Communication-Efficient Federated Learning With Data and Client Heterogeneity

Hossein Zakerinia[1]   **Shayan Talaei[2]**   **Giorgi Nadiradze [3]**   **Dan Alistarh[1]**

[1] Institute of Science and Technology Austria (ISTA)
[2] Stanford University
[3] Aptos Labs

Abstract

Federated Learning (FL) enables large-scale distributed training of machine learning models, while still allowing individual nodes to maintain data locally. However, executing FL at scale comes with inherent practical challenges: 1) heterogeneity of the local node data distributions, 2) heterogeneity of node computational speeds (asynchrony), but also 3) constraints in the amount of communication between the clients and the server. In this work, we present the first variant of the classic federated averaging (FedAvg) algorithm which, at the same time, supports data heterogeneity, partial client asynchrony, and communication compression. Our algorithm comes with a novel, rigorous analysis showing that, in spite of these system relaxations, it can provide similar convergence to FedAvg in interesting parameter regimes. Experimental results in the rigorous LEAF benchmark on setups of up to 300 nodes show that our algorithm ensures fast convergence for standard federated tasks, improving upon prior quantized and asynchronous approaches.

1   INTRODUCTION

In Federated learning (FL) (Konečnỳ et al., 2016; McMahan et al., 2017), multiple clients, orchestrated by a central authority, cooperate to jointly optimize a machine learning model given their local data. The basic FL algorithm is *federated averaging (FedAvg)* (McMahan et al., 2017), in which a central

authority periodically communicates a model to all clients; clients optimize this model locally, and send back the resulting updates to the server, which incorporates them, proceeding to the next iteration. This approach is effective in practice (Li et al., 2020), and motivates a rich line of research analyzing its convergence (Stich, 2019; Haddadpour and Mahdavi, 2019), as well as improved variants (Reddi et al., 2021; Karimireddy et al., 2020; Li and Richtárik, 2021).

Despite its popularity, scaling federated learning runs into a number of fundamental challenges (Kairouz et al., 2021). One such obstacle is *data heterogeneity*: the fact that the clients' local data distributions may be different, which can lead to difficulties in optimization Wang et al. (2020); Karimireddy et al. (2020). A second barrier is *node heterogeneity*: as practical deployments contain large node counts, which may execute *at different speeds*, it may be infeasible for a central server to orchestrate rounds across participants (Bonawitz et al., 2019; Wu et al., 2020; Nguyen et al., 2022). The third barrier is the *communication cost* of the parameter updates (Kairouz et al., 2021), which can overwhelm communication-limited clients (Jhunjhunwala et al., 2021; Li and Richtárik, 2021; Wang et al., 2022).

In a scalable FL system, all these three barriers need to be mitigated: for instance, communication-reduction may not be effective if the server has to *synchronously* wait for all of the clients to complete a communication round. Yet, supporting all these system relaxations jointly is extremely challenging: it is known (Wang et al., 2020) that one cannot allow both *data heterogeneity* and *client asynchrony* in full generality without impacting the objective; at the same time, none of the existing communication-efficient methods support asynchrony (Jhunjhunwala et al., 2021; Li and Richtárik, 2021; Wang et al., 2022). Thus, it is interesting to ask to what extent communication-compression, asynchrony, and heterogenous data *can be jointly supported* in FL.

Contribution. We address this question by proposing an algorithm for **Qu**antized **A**synchronous **F**ederated **L**earning called QuAFL, which is an extension of FedAvg supporting heterogeneous data, communication compression, and partial asynchrony. We provide a rigorous theoretical analysis of its convergence, showing that it asymptotically matches FedAvg in interesting parameter regimes, and experiments showing that it can also lead to practical gains.

Overview. At a high level, QuAFL follows the structure of FedAvg: in each "logical round," the server samples s clients uniformly at random, and sends them a (compressed) copy of its current model. As soon as a client receives the server's message, it performs two steps: first, it replies to the server with a (compressed) copy of its local progress, obtained via optimization steps on its local data, since its last server interaction. Second, the client adopts the server's model and will proceed to perform up to $K \geq 1$ local optimization steps on it in the future, at its own local speed. At the end of the round, the server collects the clients' messages and integrates them into its parameter estimate.

One key difference from FedAvg is that, in QuAFL, the entire process is *partially-asynchronous*: clients perform local steps each at their own speed, independently of the server's round structure, on their local version of the parameters. Thus, when sampled by the server, a client may still be in the middle of performing its K local steps since its last interaction, or may not yet have completed any local steps at all! In QuAFL, the clients' progress may be partial—as a client may not have completed its K local steps—and is always computed on a stale version of the parameter, previously sent by the server. The second key difference is that, in QuAFL, all client-server communication is compressed using a fast customized quantizer.

There are two analytical challenges in this setting: the first is in showing that the optimization process can still converge in this highly-decoupled setting, in which clients proceed at different speeds, can be interrupted asynchronously by the server, and sometimes do not make any progress at all. The second challenge is to interface asynchrony with communication compression: as detailed later, using standard quantizers (Alistarh et al., 2017; Karimireddy et al., 2019) induces error proportional to the second-moment gradient bound, which leads to both poor practical performance, and major difficulties in the analysis.

Our analysis circumvents these obstacles, and shows that QuAFL can provide surprisingly strong convergence guarantees, which match those of FedAvg in certain parameter regimes. We achieve this via a new potential argument, which roughly shows that, un-

der standard assumptions, the discrepancy between the client and server models is always bounded, and by leveraging an instance of position-aware quantization (Davies et al., 2021), which has the property that the compression error only depends on the *distance* between the models at the server and the clients. Our analysis approach controls the "noise" due to model inconsistency precisely, ensuring that local models are close enough to allow correct encoding and decoding using the customized positional quantizer.

We validate our algorithm experimentally in the rigorous LEAF benchmark (Caldas et al., 2018), on a range of standard tasks. We show that QuAFL can compress updates by more than $3\times$ without significant loss of convergence, and can even withstand a large fraction of "slow" clients submitting infrequent or even no updates, also in non-i.i.d. data settings. Moreover, in settings where client computation speeds are heterogenous, QuAFL provides end-to-end speedup in terms of iteration times, since the server can progress without waiting for all clients to complete their local computation, and is also competitive with asynchronous FL approaches such as FedBuff (Nguyen et al., 2022).

Related Work. There has been significant work on communication-compression for FedAvg (Philippenko and Dieuleveut, 2020; Reisizadeh et al., 2020; Jin et al., 2020; Haddadpour et al., 2021). However, virtually all prior work considers *synchronous* iterations. Reisizadeh et al. (2020) introduced a variant of FedAvg which supporting standard compressors, and provides convergence bounds under the assumption of i.i.d. client data. Jin et al. (2020) examined the signSGD quantizer (Seide et al., 2014) for FedAvg, providing convergence guarantees; however, the rates are polynomial in the *model dimension d*, rendering them less practically meaningful. Haddadpour et al. (2021) proposed a family of algorithms with communication-compression; yet, to prove convergence in the challenging heterogeneous-data setting, they require very strong technical assumptions on quantized gradients (Haddadpour et al., 2021, Assumption 5). Chen et al. (2021) also considered update compression, but under convex losses, coupled with a strong second-moment bound assumption on the gradients. Finally, Jhunjhunwala et al. (2021) adapt the degree of compression during the execution, proving convergence only under i.i.d. data sampling. *In sum, all prior work on compression for FedAvg requires at least one non-standard assumption. By contrast, our analysis works for non-convex losses, non-i.i.d. data, without second-moment gradient bounds. In addition, we support partial client asynchrony.*

A complementary approach has been to investigate FL optimizers with faster convergence (Mishchenko et al.,

2019; Karimireddy et al., 2020), or adaptive optimizers (Reddi et al., 2021; Tong et al., 2020). These approaches can be compatible with communication-compression (Gorbunov et al., 2021; Li and Richtárik, 2021; Wang et al., 2022). Specifically, for non-convex losses, MARINA (Gorbunov et al., 2021) offers theoretical guarantees both in terms of convergence and bits transmitted. However, MARINA is synchronous; moreover, it periodically computes full gradients and transmits uncompressed model updates, and requires complex synchronization and variance-reduction to compensate for quantization noise. DASHA (Tyurin and Richtárik, 2023) proposed a family of theoretical methods which extend MARINA with Momentum Variance Reduction (MVR) (Cutkosky and Orabona, 2019), partially relaxing the coupling between the server and the workers. *By contrast to this work, we focus on obtaining a practical algorithm in a highly-decoupled model, with competitive convergence relative to vanilla FedAvg: we always transmit compressed, low-precision messages, and consider asynchronous communication and client progress.*

FedBuff (Nguyen et al., 2022) is a state-of-the-art *practical* approach for asynchronous FL, where nodes aggregate their updates asynchronously in a shared buffer; once the buffer is full, the server updates the global model and communicates it. *Our convergence bounds are competitive to FedBuff, but in a more general setting, as we do not assume a gradient bound.* Experimentally, QuAFL achieves better performance relative to FedBuff in the non-i.i.d. case: the intuitive reason is that, in this case, slower clients will consistently contribute less frequently to the buffer, meaning that convergence could be "skewed" towards fast clients. Parallel work by Koloskova et al. (2022) and Mishchenko et al. (2022) provide sharp convergence bounds for asynchronous SGD in a related but different model, with arbitrary worst-case delays. Specifically, they prove convergence rates that are similar to ours in the case of a single client sampled at a time. By contrast, our work considers a *probabilistic model* on the delays, similar to Cannelli et al. (2020). The two models are incomparable: we allow the worst-case delay to be unbounded, but assume that each client i proceeds at an expected speed H_i. In addition, in our algorithm, the clients can be interrupted by the server during their local computation, which leads to further difficulties in the analysis, and practical improvements in terms of waiting times. FedNova (Wang et al., 2020) works in a *synchronous* model, but it allows each node to perform a different number of local steps at each round. One main difference between our work and FedNova is that in the asynchronous setting, the $(n - s)$ clients which are not participating in a round can still perform local updates at their

own speed, while in the synchronous scenario of FedNova, these $(n - s)$ clients will be idle. We show that asynchrony can result in faster wall-clock time convergence. Similarly, FedNova requires that, at each round, all participating nodes perform at least one local step, which results in all nodes having to wait for the slowest client; we allow nodes to perform zero steps at a round. Additionally, the bounded dissimilarity assumption in FedNova is stronger than the standard literature definition, which we use in our analysis.

2 THE ALGORITHM

2.1 System Overview

Optimization Setting. We assume a distributed system with one coordinator and n workers, jointly minimizing a d-dimensional, differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. We consider empirical risk minimization (ERM), where data samples are located at the n nodes. Each agent i has a local function f_i associated to its own data partition, i.e $\forall x \in \mathbb{R}^d$: $f(x) = \sum_{i=1}^n f_i(x)/n$. The goal is to converge on a model x^* which minimizes the empirical loss. Clients run a distributed variant of SGD, coordinated by the central node. Each client i is able to obtain *unbiased stochastic gradients* $\widetilde{g}_i$ of its own local function f_i, i.e. $\mathbb{E}[\widetilde{g}_i(x)] = \nabla f_i(x)$, sampling i.i.d. from its local distribution.

System Model. Our algorithm will follow the general pattern of FedAvg, in that the server periodically polls a subset of clients, sending them its model. However, the interaction pattern is *(partially) asynchronous*: when contacted, clients immediately communicate their local progress since the last interaction, even though it may be partial, and afterwards proceed to take local steps on the new model communicated by the server, until their next interaction. Moreover, the clients themselves may progress at *heterogeneous speeds*: the number of local steps taken by client i between server interactions t and $t + 1$ is a random variable $\mathcal{H}_i$, taking values in $\{0, 1, 2, \ldots, K\}$, where K is an upper bound on how many steps a client can take in isolation. *Our only assumption is that, for each client i and every server interaction, the expected value of $\mathcal{H}_i$, denoted by H_i, exists and is > 0.* That is, on average, each client has a fixed speed, and makes non-zero progress. *However, clients may progress at different speeds, and the individual step distributions $\mathcal{H}_i$ can be completely different. We emphasize that $\mathcal{H}_i$ can be 0, meaning that i has taken no steps between interactions.*

2.2 Algorithm Description

Overview. The pseudocode for QuAFL is given in Algorithm 1. From the server's perspective, the execu-

---

Algorithm 1 Pseudocode for the QuAFL Algorithm at Server and Clients.

% Initial models $X_0 = X^1 = X^2 = ... = X^n = 0^d$, number of local steps K
% Encoding $(Enc(A))$ and decoding $(Dec(B, Enc(A)))$ functions, with common parametrization.
% For each client i, the expected local steps between two consecutive server interactions is H_i.
% For each client i, we define weights $\eta_i = \frac{H_{\min}}{H_i}$ where $H_{\min}$ is the minimum speed among H_i.
% **At the Server**:
1: **for** $t = 0$ **to** $T - 1$ **do**   % Each round takes a fixed amount of time.
2:     Server chooses s clients uniformly at random, let S be the resulting set.
3:     **for all** clients $i \in S$ **do**
4:        Server sends $Enc(X_t)$ to the client i.
5:        Server receives $Enc(Y^i)$ from client i % Y^i is the client's model with the progress since the last interaction.
6:        $Q(Y^i) \leftarrow Dec(X_t, Enc(Y^i))$ % Decodes quantized client messages relative to X_t
7:     **end for**
8:     $X_{t+1} = \frac{1}{s+1}X_t + \frac{1}{s+1}\sum_{i \in S}Q(Y^i)$
9: **end for**
% **At Client** i:
% Upon (asynchronous) contact from the server run INTERACTWITHSERVER
% **Local variables:**
% X^i stores the base client model, following the last server interaction. Initially 0^d.
% $\widetilde{h}_i$ accumulates local gradient steps since last server interaction, initially 0^d.
1: **function** INTERACTWITHSERVER
2:     $MSG_i \leftarrow Enc(X^i - \eta\eta_i\widetilde{h}_i)$ % Client i compresses its local progress since last contacted.
3:     Client sends MSG_i to the server.
4:     Client receives $Enc(X_t)$ from the server, where t is the current server time.
5:     $Q(X_t) \leftarrow Dec(X^i, Enc(X_t))$ % Client decodes the message *using its own model* as reference point.
6:     $X^i = \frac{1}{s+1}Q(X_t) + \frac{s}{s+1}(X^i - \eta\eta_i\widetilde{h}_i)$ % The client then updates its local model
7:     LOCALUPDATES(X^i, K) % Finally, The client goes back to compute new local updates.
8:     WAIT( )
9: **end function**
1: **function** LOCALUPDATES(X^i, K)
2:     $\widetilde{h}_i = 0$ % local gradient accumulator
3:     **for** $q = 0$ **to** $K - 1$ **do**
4:        $\widetilde{h}_i^q = \widetilde{g}_i(X^i - \eta\sum_{\ell=0}^{q-1}\widetilde{h}_i^\ell)$ % compute the qth local gradient
5:        $\widetilde{h}_i = \widetilde{h}_i + \widetilde{h}_i^q$ % add it to the accumulator
6:     **end for**
7: **end function**

---

tion is similar to FedAvg: we execute logical "rounds," where in each round t the server polls a subset of s workers, sending them a compressed version of its model $Enc(X_t)$. However, the server *does not wait for workers to perform local steps over X_t in this round*: instead, it immediately receives each worker's local progress $Enc(Y^i)$ *since worker i's last server interaction.* (Thus, the server will observe progress on X_t from clients only on their *next* interaction.) The received progress is de-quantized, and integrated into the server's local model *via weighted averaging.*

From the other perspective, a contacted worker i could be either idle when polled by the server, having completed its K steps since the last server interaction, or still in the process of performing local steps. In either case, the worker *immediately* quantizes its possibly-partial local progress Y^i since the last server contact, and sends it in quantized form $Enc(Y^i)$ to the server. The worker then decodes the server's quantized message $Q(X_t)$, and updates its own local model X^i

correspondingly via weighted averaging. The worker then starts performing K local steps on top of the updated local parameters, until its next server interaction. QuAFL relaxes the FedAvg pattern as follows.

Non-blocking Communication. A key limitation of standard FedAvg is that the server has to wait for all contacted workers to compute their K updates and transmit them each round, before moving on. In QuAFL, communication is *non-blocking*: the contacted worker node i *immediately returns (a quantized version of) its local progress* since the last interaction to the server, without performing any computation. We emphasize that this progress may be incomplete, or even zero for some clients, and that it is computed with respect to the server's previously-communicated model, not the one just received. Conversely, the server does not wait for clients to take steps on its current model X_t; it will observe these updates in future interactions. This significantly reduces the server's waiting times, and allows QuAFL to

"pipeline" several communication rounds over a fixed wall-clock time, at the cost of supporting inconsistent client models in the analysis and in practice.

Model Averaging. This parameter inconsistency is handled via the server- and client-side averaging mechanism, which differs from the standard FedAvg iteration. Specifically, in each round, the server model "weight" is $\frac{1}{s+1}$, and it gets averaged with s clients. Each of them, plus the server itself, gets a $\frac{1}{s+1}$ "fraction" of the server model. Thus, the server's model is evenly distributed among $s + 1$ participants; in turn, the server receives a $\frac{1}{s+1}$ fraction of the local models of each client node it interacts with. Crucially, this average does not change, but the local models *move closer to the mean*. This idea will be reflected in our analysis, which works by first tracking convergence *at the mean of client models*, since the change in average only depends on the stochastic gradient updates at the round, and on the prior mean. We show that mean convergence implies convergence at the server.

Partial Client Asynchrony. As discussed in Section 2.1, clients progress at different speeds. Specifically, each client i is assumed to take H_i steps between two server interactions, in expectation. We support this in QuAFL as follows: the only shared information, maintained by the server, is $H_{\min}$, the lowest "speed" of any participating client. To address the difference in average speeds, each client will "dampen" its transmitted progress before transmitting it to the server, by a factor of $\eta_i = H_{\min}/H_i$. We will show that this is sufficient to maintain balance in the optimization objective, without losing any client privacy, as the server does not need to be aware of client speeds.

Fully-Quantized Communication. For quantization, we employ a customized version of the lattice-based quantizer of Davies et al. (2021); its parametrization is described formally in Section 3. Quantization works via an encoding function $Enc(A)$, which maps A to its quantized representation. To "read" message $Enc(A)$, a node calls the symmetric $Dec(B, Enc(A))$ function, which allows "decoding" of $Enc(A)$ with respect to a reference B, returning output $Q(A)$. As evident from the pseudocode, server-client communication in QuAFL is always quantized, as opposed to prior methods (Gorbunov et al., 2021), where the server still transmits full-precision updates.

The Issue with Standard Quantizers. We emphasize that compressing via standard quantizers (Alistarh et al., 2017), induces error proportional to the model's norm, which is in principle unbounded. Thus, applying direct quantization to existing methods is not theoretically-justified, and experimentally it does not always lead to good results (see Section 4). Prior work

in decentralized optimization Lu and De Sa (2020); Nadiradze et al. (2021) addressed this by only transmitting *model updates*. This requires both additional memory at the client, or an unrealistic second-moment gradient bound. Our approach avoids both issues: it requires no extra memory, and makes no extra assumptions.

3 CONVERGENCE ANALYSIS

3.1 Analytical Assumptions

We begin by stating the assumptions we make in the theoretical analysis of our algorithm. We assume the following for the global loss function f, the client losses f_i, and their stochastic gradients $\widetilde{g}_i$:

1. **Uniform Lower Bound:** There exists $f_* \in \mathbb{R}$ such that $f(x) \geq f_*$ for all $x \in \mathbb{R}^d$.

2. **Smooth Gradients**: For any client i, the gradient $\nabla f_i(x)$ is L-Lipschitz continuous for some $L > 0$, i.e. for all $x, y \in \mathbb{R}^d$: $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$.

3. **Bounded Variance**: For any client i, the variance of the stochastic gradients is bounded by some $\sigma^2 > 0$, i.e. for all $x \in \mathbb{R}^d$: $\mathbb{E}\left\|\widetilde{g}_i(x) - \nabla f_i(x)\right\|^2 \leq \sigma^2$.

4. **Bounded Dissimilarity**: There exist constants $G^2 \geq 0$ and $B^2 \geq 1$, s.t. $\forall x \in \mathbb{R}^d$:

$$\sum_{i=1}^n \frac{\|\nabla f_i(x)\|^2}{n} \leq G^2 + B^2\|\nabla f(x)\|^2.$$

The first three conditions are universal in distributed non-convex stochastic optimization, whereas the fourth encodes the fact that there must be a bound on the amount of divergence between the local distributions at the nodes in order to allow for joint optimization (Karimireddy et al., 2020; Jin et al., 2020; Gorbunov et al., 2021).

In addition, we make the following assumption on the local progress performed by each node:

5. **Probabilistic Progress:** For each client i, the expected number of local steps taken since the last interaction when contacted by the server is $0 < H_i \leq K$.

Clearly, a condition of this type is *necessary*: if a client makes zero progress on average, we cannot converge to a consistent objective in the heterogeneous setting.

Quantization Procedure. Please recall the semantics of our quantization procedure, as described in Section 2.2. In this context, the quantizer has the following guarantees (Davies et al., 2021) (Lemma 23):

Lemma 3.1. *(Lattice Quantization) Fix parameters R and $\gamma > 0$. There exists a quantization procedure defined by an encoding function $Enc_{R,\gamma} : \mathbb{R}^d \to \{0,1\}^*$ and a decoding function $Dec_{R,\gamma} = \mathbb{R}^d \times \{0,1\}^* \to \mathbb{R}^d$ such that, for any vector $x \in \mathbb{R}^d$ which we are trying to quantize, and any vector y which is used by decoding, which we call the* decoding key, *if $\|x - y\| \le R^{R^d}\gamma$ then with probability at least $1 - \log\log(\frac{\|x-y\|}{\gamma})O(R^{-d})$, the function $Q_{R,\gamma}(x) = Dec_{R,\gamma}(y, Enc_{R,\gamma}(x))$ has the following properties:*

1. *(Unbiased decoding)* $\mathbb{E}[Q_{R,\gamma}(x)] = \mathbb{E}[Dec_{R,\gamma}(y, Enc_{R,\gamma}(x))] = x$;

2. *(Error bound)* $\|Q_{R,\gamma}(x) - x\| \le (R^2 + 7)\gamma$;

3. *(Communication bound)* $O\left(d\log(\frac{R}{\gamma}\|x - y\|)\right)$ *bits are needed to send $Enc_{R,\gamma}(x)$.*

3.2 Main Results

Our proof strategy will be to show that the clients' local models stay close to the server's. This coupling is used both to show that the models converge jointly, but also that we can successfully apply the quantizer. Let $\mu_t = (X_t + \sum_{i=1}^n X^i)/(n+1)$ be the mean over all the node models in the system at a given t. Our main result shows the following:

Theorem 3.2. *Let $H_i > 0$ be client i's speed, and $H_{\min}$ be the minimum client speed. Assume the total number of server rounds is $T \ge \Omega(n^3)$, the learning rate $\eta = \frac{n+1}{H_{\min}\sqrt{sT}}$, and the quantization parameters R, and γ satisfy that $\gamma^2 = \frac{\eta^2}{(R^2+7)^2}\left((\sum_{i=1}^n \frac{H_i^2}{nH^2})\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L}\right)$, and $R = 2 + T^{\frac{3}{d}}$. Then, we have that Algorithm 1 converges at the following rate:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \le \frac{4(f(\mu_0) - f_*)}{\sqrt{sT}}$$
$$+ \frac{36KL(\sum_{i=1}^n \frac{\sigma^2}{nH_i^2} + \frac{2KG^2}{H_{\min}^2})}{\sqrt{sT}}$$
$$+ O\left(\frac{n^3K^2L^2((\sum_{i=1}^n \frac{H_{\min}^2}{nH_i^2})\sigma^2 + 2KG^2)}{H_{\min}^3 T}\right).$$

The algorithm uses $O(sT(d\log n + \log T))$ expected communication bits.

Discussion. The result shows non-trivial trade-offs between the convergence speed of the algorithm, the variance of the local distributions (given by σ and G), the sampling set size s, and individual client speeds H_i. We now examine it across relevant parameter regimes.

Let us first consider the case where the clients are *homogenous*, i.e. $H = H_i = H_{\min}, \forall i$. If assume that the

maximum number of local steps K is constant, then we also get constant H since $K \ge H$. Then, our bound is asymptotically-optimal. Specifically, the first two upper bound terms achieve the "optimal" speedup $\sqrt{sT}$ with respect to the sampling parameter s and the number of iterations T for the non-convex case. The third term contains similar "nuisance factors" as the second term, with the addition of the n^3 factor, which is directly due to asynchrony. Crucially, this larger third term is divided by T, as opposed to $\sqrt{T}$; since T is our asymptotic parameter, the whole third term is commonly assumed to be negligible (Lu and De Sa, 2020). Thus, in this case essentially the entire overhead of asynchrony and quantization is "offloaded" onto the third term, which becomes negligible as T grows.

For equal speeds $H_i = H$ and *non-constant* local steps K, it is reasonable to assume that $H = \Theta(K)$: on average, each client i will have completed a significant fraction of its local steps on the old version of the model X^i when being contacted. (Otherwise, the sampling frequency of the server is too high, and the server should simply decrease it.) If $H = \Theta(K)$, our algorithm also gets linear speedup with respect to the parameter H in terms of reducing the impact of the variance σ^2, in the second term. These dependencies match some of the best known bounds for standard FedAvg, in a similar learning rate regime: without asynchrony and quantization, we asymptotically recover the FedAvg bounds (Karimireddy et al., 2020).

Finally, take the interesting case of *heterogeneous client speeds* H_i. The first term is not affected, and the third term remains negligible as long as the "slowdown" of the slowest node is not asymptotic in the total steps T. In the second term, there is a remarkable difference between the *variance terms* σ^2/H_i^2, which decrease proportionally to the client speeds, and the *discrepancy term* $2KG^2$, which only gets decreased proportionally to the square of $H_{\min}$, the speed of the lowest node. This is inherent: since the local data distributions are *heterogenous*, the system cannot make progress in the discrepancy term without contributions from the slowest node. As such, our results are compatible with the trade-offs between asynchrony and data heterogeneity in FedNova (Wang et al., 2020).

To simplify the discussion, the Theorem statement considers a parametrization of the quantizer under which the impact of quantization noise is subsumed into the third upper bound term. For a more fine-grained upper bound result, please see Appendix B.4.

Convergence at the Server. We can obtain a similar bound for convergence *of the server's model*, as opposed to convergence of the mean of local models.

Corollary 3.3. *Assume that the total number of steps*

is $T \geq \Omega(n^4)$, whereas all the other parameter values are identical to Theorem 3.2. Then, with probability at least $1 - O(\frac{1}{T})$, the server's model X_t converges asymptotically at the same rate as the model average.

This bound is very similar to our main result, except for the larger dependency between the parameters T and n. Intuitively, this is required in the analysis due to the additional "mixing time" required for the server to converge to a similar bound to the mean μ_t. However, we do not observe such a requirement in practical experiments, and this bound is realistic given that the optimization process is usually executed over a large number of iterations.

Thus, the discussion of convergence interesting parameter regimes remains the same as for convergence of the average. In sum, QuAFL can match some of the best known rates for FedAvg. We find this surprising, since our algorithm executes in a highly-decoupled environment, in which communication is non-blocking and compressed, and clients are partially asynchronous.

3.3 Overview of the Analysis

We provide an overview of the proofs, outlining the main intermediate results. (The full analysis is given in the Appendix.) The first step is bounding the deviation between the local models and their mean. For this, we define the following potential function: $\Phi_t = \|X_t - \mu_t\|^2 + \sum_{i=1}^{n} \|X^i - \mu_t\|^2$. We can show that this potential has the following supermartingale-type property:

Lemma 3.4. *For any time step t we have:*

$$\mathbb{E}[\Phi_{t+1}] \leq (1 - \frac{1}{4n})\mathbb{E}[\Phi_t]$$
$$+ 8s\eta^2 \sum_{i=1}^{n} \eta_i^2 \mathbb{E}\|\widetilde{h}_i\|^2 + 16n(R^2 + 7)^2\gamma^2.$$

The intuition behind this result is that potential Φ_t will stay well-concentrated around its mean, except for influences from the variance due to local steps (second term) or quantization (third term). With this in place, the next lemma allows us to track the evolution of the average of the local models, with respect to local step and quantization variance:

Lemma 3.5. *For any step t,* $\mathbb{E}\|\mu_{t+1} - \mu_t\|^2 \leq \frac{2\eta^2}{(n+1)^2}\mathbb{E}\|\sum_{i \in S} \eta_i \widetilde{h}_i\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}.$

In both cases, the upper bound depends on the second moment of the nodes' local progress ($\sum_i \eta_i^2 \mathbb{E}\|\widetilde{h}_i\|^2$ or $\|\sum_{i \in S} \eta_i \widetilde{h}_i\|^2$). This is due to the fact that the server contacts s clients, which are chosen uniformly at random. Then, our main technical lemma uses properties

(2), (3) and (4), to concentrate these quantities around the true gradient $\mathbb{E}\|\nabla f(\mu_t)\|^2$, where the expectation is taken over the algorithm's randomness.

Lemma 3.6. *For any step t*

$$\mathbb{E}\|\sum_{i \in S} \eta_i \widetilde{h}_i\|^2 \leq \frac{16s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t]$$

$$+ 18sK((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + 64s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

We can use Lemmas 3.5 and 3.6 to bound $\mathbb{E}\|\mu_{t+1} - \mu_t\|^2$. Similarly to Lemma 3.6, we can get an upper bound for $\sum_i \eta_i^2 \mathbb{E}\|\widetilde{h}_i\|^2$, and by combining it with Lemma 3.4 we get the following upper bound on the potential with respect to $\mathbb{E}\|\nabla f(\mu_t)\|^2$.

Lemma 3.7. *We have that:*

$$\sum_{t=0}^{T} \mathbb{E}[\Phi_t] \leq O(Tn^2(R^2 + 7)^2\gamma^2 + K \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mu_t)\|^2))$$

$$+ n^2 sK\eta^2(T(\frac{\sum_{i=1}^{n} \eta_i^2 \sigma^2}{n} + 2KG^2).$$

Next, using the L-smoothness of the function f (2), we can show that

$$\mathbb{E}[f(\mu_{t+1})] \leq \mathbb{E}[f(\mu_t)] + \mathbb{E}\langle \nabla f(\mu_t), \mu_{t+1} - \mu_t \rangle$$
$$+ \frac{L}{2}\mathbb{E}\|\mu_{t+1} - \mu_t\|^2. \quad (1)$$

Weighting for heterogeneous clients. Using (1), and given that $\mathbb{E}[\mu_{t+1} - \mu_t] = -\frac{\eta}{n+1}\sum_{i \in S} \eta_i \widetilde{h}_i(X_t^i)$, we observe that the sum $\sum_{i=1}^{n} \mathbb{E}\langle \nabla f(\mu_t), \mu_{t+1} - \mu_t \rangle$ can be concentrated around $\mathbb{E}\|\nabla f(\mu_t)\|^2$, in similar fashion as in Lemma 3.6. While bounding this quantity we need to control $\sum_{i=1}^{n} \eta_i H_i(-\mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle)$. If data were homogeneous, for each i we had $\langle \nabla f(\mu_t), \nabla f_i(\mu_t) \rangle = \|\nabla f(\mu_t)\|^2$ and we would not need weighting. However, in the heterogeneous setting, the $f_i(\mu_t)$s are different and each of the inner products can have arbitrary sign. Thus, we choose $\eta_i = \frac{H_{\min}}{H_i}$ such that all $\eta_i H_i$s are equal, and the full term is $-nH_{\min}\|\nabla f(\mu_t)\|^2$.

Final argument. Then we can place this upper bound and the upper bound for $\mathbb{E}\|\mu_{t+1} - \mu_t\|^2$ in (1), summing over all T steps, and use Lemma 3.7 to cancel out the terms containing the potential $\sum_{t=0}^{T} \mathbb{E}[\Phi_t]$ based on $\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mu_t)\|^2$, and modulo some careful wrangling, we obtain the convergence bound in Theorem B.16. Setting the quantization parameters as stated, we get the claimed convergence rate. We discuss the impact of the quantization parameters in detail in the Appendix.
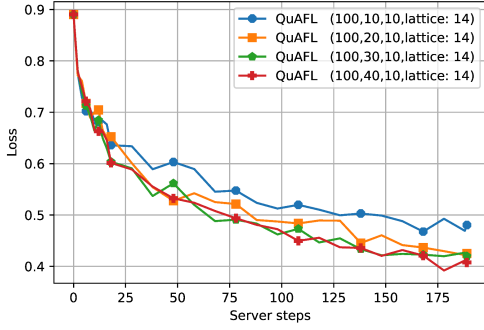
Figure 1: Impact of the number of peers $s \in \{10, 20, 30, 40\}$ on convergence, for $n = 100$ clients, 14-bit quantization, on
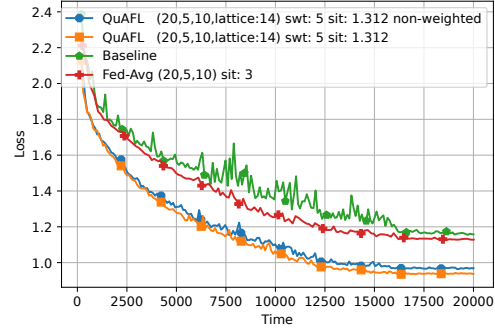

Figure 2: Convergence comparison relative to simulated time between QuAFL and FedAvg for ResNet20/CIFAR10.
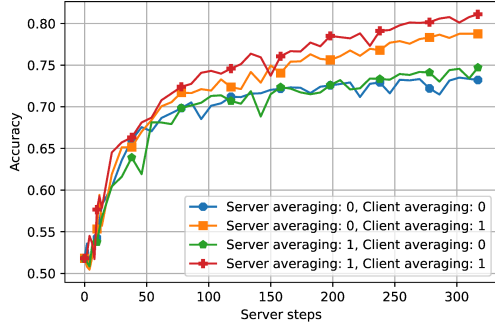

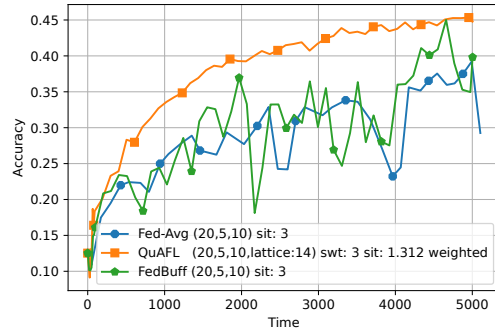**Figure 3:** The impact of averaging variants vs. validation accuracy on ResNet/CelebA, non-i.i.d data.

Figure 4: ResNet20/CIFAR10 experiment where Fast and Slow clients have non-i.i.d. data from different classes.

4 EXPERIMENTAL RESULTS

Experimental Setup. We implement QuAFL in Pytorch train neural networks for image classification tasks, specifically residual CNNs (He et al., 2016) on the MNIST (LeCun and Cortes, 2010), Fashion MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky and Hinton, 2009) and CelebA (Liu et al., 2015) datasets, in the rigorous benchmarking setup of LEAF (Caldas et al., 2018). Experiments on MNIST, FMNIST and CIFAR use a fixed random split of the training set among the nodes, while the CelebA experiments are executed in a pure non-i.i.d. setting, in which the samples are split across classes, so that each client receives a non-overlapping subset of classes. Full experimental details are presented in the Appendix. We omit error bars for readability, as the variance between experiments is low and does not impact our conclusions.

Goals and Metrics. Experiments are described by (n, s, K, b), where b is the number of bits used. In addition, swt is the server waiting time between two consecutive calls, and the server interaction time, sit, as the amount of time that server needs to send and receive necessary data. We assume a *server* and n *clients*, of which s are chosen randomly to be sampled in a round. The training data is distributed among clients so that each has access to a fixed $1/n$ partition.

We track the accuracy of the server's model on an unseen validation dataset. We measure *loss* and *accuracy* of the model with respect to *simulation time*. We update the both client and server models following QuAFL, and then increase the server time by *sit*. The server then waits for another interval of *server waiting time (swt)* to make its next call. Unless otherwise stated, all communication is quantized using the lattice quantizer of Davies et al. (2021), which is simply implemented via a random rotation followed by direct quantization/dequantization.

We run two timing experiments: *uniform* experiments assume all clients take the same amount of time for a gradient step; *non-uniform* timing experiments differentiate between *fast* or *slow* clients. Specifically, the length of each client step is taken to be a random variable $X \sim exp(\lambda)$, where λ is $1/2$ for *fast clients* and $1/8$ for *slow* clients; the expected runtime $\mathbb{E}(X)$ would be 2 and 8, respectively. In each experiment, we assumed 30% of clients to be *slow*. Unless otherwise noted, we employ the *unweighted* version of QuAFL, meaning that clients set their weighting parameter to $\eta_i = 1$ in the algorithm. All code is available at https://github.com/ShayanTalaei/QuAFL.

Results. Figure 1 examines the impact of the number of sampled peers s when training ResNet18 on the
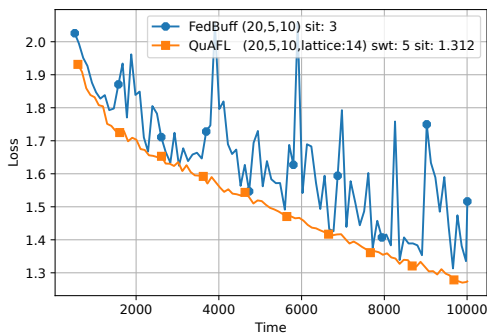
Figure 5: Experiment in which the average time per client per local step is uniformly random between 2 and 9, showing superior performance relative to FedBuff, which becomes unstable when both data and client speeds are heterogeneous (ResNet20/CIFAR10).

Figure 6: Comparison with FedNova in the same setup as Figure 5. FedNova can handle different client speeds, but is synchronous, therefore there are many idle clients during training, leading to slower convergence vs time.

(non-i.i.d) CelebA dataset, where 30% of clients are slow. Observe that convergence speed clearly follows the ordering of the number of peers s, confirming our analysis. Interestingly, timings in this experiment lead to a 27% probability that a slow client *will not have taken any steps* when interacting with the server, i.e. *its progress Y_i is zero*. This shows that QuAFL is robust to such slow clients, although their proportion impacts convergence.

In Figure 2 we examine the convergence of FedAvg and QuAFL in *simulated execution time*, in a system with 20 clients, out of which 25% are slow. (The *Baseline* is a single slow node that performs an optimization step per round.) Here, it is evident that non-blocking communication in QuAFL leads to faster convergence in terms of wall-clock time. The figure also shows better convergence for the *weighted* version of QuAFL, where agents set η_i according to speed.

In Figure 3, we examine the impact of different types of averaging on the convergence of the basic QuAFL pattern, on the non-i.i.d. CelebA dataset, with $n = 100$ clients. We clearly observe that the variant where averaging is applied *both at the server and at the client performs the best*.

In Figure 4, we compare QuAFL with FedBuff, a SOTA *asynchronous* FL protocol (Nguyen et al., 2022) and FedAvg in a non-i.i.d. experiment, in which the clients' data is created in a heterogeneous way. The experiment consists of 20 clients, of which 5 of them are slow ones, and the rest are fast ones. The task is CIFAR10 classification, in which slow clients have access to three specific classes and the rest have the other classes. FedAvg has to wait for the slow clients to perform all of their local steps, and therefore in the same amount of time as others, it performs a small number of global steps. FedBuff is asynchronous and
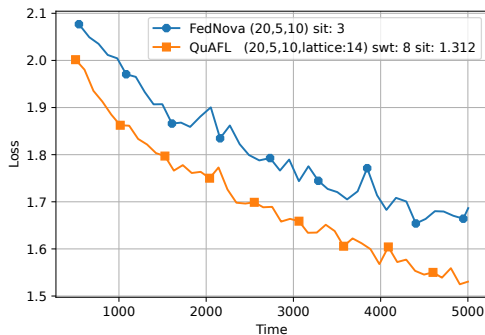
does not have to wait for slow clients, however, the fast clients contribute more often to the optimization, and the algorithm observes the seven classes from fast clients compared more frequently, relative to the three classes from slow clients. QuAFL on the other hand is asynchronous and balances the imbalance between different clients. As you can see in the figure, QuAFL outperforms both models in this comparison.

Figures 5 and 6 provide comparisons with FedBuff and FedNova, in a setup where both client speeds and client data are heterogenous. QuAFL provides faster convergence vs. time relative to FedBuff since it can handle heterogenous client speeds, whereas in FedBuff convergence is biased towards faster clients, leading to loss spikes; FedNova does not have this problem, but is *synchronous* and thus incurs slow-downs due to delays required for all clients to synchronize at every round. We present additional results in the Appendix, specifically higher node counts (up to 300), full-convergence experiments, as well as other tasks.

5 CONCLUSION AND LIMITATIONS

We have provided the first variant of FedAvg which incorporates both asynchronous and compressed communication, and have shown that this algorithm can still provide good convergence guarantees in a setting with heterogeneous data and client speeds. Our analysis is extensible to more complex federated optimizers, such as gradient tracking, e.g. (Haddadpour et al., 2021), controlled averaging (Karimireddy et al., 2020), heterogeneous clients (Diao et al., 2021) or variance-reduced variants (Gorbunov et al., 2021). In future work, we plan to investigate the limitation of our analysis in terms of the relationship between time T and the number of nodes n, applications to additional federated optimizers, as well as larger-scale deployments.

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. (2017). QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J. (2019). Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems*, volume 1, pages 374–388.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Cannelli, L., Facchinei, F., Kungurtsev, V., and Scutari, G. (2020). Asynchronous parallel algorithms for nonconvex optimization. *Mathematical Programming*, 184(1):121–154.

Chen, M., Shlezinger, N., Poor, H. V., Eldar, Y. C., and Cui, S. (2021). Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17).

Cutkosky, A. and Orabona, F. (2019). Momentum-based variance reduction in non-convex SGD. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Davies, P., Gurunanthan, V., Moshrefi, N., Ashkboos, S., and Alistarh, D. (2021). New bounds for distributed mean estimation and variance reduction. In *International Conference on Learning Representations (ICLR)*.

Diao, E., Ding, J., and Tarokh, V. (2021). HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations (ICLR)*.

Gorbunov, E., Burlachenko, K. P., Li, Z., and Richtárik, P. (2021). MARINA: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning (ICML)*.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. (2021). Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Haddadpour, F. and Mahdavi, M. (2019). On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jhunjhunwala, D., Gadhikar, A., Joshi, G., and Eldar, Y. C. (2021). Adaptive quantization of model updates for communication-efficient federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jin, R., Huang, Y., He, X., Dai, H., and Wu, T. (2020). Stochastic-sign SGD for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940*.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. (2019). Error feedback fixes signSGD and other gradient compression schemes. In *International Conference on Machine Learning (ICML)*.

Koloskova, A., Stich, S. U., and Jaggi, M. (2022). Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

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

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60.

Li, Z. and Richtárik, P. (2021). CANITA: Faster rates for distributed convex optimization with communication compression. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*.

Lu, Y. and De Sa, C. (2020). Moniqua: Modulo quantized communication in decentralized SGD. In *International Conference on Machine Learning (ICML)*.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Mishchenko, K., Bach, F., Even, M., and Woodworth, B. E. (2022). Asynchronous SGD beats minibatch SGD under arbitrary delays. In *Conference on Neural Information Processing Systems (NeurIPS)*.

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

Nadiradze, G., Sabour, A., Davies, P., Li, S., and Alistarh, D. (2021). Asynchronous decentralized SGD with quantized and local updates. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., and Huba, D. (2022). Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Philippenko, C. and Dieuleveut, A. (2020). Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees. *arXiv preprint arXiv:2006.14591*.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. (2021). Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. (2014). 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth annual conference of the international speech communication association*.

Stich, S. U. (2019). Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*.

Tong, Q., Liang, G., and Bi, J. (2020). Effective federated adaptive gradient methods with non-iid decentralized data. *arXiv preprint arXiv:2009.06557*.

Tyurin, A. and Richtárik, P. (2023). DASHA: Distributed nonconvex optimization with communication compression and optimal oracle complexity. In *International Conference on Learning Representations (ICLR)*.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Wang, Y., Lin, L., and Chen, J. (2022). Communication-efficient adaptive federated learning. In *International Conference on Machine Learning (ICML)*.

Wu, W., He, L., Lin, W., Mao, R., Maple, C., and Jarvis, S. (2020). SAFA: a semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Transactions on Computers*, 70(5):655–668.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

   (b) Complete proofs of all theoretical results. [Yes]

   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

    (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No]

    (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

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

    (a) Citations of the creator If your work uses existing assets. [Yes]

    (b) The license information of the assets, if applicable. [Not Applicable]

    (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

    (d) Information about consent from data providers/curators. [Not Applicable]

    (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

    (a) The full text of instructions given to participants and screenshots. [Not Applicable]

    (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

    (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Communication-Efficient Federated Learning With Data and Client Heterogeneity:
Supplementary Materials

A   Experimental setup

In this section, we describe our experimental setup in detail. We begin by defining the hyper-parameters which control the behavior of QuAFL and FedAvg. Then, we proceed by carefully describing the way in which we simulated each of the algorithms. Finally, we detail the datasets, tasks, and models used for our experiments. The experiments in this section are done with the non-weighted version of the QuAFL.

A.1   Hyper-parameters

We first define our hyper-parameters; in the later sections, we will examine their impact on algorithm behavior through ablation studies.

n: Number of the clients.

s: Number of clients interacting with the server at each step.

K: *In QuAFL*, this is the maximum number of allowed local steps by each client between two server calls. *In FedAvg*, this is the number of local steps performed by each client upon each server call.

b: Number of bits used to send a coordinate after quantization.

swt: Server waiting time, i.e. the amount of time that server waits between two consecutive calls.

sit: Server interaction time, i.e. the amount of time that server needs to send and receive necessary data (excluding computation time).

A.2   Simulation

We attempt to simulate a realistic FL deployment scenario, as follows. We assume a *server* and n *clients*, each of which initially has a model copy. The training dataset is distributed among the clients so that each of them has access to $1/n$ of the training data. We track the performance of each algorithm by evaluating the server's model, on an unseen validation dataset. We measure *loss* and *accuracy* of the model with respect to *simulation time*, *server steps*, and *total local steps performed by clients*. These setups so far were common between QuAFL and FedAvg. In the following, we are going to describe their specifications and differences.

QuAFL: Upon each server call, the server chooses s clients uniformly at random. It then sends its model to those clients and asks for their current local models. (Recall that clients send their model immediately to the server.) Each of the clients will have taken a maximum of K local steps by the time it is contacted by the server. The server then replaces its model with a carefully-computed average over the received models and its current model. This process increases time on the server by the *server interaction time (sit)*. The server then waits for another interval of *server waiting time (swt)* to make its next call. The s receiving clients replace their model with the weighted average between their current model and the received server's model. Since each client performs local steps from its last interaction time until the current server time, nodes are effectively executing asynchronously. Moreover, note that communication is compressed, as all the models get encoded in their source and decoded in their destination.

> **Quantization:** To have a lightweight but efficient communication between clients and the server, we use the well-known lattice quantization (Davies et al., 2021). Using this method, we send b instead of 32 bits for each scalar dimension. Informally, each 32-bit number maps to one of the 2^b quantized levels and can be sent using b bits only. The encoded number can then be decoded to a sufficiently close number at the destination, following the quantization protocol.

FedAvg: In the beginning of each round, server chooses s clients randomly, and sends its current model to them. Each of those clients receives the model, uncompressed, and performs exactly K local steps using this model as the starting point, and then sends back the resulting model to the server. The server then computes the average of the received models and adopts it as its model. By this synchronous structure, in each round, the server must wait for the *slowest* client to complete its local steps plus an extra *sit* for the communication time. After completing each round, the server starts the next call immediately, that means $swt = 0$ in FedAvg.

Timing Experiments. We differentiate between two types of timing experiments. *Uniform timing* experiments, presented in the paper body, assume all clients take the same amount of time for a gradient step. However, in real-world setups, different devices may require different amounts of time to perform a single local step. This is one of the main disadvantages of synchronous federated optimization algorithms. To demonstrate how this fact affects the experiments, in our *Non-uniform* timing experiments we differentiate clients to be either *fast* or *slow*. The length of each local step can be characterized as a memoryless time event. Therefore, the length of each local step can be defined by a random variable $X \sim exponential(\lambda)$. The parameter λ is $1/2$ for *fast clients* and $1/8$ for *slow* clients; the expected runtime $\mathbb{E}(X)$ would be 2 and 8, respectively. In each timing experiment, we assumed only one fourth of clients to be *slow*.

A.3   Datasets and Models

We used Pytorch to manage the training process in our algorithm. We have trained neural networks for image classification tasks on three well-known datasets, **MNIST**, **Fashion MNIST**, and **CIFAR-10**. For all the datasets, we used the default train/test split of the dataset for our training/validation dataset. In the following, we describe the model architecture and the training hyper-parameters used to train on each of these datasets.

MNIST: We used SGD optimizer with constant $lr = 0.5$ in all the training process. We used a two-layer MLP architecture with (784,32,10) nodes in its layers respectively. We used batch size 128 in each client's SGD step.

Fashion MNIST: Although this dataset has the same sample size and number of classes as **MNIST**, obtaining competitive performance on it requires a more complicated architecture. Therefore, we used a CNN model
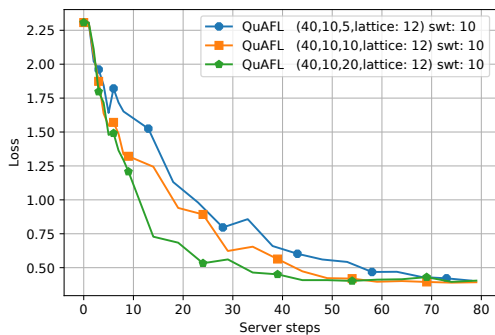
Hossein Zakerinia, Shayan Talaei, Giorgi Nadiradze, Dan Alistarh

Figure 7: Impact of the maximum number of local steps $K \in \{5, 10, 20\}$ on the QuAFL algorithm / Fashion MNIST.
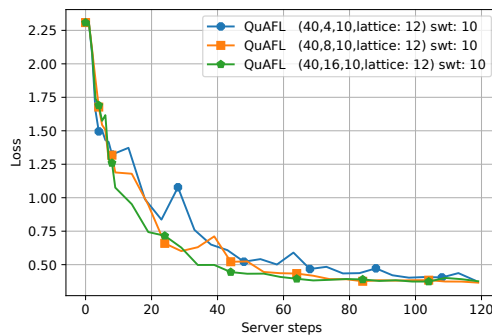

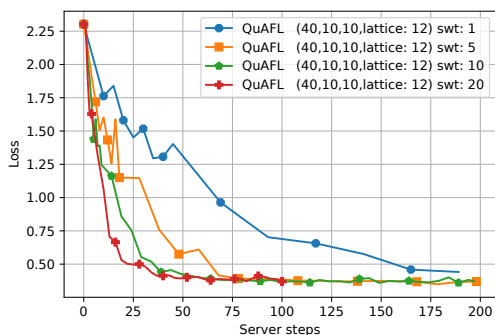**Figure 8:** Impact of the number of interacting peers $s \in \{4, 8, 16\}$ on the convergence of the algorithm.

Figure 9: Impact of the server contact frequency (controlled via server timeout swt) on the convergence of the algorithm.
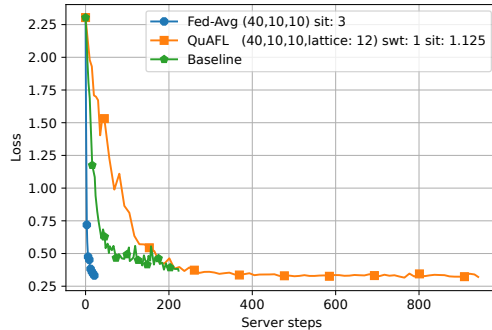

Figure 10: Convergence comparison relative to total number of rounds, between QuAFL, FedAvg, and the sequential baseline.

to train on this model and demonstrated the performance of our algorithm in a non-convex task. To optimize the models, we used Adam optimizer with constant $lr = 0.001$ and batch size 100.

CIFAR-10: To load this dataset, we used data augmentation and normalization. For this task, we trained ResNet20 models. Moreover, the SGD optimizer with constant $lr = 0.03$ is used to in the training process. The batch size $64/200$ is used for training/validation.

A.4 Results on Fashion MNIST (FMNIST)

We begin by validating our earlier results, presented in the paper body, for the slightly more complex FMNIST dataset, and on a convolutional model.

In Figures 7 and 8 we examine the impact of the parameters K and s, respectively, on the total number of interaction rounds at the server, to reach a certain training loss. As expected, we notice that higher K and s improve the convergence behavior of the algorithm. In Figures 9 we examine the impact of the server waiting time on the convergence of the algorithm relative to the number of server rounds. Again, we notice that a higher server waiting time improves convergence, as it allows the server to take advantage of additional local steps performed at the clients, as predicted by our analysis. (Higher swt means higher average number of steps completed H.)

Next, we examine the convergence, again in terms of number of optimization "rounds" at the server, between the sequential Baseline, FedAvg, and QuAFL. As expected, the Baseline is faster to converge than FedAvg, which in turn is faster than QuAFL in this measure. Specifically, the difference between QuAFL and the other algorithms comes because of the fact that, in our algorithm, nodes operate on *old* variants of the model at every step, which slows down convergence. Next, we examine convergence in terms of actual time, in the heterogeneous setting in which 25% of the clients are slow.

In Figure 11, we observe the validation accuracy ensured by various algorithms relative to the simulated execution time, whereas in Figure 12 we observe the training loss versus the same metric. (We assume that, in *Baseline*,
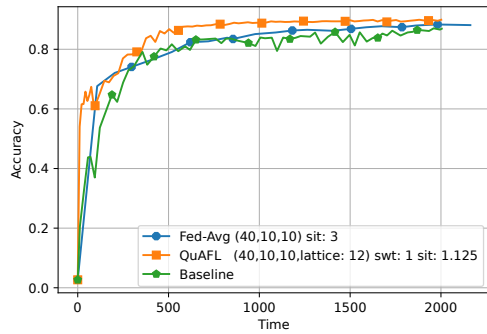
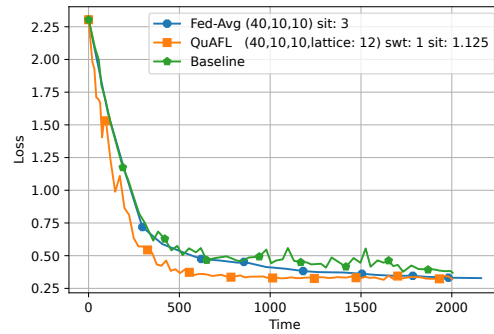**Figure 11:** Time vs. accuracy for various algorithm variants, on Fashion MNIST.

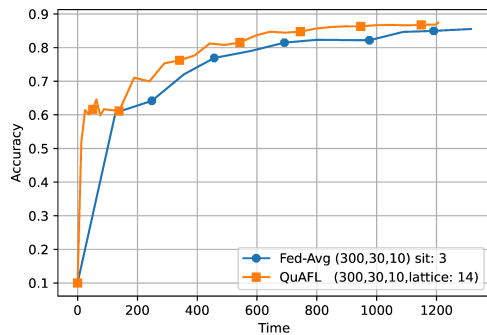**Figure 12:** Timing vs. loss for various algorithm variants, on Fashion MNIST.

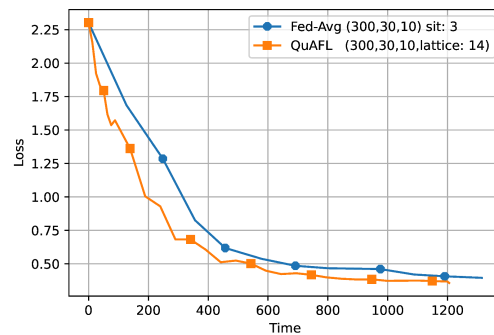**Figure 13:** Time vs. accuracy for n=300 clients, s=30 peers on Fashion MNIST.

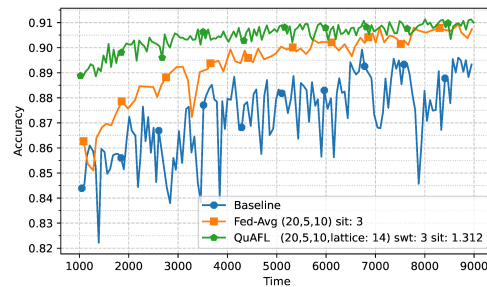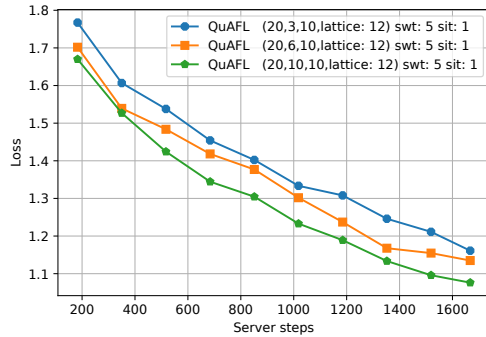**Figure 14:** Timing vs. loss for n=300 clients, s=30 peers on Fashion MNIST.

Figure 15: Full convergence result for $n = 20$ clients and $s = 5$ on F-MNIST. All methods eventually reach the sequential $\sim 91\%$ top-1 accuracy on this task, but QuAFL is the fastest to do so in terms of wall-clock time.
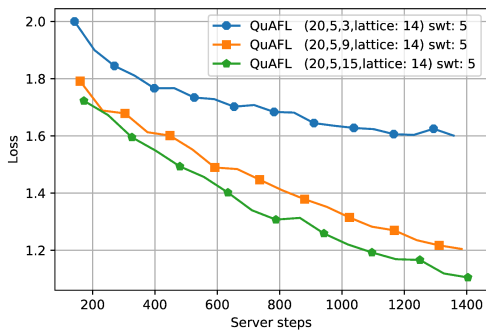
Figure 16: Impact of maximum local steps $K \in \{3, 9, 15\}$ on the QuAFL algorithm, on ResNet20/CIFAR-10.

Figure 17: Impact of the number of interacting peers $s \in \{3, 6, 10\}$ on the convergence of the algorithm.
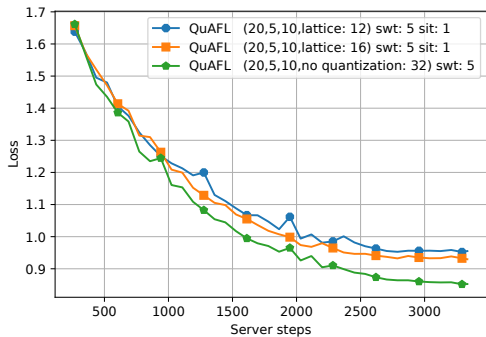
Figure 18: Impact of the number of bits for quantization $b \in \{12, 16, 32\}$ on the convergence of the algorithm.
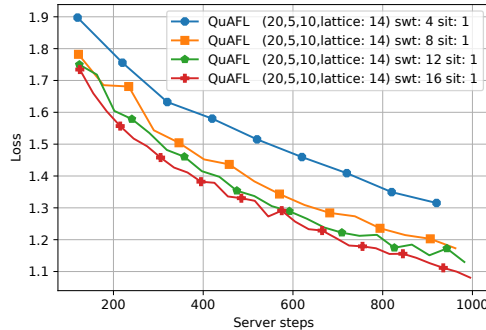
Figure 19: Impact of the server contact frequency (controlled via server timeout swt) on the convergence of the algorithm.

a single node acts as both the client and the server, and that this node is slow, i.e. has higher per-step times.) To further support the robustness of our algorithm in regimes with large number of clients, we conducted an experiment with $n = 300$ clients and $s = 30$ peers interacting with the server at each step. The validation accuracy and loss versus time regarding the mentioned experiment plotted in Figure 13 and Figure 14 respectively. We observe that, importantly, if *time* is taken into account rather than the number of server rounds, QuAFL can provide notable speedups in these metrics. This is specifically because of its asynchronous communication patters, which allow it to complete rounds faster, without having to always wait for the slow nodes to complete their local computation. While this behaviour is simulated, we believe that this reflects the algorithm's practical potential. Finally, Figure 15 shows that all methods can reach the maximum accuracy for this dataset/model combination (for the SGD baseline, this occurs later), although QuAFL is the fastest to do so in terms of wall-clock time.

A.5 Results on CIFAR-10

We now present results for a standard image classification task on the CIFAR-10 dataset, using a ResNet20 model (He et al., 2016).

Figures 16 and 17 show the decrease in training loss versus the number of server steps (or rounds) for different values of K and s respectively. As our theory suggests, increasing K and s leads to an improvement in the convergence rate of the system. Figure 18 demonstrates the impact of the number of quantization bits b, on the convergence behaviour of the algorithm. According to the definition of b, increasing the number of quantization bits improves the communication accuracy. Thus, as it can be seen in the graph, higher values of b enhance the convergence relative to the number of server steps. Finally, Figure 19 shows the impact of the server interaction frequency, again controlled via the timeout parameter swt, on the algorithm's convergence. It is apparent that a very high interaction frequency can slow the algorithm down, by not allowing it to take advantage of the clients' local steps.

In Figures 20 and 21, we examine the validation accuracy and loss, respectively, ensured by various algorithms versus the simulated execution time. (As in the F-MNIST experiments, we assumed the *Baseline* to be a single
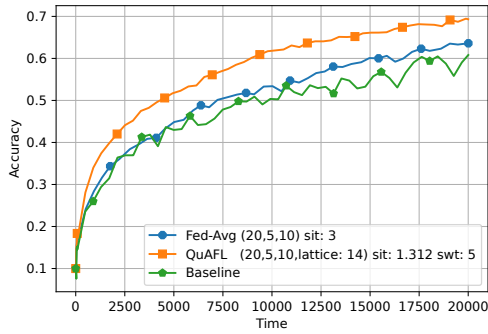
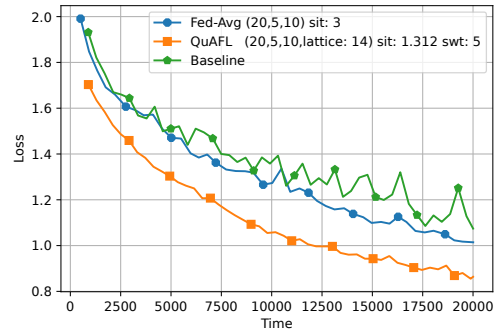**Figure 20:** Time vs. validation accuracy for various algorithm variants.

Figure 21: Timing versus validation loss for various algorithm variants.
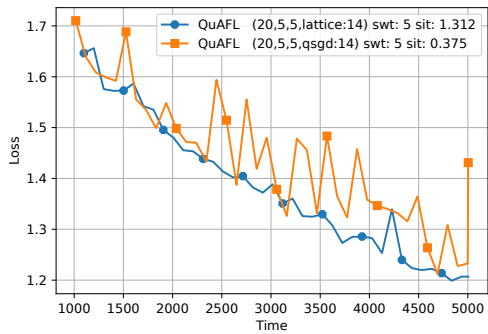

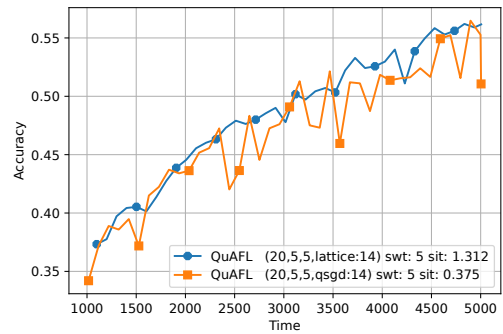**Figure 22:** Loss comparison of QuAFL with lattice quantization vs QSGD

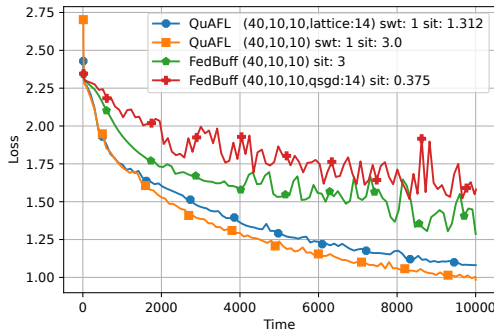**Figure 23:** Accuracy comparison of QuAFL with lattice quantization vs QSGD

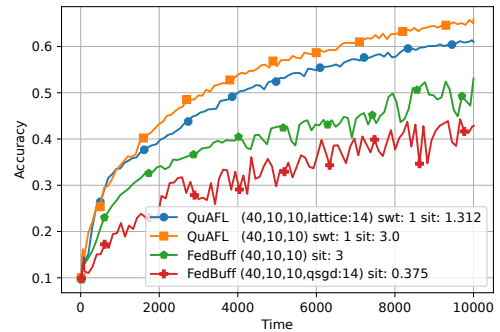**Figure 24:** QuAFL vs. SOTA asynchronous FL algorithm FedBuff with and without Quantization (ResNet20/CIFAR10).

Figure 25: Accuracy curves for Figure 24. Convergence of QuAFL vs. SOTA asynchronous FL algorithm FedBuff with and without Quantization.
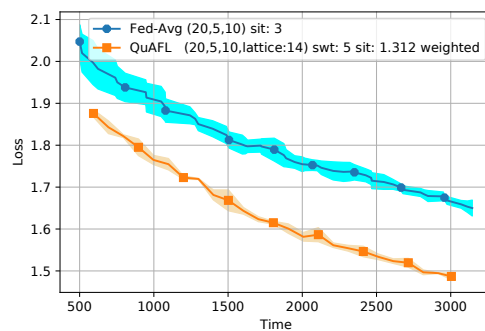

Figure 26: Comparison of QuAFL vs FedAVG on ResNet20/CIFAR10, with 5 slow clients and 15 fast clients with error bars.

slow node that performs an optimization step per round.) Again, the asynchronous nature of QuAFL provides a faster convergence rate than its synchronous counterparts; which can be clearly seen in the mentioned figures.

As discussed earlier, using standard quantization techniques such as QSGD, is not theoretically justified, because in that case, the error would be proportional to the norm of the models, which can be unbounded. Therefore, there is no guarantee that these quantization techniques will always work. To show this experimentally, we ran an experiment on CIFAR10 with lr = 0.07. As shown in Figure 22 and 23, QuAFL with lattice quantization works well in this setting, while QSGD quantization hurts the convergence. This experiment verifies our choice of lattice quantization.

In Figure 24, we compare QuAFL convergence (with and without quantization) relative to FedBuff, in terms of execution time. We have tuned each variant independently, and have also added QSGD quantization to FedBuff. (FedBuff is incompatible with lattice quantization, since nodes do not have a decoding key.) We observe that QuAFL converges faster, even with quantization: this is because QuAFL takes into account *partial progress by slow clients,* whereas in FedBuff slower clients constantly contribute less significantly to the server updates. Moreover, we observe that quantization significantly increases the variance of FedBuff.

Figure 25 shows the accuracy comparison of the experiment of Figure 24. This figure shows that the accuracy follows the same pattern as the loss. Finally, as we mentioned before we omitted error bars in our experiments as the variance is low. We show one experiment in Figure 26 with error bars to confirm this.

B The Complete Analysis

B.1 Overview and Notation

Recall that X_t denotes the model of the server at step t, and X^i is the local model of client i after its last interaction with the server. Also, $\widetilde{h}_i$ is the sum of local gradient steps for model X^i since its last interaction with the server.

For the convergence analysis, local steps of the clients that are not selected by the server don't have any effect on the server or other clients. Therefore we do not need to assume that clients are doing their local steps asynchronous, and we can assume that all clients run their local gradient steps after the server contacts them. The only thing that we should consider is the randomness of the server selecting the clients, and the fact that the server can contact nodes before they have finished their K steps. For this purpose, we assume that their number of steps is a random number H_t^i with mean H_i.

To show the analysis in this setting, we introduce new notations that consider the server round. To this end, we use X_t^i as the value of X^i when the server is running its tth iteration, And $\widetilde{h}_{i,t}$ for the sum of local steps at this time. We show each local step q with a superscript. Formally, we have

$$\widetilde{h}_{i,t}^0 = 0.$$

and for $1 \leq q \leq H_t^i$ let:

$$\widetilde{h}_{i,t}^q = \widetilde{g}_i(X_t^i - \sum_{s=0}^{q-1} \eta \widetilde{h}_{i,t}^s),$$

and

$$\widetilde{h}_{i,t} = \sum_{q=0}^{H_t^i} \widetilde{h}_{i,t}^q$$

Further , for $1 \leq q \leq H_t^i$, let

$$h_{i,t}^q = \mathbb{E}[\widetilde{g}_i(X_t^i - \sum_{s=0}^{q-1} \eta \widetilde{h}_{i,t}^s)] = \nabla f(X_t^i - \sum_{s=0}^{q-1} \eta \widetilde{h}_{i,t}^s)$$

be the expected value of $\widetilde{h}_{i,t}^q$ taken over the randomness of the stochastic gradient $\widetilde{g}_i$. Also, we have:

$$h_{i,t} = \sum_{q=0}^{H_t^i} h_{i,t}^q$$

B.2   Properties of Local Steps

Lemma B.1. *For any agent i and step t*

$$\mathbb{E}\|h_{i,t}^q\|^2 \leq \frac{\sigma^2}{K^2} + 8L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\mathbb{E}\|\nabla f_i(\mu_t)\|^2.$$

Proof.

$$
\begin{aligned}
\mathbb{E}\|h_{i,t}^q\|^2 &\leq \mathbb{E}\left\|\left(\nabla f_i(X_t^i - \sum_{s=0}^{q-1}\eta\widetilde{h}_{i,t}^s) - \nabla f_i(\mu_t)\right) + \nabla f_i(\mu_t)\right\|^2 \\
&\leq 2\mathbb{E}\left\|\nabla f_i(X_t^i - \sum_{s=0}^{q-1}\eta\widetilde{h}_{i,t}^s) - \nabla f_i(\mu_t)\right\|^2 + 2\mathbb{E}\|\nabla f_i(\mu_t)\|^2 \\
&\leq 4L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\eta^2 L^2 q\sum_{s=0}^{q-1}\mathbb{E}\|\widetilde{h}_{i,t}^s\|^2 + 2\mathbb{E}\|\nabla f_i(\mu_t)\|^2 \\
&\leq 4L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\eta^2 L^2 q\sum_{s=0}^{q-1}(\mathbb{E}\|h_{i,t}^s\|^2 + \sigma^2) + 2\mathbb{E}\|\nabla f_i(\mu_t)\|^2
\end{aligned}
$$

the rest of the proof is done by induction, and assuming $\eta < \frac{1}{4LK^2}$. $\qquad\square$

Lemma B.2. *For any step t, we have that*

$$\sum_{i=1}^n \eta_i^2 \mathbb{E}\|\widetilde{h}_{i,t}\|^2 \leq 2nK\left(\left(\frac{1}{n}\sum_{i=1}^n \eta_i^2\right)\sigma^2 + 2KG^2\right) + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2.$$

Proof. Using lemma B.1

$$
\begin{aligned}
\sum_{i=1}^n \eta_i^2 \mathbb{E}\|\widetilde{h}_{i,t}\|^2 &= \sum_{i=1}^n \eta_i^2 \sum_{h=0}^K Pr[H_t^i = h]\mathbb{E}\|\sum_{q=1}^h \widetilde{h}_{i,t}^q\|^2 \\
&\leq \sum_{i=1}^n \eta_i^2 \sum_{h=1}^K Pr[H_t^i = h]h\sum_{q=1}^h \mathbb{E}\|\widetilde{h}_{i,t}^q\|^2 \\
&\leq nK\left(\frac{1}{n}\sum_{i=1}^n \eta_i^2\right)\sigma^2 + \sum_{i=1}^n \eta_i^2 \sum_{h=1}^K Pr[H_t^i = h]h\sum_{q=1}^h \mathbb{E}\|h_{i,t}^q\|^2 \\
&\leq nK\left(\frac{1}{n}\sum_{i=1}^n \eta_i^2\right)\sigma^2 + \sum_{i=1}^n \eta_i^2 K^2\left(\frac{\sigma^2}{K^2} + 8L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\mathbb{E}\|\nabla f_i(\mu_t)\|^2\right) \\
&\leq 2nK\left(\frac{1}{n}\sum_{i=1}^n \eta_i^2\right)\sigma^2 + \sum_{i=1}^n \eta_i^2 K^2\left(8L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\mathbb{E}\|\nabla f_i(\mu_t)\|^2\right) \\
&\leq 2nK\left(\frac{1}{n}\sum_{i=1}^n \eta_i^2\right)\sigma^2 + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2G^2 + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2.
\end{aligned}
$$

$\qquad\square$

Lemma B.3. *For any local step $1 \leq q$, and agent $1 \leq i \leq n$ and step t*

$$\mathbb{E}\|\nabla f_i(\mu_t) - h_{i,t}^q\|^2 \leq 4L^2\eta^2 q^2\sigma^2 + 4L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 8L^2\eta^2 q^2\mathbb{E}\|\nabla f_i(\mu_t)\|^2.$$

Proof.

$$\mathbb{E}\|\nabla f_i(\mu_t) - h_{i,t}^q\|^2 = \mathbb{E}\|\nabla f_i(\mu_t) - \nabla f_i(X_t^i - \sum_{s=0}^{q-1} \eta \widetilde{h}_{i,t}^s)\|^2$$

$$\leq L^2 \mathbb{E}\|\mu_t - X_t^i + \sum_{s=0}^{q-1} \eta \widetilde{h}_{i,t}^s\|^2$$

$$\leq 2L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 2L^2 \eta^2 \mathbb{E}\|\sum_{s=0}^{q-1} \widetilde{h}_{i,t}^s\|^2$$

$$\leq 2L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 2L^2 \eta^2 q \sum_{s=0}^{q-1} \mathbb{E}\|\widetilde{h}_{i,t}^s\|^2$$

$$\overset{\text{Lemma (B.1)}}{\leq} 2L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 2L^2 \eta^2 q^2 \Big(2\sigma^2 + 8L^2 \mathbb{E}\|X_t^i - \mu_t\|^2$$

$$+ 4\mathbb{E}\|\nabla f_i(\mu_t)\|^2\Big)$$

$$= 4L^2 \eta^2 q^2 \sigma^2 + (2L^2 + 16L^4 \eta^2 q^2)\mathbb{E}\|X_t^i - \mu_t\|^2$$

$$+ 8L^2 \eta^2 q^2 \mathbb{E}\|\nabla f_i(\mu_t)\|^2$$

$$\leq 4L^2 \eta^2 q^2 \sigma^2 + 4L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 8L^2 \eta^2 q^2 \mathbb{E}\|\nabla f_i(\mu_t)\|^2$$

and the last inequality comes from $\eta < \frac{1}{4LK}$. □

Lemma B.4. *For any time step t*

$$\sum_{i=1}^n \mathbb{E}\langle \nabla f(\mu_t), -\eta_i h_{i,t}\rangle \leq 4KL^2 \mathbb{E}[\Phi_t] + (-\frac{3H_{\min}n}{4} + 8B^2 L^2 \eta^2 K^3 n)\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$+ 4nL^2 \eta^2 K^3 (\sigma^2 + 2G^2).$$

Proof.

$$\sum_{i=1}^n \mathbb{E}\langle \nabla f(\mu_t), -\eta_i h_{i,t}\rangle = \sum_{i=1}^n \sum_{h=1}^K Pr[H_t^i = h]\mathbb{E}\langle \nabla f(\mu_t), -\eta_i \sum_{q=1}^h h_{i,t}^q\rangle + \sum_{i=1}^n Pr[H_t^i = 0]\mathbb{E}\langle \nabla f(\mu_t), 0\rangle$$

$$= \sum_{i=1}^n \eta_i \sum_{h=1}^K Pr[H_t^i = h] \sum_{q=1}^h \Big(\mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t) - h_{i,t}^q\rangle - \mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle\Big)$$

Using Young's inequality we can upper bound $\mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t) - h_{i,t}^q\rangle$ by

$$\frac{\mathbb{E}\|\nabla f(\mu_t)\|^2}{4} + \mathbb{E}\left\|\nabla f_i(\mu_t) - h_{i,t}^q\right\|^2.$$

Plugging this in the above inequality we get:

$$\sum_{i=1}^{n} \mathbb{E}\langle \nabla f(\mu_t), -\eta_i h_{i,t}\rangle$$

$$\leq \sum_{i=1}^{n} \eta_i \sum_{h=1}^{K} Pr[H_t^i = h] \sum_{q=1}^{h} \left(\mathbb{E}\|\nabla f(\mu_t) - h_{i,t}^q\|^2 + \frac{\mathbb{E}\|\nabla f(\mu_t)\|^2}{4} - \mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle \right)$$

$$\overset{\text{Lemma B.3}}{\leq} \sum_{i=1}^{n} \eta_i \sum_{h=1}^{K} Pr[H_t^i = h] \sum_{q=1}^{h} \left(4L^2\eta^2 q^2 \sigma^2 + 4L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 8L^2\eta^2 q^2 \mathbb{E}\|\nabla f_i(\mu_t)\|^2 \right.$$

$$\left. + \frac{\mathbb{E}\|\nabla f(\mu_t)\|^2}{4} - \mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle \right)$$

$$\leq \sum_{i=1}^{n} \eta_i \sum_{h=1}^{K} Pr[H_t^i = h] h \left(4L^2\eta^2 h^2 \sigma^2 + 4L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 8L^2\eta^2 h^2 \mathbb{E}\|\nabla f_i(\mu_t)\|^2 \right.$$

$$\left. + \frac{\mathbb{E}\|\nabla f(\mu_t)\|^2}{4} - \mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle \right)$$

$$\leq \sum_{i=1}^{n} \eta_i H_i \left(4L^2\eta^2 K^2 \sigma^2 + 4L^2 \mathbb{E}\|X_t^i - \mu_t\|^2 + 8L^2\eta^2 K^2 \mathbb{E}\|\nabla f_i(\mu_t)\|^2 \right.$$

$$\left. + \frac{\mathbb{E}\|\nabla f(\mu_t)\|^2}{4} - \mathbb{E}\langle \nabla f(\mu_t), \nabla f_i(\mu_t)\rangle \right)$$

$$\leq 4KL^2 \mathbb{E}[\Phi_t] + 4nL^2\eta^2 K^3 ((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2G^2) + (8B^2 nL^2\eta^2 K^3 + \frac{H_{\min}n}{4} - H_{\min}n)\mathbb{E}\|\nabla f(\mu_t)\|^2)$$

Where in the last step we used that for each i, $\eta_i H_i = H_{\min}$, and $\sum_{i=1}^{n} \frac{f_i(x)}{n} = f(x)$, for any vector x. $\qquad\square$

Discussion. In the analysis, especially in this lemma, the value of $\eta_i H_i$s should be the same for all clients. However, we don't need them to be fixed in different rounds. Therefore, it is possible to extend the result to the case that clients' speeds would change, and they have different expected numbers of local steps in different rounds. Accordingly, the algorithm sets different values for η_i^ts. The final result would depend on the average number of local steps of each client during the whole training.

B.3 Upper Bounding Potential Functions

We proceed by proving the lemma 3.4 which upper bounds the expected change in potential:

Lemma 3.4. For any time step t we have:

$$\mathbb{E}[\Phi_{t+1}] \leq \left(1 - \frac{1}{4n}\right)\mathbb{E}[\Phi_t] + 8s\eta^2 \sum_{i=1}^{n} \mathbb{E}\|\widetilde{h}_{i,t}\|^2 + 16n(R^2+7)^2\gamma^2.$$

Proof. First we bound change in potential $\Delta_t = \Phi_{t+1} - \Phi_t$ for some fixed time step $t > 0$.

For this, let Δ_t^S be the change in potential when set S of agents wake up. for $i \in S$ define S_t^i and S_t as follows:

$$S_t^i = -\frac{s}{s+1}\eta\eta_i\widetilde{h}_{i,t} + \frac{Q(X_t) - X_t}{s+1}$$

$$S_t = -\frac{1}{s+1}\eta\eta_i \sum_{i \in S} \widetilde{h}_{i,t} + \frac{1}{s+1}\sum_{i \in S}(Q(X_t^i - \eta\eta_i\widetilde{h}_{i,t}) - (X_t^i - \eta\eta_i\widetilde{h}_{i,t}))$$

We have that:

$$X_{t+1}^i = \frac{sX_t^i + X_t}{s+1} + S_t^i$$

$$X_{t+1} = \frac{\sum_{i \in S} X_t^i + X_t}{s+1} + S_t$$

$$\mu_{t+1} = \mu_t + \frac{\sum_{j \in S} S_t^j + S_t}{n+1}$$

This gives us that for $i \in S$:

$$X_{t+1}^i - \mu_{t+1} = \frac{sX_t^i + X_t}{s+1} + S_t^i - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t$$

$$X_{t+1} - \mu_{t+1} = \frac{\sum_{i \in S} X_t^i + X_t}{s+1} + S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t$$

For $k \notin S$ we get that

$$X_{t+1}^k - \mu_{t+1} = X_t^k - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t.$$

Hence:

$$\begin{aligned}
\Delta_t^S &= \sum_{i \in S} \left(\left\| \frac{sX_t^i + X_t}{s+1} + S_t^i - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t \right\|^2 - \left\| X_t^i - \mu_t \right\|^2 \right) \\
&\quad + \left\| \frac{\sum_{i \in S} X_t^i + X_t}{s+1} + S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t \right\|^2 - \left\| X_t - \mu_t \right\|^2 \\
&\quad + \sum_{k \notin S} \left(\left\| X_t^k - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} - \mu_t \right\|^2 - \left\| X_t^k - \mu_t \right\|^2 \right) \\
&= \sum_{i \in S} \left(\left\| \frac{sX_t^i + X_t}{s+1} - \mu_t \right\|^2 + \left\| S_t^i + \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 \right. \\
&\quad + 2\left\langle \frac{sX_t^i + X_t}{s+1} - \mu_t, S_t^i - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\rangle - \left\| X_t^i - \mu_t \right\|^2 \bigg) \\
&\quad + \left(\left\| \frac{\sum_{i \in S} X_t^i + X_t}{s+1} - \mu_t \right\|^2 + \left\| S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 \right. \\
&\quad + 2\left\langle \frac{\sum_{i \in S} X_t^i + X_t}{s+1} - \mu_t, S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\rangle - \left\| X_t - \mu_t \right\|^2 \bigg) \\
&\quad + \sum_{k \notin S} 2\left\langle X_t^k - \mu_t, -\frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\rangle + \sum_{k \notin S} \left\| \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2
\end{aligned}$$

Observe that:

$$\sum_{k=0}^{n} \left\langle X_t^k - \mu_t, -\frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\rangle = 0.$$

After combining the above two equations, we get that:

$$
\begin{aligned}
\Delta_t^S &= \sum_{i \in S} \left(\left\| \frac{s(X_t^i - \mu_t) + (X_t - \mu_t)}{s+1} \right\|^2 - \frac{s}{s+1} \left\| X_t^i - \mu_t \right\|^2 - \frac{1}{s+1} \left\| X_t - \mu_t \right\|^2 \right) \\
&+ \left(\left\| \frac{\sum_{i \in S}(X_t^i - \mu_t) + (X_t - \mu_t)}{s+1} \right\|^2 - \sum_{i \in S} \frac{1}{s+1} \left\| X_t^i - \mu_t \right\|^2 - \frac{1}{s+1} \left\| X_t - \mu_t \right\|^2 \right) \\
&+ \sum_{i \in S} \left(\left\| S_t^i - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 + 2 \left\langle \frac{sX_t^i + X_t}{s+1} - \mu_t, S_t^i \right\rangle \right) \\
&+ \left\| S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 + 2 \left\langle \frac{\sum_{i \in S} X_t^i + X_t}{s+1} - \mu_t, S_t \right\rangle \\
&+ \sum_{k \notin S} \left\| \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2
\end{aligned}
$$

By simplifying the above, we get:

$$
\begin{aligned}
\Delta_t^S &= \frac{-s}{(s+1)^2} \sum_{i \in S} \| X_t^i - X_t \|^2 - \frac{1}{(s+1)^2} \sum_{i \in S} \| X_t^i - X_t \|^2 - \frac{1}{(s+1)^2} \sum_{i,j \in S} \| X_t^i - X_t^j \|^2) \\
&+ \sum_{i \in S} \left\| S_t^i - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 + \frac{2s}{s+1} \sum_{i \in S} \left\langle X_t^i - \mu_t, S_t^i \right\rangle + \frac{2}{s+1} \sum_{i \in S} \left\langle X_t - \mu_t, S_t^i \right\rangle \\
&+ \left\| S_t - \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2 + \frac{2}{s+1} \sum_{i \in S} \left\langle X_t^i - \mu_t, S_t \right\rangle + \frac{2}{s+1} \left\langle X_t - \mu_t, S_t \right\rangle \\
&+ \sum_{k \notin S} \left\| \frac{\sum_{j \in S} S_t^j + S_t}{n+1} \right\|^2
\end{aligned}
$$

Let α be a parameter we will fix later:

$$
\left\langle X_t^i - \mu_t, S_t^i \right\rangle \overset{\text{Young}}{\leq} \alpha \left\| X_t^i - \mu_t \right\|^2 + \frac{\left\| S_t^i \right\|^2}{4\alpha}
$$

Finally, we get that

$$
\begin{aligned}
\Delta_t^S \leq{} & \frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + 2 \sum_{i \in S} \left\|S_t^i\right\|^2 + \frac{2s(s+1)}{(n+1)^2} \sum_{j \in S} \left\|S_t^j\right\|^2 + \frac{2s(s+1)}{(n+1)^2} \left\|S_t\right\|^2 \\
& + \sum_{i \in S} \frac{2s\alpha}{s+1} \left\|X_t^i - \mu_t\right\|^2 + \sum_{i \in S} \frac{s\left\|S_t^i\right\|^2}{2\alpha(s+1)} + \sum_{i \in S} \frac{2\alpha}{s+1} \|X_t - \mu_t\|^2 + \sum_{i \in S} \frac{\left\|S_t^i\right\|^2}{2\alpha(s+1)} \\
& + 2\left\|S_t\right\|^2 + \frac{2(s+1)}{(n+1)^2} \sum_{j \in S} \left\|S_t^j\right\|^2 + \frac{2(s+1)}{(n+1)^2} \left\|S_t\right\|^2 + \sum_{i \in S} \frac{2\alpha}{s+1} \left\|X_t^i - \mu_t\right\|^2 + \sum_{i \in S} \frac{\left\|S_t\right\|^2}{2\alpha(s+1)} \\
& + \frac{2\alpha}{s+1} \|X_t - \mu_t\|^2 + \frac{\left\|S_t\right\|^2}{2\alpha(s+1)} + \sum_{j \in S} \frac{(n-s)(s+1)}{(n+1)^2} \left\|S_t^j\right\|^2 + \frac{(n-s)(s+1)}{(n+1)^2} \left\|S_t\right\|^2 \\
={} & \frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + (2 + \frac{2(s+1)^2}{(n+1)^2} + \frac{1}{2\alpha} + \frac{(n-s)(s+1)}{(n+1)^2}) \sum_{j \in S} \left\|S_t^j\right\|^2 + \\
& (2 + \frac{2(s+1)^2}{(n+1)^2} + \frac{1}{2\alpha} + \frac{(n-s)(s+1)}{(n+1)^2}) \left\|S_t\right\|^2 + \sum_{i \in S} 2\alpha \left\|X_t^i - \mu_t\right\|^2 + 2\alpha \left\|X_t - \mu_t\right\|^2 \\
\leq{} & \frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + (4 + \frac{1}{2\alpha}) \sum_{i \in S} \left\|S_t^i\right\|^2 + \\
& (4 + \frac{1}{2\alpha}) \left\|S_t\right\|^2 + \sum_{i \in S} 2\alpha \left\|X_t^i - \mu_t\right\|^2 + 2\alpha \left\|X_t - \mu_t\right\|^2
\end{aligned}
$$

Using definitions of S_t^i and S_t, Cauchy-Schwarz inequality and properties of quantization we get that

$$
\begin{aligned}
\|S_t^i\|^2 &\leq \frac{2s^2}{(s+1)^2} \eta^2 \eta_i^2 \|\widetilde{h}_{i,t}\|^2 + \frac{2(R^2+7)^2\gamma^2}{(s+1)^2}. \\
\|S_t\|^2 &\leq \frac{2s}{(s+1)^2} \eta^2 \eta_i^2 \sum_{i \in S} \|\widetilde{h}_{i,t}\|^2 + \frac{2s^2(R^2+7)^2\gamma^2}{(s+1)^2}
\end{aligned}
$$

Next, we plug this in the previous inequality:

$$
\begin{aligned}
\Delta_t^S \leq{} & \frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + \sum_{i \in S} 2\alpha \left\|X_t^i - \mu_t\right\|^2 + 2\alpha \left\|X_t - \mu_t\right\|^2 \\
& + (4 + \frac{1}{2\alpha}) \frac{2s^2 + 2s}{(s+1)^2} \eta^2 \eta_i^2 \|\widetilde{h}_{i,t}\|^2 + \frac{(2s^2 + 2s)(R^2+7)^2\gamma^2}{(s+1)^2} \\
\leq{} & \frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + \sum_{i \in S} 2\alpha \left\|X_t^i - \mu_t\right\|^2 + 2\alpha \left\|X_t - \mu_t\right\|^2 \\
& + (4 + \frac{1}{2\alpha})(\eta^2 \sum_{i \in S} \eta_i^2 \|\widetilde{h}_{i,t}\|^2 + 2(R^2+7)^2\gamma^2)
\end{aligned}
$$

Next, we calculate the probability of choosing the set S and upper bound Δ_t in expectation, for this we define

$\mathbb{E}_t$ as expectation conditioned on the entire history up to and including step t

$$\mathbb{E}_t[\Delta_t] = \sum_S \frac{1}{\binom{n}{s}} \mathbb{E}_t[\Delta_t^S]$$

$$\leq \sum_S \frac{1}{\binom{n}{s}} \left(\frac{-1}{s+1} \sum_{i \in S} \|X_t^i - X_t\|^2 + \sum_{i \in S} 2\alpha \left\| X_t^i - \mu_t \right\|^2 + 2\alpha \left\| X_t - \mu_t \right\|^2 \right.$$

$$\left. + (4 + \frac{1}{2\alpha})(\eta^2 \sum_{i \in S} \eta_i^2 \|\widetilde{h}_{i,t}\|^2 + 2(R^2 + 7)^2 \gamma^2) \right)$$

$$= \frac{-\binom{n-1}{s-1}}{(s+1)\binom{n}{s}} \sum_i \|X_t^i - X_t\|^2 + \sum_i \frac{2\alpha \binom{n-1}{s-1}}{\binom{n}{s}} \left\| X_t^i - \mu_t \right\|^2 + 2\alpha \left\| X_t - \mu_t \right\|^2$$

$$+ (4 + \frac{1}{2\alpha})(\eta^2 \frac{\binom{n-1}{s-1}}{\binom{n}{s}} \sum_i \eta_i^2 \|\widetilde{h}_{i,t}\|^2 + 2(R^2 + 7)^2 \gamma^2)$$

$$\leq -\sum_i \frac{s\|X_t^i - \mu_t\|^2}{(s+1)n} + \sum_i 2\frac{s\alpha}{n} \left\| X_t^i - \mu_t \right\|^2 + 2\alpha \left\| X_t - \mu_t \right\|^2$$

$$+ (8 + \frac{1}{\alpha})(R^2 + 7)^2 \gamma^2 + \sum_i \frac{s}{n}(4 + \frac{1}{2\alpha})\eta^2 \eta_i^2 \mathbb{E}_t \|\widetilde{h}_{i,t}\|^2$$

$$\leq (\frac{-s}{(s+1)n} + 2\alpha)\Phi_t + (8 + \frac{1}{\alpha})(R^2 + 7)^2 \gamma^2 + \sum_i \frac{s}{n}(4 + \frac{1}{2\alpha})\eta^2 \eta_i^2 \mathbb{E}_t \|\widetilde{h}_{i,t}\|^2$$

By setting $\alpha = \frac{3s-1}{n(8s+8)} \geq \frac{1}{8n}$, we get that:

$$\mathbb{E}_t[\Delta_t] \leq -\frac{1}{4n}\Phi_t + 16n(R^2 + 7)^2 \gamma^2 + \sum_i 8s\eta^2 \eta_i^2 \mathbb{E}_t \|\widetilde{h}_{i,t}\|^2.$$

Next we remove the conditioning , and use the definitions of Δ_i and S_t^i (for S_t^i we also use upper bound which come from the properties of quantization).

$$\mathbb{E}[\mathbb{E}_t[\Phi_{t+1}]] = \mathbb{E}[\Delta_t + \Phi_t] \leq (1 - \frac{1}{4n})\mathbb{E}[\Phi_t] + 16n(R^2 + 7)^2 \gamma^2 + 8s\eta^2 \sum_i \eta_i^2 \mathbb{E}\|\widetilde{h}_{i,t}\|^2$$

$\square$

Lemma B.5. *For any time step t we have:*

$$\mathbb{E}[\Phi_{t+1}] \leq (1 - \frac{1}{5n})\mathbb{E}[\Phi_t] + 16n(R^2 + 7)^2 \gamma^2 + 16nsK\eta^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + 32B^2 nsK^2\eta^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof. By combining Lemma 3.4 and B.2 we have:

$$\mathbb{E}[\Phi_{t+1}] \leq (1 - \frac{1}{4n})\mathbb{E}[\Phi_t] + 16n(R^2 + 7)^2 \gamma^2 +$$

$$8s\eta^2 \left(2nK((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + 8L^2 K^2 \mathbb{E}[\Phi_t] + 4nK^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2 \right)$$

$$= (1 - \frac{1}{4n} + 64sL^2 K^2 \eta^2)\mathbb{E}[\Phi_t] + 16n(R^2 + 7)^2 \gamma^2 +$$

$$16nsK\eta^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + 32B^2 nsK^2\eta^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq (1 - \frac{1}{5n})\mathbb{E}[\Phi_t] + 16n(R^2 + 7)^2 \gamma^2 +$$

$$16nsK\eta^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + 32B^2 nsK^2\eta^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

□

Lemma B.6. *For the sum of potential functions in all T steps we have:*

$$\sum_{t=0}^{T}\mathbb{E}[\Phi_t] \leq 80Tn^2(R^2+7)^2\gamma^2 + 80Tn^2sK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + 160B^2n^2sK^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof.

$$\sum_{t=0}^{T-1}\mathbb{E}[\Phi_{t+1}] \leq \sum_{t=0}^{T-1}\left((1-\frac{1}{5n})\mathbb{E}[\Phi_t] + 16n(R^2+7)^2\gamma^2 + \right.$$

$$\left. 16nsK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + 32B^2nsK^2\eta^2\mathbb{E}\|\nabla f(\mu_t)\|^2\right)$$

$$\leq (1-\frac{1}{5n})\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 16Tn(R^2+7)^2\gamma^2 + 16TnsK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 32B^2nsK^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\sum_{t=0}^{T}\mathbb{E}[\Phi_t] \leq 5n\big(16Tn(R^2+7)^2\gamma^2 + 16TnsK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 32B^2nsK^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2\big)$$

$$= 80Tn^2(R^2+7)^2\gamma^2 + 80Tn^2sK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 160B^2n^2sK^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

□

Lemma B.7. *For any step t, we have that*

$$\sum_{i=1}^{n}\eta_i^2\mathbb{E}\|\sum_{s=0}^{k-1}\widetilde{h}_{i,t}^s\|^2 \leq 2nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2G^2 + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof. Using lemma B.1

$$\sum_{i=1}^{n}\eta_i^2\mathbb{E}\|\sum_{s=0}^{k-1}\widetilde{h}_{i,t}^s\|^2 \leq nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + \sum_{i=1}^{n}\eta_i^2\mathbb{E}\|\sum_{q=0}^{k-1}h_{i,t}^q\|^2$$

$$\leq nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + K\sum_{i=1}^{n}\eta_i^2\sum_{q=0}^{k-1}\mathbb{E}\|h_{i,t}^q\|^2$$

$$\leq nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + K\sum_{i=1}^{n}\eta_i^2\sum_{q=0}^{k-1}\left(\frac{\eta_i^2\sigma^2}{K^2} + 8L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4\mathbb{E}\|\nabla f_i(\mu_t)\|^2\right)$$

$$\leq 2nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + \sum_{i=1}^{n}\left(8K^2L^2\mathbb{E}\|X_t^i - \mu_t\|^2 + 4K^2\mathbb{E}\|\nabla f_i(\mu_t)\|^2\right)$$

$$\leq 2nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2G^2 + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2.$$

□

For the following lemmas we define parameter $\zeta_{i,k}$ as it is equal to 1 if the client i did its kth step and otherwise is Zero.

Lemma B.8. *For any step t*

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}\nabla f_i(X_t^i - \sum_{s=0}^{k-1}\eta\widetilde{h}_{i,t}^s) - \sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}\nabla f_i(X_t^i)\|^2$$
$$\leq K((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{K^2L^2}{n}\mathbb{E}[\Phi_t] + 2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2.$$

Proof.

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}\nabla f_i(X_t^i - \sum_{s=0}^{k-1}\eta\widetilde{h}_{i,t}^s) - \sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}\nabla f_i(X_t^i)\|^2$$

$$\leq \mathbb{E}\Big[sK\sum_{i\in S}\eta_i^2\sum_{k=0}^{K}\zeta_{i,k}^2\|\nabla f_i(X_t^i - \sum_{s=0}^{k-1}\eta\widetilde{h}_{i,t}^s) - \nabla f_i(X_t^i)\|^2\Big]$$

$$\leq \mathbb{E}\Big[sKL^2\sum_{i\in S}\eta_i^2\sum_{k=0}^{K}\|X_t^i - \sum_{s=0}^{k-1}\eta\widetilde{h}_{i,t}^s - X_t^i\|^2\Big]$$

$$\leq \mathbb{E}\Big[sKL^2\eta^2\sum_{i\in S}\eta_i^2\sum_{k=0}^{K}\|\sum_{s=0}^{k-1}\widetilde{h}_{i,t}^s\|^2\Big] = \frac{s^2KL^2\eta^2}{n}\sum_{k=0}^{K}\sum_{i=1}^{n}\eta_i^2\mathbb{E}\|\sum_{s=0}^{k-1}\widetilde{h}_{i,t}^s\|^2$$

$$\leq \frac{s^2K^2L^2\eta^2}{n}(2nK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2G^2 + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2)$$

$$= 2s^2K^3L^2\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{8s^2K^4L^4\eta^2}{n}\mathbb{E}[\Phi_t] + 4s^2K^4L^2\eta^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq K((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{K^2L^2}{n}\mathbb{E}[\Phi_t] + 2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$\square$

Lemma B.9. *For any step t*

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}(\nabla f_i(X_t^i) - \nabla f_i(\mu_t))\|^2 \leq \frac{s^2K^2L^2}{n}\mathbb{E}[\Phi_t]$$

Proof.

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}(\nabla f_i(X_t^i) - \nabla f_i(\mu_t))\|^2 \leq sK\mathbb{E}\Big[\sum_{i\in S}\sum_{k=0}^{K}\zeta_{i,k}^2\|\nabla f_i(X_t^i) - \nabla f_i(\mu_t)\|^2\Big]$$

$$\leq \frac{s^2K^2}{n}\sum_{i=1}^{n}\mathbb{E}\|\nabla f_i(X_t^i) - \nabla f_i(\mu_t)\|^2$$

$$\leq \frac{s^2K^2L^2}{n}\sum_{i=1}^{n}\mathbb{E}\|X_t^i - \mu_t\|^2 = \frac{s^2K^2L^2}{n}\mathbb{E}[\Phi_t]$$

$\square$

Lemma B.10. *For any step t*

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}(\nabla f_i(\mu_t) - \nabla f(\mu_t))\|^2 \leq 2sK^2G^2 + 4sK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof.

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^K\zeta_{i,k}(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2 = \mathbb{E}\Big[\sum_{i\in S}\eta_i^2(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$+\mathbb{E}\Big[\sum_{i\neq j\in S}\eta_i\eta_j(\sum_{k=0}^K\zeta_{i,k})(\sum_{k=0}^K\zeta_{j,k})\langle\nabla f_i(\mu_t)-\nabla f(\mu_t),\nabla f_i(\mu_t)-\nabla f(\mu_t)\rangle\Big]$$

$$=\mathbb{E}\Big[\sum_{i\in S}\eta_i^2(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$+\mathbb{E}\Big[\sum_{i\neq j\in S}\eta_i\eta_j\mathbb{E}[(\sum_{k=0}^K\zeta_{i,k})]\mathbb{E}[(\sum_{k=0}^K\zeta_{j,k})]\langle\nabla f_i(\mu_t)-\nabla f(\mu_t),\nabla f_i(\mu_t)-\nabla f(\mu_t)\rangle\Big]$$

$$=\mathbb{E}\Big[\sum_{i\in S}\eta_i^2(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$+\mathbb{E}\Big[\sum_{i\neq j\in S}\eta_i H_i\eta_j H_j\langle\nabla f_i(\mu_t)-\nabla f(\mu_t),\nabla f_i(\mu_t)-\nabla f(\mu_t)\rangle\Big]$$

$$=\mathbb{E}\Big[\sum_{i\in S}\eta_i^2(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$+H_{\min}^2\mathbb{E}\Big[\sum_{i\neq j\in S}\langle\nabla f_i(\mu_t)-\nabla f(\mu_t),\nabla f_i(\mu_t)-\nabla f(\mu_t)\rangle\Big]$$

$$\leq\frac{\binom{n-1}{s-1}}{\binom{n}{s}}\sum_{i=1}^n\mathbb{E}\Big[(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$+H_{\min}^2\frac{\binom{n-2}{s-2}}{\binom{n}{s}}\sum_{i\neq j}\mathbb{E}\Big[\langle\nabla f_i(\mu_t)-\nabla f(\mu_t),\nabla f_i(\mu_t)-\nabla f(\mu_t)\rangle\Big]$$

$$=\frac{s}{n}\sum_{i=1}^n\mathbb{E}\Big[(\sum_{k=0}^K\zeta_{i,k})^2\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$-H_{\min}^2\frac{\binom{n-2}{s-2}}{\binom{n}{s}}\sum_{i=1}^n\mathbb{E}\Big[\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\Big]$$

$$=\frac{sK^2}{n}\sum_{i=1}^n\mathbb{E}\|(\nabla f_i(\mu_t)-\nabla f(\mu_t))\|^2\leq\frac{2sK^2}{n}\sum_{i=1}^n\mathbb{E}\|\nabla f_i(\mu_t)\|^2+2sK^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 2sK^2(G^2+B^2\mathbb{E}\|\nabla f(\mu_t)\|^2)+2sK^2\mathbb{E}\|\nabla f(\mu_t)\|^2\leq 2sK^2G^2+4sK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

□

Lemma B.11. *For any step t*

$$\mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^K\zeta_{i,k}\nabla f_i(X_t^i)\|^2\leq\frac{3s^2K^2L^2}{n}\mathbb{E}[\Phi_t]+3sK((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2+2KG^2)+15s^2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof.

$$\mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i)\|^2 = \mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} (\nabla f_i(X_t^i) - \nabla f_i(\mu_t) + \nabla f_i(\mu_t) - \nabla f(\mu_t) + \nabla f(\mu_t))\|^2$$

$$\leq 3\mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} (\nabla f_i(X_t^i) - \nabla f_i(\mu_t))\|^2 + 3\mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} (\nabla f_i(\mu_t) - \nabla f(\mu_t))\|^2$$

$$+ 3s^2 K^2 \mathbb{E}\|\nabla f(\mu_t))\|^2$$

$$\leq \frac{3s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 6sK^2 G^2 + 12sK^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2 + 3s^2 K^2 \mathbb{E}\|\nabla f(\mu_t))\|^2$$

$$\leq \frac{3s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 6sK^2 G^2 + 15s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq \frac{3s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 3sK((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + 15s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

$\square$

Lemma B.12. *For any step t*

$$\mathbb{E}\|\sum_{i \in S} \eta_i h_{i,t}\|^2 \leq \frac{8s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 8sK((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + 32s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof.

$$\mathbb{E}\|\sum_{i \in S} \eta_i h_{i,t}\|^2 = \mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} h_{i,t}^k\|^2 = \mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i - \sum_{s=0}^{k-1} \eta \widetilde{h}_{i,t}^s)\|^2$$

$$= \mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i - \sum_{s=0}^{k-1} \eta \widetilde{h}_{i,t}^s) - \sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i) + \sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i)\|^2$$

$$\leq 2\mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i - \sum_{s=0}^{k-1} \eta \widetilde{h}_{i,t}^s) - \sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i)\|^2 + 2\mathbb{E}\|\sum_{i \in S} \eta_i \sum_{k=0}^{K} \zeta_{i,k} \nabla f_i(X_t^i)\|^2$$

$$\leq 2K((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + \frac{2K^2 L^2}{n} \mathbb{E}[\Phi_t] + 2K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2)$$

$$+ \frac{6s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 6sK((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + 30s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq \frac{8s^2 K^2 L^2}{n} \mathbb{E}[\Phi_t] + 8sK((\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2 + 2KG^2) + 32s^2 K^2 B^2 \mathbb{E}\|\nabla f(\mu_t)\|^2$$

$\square$

Lemma B.13. *For any step t*

$$\mathbb{E}\|\sum_{i \in S} \eta_i (\widetilde{h}_{i,t} - h_{i,t})\|^2 \leq sK(\frac{1}{n}\sum_{i=1}^{n} \eta_i^2)\sigma^2$$

Proof.

$$\mathbb{E}\|\sum_{i\in S}\eta_i(\widetilde{h}_{i,t}-h_{i,t})\|^2 = \mathbb{E}\|\sum_{i\in S}\eta_i\sum_{k=0}^{K}\zeta_{i,k}(\widetilde{h}_{i,t}^k-h_{i,t}^k)\|^2 = \mathbb{E}\Big[\sum_{i\in S}\eta_i^2\sum_{k=0}^{K}\zeta_{i,k}^2\|(\widetilde{h}_{i,t}^k-h_{i,t}^k)\|^2\Big]$$

$$+ \mathbb{E}\Big[\sum_{i,j\in S,(i,k)\neq(j,k')}\eta_i\eta_j\zeta_{i,k}\zeta_{j,k'}\langle\widetilde{h}_{i,t}^k-h_{i,t}^k,\widetilde{h}_{j,t}^{k'}-h_{j,t}^{k'}\rangle\Big]$$

$$= \mathbb{E}\Big[\sum_{i\in S}\eta_i^2\sum_{k=0}^{K}\zeta_{i,k}^2\|(\widetilde{h}_{i,t}^k-h_{i,t}^k)\|^2\Big] \le sK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2$$

$\square$

Lemma 3.6. For any step t

$$\mathbb{E}\|\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 \le \frac{16s^2K^2L^2}{n}\mathbb{E}[\Phi_t] + 18sK((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2+2KG^2) + 64s^2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Proof.

$$\mathbb{E}\|\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 \le 2\mathbb{E}\|\sum_{i\in S}\eta_i h_{i,t}\|^2 + 2\mathbb{E}\|\sum_{i\in S}\eta_i(\widetilde{h}_{i,t}-h_{i,t})\|^2$$

$$\le 2\Big(\frac{8s^2K^2L^2}{n}\mathbb{E}[\Phi_t] + 8sK((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2+2KG^2)+32s^2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2\Big) + 2sK(\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2$$

$$\le \frac{16s^2K^2L^2}{n}\mathbb{E}[\Phi_t] + 18sK((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2+2KG^2) + 64s^2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$\square$

Lemma B.14. *For any step t*

$$\mathbb{E}\|\mu_{t+1}-\mu_t\|^2 \le \frac{2\eta^2}{(n+1)^2}\mathbb{E}\|\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

Proof.

$$\mathbb{E}\|\mu_{t+1}-\mu_t\|^2 = \frac{1}{(n+1)^2}\mathbb{E}\|-\eta\sum_{i\in S}\eta_i\widetilde{h}_{i,t} + \frac{Q(X_t)-X_t}{s+1}$$

$$+ \frac{1}{s+1}\sum_{i\in S}(Q(X_t^i-\eta\eta_i\widetilde{h}_{i,t})-(X_t^i-\eta\eta_i\widetilde{h}_{i,t}))\|^2$$

$$\le \frac{2}{(n+1)^2}\mathbb{E}\|-\eta\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 + \frac{2}{(n+1)^2}\mathbb{E}\|\frac{Q(X_t)-X_t}{s+1}$$

$$+ \frac{1}{s+1}\sum_{i\in S}(Q(X_t^i-\eta\eta_i\widetilde{h}_{i,t})-(X_t^i-\eta\eta_i\widetilde{h}_{i,t}))\|^2$$

$$\le \frac{2}{(n+1)^2}\mathbb{E}\|-\eta\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 + \frac{2}{(n+1)^2}\Big(\frac{1}{s+1}\mathbb{E}\Big\|Q(X_t)-X_t\Big\|^2$$

$$+ \frac{1}{s+1}\sum_{i\in S}\mathbb{E}\Big\|(Q(X_t^i-\eta\eta_i\widetilde{h}_{i,t})-(X_t^i-\eta\eta_i\widetilde{h}_{i,t}))\Big\|^2\Big)$$

$$\le \frac{2\eta^2}{(n+1)^2}\mathbb{E}\|\sum_{i\in S}\eta_i\widetilde{h}_{i,t}\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

$\square$

Lemma B.15. *For any step t*

$$\mathbb{E}\|\mu_{t+1} - \mu_t\|^2 \leq \frac{32\eta^2 s^2 K^2 L^2}{n(n+1)^2}\mathbb{E}[\Phi_t] + \frac{36sK\eta^2}{(n+1)^2}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)$$
$$+ \frac{128\eta^2 s^2 K^2 B^2}{(n+1)^2}\mathbb{E}\|\nabla f(\mu_t)\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

Proof.

$$\mathbb{E}\|\mu_{t+1} - \mu_t\|^2 \leq \frac{2\eta^2}{(n+1)^2}\mathbb{E}\|\sum_{i\in S}\eta_i^2\widetilde{h}_{i,t}\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

$$\leq \frac{2\eta^2}{(n+1)^2}\Big(\frac{16s^2K^2L^2}{n}\mathbb{E}[\Phi_t] + 18sK((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + 64s^2K^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2\Big)$$

$$+ \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

$$\leq \frac{32\eta^2 s^2 K^2 L^2}{n(n+1)^2}\mathbb{E}[\Phi_t] + \frac{36sK\eta^2}{(n+1)^2}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)$$

$$+ \frac{128\eta^2 s^2 K^2 B^2}{(n+1)^2}\mathbb{E}\|\nabla f(\mu_t)\|^2 + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}$$

$\square$

B.4 Convergence

Theorem B.16. *For learning rate $\eta = \frac{n+1}{sH_{\min}\sqrt{T}}$, Algorithm 1 converges at rate:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{2(f(\mu_0) - f_*)}{\sqrt{T}} + \frac{800nKL^2(R^2+7)^2\gamma^2}{H_{\min}} + \frac{6KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{T}}$$

$$+ \frac{808n(n+1)^2K^2L^2}{sH_{\min}^3 T}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + \frac{2(R^2+7)^2\gamma^2 L\sqrt{T}}{(n+1)^2 sH_{\min}}$$

Proof. Let $\mathbb{E}_t$ denote expectation conditioned on the entire history up to and including step t. By L-smoothness we have that

$$\mathbb{E}_t[f(\mu_{t+1})] \leq f(\mu_t) + \mathbb{E}_t\langle\nabla f(\mu_t), \mu_{t+1} - \mu_t\rangle + \frac{L}{2}\mathbb{E}_t\|\mu_{t+1} - \mu_t\|^2. \tag{2}$$

First we look at $\mathbb{E}_t\langle\nabla f(\mu_t), \mu_{t+1} - \mu_t\rangle = \langle\nabla f(\mu_t), \mathbb{E}_t[\mu_{t+1} - \mu_t]\rangle$. If set S is chosen at step $t+1$, We have that

$$\mu_{t+1} - \mu_t = \frac{1}{n+1}(-\eta\sum_{i\in S}\eta_i\widetilde{h}_{i,t} + \frac{Q(X_t) - X_t}{s+1} + \frac{1}{s+1}\sum_{i\in S}(Q(X_t^i - \eta\eta_i\widetilde{h}_{i,t}) - X_t^i - \eta\eta_i\widetilde{h}_{i,t})))$$

Thus, in this case:

$$\mathbb{E}_t[\mu_{t+1} - \mu_t] = -\frac{\eta}{n+1}\sum_{i\in S}\eta_i h_{i,t}.$$

Where we used unbiasedness of quantization and stochastic gradients. We would like to note that even though we do condition on the entire history up to and including step t and this includes conditioning on X_t^i, the algorithm has not yet used $\widetilde{h}_{i,t}$ (it does not count towards computation of μ_t), thus we can safely use all properties of stochastic gradients. Hence, we can proceed by taking into the account that each set of agents S is chosen as initiator with probability $\frac{1}{\binom{n}{s}}$:

$$\mathbb{E}_t[\mu_{t+1} - \mu_t] = \sum_S \frac{1}{\binom{n}{s}}\sum_{i\in S} -\frac{\eta}{n+1}\eta_i h_{i,t} = -\frac{s\eta}{n(n+1)}\sum_{i=1}^n \eta_i h_{i,t}.$$

and subsequently

$$\mathbb{E}_t\langle\nabla f(\mu_t), \mu_{t+1} - \mu_t\rangle = \sum_{i=1}^{n}\frac{s\eta}{n(n+1)}\mathbb{E}_t\langle\nabla f(\mu_t), -\eta_i h_{i,t}\rangle.$$

Hence, we can rewrite (2) as:

$$\mathbb{E}_t[f(\mu_{t+1})] \leq f(\mu_t) + \sum_{i=1}^{n}\frac{s\eta}{n(n+1)}\mathbb{E}_t\langle\nabla f(\mu_t), -\eta_i h_{i,t}\rangle + \frac{L}{2}\mathbb{E}_t\|\mu_{t+1} - \mu_t\|^2.$$

Next, we remove the conditioning

$$\mathbb{E}[(\mu_{t+1})] = \mathbb{E}[\mathbb{E}_t[f(\mu_{t+1})]] \leq \mathbb{E}[f(\mu_t)] + \sum_{i=1}^{n}\frac{s\eta}{n(n+1)}\mathbb{E}\langle\nabla f(\mu_t), -\eta_i h_{i,t}\rangle$$
$$+ \frac{L}{2}\mathbb{E}\|\mu_{t+1} - \mu_t\|^2.$$

This allows us to use Lemmas B.15 and B.4:

$$\mathbb{E}[f(\mu_{t+1})] - \mathbb{E}[f(\mu_t)] \leq \frac{s\eta}{n(n+1)}\left(4KL^2\mathbb{E}[\Phi_t] + (-\frac{3H_{\min}n}{4} + 8B^2L^2\eta^2K^3n)\mathbb{E}\|\nabla f(\mu_t)\|^2\right.$$
$$\left. + 4nL^2\eta^2K^3((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2G^2)\right)$$
$$+ \frac{L}{2}\left(\frac{36sK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{(n+1)^2} + \frac{32s^2L^2K^2\eta^2\mathbb{E}[\Phi_t]}{n(n+1)^2}\right.$$
$$\left. + \frac{128B^2s^2K^2\eta^2\mathbb{E}\|\nabla f(\mu_t)\|^2}{(n+1)^2} + \frac{2(R^2+7)^2\gamma^2}{(n+1)^2}\right)$$
$$= (\frac{4\eta sKL^2}{n(n+1)} + \frac{32s^2K^2L^3\eta^2}{n(n+1)^2})\mathbb{E}[\Phi_t]$$
$$+ (\frac{4sL^2\eta^3K^3}{n+1} + \frac{18sK\eta^2L}{(n+1)^2})((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{(R^2+7)^2\gamma^2L}{(n+1)^2}$$
$$+ (\frac{-3\eta sH}{4(n+1)} + \frac{8B^2L^2\eta^3sK^3}{n+1} + \frac{64B^2s^2K^2L\eta^2}{(n+1)^2})\mathbb{E}\|\nabla f(\mu_t)\|^2$$

By simplifying the above inequality we get:

$$\mathbb{E}[f(\mu_{t+1})] - \mathbb{E}[f(\mu_t)] \leq \frac{5\eta sKL^2\mathbb{E}[\Phi_t]}{n(n+1)} + (\frac{4sL^2\eta^3K^3}{n+1} + \frac{18sK\eta^2L}{(n+1)^2})((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$
$$+ \frac{(R^2+7)^2\gamma^2L}{(n+1)^2} + (\frac{-3\eta sH_{\min}}{4(n+1)} + \frac{8B^2L^2\eta^3sK^3}{n+1}$$
$$+ \frac{64B^2s^2K^2L\eta^2}{(n+1)^2})\mathbb{E}\|\nabla f(\mu_t)\|^2$$

by summing the above inequality for $t = 0$ to $t = T - 1$, we get that

$$\mathbb{E}[f(\mu_T)] - f(\mu_0) \leq \frac{5\eta sKL^2}{n(n+1)}\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + (\frac{4sL^2\eta^3K^3T}{n+1} + \frac{18sK\eta^2LT}{(n+1)^2})((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$
$$+ (\frac{-3\eta sH_{\min}}{4(n+1)} + \frac{8B^2L^2\eta^3sK^3}{n+1} + \frac{64B^2s^2K^2L\eta^2}{(n+1)^2})\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$
$$+ \frac{(R^2+7)^2\gamma^2LT}{(n+1)^2}$$

Further, we use Lemma B.6:

$$\mathbb{E}[f(\mu_T)] - f(\mu_0) \leq \frac{5\eta s K L^2}{n(n+1)} \Bigg(80Tn^2(R^2+7)^2\gamma^2 + 80Tn^2 sK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 160B^2 n^2 sK^2\eta^2 \sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \Bigg)$$

$$+ \big(\frac{4sL^2\eta^3 K^3 T}{n+1} + \frac{18sK\eta^2 LT}{(n+1)^2}\big)((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{(R^2+7)^2\gamma^2 LT}{(n+1)^2} +$$

$$\big(\frac{-3\eta s H}{4(n+1)} + \frac{8B^2 L^2\eta^3 sK^3}{n+1} + \frac{64B^2 s^2 K^2 L\eta^2}{(n+1)^2}\big)\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq \frac{400\eta snKL^2 T(R^2+7)^2\gamma^2}{n+1} + \frac{404Tns^2 K^2 L^2\eta^3((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{n+1}$$

$$+ \frac{18sK\eta^2 LT((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{(n+1)^2} + \frac{(R^2+7)^2\gamma^2 LT}{(n+1)^2}$$

$$+ \big(\frac{-3\eta s H_{\min}}{4(n+1)} + \frac{8B^2 L^2\eta^3 sK^3}{n+1}$$

$$+ \frac{64B^2 s^2 K^2 L\eta^2}{(n+1)^2} + \frac{800B^2 ns^2 K^3\eta^3 L^2}{n+1}\big)\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

by assuming $\eta < \frac{1}{100B\sqrt{nsk^2 L}}$ we get:

$$\mathbb{E}[f(\mu_T)] - f(\mu_0) \leq \frac{400\eta snKL^2 T(R^2+7)^2\gamma^2}{n+1} +$$

$$+ \big(\frac{18sK\eta^2 LT}{(n+1)^2} + \frac{404Tns^2 K^2 L^2\eta^3}{n+1}\big)((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{(R^2+7)^2\gamma^2 LT}{(n+1)^2}$$

$$+ \frac{-\eta s H_{\min}}{2(n+1)}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

Next, we regroup terms, multiply both sides by $\frac{2(n+1)}{\eta s H_{\min}T}$ and use the fact that $f(\mu_T) \geq f_*$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{2(n+1)(f(\mu_0)-f_*)}{sH_{\min}\eta T} + \frac{800nKL^2(R^2+7)^2\gamma^2}{H} +$$

$$+ \big(\frac{36K\eta L}{H_{\min}(n+1)} + \frac{808nsK^2 L^2\eta^2}{H_{\min}}\big)((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{2(R^2+7)^2\gamma^2 L}{(n+1)sH_{\min}\eta}$$

Finally, we set $\eta = \frac{n+1}{H_{\min}\sqrt{sT}}$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{2(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{800nKL^2(R^2+7)^2\gamma^2}{H_{\min}} + \frac{36KL((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}} \tag{3}$$

$$+ \frac{808n(n+1)^2 K^2 L^2}{H_{\min}^3 T}((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{2(R^2+7)^2\gamma^2 L\sqrt{T}}{\sqrt{s}(n+1)^2} \tag{4}$$

$\square$

Lemma B.17. *For quantization parameters* $(R^2 + 7)^2 \gamma^2 = \frac{(n+1)^2}{sH_{\min}^2 T}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L})$ *we have:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{4(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{36KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$
$$+ \frac{1608n(n+1)^2K^2L^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^3 T}$$
$$+ \frac{800n(n+1)^2KL(f(\mu_0)-f_*)}{sH_{\min}^3 T}$$

Proof.

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{2(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{800nKL^2(R^2+7)^2\gamma^2}{H_{\min}} + \frac{36KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$
$$+ \frac{808n(n+1)^2K^2L^2}{H^3 T}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2) + \frac{2(R^2+7)^2\gamma^2 L\sqrt{T}}{\sqrt{s}(n+1)^2}$$
$$= \frac{2(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{800nKL^2(n+1)^2}{sH_{\min}^3 T}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L})$$
$$+ \frac{36KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}} + \frac{808n(n+1)^2K^2L^2}{H_{\min}^3 T}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)$$
$$+ \frac{2L}{s\sqrt{s}H_{\min}^2\sqrt{T}}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L})$$
$$\leq \frac{4(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{36KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$
$$+ \frac{1608n(n+1)^2K^2L^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^3 T}$$
$$+ \frac{800n(n+1)^2KL(f(\mu_0)-f_*)}{sH_{\min}^3 T}$$

$\square$

Lemma B.18. *We have:*

$$5s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 3\eta^2\sum_{t=0}^{T-1}\sum_i \mathbb{E}\|\widetilde{h}_{i,t}\|^2 \leq 1000Tn^3 s(R^2+7)^2\gamma^2$$
$$+ 10000B^2n^3s^3H_{\min}^2K^2LT(R^2+7)^2\gamma^2$$

Proof.

$$5s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 3\eta^2\sum_{t=0}^{T-1}\sum_i\mathbb{E}\|\widetilde{h}_{i,t}\|^2$$

$$\leq 5s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 3\eta^2\sum_{t=0}^{T-1}\left(2nK((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2) + 8L^2K^2\mathbb{E}[\Phi_t] + 4nK^2B^2\mathbb{E}\|\nabla f(\mu_t)\|^2\right)$$

$$\leq 5s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 6nT\eta^2K((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2) + 24\eta^2L^2K^2\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t]$$

$$+ 12nB^2\eta^2K^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 6s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 6nT\eta^2K((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2) + 12B^2n\eta^2K^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 6s\Big(80Tn^2(R^2+7)^2\gamma^2 + 80Tn^2sK\eta^2((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 160B^2n^2sK^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2\Big)$$

$$+ 6nT\eta^2K((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2) + 12B^2n\eta^2K^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 480Tn^2s(R^2+7)^2\gamma^2 + (480Tn^2s^2K\eta^2 + 6nT\eta^2K)((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ (960n^2s^2K^2B^2\eta^2 + 12B^2n\eta^2K^2)\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 480Tn^2s(R^2+7)^2\gamma^2 + 486Tn^2s^2K\eta^2((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 1000B^2n^2s^2K^2\eta^2\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2$$

$$\leq 480Tn^2s(R^2+7)^2\gamma^2 + 486Tn^2s^2K\eta^2((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 1000B^2n^2s^2K^2\eta^2\Big(\frac{2(n+1)(f(\mu_0) - f_*)}{sH_{\min}\eta} + \frac{800TnKL^2(R^2+7)^2\gamma^2}{H_{\min}} +$$

$$+ (\frac{36TK\eta L}{H_{\min}(n+1)} + \frac{808TnsK^2L^2\eta^2}{H_{\min}})((\frac{1}{n}\sum_{i=1}^n\eta_i^2)\sigma^2 + 2KG^2) + \frac{2T(R^2+7)^2\gamma^2L}{(n+1)sH_{\min}\eta}\Big)$$

$$\leq 480Tn^2s(R^2+7)^2\gamma^2 + 486Tn^2s^2K\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ \frac{2000B^2n^2(n+1)sK^2\eta(f(\mu_0)-f_*)}{H_{\min}} + \frac{800000TB^2n^3s^2K^3\eta^2L^2(R^2+7)^2\gamma^2}{H_{\min}} +$$

$$+ (\frac{36000TB^2n^2s^2K^3\eta^3L}{H_{\min}(n+1)} + \frac{808000TB^2n^3s^3K^4\eta^4L^2}{H_{\min}})((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ \frac{4000TB^2nsK^2\eta L(R^2+7)^2\gamma^2}{H_{\min}}$$

$$\leq 1000Tn^3s(R^2+7)^2\gamma^2 + \frac{2000B^2n^2(n+1)^2sK^2}{\sqrt{T}}(f(\mu_0)-f_*)$$

$$+ \frac{10000Tn^3(n+1)^2s^2KL}{T}((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

Therefore we have

$$5s\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + 3\eta^2\sum_{t=0}^{T-1}\sum_{i}\mathbb{E}\|\widetilde{h}_{i,t}\|^2 \leq 1000Tn^3s(R^2+7)^2\gamma^2$$

$$+ 10000B^2n^3(n+1)^2s^2K^2L(\frac{f(\mu_0)-f_*}{L} + (\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$= 1000Tn^3s(R^2+7)^2\gamma^2 + 10000B^2n^3s^3H_{\min}^2K^2LT(R^2+7)^2\gamma^2$$

$$\square$$

Lemma B.19. *Let* $T \geq O(n^3)$*, then for quantization parameters* $R = 2 + T^{\frac{3}{d}}$ *and* $\gamma^2 = \frac{(n+1)^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2+2KG^2+\frac{f(\mu_0)-f_*}{L})}{sH_{\min}^2T(R^2+7)^2}$ *we have that the probability of quantization never failing during the entire run of the Algorithm 1 is at least* $1 - O\left(\frac{1}{T}\right)$*.*

Proof. Let $\mathcal{L}_t$ be the event that quantization does not fail during step t. Our goal is to show that $Pr[\cup_{t=1}^{T}\mathcal{L}_t] \geq 1 - O\left(\frac{1}{T}\right)$. In order to do this, we first prove that $Pr[\neg\mathcal{L}_{t+1}|\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_t] \leq O\left(\frac{1}{T^2}\right)$ (O is with respect to T here).

We need need to lower bound probability that :

$$\forall i \in S : \|X_t - X_t^i\|^2 \leq (R^{R^d}\gamma)^2 \tag{5}$$

$$\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2 \leq (R^{R^d}\gamma)^2 \tag{6}$$

$$\|X_t - X_t^i\|^2 = O\left(\frac{\gamma^2(poly(T))^2}{R^2}\right) \tag{7}$$

$$\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2 = O\left(\frac{\gamma^2(poly(T))^2}{R^2}\right) \tag{8}$$

We would like to point out that these conditions are necessary for decoding to succeed, we ignore encoding since it will be counted when someone will try to decode it. Since, $R = 2 + T^{\frac{3}{d}}$ this means that $(R^{R^d})^2 \geq 2^{2T^3} \geq T^{30}$, for large enough T. Hence, it is suffices to upper bound the probability that $\sum_{i \in S}\|X_t - X_t^i\|^2 + \sum_{i \in S}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2 \geq T^{30}\gamma^2$. To prove this, we have:

$$\sum_{i \in S}\|X_t - X_t^i\|^2 + \sum_{i \in S}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2 \leq \sum_{i \in S}(5\|X_t - \mu_t\|^2$$

$$+ 5\|\mu_t - X_t^i\|^2 + 3\eta^2\|\widetilde{h}_{i,t}\|^2) \leq 5s\Phi_t + 3\eta^2\sum_{i}\|\widetilde{h}_{i,t}\|^2$$

Now, we use Markov's inequality, and Lemma B.18:

$$Pr[5s\Phi_t + 3\eta^2\|\widetilde{h}_{i,t}\|^2 \geq T^{30}\gamma^2|\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_t] \leq \frac{\mathbb{E}[5s\Phi_t + 3\eta^2\sum_i\|\widetilde{h}_{i,t}\|^2|\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_t]}{T^{30}\gamma^2}$$

$$\leq \frac{1000Tn^3s(R^2+7)^2\gamma^2 + 10000B^2n^3s^3H_{\min}^2K^2LT(R^2+7)^2\gamma^2}{T^{30}\gamma^2} \leq O(\frac{1}{T^2})$$

Thus, the failure probability due to the models not being close enough for quantization to be applied is at most $O\left(\frac{1}{T^2}\right)$. Conditioned on the event that $\|X_t - X_t^i\|$ and $\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|$ are upper bounded by $T^{15}\gamma$ (This is what we actually lower bounded the probability for using Markov), we get that the probability of quantization algorithm failing is at most

$$\sum_{i \in S} \log\log(\frac{1}{\gamma}\|X_t - X_t^i\|) \cdot O(R^{-d})$$

$$+ \sum_{i \in S} \log\log(\frac{1}{\gamma}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|) \cdot O(R^{-d})$$

$$\leq O\left(\frac{s\log\log T}{T^3}\right) \leq O\left(\frac{1}{T^2}\right).$$

By the law of total probability (to remove conditioning) and the union bound we get that the total probability of failure, either due to not being able to apply quantization or by failure of quantization algorithm itself is at most $O\left(\frac{1}{T^2}\right)$. Finally we use chain rule to get that

$$Pr[\cup_{t=1}^T \mathcal{L}_t] = \prod_{t=1}^T Pr[\mathcal{L}_t| \cup_{s=0}^{t-1}\mathcal{L}_s] = \prod_{t=1}^T\left(1 - Pr[\neg\mathcal{L}_t| \cup_{s=0}^{t-1}\mathcal{L}_s]\right)$$

$$\geq 1 - \sum_{t=1}^T Pr[\neg\mathcal{L}_t| \cup_{s=0}^{t-1}\mathcal{L}_s] \geq 1 - O\left(\frac{1}{T}\right).$$

□

Lemma B.20. *Let $T \geq O(n^3)$, then for quantization parameters $R = 2 + T^{\frac{3}{d}}$ and $\gamma^2 = \frac{\eta^2}{(R^2+7)^2}((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L})$ we have that the expected number of bits used by Algorithm 1 per communication is $O(d\log(n) + \log(T))$.*

Proof. At step $t + 1$, by Corollary 3.3, we know that the total number of bits used is at most

$$\sum_{i \in S} O\left(d\log(\frac{R}{\gamma}\|X_t^i - X_t\|)\right) + O\left(d\log(\frac{R}{\gamma}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|)\right)$$

By taking the randomness of agent interaction at step $t + 1$ into the account, we get that the expected number

of bits used is at most:

$$\sum_S \frac{1}{\binom{n}{s}} \sum_{i \in S} \left(O\left(d\log(\frac{R}{\gamma}\|X_t^i - X_t\|)\right) + O\left(d\log(\frac{R}{\gamma}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|)\right)\right)$$

$$= \sum_i \frac{s}{n} \left(O\left(d\log(\frac{R}{\gamma}\|X_t^i - X_t\|)\right) + O\left(d\log(\frac{R}{\gamma}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|)\right)\right)$$

$$=\leq \sum_i \frac{s}{n} \left(O\left(d\log(\frac{R^2}{\gamma^2}\|X_t^i - X_t\|^2)\right) + O\left(d\log(\frac{R^2}{\gamma^2}\|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2)\right)\right)$$

$$\overset{Jensen}{\leq} s\left(O\left(d\log(\frac{R^2}{\gamma^2}\sum_i \frac{1}{n}(\|X_t^i - X_t\|^2 + \|X_t - (X_t^i - \eta\widetilde{h}_{i,t})\|^2))\right)\right)$$

$$\leq s\left(O\left(d\log(\frac{R^2}{\gamma^2}\sum_i \frac{1}{n}(\|X_t - \mu_t\|^2 + \|X_t^i - \mu_t\|^2 + \eta^2\|\widetilde{h}_{i,t}\|^2))\right)\right)$$

$$\leq s\left(O\left(d\log(\frac{R^2}{\gamma^2}(\Phi_t + \frac{\eta^2}{n}\sum_i \|\widetilde{h}_{i,t}\|^2))\right)\right)$$

So the expected number of bits per communication in all rounds is at most:

$$\frac{1}{sT}\sum_{t=0}^{T-1} s\left(O\left(d\log(\frac{R^2}{\gamma^2}(\Phi_t + \frac{\eta^2}{n}\sum_i \|\widetilde{h}_{i,t}\|^2))\right)\right)$$

$$\leq \left(O\left(d\log(\frac{R^2}{\gamma^2}(\frac{1}{T}\sum_{t=0}^{T-1}\Phi_t + \frac{1}{T}\sum_{t=0}^{T-1}\frac{\eta^2}{n}\sum_i \|\widetilde{h}_{i,t}\|^2))\right)\right)$$

Next, By Jensen inequality and Lemma B.18, We get that the expected number of bits used is at most,

$$O\left(d\mathbb{E}\left[\log(\frac{R^2}{\gamma^2}(\frac{1}{T}\sum_{t=0}^{T-1}\Phi_t + \frac{1}{T}\sum_{t=0}^{T-1}\frac{\eta^2}{n}\sum_i \|\widetilde{h}_{i,t}\|^2)]\right)\right)$$

$$\overset{Jensen}{\leq} O\left(d\log(\frac{R^2}{\gamma^2}(\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + \frac{1}{T}\sum_{t=0}^{T-1}\frac{\eta^2}{n}\sum_i \mathbb{E}\|\widetilde{h}_{i,t}\|^2)\right)$$

$$\leq O\left(d\log(\frac{R^2}{\gamma^2}(\frac{1}{T}(1000Tn^3s(R^2+7)^2\gamma^2 + 10000B^2n^3s^3H^2K^2LT(R^2+7)^2\gamma^2)))\right)$$

$$\leq O\left(d\log(R^2(1000n^3s(R^2+7)^2 + 10000B^2n^3s^3H^2K^2L(R^2+7)^2))\right) = O(d\log(n) + \log(T))$$

$\square$

Theorem 3.2. Assume the total number of steps $T \geq \Omega(n^3)$, the learning rate $\eta = \frac{n+1}{H_{\min}\sqrt{sT}}$, and quantization parameters $R = 2 + T^{\frac{3}{d}}$ and $\gamma^2 = \frac{\eta^2}{(R^2+7)^2}\left((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L}\right)$. Let $H_{\min} > 0$ be the minimum H_i. Then, with probability at least $1 - O(\frac{1}{T})$ we have that Algorithm 1 converges at the following rate

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2 \leq \frac{4(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{36KL((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}} +$$
$$O\left(\frac{n^3K^2L^2((\frac{1}{n}\sum_{i=1}^n \eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^3 T}\right).$$

and uses $O\left(sT(d\log n + \log T)\right)$ expected communication bits in total.

Proof. The proof simply follows from combining Lemmas B.17, B.19 and B.20

$\square$

Lemma B.21. *For the convergence of the server, we have:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(X_t)\|^2 \le \frac{12(f(X_0)-f_*)}{\sqrt{sT}} + \frac{108KL((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$

$$+ \left(\frac{4824n(n+1)^2K^2L^2}{H_{\min}^3 T} + \frac{320n^2(n+1)^2KL^2}{H_{\min}^2 T}\right)\left((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2\right)$$

$$+ \left(\frac{2400n(n+1)^2KL}{sH_{\min}^3 T} + \frac{160n^2(n+1)^2L^2}{sH_{\min}^2 T}\right)(f(X_0)-f_*)$$

Proof.

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(X_t)\|^2 \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(X_t) - \nabla f(\mu_t) + \nabla f(\mu_t)\|^2$$

$$\leq \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(X_t) - \nabla f(\mu_t)\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}\|\nabla f(\mu_t)\|^2$$

$$\leq \frac{2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|X_t - \mu_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}\|\nabla f(\mu_t)\|^2$$

$$\leq \frac{2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Phi_t] + \frac{2}{T}\sum_{t=0}^{T-1}\|\nabla f(\mu_t)\|^2$$

$$\leq 2L^2\big(80n^2(R^2+7)^2\gamma^2 + 80n^2sK\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ 160B^2n^2sK^2\eta^2\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\mu_t)\|^2) + \frac{2}{T}\sum_{t=0}^{T-1}\|\nabla f(\mu_t)\|^2$$

$$\leq 160n^2L^2(R^2+7)^2\gamma^2 + 160n^2sKL^2\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{3}{T}\sum_{t=0}^{T-1}\|\nabla f(\mu_t)\|^2$$

$$\leq 160n^2L^2(R^2+7)^2\gamma^2 + 160n^2sKL^2\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) + \frac{12(f(\mu_0)-f_*)}{\sqrt{sT}}$$

$$+ \frac{108KL((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$

$$+ \frac{4824n(n+1)^2K^2L^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^3 T} + \frac{2400n(n+1)^2KL(f(\mu_0)-f_*)}{sH_{\min}^3 T}$$

$$\leq 160n^2L^2\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2 + \frac{f(\mu_0)-f_*}{L}) + 160n^2sKL^2\eta^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2) +$$

$$\frac{12(f(\mu_0)-f_*)}{\sqrt{sT}} + \frac{108KL((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$

$$+ \frac{4824n(n+1)^2K^2L^2((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^3 T}$$

$$+ \frac{2400n(n+1)^2KL(f(\mu_0)-f_*)}{sH_{\min}^3 T}$$

$$\leq \frac{12(f(X_0)-f_*)}{\sqrt{sT}} + \frac{108KL((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)}{H_{\min}^2\sqrt{sT}}$$

$$+ (\frac{4824n(n+1)^2K^2L^2}{H_{\min}^3 T} + \frac{320n^2(n+1)^2KL^2}{H_{\min}^2 T})((\frac{1}{n}\sum_{i=1}^{n}\eta_i^2)\sigma^2 + 2KG^2)$$

$$+ (\frac{2400n(n+1)^2KL}{sH_{\min}^3 T} + \frac{160n^2(n+1)^2L^2}{sH_{\min}^2 T})(f(X_0)-f_*)$$

<div align="right">□</div>

Finally, the **proof of Corollary 3.3** follows from combining Lemmas B.21, B.19 and B.20