Quick UnbIased Compression for Federated Learning (QUIC-FL): A Report

Gautam Singh
CS21BTECH11018

CONTENTS

***Abstract*—This document is a report of the paper [1]. It summarizes the main contributions of the authors and analyzes the results obtained. This report also lists out possible future research in the area of Federated Learning (FL).**

1 PROBLEM STATEMENT

The authors of [1] consider the **Distributed Mean Estimation** problem, illustrated in Figure 1. Specifically, each client C_i sends data $\mathbf{x}_i$, quantized as $\mathbf{Y}_i \in \{0,1\}^b$, for $1 \leq i \leq n$. Here, b represents the number of bits used for quantization per client.

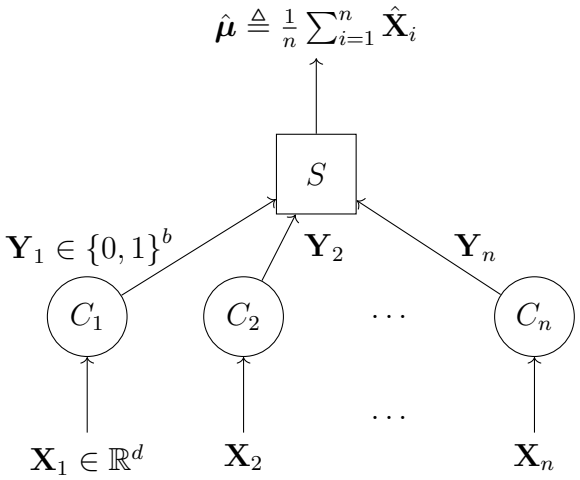

Fig. 1. An Illustration of the DME Problem.

The server computes the estimates $\hat{\mathbf{X}}_i$ using the obtained $\mathbf{Y}_i$ as

$$\hat{\boldsymbol{\mu}} \triangleq \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{X}}_i. \tag{1}$$

The aim of the DME problem is to estimate the true mean $\boldsymbol{\mu} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$ using $\hat{\boldsymbol{\mu}}$ as defined in (1) with *minimal error* (see section 3.1).

2 GOALS OF THE PAPER

The goals of [1] are to develop a quantization scheme that, compared to other state-of-the-art methods (see section 8.1 for a list of such methods).

1) Less computational complexity compared to other methods, at *both* client and server.
2) Same (asymptotic) NMSE of $\mathcal{O}\left(\frac{1}{n}\right)$.

3) Same convergence rate.
4) Better compression ratio.

3 Preliminaries

3.1 vNMSE and NMSE

The main performance metric used to assess a quantization scheme is the *squared error* of the estimated mean from the actual mean. To perform such an assessment, the authors define two quantities that will be useful.

Definition 1 (vNMSE)**.** *The* vector Normalized Mean Square Error *of a vector* $\mathbf{x}$ *is defined as*

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

Definition 2 (NMSE)**.** *For the DME problem, the* Normalized Mean Square Error *is defined as*

$$NMSE \triangleq \frac{\mathbb{E}\left[\|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\|_2^2\right]}{\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i\|_2^2} \tag{3}$$

$$= \frac{\mathbb{E}\left[\left\|\hat{\boldsymbol{\mu}}-\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|_2^2\right]}{\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i\|_2^2}. \tag{4}$$

From Definition 1 and Definition 2, we can perform simple algebraic manipulations to obtain the following result.

Lemma 1. *For n clients with unbiased and independent estimates, we have [2]*

$$NMSE = \frac{\sum_{i=1}^n vNMSE(i)\|\mathbf{x}_i\|_2^2}{n\sum_{i=1}^n \|\mathbf{x}_i\|_2^2} \tag{5}$$

Proof. From (4), we obtain

$$NMSE = \frac{\mathbb{E}\left[\|\sum_{i=1}^n (\hat{\mathbf{x}}_i-\mathbf{x}_i)\|_2^2\right]}{n\sum_{i=1}^n \|\mathbf{x}_i\|_2^2} \tag{6}$$

$$= \frac{\sum_{i=1}^n \mathbb{E}\left[\|\hat{\mathbf{x}}_i-\mathbf{x}_i\|_2^2\right]}{n\sum_{i=1}^n \|\mathbf{x}_i\|_2^2} \tag{7}$$

$$= \frac{\sum_{i=1}^n vNMSE(i)\|\mathbf{x}_i\|_2^2}{n\sum_{i=1}^n \|\mathbf{x}_i\|_2^2}. \tag{8}$$

where (7) is due to the fact that the estimates are unbiased and independednt, thus eliminating cross terms, and (8) is by (2). □

We also obtain this simple corollary as a result of (8).

Corollary 3.1. *In Lemma 1, if all clients have the same $vNMSE$ which is independent of the L2-norm of its vector, then*

$$NMSE = \frac{vNMSE}{n}. \tag{9}$$

3.2 Randomized Hadamard Transform (RHT)

QUIC-FL uses the RHT to gain speed asymptotically. In this subsection, we present a few definitions and important properties of the RHT.

Definition 3 (Walsh-Hadamard Matrix)**.** *The Walsh-Hadamard matrix $\mathbf{H}_{2^k}$ with $\mathbf{H}_1 = \begin{pmatrix}1\end{pmatrix}$ is recursively defined as*

$$\mathbf{H}_{2^k} = \begin{pmatrix}\mathbf{H}_{2^{k-1}} & \mathbf{H}_{2^{k-1}} \\ \mathbf{H}_{2^{k-1}} & -\mathbf{H}_{2^{k-1}}\end{pmatrix}. \tag{10}$$

Definition 4 (Randomized Hadamard Transform)**.** *Let $\mathbf{H} \in \{-1,+1\}^{d\times d}$ be a Walsh-Hadamard Matrix and $\mathbf{D}$ be a diagonal matrix with uniform iid Rademacher entries* i.e.*, entries that are ± 1 with equal probability. Then, the* randomized Hadamard transform *(RHT) of $\mathbf{x} \in \mathbb{R}^d$ is given by*

$$\mathcal{R}_{\mathbf{H}}(\mathbf{x}) \triangleq \left(\frac{1}{\sqrt{d}}\mathbf{H}\mathbf{D}\right)\mathbf{x}. \tag{11}$$

We can also define the inverse of the transform as follows.

Definition 5 (Inverse RHT)**.** *The* inverse randomized Hadamard transform *of* $\mathbf{x}$ *is given by*

$$\mathcal{R}_{\mathbf{H}}^{-1}(\mathbf{x}) \triangleq \left(\frac{1}{\sqrt{d}}\mathbf{D}\mathbf{H}\right)\mathbf{x}. \tag{12}$$

A key property of the RHT is obtained in the limit that $d \to \infty$, which we state without proof.

Theorem 3.2. *Define $F_{i,d}(x)$ to be the CDF of the random variable $\frac{1}{\sigma}\mathcal{R}_{\mathbf{H}}(\mathbf{x})_i$, where $\mathbb{E}[x_i^2] = \sigma^2$, and $\Phi(x)$ the CDF of the standard normal distribution. Then, as $d \to \infty$, $\forall\ 1 \le j \le d$,*

$$F_{i,d}(x_j) \xrightarrow{\mathrm{d}} \Phi(x_j). \tag{13}$$

Theorem 3.2 establishes a weak convergence between the RHT and the standard normal distribution, which is important for QUIC-FL, as we shall see.

4 Bounded Support Quantization (BSQ)

In this section, we describe the Bounded Support Quantization (BSQ) algorithm, which is used in QUIC-FL. We then proceed to analyze its NMSE, and comment on its utility. The BSQ encoding algorithm is described in Algorithm 1 and the BSQ decoding algorithm is described in Algorithm 2.

Algorithm 1 BSQ Encoder

Require: BSQ parameter $p \in (0, 1]$, stochastic quantization algorithm Q.

procedure BSQ-Encoder($\mathbf{x}$)
- $t_p \leftarrow$ threshold such that at most dp points in $\mathbf{x}$ lie outside $[-t_p, t_p]$.
- $\mathbf{U} \leftarrow \{x_i \mid |x_i| > t_p\}$
- $\mathbf{I} \leftarrow \{i \mid |x_i| > t_p\}$
- $\mathbf{V} \leftarrow \{x_i \mid |x_i| \leq t_p\}$
- $\mathbf{X} \leftarrow Q(\mathbf{V})$.
- Output $(\mathbf{U}, \mathbf{I}, \mathbf{X})$.

Algorithm 2 BSQ Decoder

Require: BSQ parameter $p \in (0, 1]$.

procedure BSQ-Decoder($\mathbf{U}, \mathbf{I}, \mathbf{X}$)
- $\hat{\mathbf{X}} \leftarrow$ reconstructed points from $\mathbf{X}$.
- $\hat{\mathbf{x}} \leftarrow \textsc{Merge}\left(\mathbf{U}, \mathbf{I}, \hat{\mathbf{X}}\right)$.
- Output $\hat{\mathbf{x}}$.

The BSQ parameter p is chosen by the designer. The essence of BSQ lies in the fact that errors for larger components may incur a larger NMSE. Thus, by quantizing in a *bounded support* $[-t_p, t_p]$, we can limit the NMSE. The main result of BSQ stated without proof is given below.

Theorem 4.1. *BSQ, without further assumptions, admits a worst-case NMSE of* $\frac{1}{np\left(2^b-1\right)^2}$ *with* $\mathcal{O}(d)$ *and* $\mathcal{O}(nd)$ *complexity for encoding and decoding respectively.*

One can clearly see that selecting arbitrarily small p can lead to a large constant factor in the NMSE. QUIC-FL performs a random rotation of $\mathbf{x}$ before applying BSQ to bound this constant factor.

5 Distribution-Aware Unbiased Quantization

The key idea of this quantization scheme lies in the fact that $\forall\ 1 \leq i \leq d$, where $\mathbf{R}$ denotes a random rotation matrix [3],

$$\frac{\sqrt{d}}{\|\mathbf{x}\|_2} (\mathbf{R}\mathbf{x})_i \sim \mathcal{N}(0, 1). \tag{14}$$

Thus, we only need to optimize the reconstruction values for when the data is sampled from the *standard normal distribution*. This will yield a *near-optimal* quantization for the original vectors $\mathbf{x}_i$. Notice that this is possible since

1) We know the distribution of the transformed vector by (14), as well as the inverse transform.
2) We know the bounded support range $[-t_p, t_p]$ in which stochastic quantization must be performed.

Therefore, we can format this new quantization problem as an optimization problem. The notation used for this section is shown in Table 1.

5.1 The Optimization Problem

We want to minimize the MSE of this quantizer while keeping it unbiased. Thus, the optimization problem for the distribution-aware unbiased quantizer is

$$\min_{S,R} \int_{-t_p}^{t_p} \sum_{x \in \mathcal{M}} S(z, x) (z - R(x))^2 e^{-\frac{z^2}{2}} dz \tag{15}$$

$$\text{s.t (Probability)} \sum_{x \in \mathcal{M}} S(z, x) = 1 \tag{16}$$

$$\text{(Unbiasedness)} \sum_{x \in \mathcal{M}} S(z, x) R(x) = z. \tag{17}$$

Notice that without the constraint in (17), this optimization problem corresponds to that of the Lloyd-Max quantizer.

5.2 Discretization

The optimization problem in (15) is quite difficult to solve analytically given the constraints (16) and (17). Thus, the authors propose a discretization of the problem, which makes this a linear programming problem to which solvers such as APOPT [4] and IPOPT [5] can be applied. The discretized changed optimization problem is as follows.

Notation	Definition
p	BSQ parameter
t_p	BSQ threshold
b	Number of bits per quantized value
$\mathcal{M}$	Message space $\{0, 1, \ldots, 2^b - 1\}$
$R(x)$	Reconstruction point associated with x at the receiver
$\mathcal{Q}_{b,p}$	Set of all reconstruction points $\{R(x) \mid x \in \mathcal{M}\}$
$\mathcal{I}_m$	$\{0, 1, \ldots, m-1\}$
$\mathcal{A}_{p,m}(i)$	$\Pr(Z \leq \mathcal{A}_{p,m}(i) \mid Z \in [-t_p, t_p],\ Z \sim \mathcal{N}(0,1)) = \frac{i}{m-1}$
$\mathcal{A}_{p,m}$	Right endpoints of quantiles $\{\mathcal{A}_{p,m}(i) \mid i \in \mathcal{I}_m\}$
$S(z, x)$	$\Pr(z$ is quantized to $R(x))$
$S'(i, x)$	$S(\mathcal{A}_{p,m}(i), x)$

TABLE 1
NOTATION FOR DISTRIBUTION-AWARE UNBIASED QUANTIZATION.

$$\min_{S', R} \sum_{\substack{i \in \mathcal{I}_m \\ x \in \mathcal{M}}} S'(i,x) \left(\mathcal{A}_{p,m}(i) - R(x)\right)^2 \text{ s.t.} \tag{18}$$

$$\text{(Probability)} \sum_{x \in \mathcal{M}} S'(i,x) = 1 \tag{19}$$

$$\text{(Unbiasedness)} \sum_{x \in \mathcal{M}} S'(i,x)\, R(x) = \mathcal{A}_{p,m}(i). \tag{20}$$

The reconstruction points $\mathcal{Q}_{b,p}$ obtained correspond to a *near-optimal* solution, since

1) as $d \to \infty$, the randomly rotated coordinates converge to the normal distribution, from (14).
2) as $m \to \infty$, the discrete problem converges to the continuous version.

6 THE QUIC-FL ALGORITHM

The authors piece the two quantizers above together to create the QUIC-FL client and server algorithms, illustrated in Algorithm 3 and Algorithm 4

Algorithm 3 QUIC-FL Client

Require: BSQ parameter p and threshold t_p, bits per symbol b, global randomness T, precomputed reconstruction values $\mathcal{Q}_{b,p}$, stochastic quantization algorithm Q.

procedure QUICFLCLIENT($\mathbf{x}_c$)
  $\mathbf{Z}_c \leftarrow \frac{\sqrt{d}}{\|\mathbf{x}_c\|_2} T(\mathbf{x}_c)$
  $\mathbf{U}_c \leftarrow \{Z_{ci} \mid |Z_{ci}| > t_p\}$
  $\mathbf{I}_c \leftarrow \{i \mid |Z_{ci}| > t_p\}$
  $\mathbf{V}_c \leftarrow \{Z_{ci} \mid |Z_{ci}| \leq t_p\}$
  $\mathbf{X}_c \leftarrow Q(\mathbf{V}_c)$ using $\mathcal{Q}_{b,p}$
  Send $(\|\mathbf{x}_c\|_2, \mathbf{U}_c, \mathbf{I}_c, \mathbf{X}_c)$ to server

Algorithm 4 QUIC-FL Server

Require: BSQ parameter p and threshold t_p, bits per symbol b, global randomness T, precomputed reconstruction values $\mathcal{Q}_{b,p}$, stochastic quantization algorithm Q.

procedure QUICFLSERVER
  for all $1 \leq c \leq n$ **do**
    $\hat{\mathbf{V}}_c \leftarrow \{\mathcal{Q}_{b,p}(x) \text{ for } x \text{ in } \mathbf{X}_c\}$
    $\hat{\mathbf{Z}}_c \leftarrow \text{MERGE}\left(\hat{\mathbf{V}}_c, \mathbf{U}_c, \mathbf{I}_c\right)$
  $\hat{\boldsymbol{\mu}}_Z \leftarrow \frac{1}{n} \sum_{c=1}^{n} \frac{\|\mathbf{x}_c\|_2}{\sqrt{d}} \hat{\mathbf{Z}}_c$
  $\hat{\boldsymbol{\mu}} \leftarrow T^{-1}(\hat{\boldsymbol{\mu}}_Z)$

This proposed algorithm gives the following NMSE bound.

Theorem 6.1. *Let $Z \sim \mathcal{N}(0,1)$ and $\hat{Z}$ be its estimate using the distribution-aware unbiased quantization scheme. Then, for any number of clients n and any set of input vectors $\{\mathbf{x}_i \in \mathbb{R}^d \mid i \in \{0, 1, \ldots, n-1\}\}$, the NMSE of QUIC-FL is given by*

$$NMSE = \frac{1}{n} \mathbb{E}\left[\left(Z - \hat{Z}\right)^2\right] + \mathcal{O}\left(\frac{1}{n}\sqrt{\frac{\log d}{d}}\right) \tag{21}$$

When $d \to \infty$, the NMSE respects $\mathcal{O}\left(\frac{1}{n}\right)$.

7 OPTIMIZATIONS TO QUIC-FL

In this section, we list out some optimizations to QUIC-FL to improve its performance.

7.1 Client-Specific Shared Randomness

A more general DME problem involves using shared randomness between clients and the server, with each client having their specific randomness. It

is possible to introduce a negative correlation using private randomness which can lower the NMSE.

In addition to the notation of (1), we introduce the following notation as well.

1) l, the number of random bits used.
2) $\mathcal{H}_l \triangleq \{0, 1, \ldots, 2^l - 1\}$, the set of all possible random bitstrings of length l.
3) $H \sim \mathcal{U}[\mathcal{H}_l]$, the random variable representing the shared randomness, selected uniformly from $\mathcal{H}_l$.
4) $S'(h, i, x)$, which is defined as $\Pr(\mathcal{A}_{p,m}(i) \text{ quantized to } R(H, x) \mid H = h)$.
5) $R(h, x)$, the reconstruction point for $x \in \mathcal{M}$ given shared randomness h.

The discrete optimization problem to be solved takes the following form.

$$\min_{S', R} \sum_{\substack{h \in \mathcal{H}_l \\ i \in \mathcal{I}_m \\ x \in \mathcal{M}}} S'(h, i, x) \left(\mathcal{A}_{p,m}(i) - R(h, x)\right)^2 \text{ s.t.} \tag{22}$$

$$\text{(Probability)} \sum_{x \in \mathcal{M}} S'(h, i, x) = 1 \tag{23}$$

$$\text{(Unbiasedness)} \frac{1}{2^l} \sum_{\substack{h \in \mathcal{H}_l \\ x \in \mathcal{M}}} S'(h, i, x) R(h, x) = \mathcal{A}_{p,m}(i). \tag{24}$$

Just like the number of quantiles m, we see a tradeoff in cost of generating random bits and reducing the NMSE with l.

7.2 Using the RHT

The advantages of using the RHT instead of a uniform random rotation are as follows.

1) Accelerates both client and server algorithms.
2) Same asymptotic guarantees but with larger constants.
   a) For the RHT, $\mathbb{E}\left[\left(Z - \hat{Z}\right)^2\right] = \mathcal{O}(p)$.
   b) Can compensate by slight increase in b.
   c) The guarantees on the constant are still better than DRIVE and EDEN.
3) Fastest complexity in comparison to other state-of-the-art methods, at both client and server.

8 Results

In this section, we analyze the results of QUIC-FL on various flavours of FL and various FL tasks.

8.1 Complexity and NMSE Analysis

The complexities and asymptotic NMSE of various state-of-the-art methods are illustrated in Table 2. Clearly, QUIC-FL has the fastest encoding and decoding complexity while not compromising on the NMSE of $\mathcal{O}\left(\frac{1}{n}\right)$.

Since this quantization scheme is unbiased with an NMSE of $\mathcal{O}\left(\frac{1}{n}\right)$, it converges as fast as vanilla stochastic gradient descent (SGD) [6].

8.2 Accuracy

In this section, we analyze the author's report on accuracy of QUIC-FL on various FL tasks.

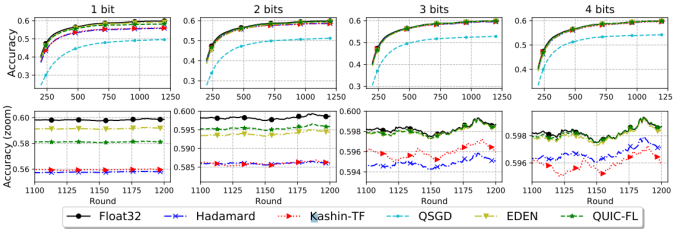

Fig. 2. *FedAvg* over the Shakespeare next-word prediction task at various bit budgets (rows). Training accuracy per round reported as a rolling mean of 200 rounds. [1]

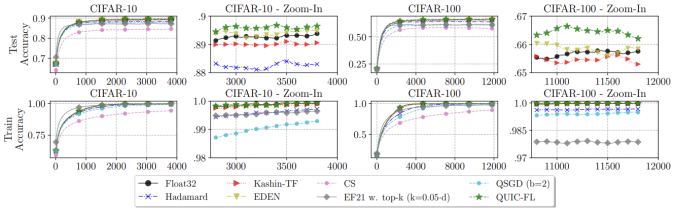

Fig. 3. Cross-device Federated Learning with Various Methods of Unbiased Quantization. [1]

In the next-word prediction task in Figure 2, QUIC-FL performs the best for larger communication budgets, but not for smaller communication budgets. QUIC-FL completes with EDEN for lower communication budgets in terms of accuracy.

In the cross-device FL task, the CIFAR-10 and CIFAR-100 datasets are used. QUIC-FL performs the best on both datasets in terms of test and train accuracy. There is some tapering down with the number of rounds in test accuracy which might be doubtful though.

Algorithm	Enc. Complexity	Dec. Complexity	NMSE
QSGD	$\mathcal{O}(d)$	$\mathcal{O}(nd)$	$\mathcal{O}(d/n)$
Hadamard	$\mathcal{O}(d\log d)$	$\mathcal{O}(nd + d\log d)$	$\mathcal{O}(\log d/n)$
Kashin	$\mathcal{O}(d\log d\log(nd))$	$\mathcal{O}(nd + d\log d)$	$\mathcal{O}(1/n)$
EDEN	$\mathcal{O}(d\log d)$	$\mathcal{O}(nd\log d)$	$\mathcal{O}(1/n)$
QUIC-FL	$\mathcal{O}(d\log d)$	$\mathcal{O}(nd + d\log d)$	$\mathcal{O}(1/n)$

TABLE 2
COMPLEXITIES AND NMSE OF UNBIASED DME ALGORITHMS.

9 FUTURE WORKS

After a thorough reading of the paper, the following concerns and suggestions are raised by the authors of the paper as well as this report.

1) (Open problem) Does using more than one RHT lead to better bounds?
2) Can QUIC-FL be modified to perform better for cases using limited bits b and dimensions d.
3) Can the NMSE be lowered using a client-server feedback system?
4) Security concerns of QUIC-FL.
    a) Does the random transformation T in Algorithm 3 and Algorithm 4 act as a "secret key"?
    b) What security guarantees does QUIC-FL have against various adversarial attacks?
5) Can shared randomness be introduced to reduce the NMSE? What are the security and storage concerns regarding its use?

REFERENCES

[1] R. B. Basat, S. Vargaftik, A. Portnoy, G. Einziger, Y. Ben-Itzhak, and M. Mitzenmacher, "QUIC-FL: Quick unbiased compression for federated learning," 2023.

[2] S. Vargaftik, R. Ben Basat, A. Portnoy, G. Mendelson, Y. Ben-Itzhak, and M. Mitzenmacher, "EDEN: Communication-Efficient and Robust Distributed Mean Estimation for Federated Learning," in *International Conference on Machine Learning*, 2022.

[3] ——, "DRIVE: One-bit Distributed Mean Estimation," in *NeurIPS*, 2021.

[4] "Advanced Process OPTimizer (APOPT) Solver," https://github.com/APMonitor/apopt.

[5] "Interior Point Optimizer (IPOPT) Solver," https://coin-or.github.io/Ipopt/.

[6] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," *CoRR*, vol. abs/1901.09847, 2019. [Online]. Available: http://arxiv.org/abs/1901.09847