdot \mathbb{E}\left[\left\| \mathcal{Q}\left(\frac{Z}{\sqrt{1-\alpha}}\right)\right\|_2^2\right]\right) \,.$$

Our goal is to use the following result from Chung & Lu (2006) (Theorem 3.5) which we restate here for clarity:

If $X_1, X_2, ..., X_n$ are nonnegative independent random variables, we have the following bounds for the sum $X = \sum_{j=1}^n X_j$:

$$\mathbb{P}(X \leq \mathbb{E}[X] - \lambda) \leq e^{-\frac{\lambda^2}{2\sum_{j=1}^n \mathbb{E}\left[X_j^2\right]}} \,.$$

To do so, we use that, according to Lemmas D.4 and D.5,

- $\mathbb{E}\left[\left\langle \frac{Z}{\sqrt{1+\alpha}}, \mathcal{Q}\left(\frac{Z}{\sqrt{1+\alpha}}\right)\right\rangle\right] \geq \frac{1}{\sqrt{1+\alpha}} \cdot \mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] - (\sqrt{1+\alpha} - 1) \cdot M_1(\mathcal{I}) \cdot d \,.$
- $\mathbb{E}\left[\left\| \mathcal{Q}\left(\frac{Z}{\sqrt{1-\alpha}}\right)\right\|_2^2\right] \leq \frac{1}{\sqrt{1-\alpha}} \mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] + (1 - \sqrt{1-\alpha}) \cdot M_2(\mathcal{I}) \cdot d \,.$

where $M_i(\mathcal{I})$ for $i = 1, 2$ are finite constants that depend on $\mathcal{I}$.

Next, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \frac{Z}{\sqrt{1+\alpha}}, \mathcal{Q}\left(\frac{Z}{\sqrt{1+\alpha}}\right)\right\rangle\right] - \frac{1}{\sqrt{1+\beta}} \cdot \mathbb{E}\left[\left\|\mathcal{Q}\left(\frac{Z}{\sqrt{1-\alpha}}\right)\right\|_2^2\right] \\
&\geq \left(\frac{1}{\sqrt{1+\alpha}}\mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] - (\sqrt{1+\alpha}-1)M_1(\mathcal{I})\cdot d\right) - \frac{1}{\sqrt{1+\beta}}\left(\frac{1}{\sqrt{1-\alpha}}\mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] + (1-\sqrt{1-\alpha})M_2(\mathcal{I})\cdot d\right) \\
&= \mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right]\cdot\left(\frac{1}{\sqrt{1+\alpha}} - \frac{1}{\sqrt{1+\beta}\sqrt{1-\alpha}}\right) - d\cdot\left(M_2(\mathcal{I})\cdot\frac{(1-\sqrt{1-\alpha})}{\sqrt{1+\beta}} + M_1(\mathcal{I})\cdot(\sqrt{1+\alpha}-1)\right) \\
&= \left(\frac{0.1}{\sqrt{1+\alpha}} - \frac{0.1}{\sqrt{1+\beta}\sqrt{1-\alpha}}\right) - d\cdot\left(M_2(\mathcal{I})\cdot\frac{(1-\sqrt{1-\alpha})}{\sqrt{1+\beta}} + M_1(\mathcal{I})\cdot(\sqrt{1+\alpha}-1)\right) \\
&= d\cdot\left(\frac{0.1}{\sqrt{1+\alpha}} - \frac{0.1}{\sqrt{1+\beta}\sqrt{1-\alpha}} - M_2(\mathcal{I})\cdot\frac{(1-\sqrt{1-\alpha})}{\sqrt{1+\beta}} - M_1(\mathcal{I})\cdot(\sqrt{1+\alpha}-1)\right) \triangleq d\cdot\Phi(\alpha,\beta,\mathcal{I})\ .
\end{aligned}$$

Here, we used that by Lemma D.7 it holds that $\mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] \geq 0.1\cdot d$ and that our choice of α, β results in $\frac{0.1}{\sqrt{1+\alpha}} - \frac{0.1}{\sqrt{1+\beta}\sqrt{1-\alpha}} > 0$ (we later show that the constant $M(\mathcal{I})$ is lower bounded by 0.0065). Now, we use Taylor expansions around 0 to simplify $\Phi(\alpha,\beta,\mathcal{I})$. In particular, for $0 \leq a \leq \frac{1}{2}$:

- $1 + \frac{a}{4} \leq \sqrt{1+a} \leq 1 + \frac{a}{2}$.
- $1 - a \leq \sqrt{1-a} \leq 1 - \frac{a}{2}$.

Also, recall that $\alpha, \beta \leq \frac{1}{2}$. This yields

$$\begin{aligned}
\Phi(\alpha,\beta,\mathcal{I}) &= \frac{0.1}{\sqrt{1+\alpha}} - \frac{0.1}{\sqrt{1+\beta}\sqrt{1-\alpha}} - M_2(\mathcal{I})\cdot\frac{(1-\sqrt{1-\alpha})}{\sqrt{1+\beta}} - M_1(\mathcal{I})\cdot(\sqrt{1+\alpha}-1) \\
&\geq \frac{0.1}{(1+\frac{\alpha}{2})} - \frac{0.1}{(1+\frac{\beta}{4})(1-\alpha)} - (M_1(\mathcal{I})+M_2(\mathcal{I}))\cdot(\alpha+\frac{\alpha}{2}) \\
&\geq \frac{0.1(1+\frac{\beta}{4})(1-\alpha) - 0.1(1+\frac{\alpha}{2}) - \frac{3}{2}(M_1(\mathcal{I})+M_2(\mathcal{I}))\alpha(1+\frac{\beta}{4})(1-\alpha)(1+\frac{\alpha}{2})}{(1+\frac{\beta}{4})(1-\alpha)(1+\frac{\alpha}{2})} \\
&\geq \frac{0.025\cdot\beta - \alpha\cdot(0.1625 + 2.109375(M_1(\mathcal{I})+M_2(\mathcal{I})))}{0.84375} \\
&\geq 0.029\cdot\beta - \alpha\cdot(0.2 + 2.5(M_1(\mathcal{I})+M_2(\mathcal{I})))\ .
\end{aligned}$$

Now, if $\alpha \leq \frac{0.0145\cdot\beta}{0.2+2.5(M_1(\mathcal{I})+M_2(\mathcal{I}))}$, we have $\Phi(\alpha,\beta,\mathcal{I}) \geq 0.0145\cdot\beta$. Now we can use the concentration bound and obtain

$$\begin{aligned}
\mathbb{P}(A\cap B^c) \leq\ &\mathbb{P}\left(\left\langle \frac{Z}{\sqrt{1+\alpha}}, \mathcal{Q}\left(\frac{Z}{\sqrt{1+\alpha}}\right)\right\rangle \leq \frac{1}{\sqrt{1+\beta}}\cdot\mathbb{E}\left[\left\|\mathcal{Q}\left(\frac{Z}{\sqrt{1-\alpha}}\right)\right\|_2^2\right]\right) \\
&\leq e^{-\frac{(0.0145\cdot\beta)^2\cdot d}{6}} = e^{-M(\mathcal{I})\cdot\alpha^2\cdot d}\ .
\end{aligned}$$

Here, we denoted $M(\mathcal{I}) = \frac{(0.2+2.5(M_1(\mathcal{I})+M_2(\mathcal{I})))^2}{6}$ and used that according to Lemma D.6, $\mathbb{E}\left[\left(\frac{z}{\sqrt{1+\alpha}}\cdot\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right)^2\right] \leq \mathbb{E}\left[(z\cdot\mathcal{Q}(z))^2\right] = 3$. Observe that $M(\mathcal{I}) \geq \frac{(0.2)^2}{6} \geq 0.0065$ and thus $\alpha = \beta\cdot\frac{0.005}{\sqrt{M(\mathcal{I})}}$ respects both $\frac{0.1}{\sqrt{1+\alpha}} - \frac{0.1}{\sqrt{1+\beta}\sqrt{1-\alpha}} > 0$ and $\alpha \leq \frac{0.0145\cdot\beta}{0.2+2.5(M_1(\mathcal{I})+M_2(\mathcal{I}))}$. □

Lemma D.4. *It holds that,*

- $\mathbb{E}\left[\left\langle \frac{Z}{\sqrt{1+\alpha}}, \mathcal{Q}\left(\frac{Z}{\sqrt{1+\alpha}}\right)\right\rangle\right] \geq \frac{1}{\sqrt{1+\alpha}} \cdot \mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] - (\sqrt{1+\alpha}-1)\cdot M_1(\mathcal{I})\cdot d\,.$
- $\mathbb{E}\left[\left\|\mathcal{Q}(\frac{Z}{\sqrt{1+\alpha}})\right\|_2^2\right] \geq \mathbb{E}\left[\|\mathcal{Q}(Z)\|_2^2\right] - (\sqrt{1+\alpha}-1)\cdot M_1(\mathcal{I})\cdot d\,.$

Proof. Due to the linearity of expectation, it is sufficient to show that

$$\mathbb{E}\left[\frac{z}{\sqrt{1+\alpha}}\cdot\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right] \geq \frac{1}{\sqrt{1+\alpha}}\cdot\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right] - (\sqrt{1+\alpha}-1)\cdot M_1(\mathcal{I})\,,$$

and

$$\mathbb{E}\left[\left(\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right)^2\right] \geq \mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right] - (\sqrt{1+\alpha}-1)\cdot M_1(\mathcal{I})\,.$$

Recall the set of intervals $\mathcal{I}$ and denote:

- $\mathcal{I}^- = \{I\in\mathcal{I} | I\subset\mathbb{R}^-\cup\{0\}\}\,.$
- $\mathcal{I}^+ = \{I\in\mathcal{I} | I\subset\mathbb{R}^+\cup\{0\}\}\,.$

First, using the law of total expectation,

$$\begin{aligned}\mathbb{E}\left[\frac{z}{\sqrt{1+\alpha}}\cdot\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right] &= \mathbb{E}\left[q_I\cdot\mathbb{E}\left[\frac{z}{\sqrt{1+\alpha}}\right] \;\middle|\; \frac{z}{\sqrt{1+\alpha}}\in I\right]\\ &\geq \frac{1}{\sqrt{1+\alpha}}\sum_{I\in\mathcal{I}} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I) = \frac{1}{\sqrt{1+\alpha}}\cdot\mathbb{E}\left[\left(\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right)^2\right]\,.\end{aligned}$$

Here, we used $\mathbb{E}[z|\frac{z}{\sqrt{1+\alpha}}\in I]\geq\mathbb{E}[z|z\in I]=q_I$.

Next, by definition, $\mathbb{E}\left[\left(\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right)^2\right] = \sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I) + \sum_{I\in\mathcal{I}^-} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I)$. Also, since the distribution of z and the set of intervals $\mathcal{I}$ are symmetric around 0, $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I) = \sum_{I\in\mathcal{I}^-} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I)$. For an interval $I\in\mathcal{I}^+$, denote $a_I=\min(I)$. Now, we can write $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1+\alpha}}\in I) \geq \sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}(z\in I) - \mathbb{P}\left(z\in\left[a_I, a_I\cdot\sqrt{1+\alpha}\right]\right)\right)$.

Next, we upper-bound $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}\left(z\in\left[a_I, a_I\cdot\sqrt{1+\alpha}\right]\right)\right)$. First, $q_I = \frac{\int_{x\in I} x\cdot e^{-\frac{x^2}{2}}dx}{\int_{x\in I} e^{-\frac{x^2}{2}}dx} \leq \frac{\int_a^\infty x\cdot e^{-\frac{x^2}{2}}dx}{\int_a^\infty e^{-\frac{x^2}{2}}dx} \leq a_I + \sqrt{\frac{2}{\pi}}$. Here, the last inequality follows from the fact that the derivative of the hazard rate function of the normal distribution is bounded above by 1 on the positive reals. Next, we obtain

$$q_I^2\cdot\mathbb{P}(z\in\left[a_I, a_I\cdot\sqrt{1+\alpha}\right]) \leq \left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot\left(\frac{1}{\sqrt{2\pi}}\int_{a_I}^{a_I\cdot\sqrt{1+\alpha}} e^{-\frac{x^2}{2}}dx\right) \leq \frac{\sqrt{1+\alpha}-1}{\sqrt{2\pi}}\cdot a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{2}}\,.$$

Thus, $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}\left(z\in\left[a_I, a_I\cdot\sqrt{1+\alpha}\right]\right)\right) \leq \frac{\sqrt{1+\alpha}-1}{\sqrt{2\pi}}\sum_{I\in\mathcal{I}^+}\left(a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{2}}\right)$.

Denoting $M_1(\mathcal{I}) = 2\cdot\frac{1}{\sqrt{2\pi}}\sum_{I\in\mathcal{I}^+}\left(a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{2}}\right)$ (we omitted $\sqrt{1+\alpha}$ from the denominator since it only decreases this term) concludes the proof.

We note that $M_1(\mathcal{I})$ is finite for any $\mathcal{I}$ with a strictly positive and fixed lower bound $\delta_{\mathcal{I}}$ on an interval size. To see this, denote $M = \max_{a\in\mathbb{R}^+}\left(\left(a\cdot\left(a+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a^2}{2}}\right)\right)$, ($M\approx 0.589505$) and a^* the corresponding argument ($a^*\approx 0.69479$).

Then, since for any $a > a^*$ this function is monotonically decreasing,

$$\sum_{I\in\mathcal{I}^+}\left(a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{2}}\right)\leq\frac{M+\delta_{\mathcal{I}}}{\delta_{\mathcal{I}}}+\sum_{n=0}^{\infty}(a^*+n\cdot\delta_{\mathcal{I}}+\sqrt{\frac{2}{\pi}})^3\cdot e^{\frac{-(a^*+n\delta_{\mathcal{I}})^2}{2}}\ ,$$

and the summation converges for any fixed $\delta_{\mathcal{I}} > 0$. □

Lemma D.5. $\mathbb{E}\left[\left\|\mathcal{Q}\left(\frac{Z}{\sqrt{1-\alpha}}\right)\right\|_2^2\right]\leq\frac{1}{\sqrt{1-\alpha}}\mathbb{E}\left[\left\|\mathcal{Q}\left(Z\right)\right\|_2^2\right]+(1-\sqrt{1-\alpha})\cdot M_2(\mathcal{I})\cdot d\,.$

Proof. Recall the sets of intervals $\mathcal{I}, \mathcal{I}^-, \mathcal{I}^+$ and for an interval $I \in \mathcal{I}^+$, denote $a_I = \min(I)$.

By definition, $\mathbb{E}\left[\left(\mathcal{Q}(\frac{z}{\sqrt{1-\alpha}})\right)^2\right] = \sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1-\alpha}}\in I)+\sum_{I\in\mathcal{I}^-} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1-\alpha}}\in I)$. Also, since the distribution of z and the set of intervals $\mathcal{I}$ are symmetric around 0, $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1-\alpha}}\in I) = \sum_{I\in\mathcal{I}^-} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1-\alpha}}\in I)$. Now, we can write $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\mathbb{P}(\frac{z}{\sqrt{1-\alpha}}\in I)\leq\sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}(z\in I)+\mathbb{P}\left(z\in\left[a_I\cdot\sqrt{1-\alpha},a_I\right]\right)\right)$.

Next, we upper-bound $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}\left(z\in\left[a_I\cdot\sqrt{1-\alpha},a_I\right]\right)\right)$. First, we again use $q_I\leq a_I+\sqrt{\frac{2}{\pi}}$ and obtain

$$q_I^2\cdot\mathbb{P}(z\in\left[a_I\cdot\sqrt{1-\alpha},a_I\right])\leq\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot\left(\frac{1}{\sqrt{2\pi}}\int_{a_I\cdot\sqrt{1-\alpha}}^{a_I}e^{-\frac{x^2}{2}}dx\right)\leq\frac{1-\sqrt{1-\alpha}}{\sqrt{2\pi}}\cdot a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2\cdot(1-\alpha)}{2}}\ .$$

Thus, $\sum_{I\in\mathcal{I}^+} q_I^2\cdot\left(\mathbb{P}\left(z\in\left[a_I\cdot\sqrt{1-\alpha},a_I\right]\right)\right)\leq\frac{1-\sqrt{1-\alpha}}{\sqrt{2\pi}}\sum_{I\in\mathcal{I}^+}\left(a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{4}}\right)$, where we used $\alpha < \frac{1}{2}$.

Denoting $M_2(\mathcal{I}) = 2\cdot\frac{1}{\sqrt{2\pi}}\sum_{I\in\mathcal{I}^+}\left(a_I\cdot\left(a_I+\sqrt{\frac{2}{\pi}}\right)^2\cdot e^{-\frac{a_I^2}{4}}\right)$ concludes the proof.

Similarly to the argument in Lemma D.4, for any $\mathcal{I}$ with a fixed lower bound on an interval size, $M_2(\mathcal{I})$ is finite.

□

Lemma D.6. *For $z\sim\mathcal{N}(0,1)$ it holds that* $\mathbb{E}\left[\left(\frac{z}{\sqrt{1+\alpha}}\cdot\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)\right)^2\right]\leq\mathbb{E}\left[\left(z\cdot\mathcal{Q}\left(z\right)\right)^2\right]\leq 3.$

Proof. It holds that $\left(z^2-(\mathcal{Q}(z))^2\right)^2 = z^4-2\cdot z^2\cdot(\mathcal{Q}(z))^2+(\mathcal{Q}(z))^4\geq 0$.

Also, $\mathbb{E}\left[z^2\cdot(\mathcal{Q}(z))^2\right] = \mathbb{E}\left[(\mathcal{Q}(z))^2\cdot\mathbb{E}\left[z^2\mid Q(z)\right]\right]$ and by Jensen's inequality and property (v) $\mathbb{E}\left[z^2\mid Q(z)\right]\geq\left(\mathbb{E}\left[z\mid Q(z)\right]\right)^2 = (Q(z))^2$. Thus, $\mathbb{E}\left[z^2\cdot(\mathcal{Q}(z))^2\right]\geq\mathbb{E}\left[(\mathcal{Q}(z))^4\right]$. This yields

$$0<\mathbb{E}\left[\left(z^2-(\mathcal{Q}(z))^2\right)^2\right]\leq\mathbb{E}\left[z^4\right]-\mathbb{E}\left[(\mathcal{Q}(z))^4\right]\ .$$

Next,

$$\mathbb{E}\left[(z\cdot\mathcal{Q}(z))^2\right]\leq\frac{1}{2}\cdot\mathbb{E}\left[(z)^4\right]+\frac{1}{2}\cdot\mathbb{E}\left[(\mathcal{Q}(z))^4\right]\leq\mathbb{E}\left[(z)^4\right]=3\,.$$

Observing that $z > \frac{z}{\sqrt{1+\alpha}}$ and $\mathcal{Q}(z)\geq\mathcal{Q}\left(\frac{z}{\sqrt{1+\alpha}}\right)$ for any $\alpha > 0$ concludes the proof. □

Lemma D.7. $\mathbb{E}\left[\left\|\mathcal{Q}(Z)\right\|_2^2\right]\geq 0.1\cdot d.$

Proof. Due to the linearity of expectation, it is sufficient to show that $\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right]\geq\left(\mathbb{E}\left[\mathcal{Q}(z)\right]\right)^2\geq 0.1$.

Also, $\mathbb{E}\left[\mathcal{Q}(z)\right] = \sum_{I\in\mathcal{I}} q_I\cdot P(z\in I) = \sum_{I\in\mathcal{I}}\frac{\int_I t\cdot e^{-\frac{t^2}{2}}dt}{\int_I e^{-\frac{t^2}{2}}dt}\cdot\int_I e^{-\frac{t^2}{2}}dt = \sum_{I\in\mathcal{I}}\int_I t\cdot e^{-\frac{t^2}{2}}dt$.

Now,we divide into two cases.

Case 1: $[-a,a] \in \mathcal{I}$. In this case, $a < 1$ and thus $\mathbb{E}\left[(\mathcal{Q}(z))\right] \geq 2 \cdot \frac{1}{\sqrt{2\pi}} \int_a^\infty e^{\frac{-t^2}{2}} dt > 0.317$.

Case 2: $[-a,a] \notin \mathcal{I}$ for any a. Thus, $\mathbb{E}\left[(\mathcal{Q}(z))\right] = 2 \cdot \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{\frac{-t^2}{2}} dt = \sqrt{\frac{2}{\pi}}$.

For both cases, it holds that $\mathbb{E}\left[\left(\mathcal{Q}(z)\right)^2\right] \geq (0.317)^2 \geq 0.1$. □

E. Sub-bit Compression Proofs

Lemma E.1. *Consider two unbiased compression techniques $\mathcal{A}$ and $\mathcal{B}$ (i.e., $\forall x : \mathbb{E}[\mathcal{A}(x)] = \mathbb{E}[\mathcal{B}(x)] = x$) with independent randomness. Then,*

1. $\forall x : \frac{\mathbb{E}[\|x-\mathcal{A}(x)\|_2^2]}{\|x\|_2^2} \leq A$ *and* $\frac{\mathbb{E}[\|x-\mathcal{B}(x)\|_2^2]}{\|x\|_2^2} \leq B \implies \forall x : \frac{\mathbb{E}[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2]}{\|x\|_2^2} \leq A + AB + B$.
2. $\forall x : \frac{\mathbb{E}[\|x-\mathcal{A}(x)\|_2^2]}{\|x\|_2^2} \geq A$ *and* $\frac{\mathbb{E}[\|x-\mathcal{B}(x)\|_2^2]}{\|x\|_2^2} \geq B \implies \forall x : \frac{\mathbb{E}[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2]}{\|x\|_2^2} \geq A + AB + B$.

Proof. For ease of exposition we denote $y = \mathcal{A}(x)$ and $z = \mathcal{B}(\mathcal{A}(x))$. Using unbiasedness we obtain:

$$\begin{aligned}\mathbb{E}\left[\|y-z\|_2^2 \mid y\right] &= \mathbb{E}\left[\|z\|_2^2 \mid y\right] - 2 \cdot \mathbb{E}\left[\langle z, y\rangle \mid y\right] + \mathbb{E}\left[\|y\|_2^2\right] \\ &= \mathbb{E}\left[\|z\|_2^2 \mid y\right] - \|y\|_2^2 \implies \mathbb{E}\left[\|y-z\|_2^2\right] = \mathbb{E}\left[\|z\|_2^2\right] - \mathbb{E}\left[\|y\|_2^2\right] .\end{aligned}$$

Similarly, we have that $\mathbb{E}\left[\|x-y\|_2^2\right] = \mathbb{E}\left[\|y\|_2^2\right] - \|x\|_2^2$. Thus, it holds that

$$\mathbb{E}\left[\|x-y\|_2^2\right] + \mathbb{E}\left[\|y-z\|_2^2\right] = \mathbb{E}\left[\|z\|_2^2\right] - \|x\|_2^2 .$$

Also,

$$\mathbb{E}\left[\|x-z\|_2^2\right] = \mathbb{E}\left[\|z\|_2^2\right] - 2 \cdot \mathbb{E}\left[\langle x, z\rangle\right] + \|x\|_2^2 .$$

And since,

$$\mathbb{E}\left[\langle x, z\rangle\right] = \mathbb{E}\left[\mathbb{E}\left[\langle x, z\rangle\right] \mid y\right] = \mathbb{E}\left[\langle x, y\rangle\right] = \|x\|_2^2 ,$$

we obtain,

$$\mathbb{E}\left[\|x-z\|_2^2\right] = \mathbb{E}\left[\|z\|_2^2\right] - \|x\|_2^2 = \mathbb{E}\left[\|x-y\|_2^2\right] + \mathbb{E}\left[\|y-z\|_2^2\right] .$$

Proof of part 1: we can write:

1. $\frac{\mathbb{E}\left[\|x-\mathcal{A}(x)\|_2^2\right]}{\|x\|_2^2} = \frac{\mathbb{E}\left[\|x-y\|_2^2\right]}{\|x\|_2^2} \leq A$.
2. $\mathbb{E}\left[\frac{\|\mathcal{A}(x)-\mathcal{B}(\mathcal{A}(x))\|_2^2}{\|\mathcal{A}(x)\|_2^2} \middle| \mathcal{A}(x)\right] = \mathbb{E}\left[\frac{\|y-z\|_2^2}{\|y\|_2^2} \middle| y\right] \leq B \implies \mathbb{E}\left[\|y-z\|_2^2\right] \leq B \cdot \mathbb{E}\left[\|y\|_2^2\right]$.

Using unbiasedness we obtain $\mathbb{E}\left[\|x-y\|_2^2\right] = \mathbb{E}\left[\|y\|_2^2\right] - \|x\|_2^2 \leq A \cdot \|x\|_2^2$ and $\|x\|_2^2 \geq \frac{\mathbb{E}\left[\|y\|_2^2\right]}{A+1}$. Thus,

$$\begin{aligned}\frac{\mathbb{E}\left[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2\right]}{\|x\|_2^2} &= \frac{\mathbb{E}\left[\|x-z\|_2^2\right]}{\|x\|_2^2} = \frac{\mathbb{E}\left[\|x-y\|_2^2\right]}{\|x\|_2^2} + \frac{\mathbb{E}\left[\|y-z\|_2^2\right]}{\|x\|_2^2} \\ &\leq A + \frac{A+1}{\mathbb{E}\left[\|y\|_2^2\right]} \cdot \mathbb{E}\left[\|y-z\|_2^2\right] \leq A + \frac{A+1}{\mathbb{E}\left[\|y\|_2^2\right]} \cdot B \cdot \mathbb{E}\left[\|y\|_2^2\right] = A + AB + B .\end{aligned}$$

This concludes the proof of part 1.

Proof of part 2: The proof is identical to the first part by flipping all the inequalities. We can write:

1. $\frac{\mathbb{E}\left[\|x-\mathcal{A}(x)\|_2^2\right]}{\|x\|_2^2} = \frac{\mathbb{E}\left[\|x-y\|_2^2\right]}{\|x\|_2^2} \geq A$.

2. $\mathbb{E}\left[\frac{\|\mathcal{A}(x)-\mathcal{B}(\mathcal{A}(x))\|_2^2}{\|\mathcal{A}(x)\|_2^2}\middle|\mathcal{A}(x)\right] = \mathbb{E}\left[\frac{\|y-z\|_2^2}{\|y\|_2^2}\middle|y\right] \geq B \implies \mathbb{E}\left[\|y-z\|_2^2\right] \geq B\cdot\mathbb{E}\left[\|y\|_2^2\right]$.

Using unbiasedness we obtain $\mathbb{E}\left[\|x-y\|_2^2\right] = \mathbb{E}\left[\|y\|_2^2\right] - \|x\|_2^2 \geq A\cdot\|x\|_2^2$ and $\|x\|_2^2 \geq \frac{\mathbb{E}\left[\|y\|_2^2\right]}{A+1}$. Thus,

$$\begin{aligned}\frac{\mathbb{E}\left[\|x-\mathcal{B}(\mathcal{A}(x))\|_2^2\right]}{\|x\|_2^2} &= \mathbb{E}\left[\frac{\|x-z\|_2^2}{\|x\|_2^2}\right] = \mathbb{E}\left[\frac{\|x-y\|_2^2}{\|x\|_2^2}\right] + \mathbb{E}\left[\frac{\|y-z\|_2^2}{\|x\|_2^2}\right] \\ &\geq A + \frac{A+1}{\mathbb{E}\left[\|y\|_2^2\right]}\cdot\mathbb{E}\left[\|y-z\|_2^2\right] \geq A + \frac{A+1}{\mathbb{E}\left[\|y\|_2^2\right]}\cdot B\cdot\mathbb{E}\left[\|y\|_2^2\right] = A + AB + B\,.\end{aligned}$$

This concludes the proof of part 2. □

F. Lossy Networks Proofs

Lemma F.1. *Let $x \in \mathbb{R}^d$ and let $m_{ds} \in \{0,1\}^d$ be a* deterministic *mask. Denote $p = \frac{\|m_{ds}\|_1}{d}$ and let $\mathcal{R}_{ds}(x) = \frac{1}{p}\cdot m_{ds}\circ\mathcal{R}(x)$. Then, using EDEN with $\mathcal{R}_{ds}(x)$ instead of $\mathcal{R}(x)$ results in:*

1. $\mathbb{E}\left[\widehat{x}\right] = x$.
2. *vNMSE* $\leq \frac{1}{p\cdot\mathbb{E}\left[(\mathcal{Q}(z))^2\right]} - 1 + O\left(\sqrt{\frac{\log d}{d\cdot p^2}}\right)$.

Proof. The receiver uses $\frac{1}{\|m_{ds}\|_1}\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m_{ds}$ instead of $\mathcal{Q}(\eta_x\cdot R\cdot x)$.

Proof of 1: We revisit Equation (1) in Theorem 2.1 and obtain

$$\widehat{x} = R^{-1}_{x\to x'}\cdot\|x\|_2\cdot\left(\frac{\left\langle C_0, \frac{1}{\|m_{ds}\|_1}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}, \ldots, \frac{\left\langle C_{d-1}, \frac{1}{\|m_{ds}\|_1}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}\right)^T .$$

The proof continues similarly to that of Theorem 2.1.

We again consider algorithm EDEN' from the proof of Theorem 2.1. It holds that:

$$\widehat{x} + \widehat{x}' = 2\cdot R^{-1}_{x\to x'}\cdot\|x\|_2\cdot\left(\frac{\left\langle C_0, \frac{1}{\|m_{ds}\|_1}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}, 0, \ldots, 0\right)^T .$$

In the numerator, we have a sum of random variables whose all subsets of size $\|m_{ds}\|_1$ follows the same distribution. This means that for any two deterministic masks m_{ds} and m'_{ds} such that $\|m_{ds}\|_1 = \left\|m'_{ds}\right\|_1$, we have that

$$\mathbb{E}\left[\frac{\left\langle C_0, \frac{1}{\|m_{ds}\|_1}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}\right] = \mathbb{E}\left[\frac{\left\langle C_0, \frac{1}{|m'_{ds}|}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m'_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}\right] .$$

Also, due to linearity of expectation,

$$\sum_{\substack{m'_{ds}:\\ \left\|m'_{ds}\right\|_1=\|m_{ds}\|_1}} \mathbb{E}\left[\frac{\left\langle C_0, \frac{1}{p}\cdot\mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m'_{ds}\right\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}\right] =$$

$$\frac{1}{p}\cdot\mathbb{E}\left[\sum_{\substack{m'_{ds}:\\ \left\|m'_{ds}\right\|_1=\|m_{ds}\|_1}} \frac{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\circ m'_{ds}\rangle}{\langle C_0, \mathcal{Q}\left(\eta_x\cdot\|x\|_2\cdot C_0\right)\rangle}\right] = \frac{1}{p}\cdot p\cdot M_{ds}\ ,$$

where $M_{ds}=\binom{d}{\|m_{ds}\|_1}$ is the number of different masks with the same number of 1's. This means that $\mathbb{E}\left[\widehat{x}+\widehat{x}'\right]=2\cdot x$. Recall that $\widehat{x}$ and $\widehat{x}'$ follow the same distribution. This concludes the proof.

Intuitively, the reason why unbiasedness is preserved with a deterministic mask is that all coordinates of the rotated and quantized vector follow the same distribution, and thus after the inverse rotation and scaling, the distribution of the reconstructed vector depends only on the number of zeros in the mask but not on their indices.

Proof of 2: Due to unbiasedness (i.e., Part 1), it holds that

$$vNMSE\cdot\|x\|_2^2=\mathbb{E}\left[\|x-\widehat{x}\|_2^2\right]=\|\widehat{x}\|_2^2-\|x\|_2^2\ .$$

Now, we examine $\|\widehat{x}\|_2^2=\|\mathcal{R}(\widehat{x})\|_2^2=\frac{1}{\|m_{ds}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m_{ds}\|_2^2$.

Similarly to Case 1, we have a sum of random variables whose all subsets of size $\|m_{ds}\|_1$ follows the same distribution. This means that for any two deterministic masks m_{ds} and m'_{ds} such that $\|m_{ds}\|_1=\left\|m'_{ds}\right\|_1$, we obtain $\mathbb{E}\left[\frac{1}{\|m_{ds}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m_{ds}\|_2^2\right]=\mathbb{E}\left[\frac{1}{\|m'_{ds}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m'_{ds}\|_2^2\right]$.

Let m_{rs} be a random mask such that $\|m_{rs}\|_1=\|m_{ds}\|_1$. Then,

$$\mathbb{E}\left[\frac{1}{\|m_{rs}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m_{rs}\|_2^2\right]$$

$$=\frac{1}{M_{ds}}\cdot\sum_{\substack{m'_{ds}:\\ \left\|m'_{ds}\right\|_1=\|m_{ds}\|_1}}\mathbb{E}\left[\frac{1}{\|m'_{ds}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m'_{ds}\|_2^2\ \middle|\ m_{rs}=m'_{ds}\right]$$

$$=\mathbb{E}\left[\frac{1}{\|m_{ds}\|_1^2}\cdot\|S\cdot\mathcal{Q}(\eta_x\cdot R\cdot x)\circ m_{ds}\|_2^2\right]\ .$$

Using the above with Lemma 4.1 concludes the proof. □

G. Entropy Compressed EDEN

G.1. Evaluation

As discussed in §4.3, when using Entropy Encoding (EE), EDEN uses the quantization interval set $\mathcal{I}_{\Delta_b}=\left\{\left[\Delta_b\cdot\left(n-\frac{1}{2}\right),\Delta_b\cdot\left(n+\frac{1}{2}\right)\right]\ \middle|\ n\in\mathbb{Z}\right\}$, for the smallest Δ_b such that $H_{\mathcal{I}_{\Delta_b}}\leq b$.

Figure 4 shows the $vNMSE$ of:

- EDEN with the super-bit compression (§4.1) for $b\geq 1$ and the sub-bit compression (§4.1) for $b\in(0,1]$.

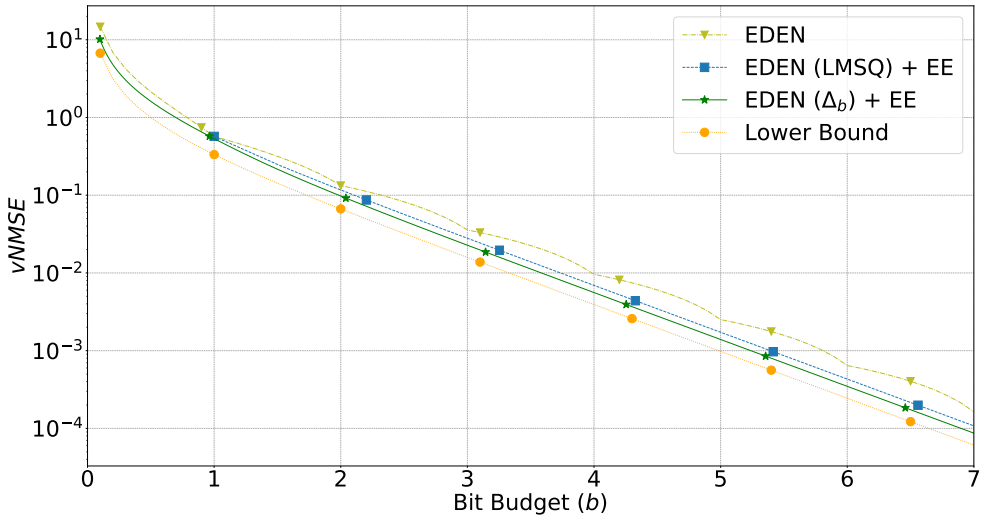


Figure 4. EDEN's $vNMSE$ with and without Entropy Encoding for the different quantization schemes. (displayed for $b \in \{0.1 \cdot i \mid i \in \{1, \ldots, 80\}\}$)

- EDEN, with EE applied on the vector resulting from the Lloyd-Max Scalar Quantizer.
- EDEN, with EE applied on the vector resulting from the $\mathcal{Q}_{\mathcal{I}_{\Delta_b}}$ quantization.
- A lower bound on EDEN, for any $\mathcal{I}$, derived from the Rate-distortion theory (Cover, 1999) over the normal distribution.

As shown, even without EE, EDEN requires less than one bit more than the lower bound. Indeed, using more quantization values and compressing the resulting vector with EE reduces the error. Switching to our tailored quantization $\mathcal{I}_{\Delta_b}$ reduces the error further. Also, $\mathcal{I}_{\Delta_b}$ requires at most 0.25 bits more than the rate-distortion lower bound.

Next, we compare EDEN, with and without EE, to two previously suggested variable-length encoding DME schemes. Specifically, we show the $vNMSE$ of QSGD with Elias Omega encoding (Alistarh et al., 2017), and optimized stocastic quantization with Huffman encoding (Suresh et al., 2017). Without EE, EDEN uses the Lloyd Max Scalar Quantizer $\mathcal{I}_b$ for $b \geq 1$ (§3) while for $b \in (0, 1]$ it uses the sub-bit compression (§4.1). When EE is applied, EDEN uses $\mathcal{I}_{\Delta_b}$ (see §4.3).

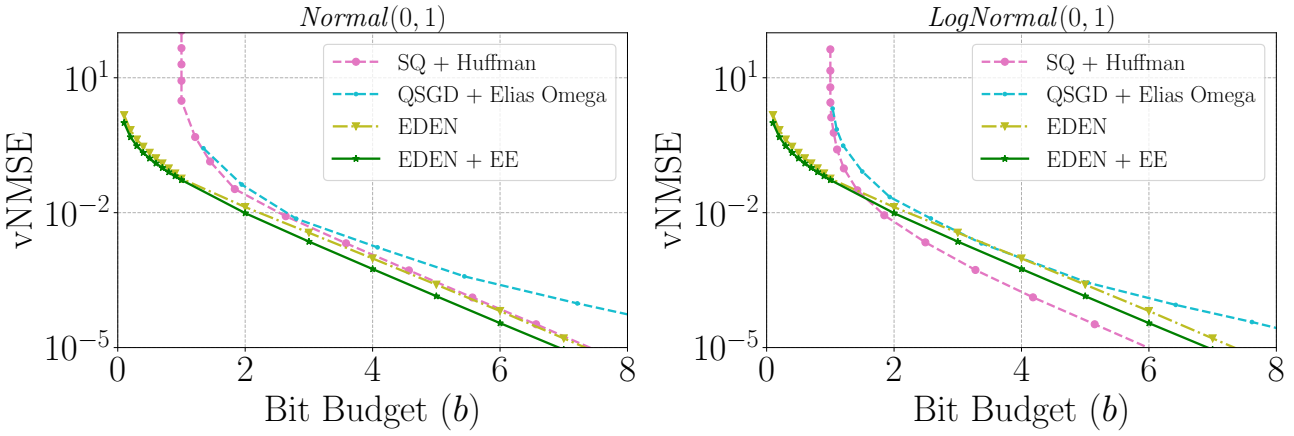


Figure 5. EDEN's $vNMSE$ with and without Entropy Encoding compared to other Variable-Length schemes. (displayed for $b \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8\}$)

As depicted in Figure 5, EDEN, which has a GPU-friendly implementation (i.e., without EE), improves the worst-case $vNMSE$. Also, EDEN is robust and has the same error for all distributions (which is aligned with the theoretical results) and can be further optimized using EE, at the cost of additional computation. The stochastic quantization with Huffman encoding has a lower $vNMSE$ for some input distributions, e.g., the LogNormal, but its error is input dependent.

G.2. Variable-Length Encoding Representation Length

In practice, the frequency of a quantization value $I \in \mathcal{I}$ may not be exactly $p_I \cdot d$. Nonetheless, using arithmetic-coding, we can get an encoding that uses $(H_{\mathcal{I}} + \epsilon)(1 + o(1))$ bits per coordinate on average. A proof sketch, which assumes that the coordinates are independent (in practice, they are weakly dependent for a sufficiently large dimension d) follows. We defer the formal proof to future work. As indicated by Mitzenmacher & Upfal (2017) (see Chapter 10), from the fact that each coordinate j of a vector $y \in \mathbb{R}^d$, decreases the length of the encoded interval by a factor of $\mathcal{L}_{\mathcal{Q}(y[j])}$, where $\mathcal{L}_{\mathcal{Q}(y[j])}$ is the random variable that represents the quantization value of the interval of $y[j]$. Therefore, the length of the interval of the vector is $\mathcal{L}_y \triangleq \prod_{j=0}^{d-1} \mathcal{L}_{\mathcal{Q}(y[j])}$ which means that the representation of y requires $\left\lceil \log_2\left(\frac{1}{\mathcal{L}_y}\right) + 1 \right\rceil \leq 2 + \sum_{j=0}^{d-1} \log_2 \frac{1}{\mathcal{L}_{\mathcal{Q}(y[j])}}$. A standard application of the Chernoff bound suffices to complete the argument.

H. Structured Rotation

To improve computational efficiency, the randomized Hadamard transform is used in different domains to replace computationally extensive matrix multiplications. This is now common. For example: in Rader (1969); Thomas (2013); Herendi et al. (1997), it is used to develop a computationally cheap methods to generate independent normally distributed variables from simpler (e.g., uniform) distributions; in Yu et al. (2016), it is used in the context of Gaussian kernel approximation, replacing the random Gaussian matrix; in Choromanski et al. (2018), it is used for gradient estimation in derivative-free optimization reinforcement learning; in Choromanski et al. (2017), it is used for efficient computation of embeddings.

In our context, it is used by recent DME techniques (Hadamard + SQ (Suresh et al., 2017; Konečnỳ & Richtárik, 2018), Kashin + SQ (Lyubarskii & Vershynin, 2010; Caldas et al., 2018b; Safaryan et al., 2020) and DRIVE (Vargaftik et al., 2021)). While (Vargaftik et al., 2021) pointed out an adversarial example for a single transform, we are not aware of adversarial inputs for more than a single transform. For some use cases, previous works (e.g., Yu et al. (2016); Andoni et al. (2015)) suggest using 2-3 transforms to avert the dependency on the input. In the case of neural networks, as was indicated by (Vargaftik et al., 2021), when the input vector is high dimensional and admits finite moments, a single transform performs sufficiently similar to a uniform rotation. As recently reported by several works, this is indeed the case in common DNN workloads where gradients and network parameters follow, e.g., the lognormal (Chmiel et al., 2021) or normal (Banner et al., 2019; Ye et al., 2020) distributions.

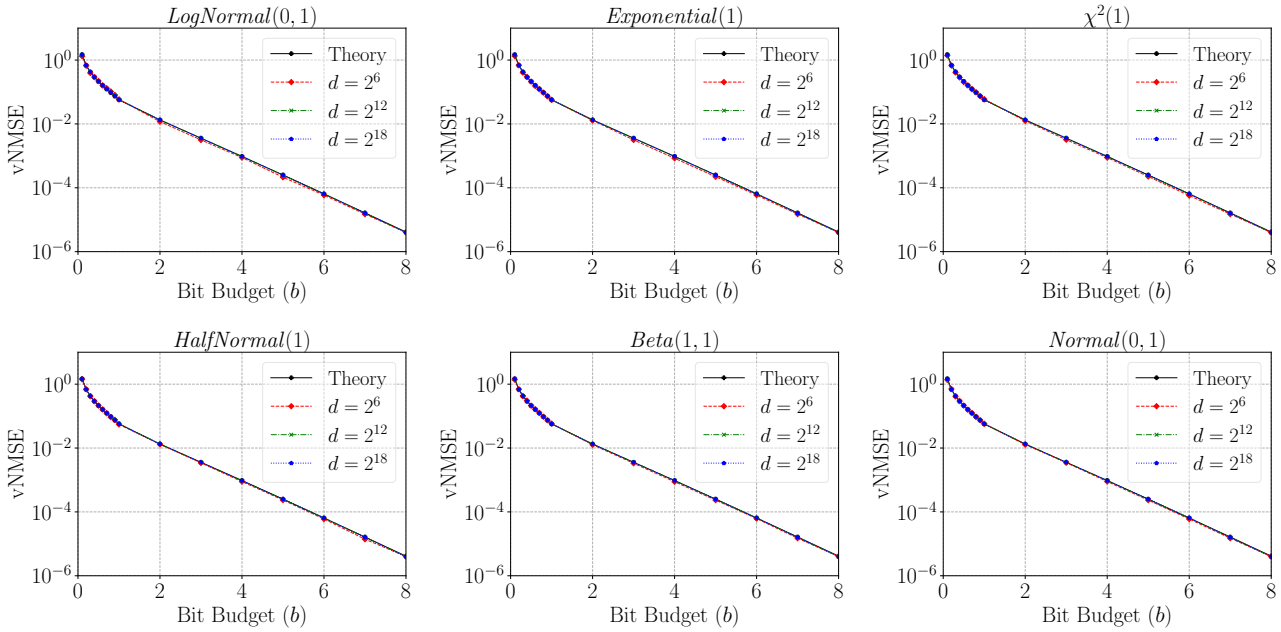


Figure 6. Using a single Hadamard transform (instead of a uniform random rotation) results in a *vNMSE* that coinsides with the theoretical bound of Corollary 2.4 for these tested distributions (displayed for $b \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8\}$).

In Figure 6, we show how EDEN's *vNMSE* with a single transform aligns with the theoretical bound (Corollary 2.4) for all tested input distributions (LogNormal, Normal, Exponential, χ^2, Half-Normal, Beta, and others not shown) and vector

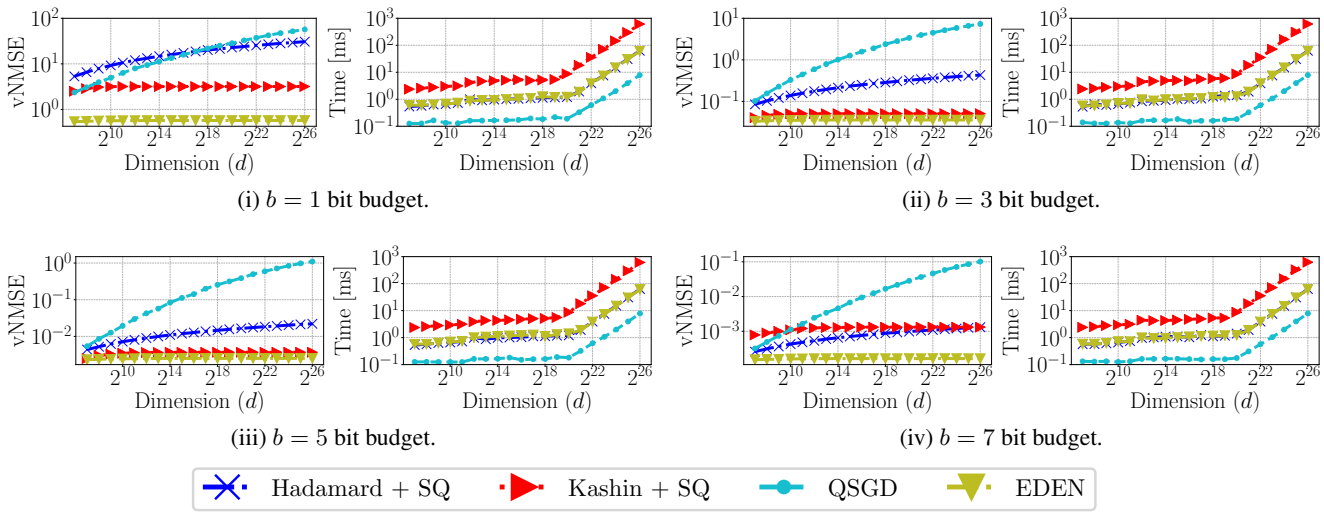


Figure 7. The *vNMSE* and compression time as a function of the dimension d for LogNormal(0,1) distribution.

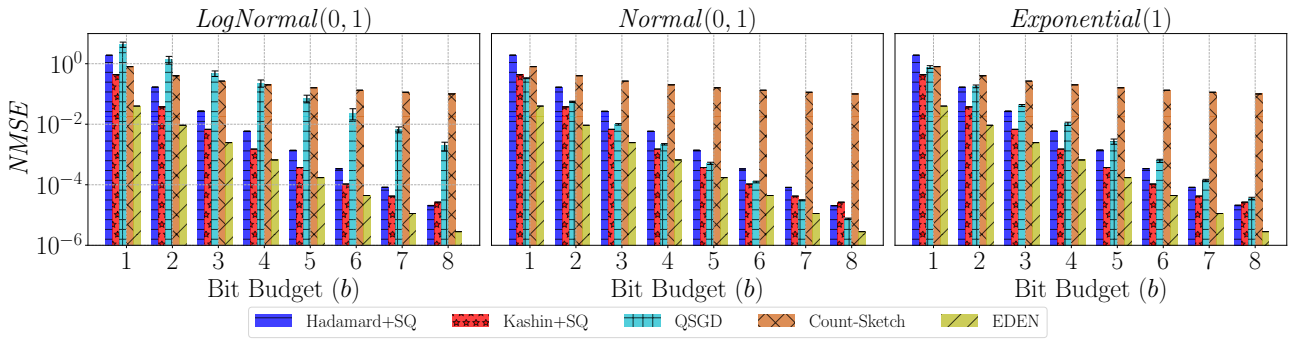


Figure 8. NMSE evaluation with $c = 10$ clients and vector dimension of $d = 11,511,784$ (ResNet18). Sweeping over a bit budget of 1-8 bits per coordinate.

dimensions. As can be seen, even a single randomized Hadamard transform aligns extremely closely with the theoretical results for a uniform random rotation, so that one cannot visually distinguish the results. For even smaller dimensions (e.g., $d = 16$), we observed a slightly higher error for the implementation than the theoretical asymptotic bound, as the $O\left(\sqrt{\frac{\log d}{d}}\right)$ term of Theorem 2.3 is not negligible. However, since our interest is in neural network gradients of large dimension, this is not important for this application.

I. Additional Simulation Details and Results

I.1. *vNMSE*, *NMSE*, and encoding speed

Here, we run the experiment of Figure 2 with different bit budgets b and depict the results in Figure 7. As shown, EDEN has the lowest *vNMSE* for all dimension and bit budget combinations, and is also significantly faster than the second most accurate solution, Kashin + SQ. As in Figure 2, the *vNMSE* of EDEN and Kashin + SQ is bounded independently of the dimension while Hadamard + SQ's and QSGD's error increases with d.

Our encoding speed measurements are performed using NVIDIA GeForce RTX 3090 GPU. The machine has Intel Core i9-10980XE CPU (18 cores, 3.00 GHz, and 24.75 MB cache) and 128 GB RAM. We use Ubuntu 20.04.2 LTS operating system, CUDA release 11.1 (V11.1.105), and PyTorch version 1.10.1.

Next, we conduct experiments with a vector of size $d = 11511784$ (the number of parameters in a ResNet18 architecture) in

a DME setting where $c = 10$ senders send their vectors to a receiver for averaging. We test the different DME techniques over different distributions. Each data point is repeated 100 times and we report the mean and standard deviation. In addition, here we also compare against *Count-Sketch* (Charikar et al., 2002), which is the main building block for some recent distributed and federated compression schemes (e.g., Ivkin et al. (2019)).

We sweep over a bit budget of 1-8 bits per coordinate. The results, shown in Figure 8, indicate that: (1) as dictated by theory, EDEN's $NMSE$ is not sensitive to the specific distribution; (2) EDEN significantly improves over all techniques for all bit budgets (notice the logarithmic $NMSE$ scale). Moreover, there is no consistent runner-up in this DME experiment since different competitors are more accurate for different distributions and bit budgets.

Several additional trends are evident: (1) Count-Sketch is competitive for a low communication budget but not for a higher ones. This is expected since its error guarantee decreases polynomially in b (which linearly affects its number of counters). In contrast, for other techniques, the error decreases exponentially (i.e., doubling the number of quantization values for each added bit); (2) For most bit budgets, the most competitive DME algorithm to ours in terms of $NMSE$ is Kashin + SQ. However, its accuracy is less prominent for high bit budgets constraints where its coefficient representation dominates the error (which can be improved with more iterations, at the cost of additional computation); (3) QSGD employs uniform quantization values and therefore performs better for light tailed distributions. For example, for the heavy-tailed Log-Normal distribution, which is common in machine learning workloads and, in particular, neural network gradients (e.g., Chmiel et al. (2021)), its performance degrades notably.

Finally, recall that for $b = 1$ and no coordinate losses, EDEN and DRIVE (Vargaftik et al., 2021) admit the same performance.

I.2. EMNIST and Shakespeare experiments details

The EMNIST and Shakespeare federated learning experiments, presented in §5.3, follow precisely the setup described in Reddi et al. (2021) for the Adam server optimizer case, which we restate in Table 1.

The client partitioning of these datasets was designed to naturally simulate a realistic heterogeneous federated learning setting. Specifically, federated EMNIST (Caldas et al., 2018a) includes 749,068 handwritten characters partitioned among their 3400 writers (i.e., this is the total number of clients), and federated Shakespeare (McMahan et al., 2017) consists of 18,424 lines of text from Shakespeare plays partitioned among the respective 715 speakers (i.e., clients). For EMNIST, a CNN with two convolutional layers is used (with $\approx 1.2M$ parameters), and for Shakespeare, a standard LSTM recurrent model (Hochreiter & Schmidhuber, 1997) (with $\approx 820K$ parameters).

Task	Clients per round	Rounds	Batch size	Client lr	Server lr	Adam's ϵ
EMNIST	10	1500	20	$10^{-1.5}$	$10^{-2.5}$	10^{-4}
Shakespeare	10	1200	4	1	10^{-2}	10^{-3}

Table 1. Hyperparameters for the EMNIST and Shakespeare experiments.

Additionally, for completeness, we include Figure 9, which is a zoomed-out version of Figure 3.

I.3. Distributed Logistic Regression

While the federated learning benchmarks demonstrate the applicability of our method, they are also often noisy and generally converge with low bit budgets. In contrast, logistic regression allows us to show more fine-grained differences between the methods for super-bit budgets. We perform an experiment similar to that of Malinovskiy et al. (2020) (§4). In particular, we use UCI's Census Income binary prediction task (Kohavi et al., 1996) with the discretization described in (Platt, 1998)[7] and divide the data equally among 20 clients. With each compression strategy, we run distributed gradient descent where, in each round, clients report the full gradient over their share, and the server uses the average of these gradients to take a step towards the optimum. Given that this is a convex setup, we expect the lower variance unbiased gradient estimates to consistently move towards the optimum. We run each compression scheme for 100 rounds with different bit budgets and report the Euclidean distance of the model parameters from the model received without any compression. We present the results in Figure 10. As hypothesized, EDEN is closer to the optimum than other methods in all the bit budgets we measured.

[7] Available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a`.

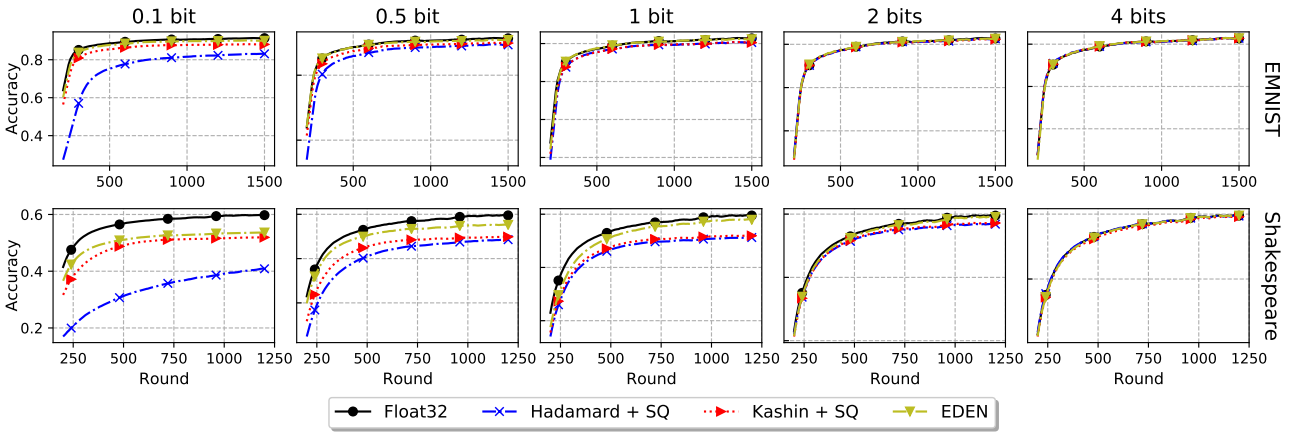


Figure 9. A fully-zoomed out of Figure 3. *FedAvg* over the EMNIST and Shakespeare tasks (columns) at various bit budgets (rows). We report training accuracy per round with a smoothing rolling mean window of 200 rounds. Sparsification is done using a random mask as described in §4.1.

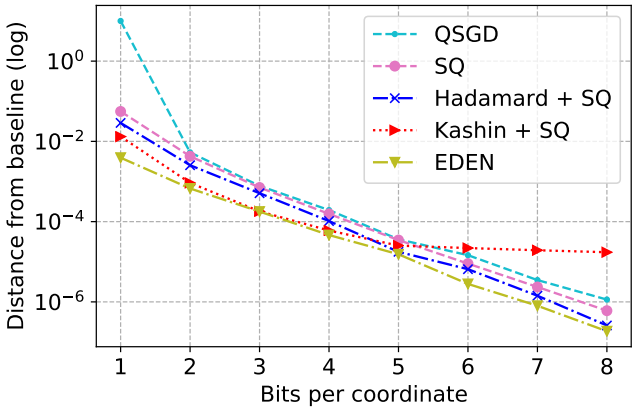


Figure 10. Logistic regression over UCI's Census Income task. For every compression scheme and bit budget, we run 100 rounds of distributed gradient descent and report the Euclidean distance from the baseline model (i.e., Float32).

I.4. Loss vs. Sub-bit

We now measure the $vNMSE$ under network loss and for sub-bit compression. Intuitively, while in both not all coordinates are received by the receiver, our sub-bit compression selects a uniform random mask of coordinates that are encoded while packet loss may be arbitrary (e.g., in blocks). A different view point is that sub-bit compression means sparsifying the vector *prior* to the random rotation while packet drops means loss of coordinates in the rotated vector. As shown in Figure 11, the empirical $vNMSE$ of the two is identical (when the same fraction of coordinates is received) and follows the theory of Corollary 4.2 and Lemma 4.3. Indeed, as shown by Lemma 4.1, the $vNMSE$ equals $A + AB + B$ (where A is EDEN's $vNMSE$ and B is the sparsification's) regardless of the orders in which $\mathcal{A}$ and $\mathcal{B}$ are applied.

I.5. Distributed Power Iteration

For some machine learning tasks (e.g., Principal Component Analysis), power iteration, which approximates the dominant eigenvalues and eigenvectors of a matrix, is used as a sub-routine. We perform an experiment with 10 clients and a server that jointly compute the top eigenvector in a matrix that is distributed among the clients. In particular, the training occurs in rounds where in each training round we have the following sequence of events:

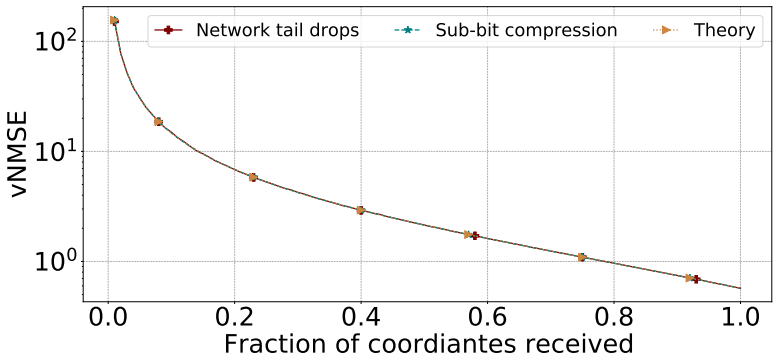


Figure 11. EDEN with sub-bit compression vs. network tail drops vs. the theoretical bound ($d = 2^{19}$).

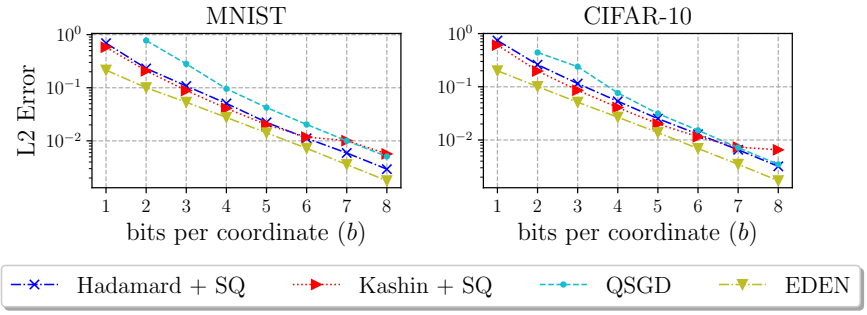


Figure 12. Distributed power iteration of MNIST and CIFAR-10 with 10 clients.

- Each client updates the top eigenvector based on its local data, compresses it, and sends it to the server.
- For each client, the server calculates the diff vector between the eigenvector from the previous round, averages the diffs and updates the eigenvector estimation using this average scaled by a *learning rate* of 0.1.
- The server sends the updated eigenvector estimation to all of the clients.

For each compression scheme, we vary the bit budget b from one bit to eight and measure the L2 norm of the diff between the final eigenvector to the optimal solution (i.e., the achieved eigenvector without compression). Figure 12 presents the results for the MNIST and CIFAR-10 (Krizhevsky, 2009; LeCun et al., 1998; 2010) datasets. It is evident that in both datasets, EDEN achieves the lowest L2 error for all values of b.

I.6. Homogeneous federated learning

We simulate two federated scenarios with 10 clients and the CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009). For both scenarios the data is uniformly distributed among the clients. We use a batch size of 128, an SGD optimizer with a Cross entropy loss criterion, and each client performs a single training step at each round. For CIFAR-10, we use the ResNet-9 (He et al., 2016) architecture with learning rate of 0.1, and the bit budget b is set to 0.1. For CIFAR-100, we use the ResNet-18 (He et al., 2016) architecture with learning rate of 0.05, and the bit budget b is set to 0.5.

Figure 13 presents the test and train accuracy of EDEN and competitive compression schemes, with a smoothing rolling mean window of 60 rounds. In both scenarios, EDEN achieves the highest accuracy. More specifically, for CIFAR-100, EDEN is the only compression scheme that converges for such a low bit budget.

I.7. Cross-device federated learning

We simulate two cross-device federated scenarios with 50 clients and the MNIST and CIFAR-10 datasets (Krizhevsky, 2009; LeCun et al., 1998; 2010). For both scenarios, 10 clients are randomly chosen at each training round, and each client

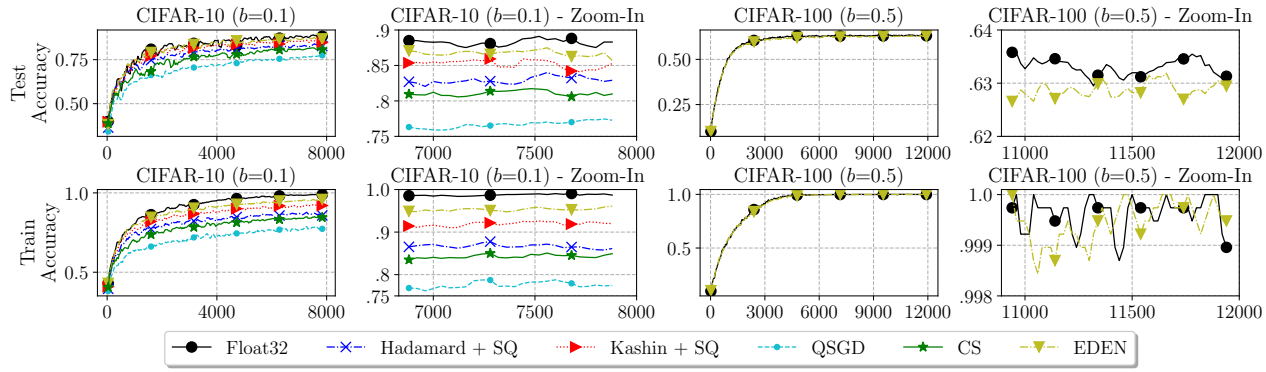


Figure 13. Homogeneous federated learning of CIFAR-10 and CIFAR-100 with 10 clients.

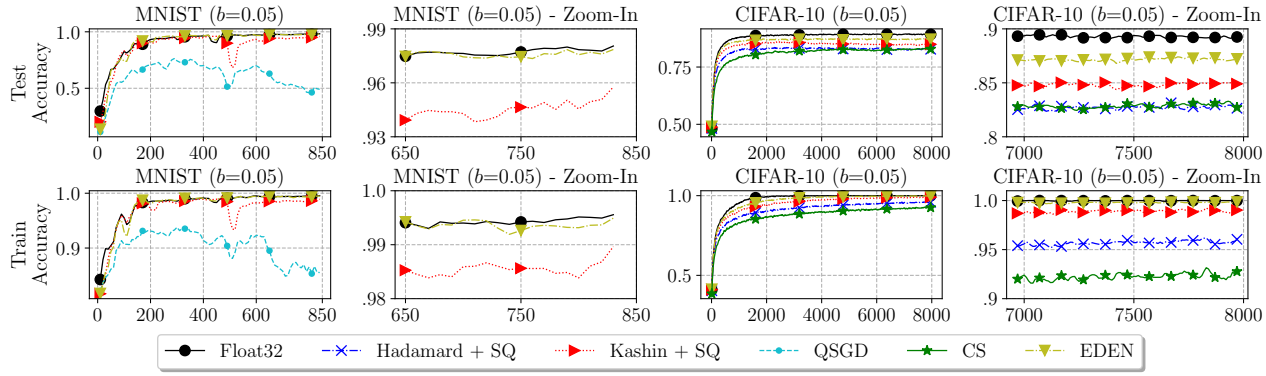


Figure 14. Cross-device federated learning of MNIST and CIFAR-10 with 50 clients.

performs five local training steps. We use a batch size of 128, an SGD optimizer with a cross entropy loss criterion, and the bit budget b is set to 0.05, testing a setting with very aggressive compression.

For MNIST, we use LeNet-5 (LeCun et al., 1998) with a learning rate of 0.05. To simulate severe heterogeneity, each client holds data instances that belong to a single class (i.e., severe label skew). For the CIFAR-10 configuration, we use ResNet-9 (He et al., 2016) with a learning rate of 0.1, and the data is uniformly distributed among the clients.

Figure 14 presents the test and train accuracy of EDEN and the competitive compression schemes, with a smoothing rolling mean window of 30 rounds. In both scenarios, EDEN achieves the highest accuracy. For MNIST, EDEN is only slightly below the baseline model (without compression).