DoCoFL: Downlink Compression for Cross-Device Federated Learning

Ron Dorfman [1 2] **Shay Vargaftik** [1] **Yaniv Ben-Itzhak** [1] **Kfir Y. Levy** [2]

Abstract

Many compression techniques have been proposed to reduce the communication overhead of Federated Learning training procedures. However, these are typically designed for compressing model updates, which are expected to decay throughout training. As a result, such methods are inapplicable to downlink (i.e., from the parameter server to clients) compression in the cross-device setting, where heterogeneous clients *may appear only once* during training and thus must download the model parameters. Accordingly, we propose DoCoFL – a new framework for downlink compression in the cross-device setting. Importantly, DoCoFL can be seamlessly combined with many uplink compression schemes, rendering it suitable for bi-directional compression. Through extensive evaluation, we show that DoCoFL offers significant bi-directional bandwidth reduction while achieving competitive accuracy to that of a baseline without any compression.

1. Introduction

In recent years, there has been an increasing interest in federated learning (FL) as a paradigm for large-scale machine learning over decentralized data (Konečnỳ et al., 2016a; Kairouz et al., 2021). FL enables organizations and/or devices, collectively termed *clients*, to jointly build better and more robust models by relying on their collective data and processing power. Importantly, the FL training procedure occurs without exchanging or sharing client-specific data, thus ensuring some degree of privacy and compliance with data access rights and regulations (e.g., the General Data Protection Regulation (GDPR) implemented by the European Union in May, 2018). Instead, in each round, clients perform local optimization using their local data and send only model updates to a central coordinator, also known as the *parameter server* (PS). The PS aggregates these updates and updates the global model, which is then utilized by the clients in subsequent rounds.

One of the main challenges in FL is the communication bottleneck introduced during the distributed training procedure. To illustrate this bottleneck, consider the following example of a real FL deployment presented by McMahan et al. (2022): their training involves a small neural network with 1.3 million parameters; in each round there are 6500 participating clients; and the model is trained over 2000 rounds. A simple calculation shows that the total required bandwidth to and from the PS during this training is ≈ 61.5 TB. Since modern machine learning models have many millions (or even billions) of parameters and we might have more participants, FL may result in excessive communication overhead.

To deal with this overhead, many bandwidth reduction techniques have been proposed. These include taking multiple (rather than a single) local optimization steps (McMahan et al., 2017), quantization techniques (Seide et al., 2014; Alistarh et al., 2017; Wen et al., 2017; Bernstein et al., 2018a;b; Karimireddy et al., 2019; Jin et al., 2020; Shlezinger et al., 2020), low-rank decomposition (Vogels et al., 2019), sketching (Ivkin et al., 2019; Rothchild et al., 2020), and distributed mean estimation (Lyubarskii & Vershynin, 2010; Suresh et al., 2017; Konečnỳ & Richtárik, 2018; Vargaftik et al., 2021; 2022; Safaryan et al., 2022). However, as we detail in §2, a direct application of these techniques is less suitable for downlink compression, i.e., from the PS to the clients, in the cross-device setup in which new and heterogeneous clients may participate at each round and thus must download the model parameters. This is in contrast to the cross-silo setup in which the PS can compress and send a global update (i.e., clients' aggregated update) to all silos.

To the best of our knowledge, only a handful of works consider bi-directional compression, i.e., compression from the clients to the PS and vice versa. These works mainly rely on per-client memory mechanism (Tang et al., 2019; Zheng et al., 2019; Liu et al., 2020; Philippenko & Dieuleveut, 2020; 2021; Gruntkowska et al., 2022) or require keeping an updated copy of the model on all clients (Horvóth et al., 2022), thus targeting either distributed learning or FL with full or partial but *recurring* participation (e.g., cross-silo FL). Such solutions are less suitable for large-scale cross-device

[1]VMware Research [2]Viterby Faculty of Electrical and Computer Engineering, Technion, Haifa, Israel. Correspondence to: †Ron Dorfman <rdorfman@campus.technion.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

FL, where a client may appear only a handful of times, or even just once, during the entire training procedure.

It is important to stress that the significance of bi-directional bandwidth reduction for cross-device FL goes far beyond cost reduction, energy efficiency, and carbon footprint considerations. In fact, inclusion, fairness, and bias are at the very heart of cross-device FL as, according to recent sources (Sumra, 2020; Howdle, 2022), the price of a wireless connection and its quality admits differences of orders of magnitude among countries. This may prevent large populations from contributing to cross-device FL training due to costly and unstable connectivity, resulting in biased and less accurate models.

Accordingly, in this work we introduce DoCoFL, a novel downlink compression framework specifically designed for cross-device FL. Importantly, it operates independently of many uplink compression techniques, making it suitable for bi-directional compression in cross-device setups.

The primary challenge addressed by DoCoFL is that clients must download model parameters (i.e., weights) instead of model updates. Unlike updates, which are proportional to gradients and thus their norm is expected to decrease during training, the model parameters do not decay, rendering low-bit compression methods undesirable. As a result, and since clients can only download the updated model weights during their designated participation round, this can lead to a network bottleneck for low-resourced clients.

To address this bottleneck, DoCoFL decomposes the download burden by utilizing previous models, referred to as *anchors*, which clients can download prior to their participation round. Then, at the designated participation round, clients only need to download the correction, i.e., the difference between the updated model and the anchor. As the correction is proportionate to the sum of previous updates, it is expected to decay, allowing for the use of low-bit compression methods. To ensure the correction term, PS memory footprint, and PS computational overhead remain manageable, the available anchors are updated periodically. This approach reduces the amount of bandwidth required by the clients *online* (i.e., at their participation round). To reduce the overall downlink bandwidth usage, we further develop and utilize an efficient anchor compression technique with an appealing bandwidth-accuracy tradeoff.

Contributions. We summarize our contributions below,

- We propose a new framework (DoCoFL) that both enlarges the time window during which clients can obtain the model parameters and reduces the total downlink bandwidth requirements for cross-device FL.
- We show that DoCoFL provably converges to a stationary point when not compressing anchors and give an asymptotic convergence rate.
- We design a new compression technique with strong empirical results, which DoCoFL uses for anchor compression and can be of independent interest. We provide the theoretical intuition and empirical evidence for why DoCoFL with anchor compression works.

Finally, we show over image classification and language processing tasks that DoCoFL consistently achieves model accuracy that is competitive with an uncompressed baseline, namely, FedAvg (McMahan et al., 2017) while reducing bandwidth usage in both directions by order of magnitude.

2. Background and Related Work

In this section, we overview mostly related work and detail the challenges in designing a bi-directional bandwidth reduction framework for cross-device FL.

2.1. Uplink vs. Downlink Compression

In the context of FL, uplink (i.e., client to PS) and downlink (i.e., PS to client) compression are inherently different and should not be treated in the same manner. In particular, many recent uplink compression solutions (e.g., Konečnỳ et al. (2016b); Alistarh et al. (2017)) partially rely on two properties to obtain their effectiveness:

Averaging. A fundamental property arises when many clients send their compressed gradients for averaging at the PS. If the clients' estimates are independent and unbiased, the error in estimating their mean by calculating their estimations' mean is decreasing linearly with respect to the number of clients. Thus, having more clients in each round allows for more aggressive and more accurate compression.

Error Decay. Essentially, unbiased compression of updates results in an increased variance in their estimation. This increase can be compensated by decreasing the learning rate. Moreover, the effect of update compression is expected to diminish since the expected update decays as the training process approaches a stationary point. This is not the case when compression model parameters.

For downlink compression, we immediately lose the averaging property since, by design, there is only one source with whom the clients communicate, namely, the PS. Regarding the error decay property, we must further distinguish between different FL setups as described next.

2.2. Cross-silo vs. Cross-device FL

FL can be divided into two types based on the nature of the participating clients (Kairouz et al. (2021), Table 1).

Silos. In cross-silo FL, the clients are typically assumed to be active throughout the training procedure and with sufficient compute and network resources. Silos are typi-

Table 1. Averaging and Error Decay in different setups.

		Averaging	Error Decay
Uplink		✓	✓
Downlink	Cross-silo	×	✓
	Cross-device	×	×

cally associated with entities such as hospitals that jointly train a model for better diagnosis and treatment (Ng et al., 2021) or banks that jointly build better models for fraud and anomalous activity detection (Yang et al., 2019).

Indeed, silos allow for the design of efficient compression techniques that rely on client persistency and per-client memory mechanisms that are used for, e.g., compressing gradient differences, employing error feedback, and learning control variates (Alistarh et al., 2018; Karimireddy et al., 2020; Philippenko & Dieuleveut, 2020; Gorbunov et al., 2021; Richtárik et al., 2021). While most of these techniques consider only uplink compression, some recent works target bi-directional compression by utilizing the same property of using per-client memory and relying on *repeated* client participation (Tang et al., 2019; Liu et al., 2020; Philippenko & Dieuleveut, 2020; 2021; Gruntkowska et al., 2022).

Devices. In cross-device FL, clients are typically assumed to be heterogeneous and not persistent to the extent that a client often participates in a *single* out of many thousands of training rounds. Also, in this setup, clients may often admit compute and network constraints. Devices are usually associated with entities such as laptops, smartphones, smartwatches, tablets, and IoT devices. A typical example of a cross-device FL application is keyboard completion for android devices (McMahan & Ramage, 2017).

Unlike in silos with full or partial but repeated participation, compression techniques for devices that appear only once or a handful of times cannot rely on having some earlier state for or on that device. This renders methods that rely on per-client memory or learned control variates less suitable for such cross-device FL setups. Indeed, recent gradient compression techniques can be readily used for bi-directional compression in the cross-silo setup or only uplink compression in both setups (Konečnỳ et al., 2016b; Alistarh et al., 2017; Suresh et al., 2017; Ramezani-Kebrya et al., 2021; Vargaftik et al., 2021; 2022; Safaryan et al., 2022).

2.3. Putting It All Together

As summarized in Table 1, differences in the clients' nature and the compression direction (i.e., uplink vs. downlink) significantly affect the efficiency of bandwidth reduction techniques. In the considered setups, downlink compression is more challenging than uplink compression due to the lack of averaging and received considerably less attention in the literature. Moreover, for the cross-device setup, the problem is more acute due to not having error decay as well.

3. DoCoFL

In this section, we present DoCoFL. We start with describing our design goals, which are derived from the challenges outlined in the previous section, followed by a formal definition of the federated optimization problem. Then, in §3.1, we give intuition and introduce our framework. In §3.2, we detail about an important element of DoCoFL, namely, the client selection process employed by the PS. Finally, we provide a theoretical convergence result in §3.3.

Design Goals. Motivated by the discussion in the previous section, we aim at achieving two goals to deal with the low bandwidth and slow and unstable connectivity conditions that edge devices may experience:

1. *Enlarging the time window* during which a client can download the model weights from the PS.
2. *Reducing the bandwidth* requirements in the downlink direction.

Achieving both these goals will enable more heterogeneous clients to participate in the training process, which in turn may reduce bias and improve fairness[1].

Preliminaries. We use $\|\cdot\|$ to denote the L_2 norm and for every $n \in \mathbb{N}$, $[n] \coloneqq \{1, \dots, n\}$. Let N be the number of clients participating in the federated training procedure. Each client $i \in [N]$ is associated with a local loss function f_i, and our goal is to minimize the loss with respect to all clients, i.e., to solve

$$\min_{w \in \mathbb{R}^d} f(w) \coloneqq \frac{1}{N} \sum_{i=1}^{N} f_i(w) \,. \tag{1}$$

Unlike in standard distributed optimization or cross-silo FL with full or partial but repeated participation, in cross-device FL, only a subset of S clients participate in each optimization round and typically $S \ll N$ (e.g., S in the hundreds/thousands and N in many millions). Thus, clients are not expected to repeatedly participate in the optimization.

Since FL mostly considers non-convex optimization (e.g., neural networks), and global loss minimization of such models is generally intractable, we focus on finding an approximate stationary point, i.e., a point w for which the expected gradient norm $\mathbb{E}\|\nabla f(w)\|$ tends to zero.

For the purpose of formal analysis, we make a few standard assumptions, namely, that f is bounded from below by

[1] By fairness, we refer to the situation where clients in regions with limited, unstable, and costly connectivity are keen to participate in the training procedure.

f^*, the local functions $\{f_i\}$ are β-smooth, i.e., $\|\nabla f_i(w) - \nabla f_i(u)\| \leq \beta\|w-u\|, \forall w,u \in \mathbb{R}^d$, and the access to each local function is done via a stochastic gradient oracle, i.e.,

Assumption 3.1. For any $w \in \mathbb{R}^d$, client i computes an unbiased gradient estimator $g^i(w)$ with a variance that is upper bounded by σ^2, i.e.,

$$\mathbb{E}[g^i(w)] = \nabla f_i(w), \;\; \mathbb{E}\|g^i(w) - \nabla f_i(w)\|^2 \leq \sigma^2 \; . \quad (2)$$

Additionally, we assume that the dissimilarity of the local gradients is bounded (i.e., limited client data heterogeneity).

Assumption 3.2. There exist constants $G, B \in \mathbb{R}_+$ such that for every $w \in \mathbb{R}^d$:

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(w)\|^2 \leq G^2 + B^2\|\nabla f(w)\|^2 \; . \quad (3)$$

While some works consider milder (Khaled & Richtárik, 2020; Haddadpour et al., 2021) or no (Gorbunov et al., 2021) assumptions on client heterogeneity in some settings (e.g., exact gradients and/or full participation), this assumption is standard in heterogeneous federated learning (Karimireddy et al., 2020; Wang et al., 2020; Reddi et al., 2021).

3.1. Overview

A naive approach to reduce bandwidth in the downlink direction is to apply some compression to the model weights and have all participating clients download the compressed weights. That is, in each round t, the participating clients $\mathcal{S}_t \subseteq [N]$ obtain a compressed version of the model weights $\hat{w}_t = \mathcal{C}_w(w_t)$, for some compression operator $\mathcal{C}_w$. The clients can then compute an unbiased gradient estimator at $\hat{w}_t$ and send it back to the PS for aggregation.

While this method is fairly simple, it has inherent disadvantages with respect to our goals. First, there is no enlarged time window during which clients can download the compressed model weights, as they can only do so at their participation round. Second, unlike with gradient compression, convergence in this setting can be guaranteed only to a proximity that is proportional to the compression error, rendering standard low-bit compression schemes unusable. Indeed, this is the case even for strongly convex functions, as shown by Chraibi et al. (2019) and reinforced by our counter-example in Appendix A.

To tackle these challenges, our approach relies on the following relation: for any $\tau \geq 0$, we can decompose w_t into two ingredients: *Anchor* and *Correction*. Formally,

$$w_t = \underbrace{w_{t-\tau}}_{\textit{(i) Anchor}} + \underbrace{w_t - w_{t-\tau}}_{\textit{(ii) Correction}} \; .$$

This implies that:

Algorithm 1 DoCoFL – Parameter Server

Input: Initial weights $w_0 \in \mathbb{R}^d$, learning rate η, weights (anchors) compression $\mathcal{C}_w$, correction compression $\mathcal{C}_c$, anchor compression rate K, compressed anchors queue $Q \leftarrow \emptyset$ with capacity $\mathcal{V}$, client participation process $\mathcal{P}(\cdot)$
for $t = 0, \ldots, T-1$ **do**

▷ Anchor Deployment
if t mod $K == 0$ **then**
Compress anchor, $\mathcal{C}_w(w_t)$
Q.enqueue($\mathcal{C}_w(w_t)$) ▷ If Q is full, Q.dequeue()
end if

▷ Client Participation Process
$\mathcal{S}_t \leftarrow \mathcal{P}(t)$ ▷ $|\mathcal{S}_t| = S$; see §3.2

▷ Optimization
for client $i \in \mathcal{S}_t$ in parallel **do**
Send compressed correction, $\hat{\Delta}_t^i$ ▷ See Algorithm 2
Obtain compressed local gradient, $\mathcal{C}_g(\hat{g}_t^i)$
end for
Aggregate local gradients, $\hat{g}_t := \frac{1}{S}\sum_{i\in\mathcal{S}_t} \mathcal{C}_g(\hat{g}_t^i)$
Update weights, $w_{t+1} = w_t - \eta\hat{g}_t$
end for

Algorithm 2 DoCoFL – Client i

Input: Gradient compression $\mathcal{C}_g$
Notification round s (by process $\mathcal{P}$):
Obtain participation round t ▷ $i \in \mathcal{P}(t), s \leq t$
Obtain latest compressed anchor, $y_t^i \leftarrow Q$.top() ▷ Within time window $[s, t]$

Participation round t:
Obtain compressed correction, $\hat{\Delta}_t^i = \mathcal{C}_c(w_t - y_t^i)$
Construct current model estimate, $\hat{w}_t^i = y_t^i + \hat{\Delta}_t^i$
Compute local gradient, $\hat{g}_t^i = g_t^i(\hat{w}_t^i)$
Compress and send local gradient, $\mathcal{C}_g(\hat{g}_t^i)$, to PS

(i) If a client is notified at round $t-\tau$ about its upcoming participation at round t, it can start downloading the anchor, that is, $w_{t-\tau}$, *ahead of its participation round*,[2]

(ii) and thus, at round t, the client only needs to download the correction, that is, $w_t - w_{t-\tau}$.

Yet, merely relying on this relation is not sufficient to achieve our goals; additionally, we seek to *compress* both *(i)* and *(ii)*. However, these terms are inherently different and therefore, should not be treated in the same manner. Essentially, the client has more time to download *(i)*, which is the main ingredient that forms the model weights. Introducing a large error in this term may prevent the model from converging. Conversely, *(ii)* must be downloaded at the participation round of the client, but it is just the sum of τ recent gradients.

For *(i)*, we develop a new compression technique (see §4), that achieves a better accuracy-bandwidth tradeoff than gradient compression techniques at the cost of higher complex-

[2]The client can start downloading $w_{t-\tau'}$ at round $t-\tau'$ for some $\tau' \leq \tau$, as long as the download is complete *before* round t.

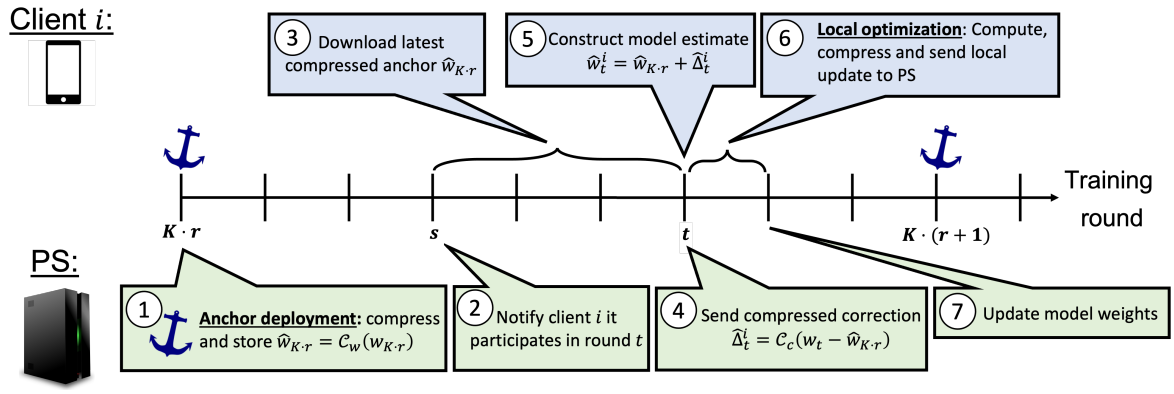

Figure 1. DoCoFL's training procedure. Here, we illustrate the interaction between the PS and a single client. Typically, multiple clients participate in each training round, and each client may be notified about its participation in a different round.

ity; we amortize its complexity over several rounds. Using this technique with only several bits per coordinate (e.g., 4) results in a negligible error. For *(ii)*, we use standard gradient compression techniques with as low as 0.5 bit per coordinate. Since *(ii)* is only a sum of τ recent gradients, this error is expected to decay as training progresses.

Overall, we achieve both goals as (1) the download time window is enlarged, with only a small bandwidth fraction that must be used online; and (2) the total downlink bandwidth usage is reduced by up to an order of magnitude compared to existing solutions and without degrading model accuracy.

In Figure 1, we show a timeline that illustrates the training procedure of DoCoFL, as we now formally detail on the role of the PS and the clients in our framework.

Parameter Server. As detailed in Algorithm 1, our PS executes three separate processes throughout the training procedure. First, it performs ① 'anchor deployment' – once in every K rounds, it compresses the model weights and stores the compressed weights in a queue Q of length $\mathcal{V}$. Second, the PS employs a client participation process, which we elaborate on in §3.2; this process determines which clients will participate in a given round. Finally, in each round, the PS obtains the local model updates (i.e., gradients) from the clients participating in that round, computes their average, and ⑦ updates the model weights.

Client. Consider a client that is ② chosen by the PS at some round s to participate in some future round t. This means that in round $s \leq t$, the client is notified about its upcoming participation. It can then start ③ downloading the latest anchor stored by the PS; note that the client has a time window of length $t - s$ rounds to download the compressed anchor. At its participation round t, the client ④ obtains the compressed correction from the PS, i.e., the compressed difference between the updated model and the compressed anchor obtained earlier, to ⑤ construct an unbiased estimate of the updated model weights.[3] It then ⑥ locally computes a stochastic gradient, compresses it, and sends the compressed gradient to the PS; see Algorithm 2.

3.2. Client Participation Process

An important element in DoCoFL is the client participation process $\mathcal{P}$. For a round number t, it returns a subset of clients $\mathcal{S}_t \subseteq [N]$ of size S to participate in that round. Crucially, in DoCoFL, clients can be notified about their participation prior to their actual participation round.

This discrepancy between the notification and the participation rounds gives DoCoFL the desired versatility that opens the door for more clients to participate in the optimization process. While current frameworks follow a selection process that notifies a client about its participation just before it takes place, it is possible to consider useful selection/notification processes where some clients (e.g., with weaker connectivity) are notified earlier than others.

Another point to consider is the bias-utility tradeoff, where some choices of $\mathcal{P}$ can allow more clients to participate but may introduce bias in the participation rounds of clients with an untractable effect on the optimization process. Instead, we focus on processes that preserve the property where at each round, the PS can obtain an unbiased estimate of the gradient, which means that we require $\mathcal{P}$ to satisfy the following property: all clients have the same probability of participating in any given round t, i.e., $\mathbb{P}(i \in \mathcal{P}(t)) = S/N$. Surprisingly, such a restriction allows for a wide range of useful selection policies.

For example, consider a simple scenario where the PS has

[3] The client obtains the (unbiasedly) compressed difference between w_t and the *compressed* anchor rather than the exact anchor. Thus, the model estimate is unbiased, even if the compressor $\mathcal{C}_w$ is biased. A similar mechanism was used by Horváth & Richtárik (2020) for gradient compression, i.e., 'induced compressor'.

predetermined time windows T_s and T_w that it associates with "strongly connected" and "weakly connected" devices, respectively. Then, at each round t, the PS randomly selects clients but assigns their participation rounds to $t+T_s$ or $t+T_w$ according to their strength. Observe that this simple, yet very useful scenario satisfies the property we seek after T_w rounds (the first T_w rounds may take longer since, during these initial rounds, the weakly connected clients cannot be notified enough rounds prior to their participation).

3.3. Theoretical Guarantee

The primary challenge in analyzing downlink compression schemes for cross-device FL is that, even when using an unbiased compression method, for which $\mathbb{E}[\mathcal{C}_w(w)]=w$, the resulting gradient estimate $\nabla f(\mathcal{C}_w(w))$ may be biased. This is because, in general, the gradient is not a linear mapping. As mentioned, the resulting bias can hinder convergence; in Appendix A we show that gradient descent with weights compression may not reach the optimal solution even for strongly convex functions.

Accordingly, we show that DoCoFL converges to a stationary point when $\mathcal{C}_w$ is the identity mapping, i.e., $\mathcal{C}_w(w)=w, \forall w\in\mathbb{R}^d$ – a setup that achieves our first goal (i.e., enlarged time window). As our analysis suggests, this identity assumption enables us to effectively bound the gradient bias resulting from the compression. For simplicity, we also assume no uplink compression (i.e., $\mathcal{C}_g$ is also identity), although including it in our analysis is straightforward for unbiased $\mathcal{C}_g$[4], and we do incorporate it in our experiments.

Following this result, and the result of Chraibi et al. (2019) in the convex case, in Appendix B we give a theoretical intuition and empirical evidence for why DoCoFL works well in setups of interest, when $\mathcal{C}_w$ is *not* the identity function – achieving our second goal (i.e., total bandwidth reduction).

Before we state our convergence result, we require an additional standard assumption about the correction compression operator $\mathcal{C}_c$, namely, that it has a bounded Normalized Mean-Squared-Error (NMSE) (Philippenko & Dieuleveut, 2020; Richtárik et al., 2021; Vargaftik et al., 2021).

Assumption 3.3. There exists an $\omega\in\mathbb{R}_+$ such that

$$\mathbb{E}\left[\|\mathcal{C}_c(w)-w\|^2\right]\leq\omega^2\|w\|^2,\quad\forall w\in\mathbb{R}^d\,. \tag{4}$$

We now give a convergence result for DoCoFL, namely, Theorem 3.4. Its full proof is deferred to Appendix C; here, we discuss the result and give a proof sketch.

[4]Unbiasedness in the uplink direction is highly desired since (together with independence) it ensures linearly decaying mean estimation error with respect to the number of clients. For biased $\mathcal{C}_g$, in light of existing results on biased compressors (Beznosikov et al., 2020), it may be the case that for some biased compressors the theoretical guarantee, with additional challenges, holds.

Theorem 3.4. *Let $M=f(w_0)-f^*$, $\tilde{\sigma}^2=\sigma^2+4\left(1-\frac{S}{N}\right)G^2$, and $\gamma=1+\left(1-\frac{S}{N}\right)\frac{B^2}{S}$. Then, DoCoFL with $\mathcal{C}_w$ and $\mathcal{C}_g$ as identity mappings (and appropriate η) guarantees*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(w_t)\|^2\right]\in\mathcal{O}\left(\sqrt{\frac{M\beta\tilde{\sigma}^2}{TS}}+\frac{(M^2\beta^2\omega^2K\mathcal{V}\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}}+\frac{\gamma M\beta(\omega K\mathcal{V}+1)}{T}\right).$$

The convergence rate in Theorem 3.4 consists of three terms:

- The first term $\sqrt{\frac{M\beta\tilde{\sigma}^2}{TS}}$ is a slow statistical term that depends only on the noise level $\tilde{\sigma}^2$ (and the objective's properties); importantly, it is independent of DoCoFL's hyperparameters, $K,\mathcal{V}$, and ω.
- The last term $\frac{\gamma M\beta(\omega K\mathcal{V}+1)}{T}$ is a fast deterministic term. When $\omega K\mathcal{V}\in\mathcal{O}(1)$ it decreases proportionally to $1/T$, and otherwise, it is proportional to $\omega K\mathcal{V}/T$.
- The middle term $\frac{(M^2\beta^2\omega^2K\mathcal{V}\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}}$ is a moderate term that depends on both noise level and DoCoFL's hyperparameters through the multiplication $\omega^2K\mathcal{V}\tilde{\sigma}^2$; it is proportionate to $(T^{2/3}S^{1/3})^{-1}$.

We next derive observations from Theorem 3.4. Henceforth, we omit from $\mathcal{O}(\cdot)$ the dependence on M,β, and γ.

Corollary 3.5. *When $\mathcal{C}_c$ is the identity mapping, i.e., $\omega=0$, clients obtain the exact model, and thus our method is equivalent to FedAvg. Indeed, we get the same asymptotic rate as FedAvg (McMahan et al., 2017; Karimireddy et al., 2019), namely, $\mathcal{O}(\sqrt{\tilde{\sigma}^2/TS}+1/T)$.*

Corollary 3.6. *Suppose $\omega K\mathcal{V}\in\Theta(1)$. In that case, we get the following asymptotic rate:*

$$\mathcal{O}\left(\sqrt{\frac{\tilde{\sigma}^2}{TS}}+\frac{(\omega\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}}+\frac{1}{T}\right).$$

Compared to Corollary 3.5, we note that the middle term is the additional cost incurred for utilizing compression. Importantly, it decreases when we improve the correction compression, i.e., reduce ω.

Corollary 3.7. *Consider $\omega=\Theta(1)$. If $K\mathcal{V}\in\mathcal{O}(\sqrt{\tilde{\sigma}^2T/S})$, the slow term dominates the rate, which is $\mathcal{O}(\sqrt{\tilde{\sigma}^2/TS})$; that is, we can set $K\mathcal{V}$ as large as $\mathcal{O}(\sqrt{\tilde{\sigma}^2T/S})$ and still get, similarly to FedAvg, a speed-up with S, the number of participating clients per-round.*

Proof Sketch. Denote: $\nabla_t\coloneqq\nabla f(w_t)$ and $\hat{\nabla}_t\coloneqq\mathbb{E}[\hat{g}_t]$. By the update rule, the smoothness of the objective and standard arguments, we obtain that

$$\mathbb{E}[f(w_{t+1})-f(w_t)]\leq-\frac{\eta}{2}\mathbb{E}\|\nabla_t\|^2+\frac{\beta\eta^2}{2}\mathbb{E}\|\hat{g}_t\|^2+\frac{\eta}{2}\mathbb{E}\|\hat{\nabla}_t-\nabla_t\|^2\,. \tag{5}$$

Using the smoothness of f, we can bound the last term in the right-hand side, corresponding to the gradient bias, by the clients' average compression error:

$$\mathbb{E}\|\hat{\nabla}_t - \nabla_t\|^2 \leq \beta^2 \cdot \mathbb{E}\left[\frac{1}{S}\sum_{i\in\mathcal{S}_t}\|\hat{w}_t^i - w_t\|^2\right]. \quad (6)$$

Additionally, in Lemma C.1, we derive the following bound on the second moment of the stochastic aggregated gradient:

$$\mathbb{E}\|\hat{g}_t\|^2 \leq \frac{\tilde{\sigma}^2}{S} + 4\gamma\mathbb{E}\|\nabla_t\|^2 + \frac{2\beta^2}{S}\mathbb{E}\sum_{i\in\mathcal{S}_t}\|\hat{w}_t^i - w_t\|^2. \quad (7)$$

Plugging these bounds back to Eq. (5), we obtain:

$$\mathbb{E}[f(w_{t+1}) - f(w_t)] \leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right)\mathbb{E}\|\nabla_t\|^2 + \frac{\beta\eta^2\tilde{\sigma}^2}{2S} + \left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\mathbb{E}\left[\frac{1}{S}\sum_{i\in\mathcal{S}_t}\|\hat{w}_t^i - w_t\|^2\right]. \quad (8)$$

Recall that each client constructs the current model estimate by summing an anchor and a compressed correction, i.e., $\hat{w}_t^i = y_t^i + \mathcal{C}_c(w_t - y_t^i)$, where y_t^i (i.e., the anchor) is some model from up to $K\mathcal{V}$ rounds ago; for simplicity, assume that all clients obtain the oldest anchor, i.e., $y_t^i = w_{t-K\mathcal{V}}$. Therefore, using Assumption 3.3, we can bound the compression error by the difference between the current model and the obtained anchor, which is proportional to the sum of the last few (aggregated) gradients:

$$\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \leq \omega^2\mathbb{E}\|w_t - y_t^i\|^2 = \omega^2\eta^2\mathbb{E}\left\|\sum_{k=t-K\mathcal{V}}^{t-1}\hat{g}_k\right\|^2.$$

Denote the client compression error by $e_t^i \coloneqq \mathbb{E}\|\hat{w}_t^i - w_t\|^2$. Decomposing each gradient into bias and variance as $\hat{g}_k = \hat{\nabla}_k + \hat{\xi}_k$, where $\mathbb{E}[\hat{\xi}_k] = 0$, we get:

$$\begin{aligned} e_t^i &\leq 2\omega^2\eta^2\mathbb{E}\left\|\sum_{k=t-K\mathcal{V}}^{t-1}\hat{\nabla}_k\right\|^2 + 2\omega^2\eta^2\mathbb{E}\left\|\sum_{k=t-K\mathcal{V}}^{t-1}\hat{\xi}_k\right\|^2 \\ &\leq 2\omega^2\eta^2 K\mathcal{V}\sum_{k=t-K\mathcal{V}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 + 2\omega^2\eta^2\sum_{k=t-K\mathcal{V}}^{t-1}\mathbb{E}\|\hat{\xi}_k\|^2, \end{aligned}$$

where we used the orthogonality of the noises, i.e., $\mathbb{E}[\hat{\xi}_k^\top\hat{\xi}_\ell] = 0$ for $k \neq \ell$. Plugging-in $\hat{\xi}_k = \hat{g}_k - \hat{\nabla}_k$, we obtain:

$$e_t^i \leq 6\omega^2\eta^2 K\mathcal{V}\sum_{k=t-K\mathcal{V}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 + 4\omega^2\eta^2\sum_{k=t-K\mathcal{V}}^{t-1}\mathbb{E}\|\hat{g}_k\|^2.$$

Using Eq. (6) and (7) to bound $\mathbb{E}\|\hat{\nabla}_k\|^2$ and $\mathbb{E}\|\hat{g}_k\|^2$, respectively, we get a recursive relation as the client compression error at round t depends on all prior errors. This is due to error accumulation from computing the aggregated gradients at inaccurate iterates. Lemma C.2 provides a (non-recursive) bound on the compression error at round t. Plugging this bound back to Eq. (8), summing over $t = 0, \ldots, T-1$, and using some algebra, we get:

$$\mathbb{E}[f(w_T) - f(w_0)] \leq -\frac{\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 12\beta^2\omega^2 K\mathcal{V}\eta^3\right)\frac{T\tilde{\sigma}^2}{S}.$$

Rearranging terms and tuning η concludes the proof. □

It is important to note that our framework may also introduce some opportunities for system-wise improvements that are not captured by standard analysis. For example, with a larger pool of clients that are able to participate in a training procedure, it may be easier and faster to reach the desired threshold of participants in each round. Also, it may offer access to more data overall with a different resulting model. How to capture and model such potential benefits in a way that is consistent with and useful in real deployments? Indeed, this is an interesting and significant challenge for future work that may yield new FL policies.

4. Anchor Compression

Compressing the anchors is an essential building block of DoCoFL for reducing the total downlink bandwidth. While many compression techniques exist, most techniques were designed for gradient compression. Although we can use many such methods in our framework, it is less desirable to use a gradient compression scheme for anchor compression since the compression error of the anchor has a larger impact on the resulting model accuracy than the correction error; recall that the model weights, unlike the correction, do not decay throughout training. Accordingly, we designed a compression technique for that purpose.

We first observe that this technique is considerably less restricted on the PS side (i.e., compression) than on the client's side (i.e., decompression). On the PS side, we typically have more resources and time (a new anchor is deployed only every K rounds) to employ more complex calculations, where at the client side we seek speed and lighter computations.

Consequently, we devised a compression method called *Entropy-Constrained Uniform Quantization (ECUQ)*. The main idea behind this approach is to approximate Entropy-Constrained Quantization (ECQ), which is an optimal scheme among a large family of quantization techniques (Chou et al., 1989). Intuitively, given some vector, ECQ finds the best quantization values (i.e., those that minimize the mean squared error) such that after quantization and entropy encoding (e.g., Huffman coding, Huffman (1952)) of the resulting quantized vector, a given budget constraint is respected. However, this approach is slow,

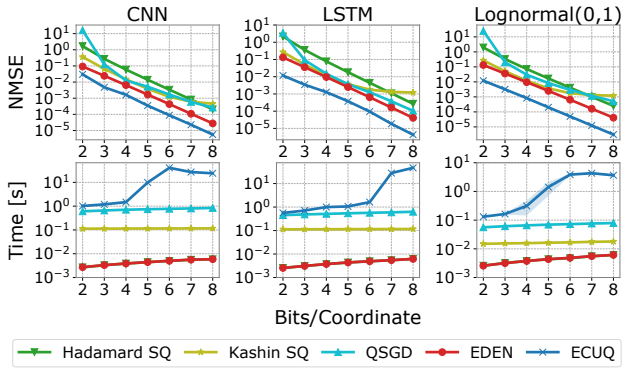

Figure 2. ECUQ vs. gradient compression methods: NMSE (**top**) and encoding time (**bottom**) for three different cases – recorded model parameters of a CNN (**left**); LSTM (**middle**); and vectors drawn from synthetic $LogNormal(0,1)$ distribution (**right**).

complex, and unstable (sensitive to hyperparameters), which renders it unsuitable for online compression of large vectors.

As we detail in Appendix D, ECUQ employs a double binary search to efficiently find the maximal number of *uniformly spaced* quantization values (between the minimal and maximal values of the input vector), such that after quantization, the entropy of the vector would be within a small threshold from a given bandwidth constraint. Since computing the entropy of a vector does not require to actually encode it, the double binary search is executed fast, and only after finding these quantization values, we encode the vector.

In Appendix D, we compare ECUQ, ECQ and a technique based on K-Means clustering (ECK-Means), which also approximates ECQ (see Figure 6); our results indicate that ECUQ is always better than ECK-Means and competitive with ECQ while being orders of magnitude faster.

We also compare ECUQ with four recent gradient compression techniques: **(1)** Hadamard followed by stochastic quantization (SQ) (Suresh et al., 2017); **(2)** Kashin's representation followed by SQ (Lyubarskii & Vershynin, 2010; Safaryan et al., 2022); **(3)** QSGD followed by Elias Gamma encoding (Alistarh et al., 2017); and **(4)** EDEN (Vargaftik et al., 2022). We test these in three different scenarios: **(1)** Model parameters of a convolutional neural network (CNN) with $\approx$ 11M parameters; **(2)** Model parameters of an LSTM network with $\approx$ 8M parameters; and **(3)** Vectors from a $LogNormal(0,1)$ distribution with 1M entries. We repeat each experiment ten times and report the mean.

As shown in Figure 2, ECUQ consistently offers the best NMSE, which is by up to an order of magnitude better than that of the second best. We also find that ECUQ is sufficiently fast to be used by the PS every several rounds (a typical cross-device FL round may take minutes to hours).

While our comparison here focuses on quantization-based methods, in Appendix E we compare ECUQ with three popular compression techniques that do not rely on quantization, namely, Rand-K, Top-K (Alistarh et al., 2018), and Count-Sketch (Charikar et al., 2002) and show similar trends. Nevertheless, we note that quantization is mostly orthogonal to such techniques and they can be used in conjunction[5].

Table 2. Tasks configuration.

Dataset	Net. (# params)	# clients (S)	Partition
CIFAR-100	ResNet-9 (4.9M)	200 (10)	I.I.D
EMNIST	LeNet (65K)	1000 (20)	Non-I.I.D
Amazon	LSTM (8.3M)	500 (10)	I.I.D
Shakespeare	LSTM (820K)	1129 (20)	Non-I.I.D

5. Experiments

As previously mentioned, most prior downlink compression methods rely on repeated client participation and/or control variates and are, therefore, less suitable for large-scale cross-device FL where a client may participate only once or a handful of times during the training procedure. Also, there are prior methods that target model size reduction via sketching (Rabbani et al., 2021) and sparsification (Shah & Lau, 2021), but rely on restrictive assumptions and typically result in longer training times and lower accuracy with an increasing number of clients and decreasing participation ratio. Some other model size reduction methods, such as low-rank approximation (e.g., Tai et al. (2016)) are orthogonal to DoCoFL and they can be used in conjunction.

Accordingly, we compare DoCoFL with an uncompressed baseline obtained by running FedAvg (McMahan et al., 2017) without any (i.e., uplink or downlink) compression, utilizing full precision (i.e., 32-bit floats) in both directions. Then, we perform an ablation study that shows the consistency of DoCoFL with respect to its hyperparameters.

We cover a wide range of use cases that include two image classification and two language processing tasks with different configurations and data partitioning, as shortly summarized in Table 2 and further detailed in Appendix F.

Image Classification. We use the CIFAR-100 and EMNIST datasets. For CIFAR-100 (Krizhevsky et al., 2009), the data distribution among the clients is i.i.d. For EMNIST (Cohen et al., 2017), the dataset of each client is composed of 10% i.i.d samples from the entire dataset and 90% i.i.d samples of 2 out of 47 classes (Karimireddy et al., 2020).

Language Processing. For language processing, we perform a sentiment analysis task on the Amazon Reviews dataset (Zhang et al., 2015) with i.i.d data partitioning; and a next-character prediction task on the Shakespeare

[5]For example, Vargaftik et al. (2022) use Rand-K as a subroutine alongside quantization to reach a sub-bit compression ratio.

Table 3. Best validation accuracy for different tasks. The configuration triplet (b_w, b_c, b_g) means using b_w, b_c, and b_g bits per coordinate for the anchor, correction, and gradient (uplink) compression, respectively. For all tasks, we use $K = 10$ and $\mathcal{V} = 3$.

		CIFAR-100		EMNIST		Amazon		Shakespeare	
		(b_w, b_c, b_g)	Accuracy	(b_w, b_c, b_g)	Accuracy	(b_w, b_c, b_g)	Accuracy	(b_w, b_c, b_g)	Accuracy
FedAvg		–	65.03	–	85.85	–	**92.59**	–	46.10
DoCoFL	Config 1	$(2, 2, 1)$	64.94	$(4, 4, 3)$	85.94	$(6, 6, 2)$	92.51	$(4, 4, 4)$	45.86
	Config 2	$(2, 1/2, 1)$	**65.81**	$(2, 2, 3)$	**86.83**	$(4, 4, 2)$	92.24	$(2, 2, 4)$	**46.55**

dataset (McMahan et al., 2017), where each client holds data associated with a single role and play.

In all simulations, we run DoCoFL with ECUQ for anchor compression (i.e., $\mathcal{C}_w$), and EDEN (Vargaftik et al., 2022) for correction and uplink compression (i.e., $\mathcal{C}_c$ and $\mathcal{C}_g$).

Main Results. In Table 3, we report the best validation accuracy achieved during training for FedAvg and two representative configurations of DoCoFL. It is evident that the validation accuracy of DoCoFL and FedAvg is always competitive; in some tasks, DoCoFL performs somewhat better. For example, for EMNIST, DoCoFL reduces the online and total downlink bandwidth by $16\times$ and $8\times$, respectively, while achieving higher validation accuracy.

As is often the case in FL, our evaluation indicates that using more bandwidth does not necessarily lead to higher validation accuracy. While using less bandwidth usually impacts the train accuracy, as it implies a larger compression error, it may positively affect the model's generalization ability. We further reinforce these observations in Appendix G.

Hyperparameters Ablation. In Figure 3, we report the final *train* accuracy of DoCoFL for the CIFAR-100 task with varying values of $K \in \{10, 50, 100, 500\}$ and $\mathcal{V} \in \{3, 5, 10\}$ under two bandwidth configurations. The results indicate that our framework performs as expected for a wide range of anchor deployment rates and queue capacities. Additionally, in line with our theoretical findings, when the multiplication $K\mathcal{V}$ is too large, the norm of the correction becomes sizable, which can hinder the final accuracy and even convergence. To allow the use of large $K\mathcal{V}$, one may increase the correction bandwidth, trading online bandwidth for a larger anchor download time window. We defer an ablation study of the anchor and correction bandwidth budgets to Appendix G.2. These results indicate that DoCoFL performs well for a wide range of budgets and provide further intuition for configuring these parameters.

The Value of the Correction Term. When ignoring the correction, DoCoFL may resemble other frameworks such as delayed gradients (e.g., Stich & Karimireddy (2019)) and asynchronous SGD (e.g., Lian et al. (2015)). In Appendix G.3 we discuss this similarity and convey that ignoring the correction leads to a significant performance drop.

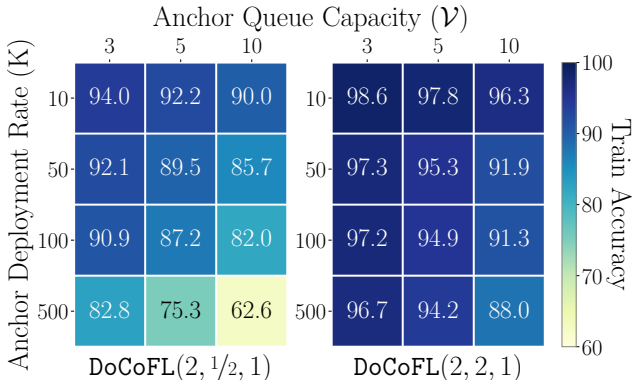

Figure 3. Final train accuracy of DoCoFL for two bandwidth configurations and various values of K and $\mathcal{V}$ on the CIFAR-100 task.

DoCoFL and EF21. In Appendix G.4, we focus on recent advancements based on the EF21 technique (Richtárik et al., 2021), which relies on client-side memory. Specifically, we extend EF21-PP (Fatkhullin et al., 2021) to support downlink bandwidth reduction using DoCoFL while matching baseline accuracy, where naive model compression results in performance degradation. Also, we discuss some similarities with EF21-P + DIANA (Gruntkowska et al., 2022).

6. Conclusion

In this work, we presented DoCoFL, a framework for downlink compression in the challenging cross-device FL setup. By enlarging the clients' model download time window, reducing total downlink bandwidth requirements, and allowing for uplink compression, DoCoFL is designed to allow more resource-constrained and diverse clients to participate in the training procedure. Experiments over various tasks indicate that DoCoFL indeed significantly reduces bi-directional bandwidth usage while performing competitively with an uncompressed baseline. In Appendix H, we discuss some directions for future research.

Acknowledgements

We thank the reviewers and the area chair for their helpful suggestions. KYL is supported by the Israel Science Foundation (grant No. 447/20) and the Technion Center for Machine Learning and Intelligent Systems (MLIS).

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

Arjevani, Y., Shamir, O., and Srebro, N. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pp. 111–132. PMLR, 2020.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018a.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. signSGD with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2018b.

Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

Charikar, M., Chen, K., and Farach-Colton, M. Finding frequent items in data streams. In *Automata, Languages and Programming: 29th International Colloquium, ICALP 2002 Málaga, Spain, July 8–13, 2002 Proceedings 29*, pp. 693–703. Springer, 2002.

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

Chraibi, S., Khaled, A., Kovalev, D., Richtárik, P., Salim, A., and Takáč, M. Distributed fixed point methods with compressed iterates. *arXiv preprint arXiv:1912.09925*, 2019.

Cohen, A., Daniely, A., Drori, Y., Koren, T., and Schain, M. Asynchronous stochastic optimization robust to arbitrary delays. *Advances in Neural Information Processing Systems*, 34:9024–9035, 2021.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Fatkhullin, I., Sokolov, I., Gorbunov, E., Li, Z., and Richtárik, P. EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback. *arXiv preprint arXiv:2110.03294*, 2021.

Giladi, N., Nacson, M. S., Hoffer, E., and Soudry, D. At Stability's Edge: How to Adjust Hyperparameters to Preserve Minima Selection in Asynchronous Training of Neural Networks? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Bkeb7lHtvH.

Gorbunov, E., Burlachenko, K. P., Li, Z., and Richtárik, P. MARINA: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning*, pp. 3788–3798. PMLR, 2021.

Gruntkowska, K., Tyurin, A., and Richtárik, P. EF21-P and Friends: Improved Theoretical Communication Complexity for Distributed Optimization with Bidirectional Compression. *arXiv preprint arXiv:2209.15218*, 2022.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pp. 2350–2358. PMLR, 2021.

Horváth, S. and Richtárik, P. A better alternative to error feedback for communication-efficient distributed learning. *arXiv preprint arXiv:2006.11077*, 2020.

Horvóth, S., Ho, C.-Y., Horvath, L., Sahu, A. N., Canini, M., and Richtárik, P. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pp. 129–141. PMLR, 2022.

Howdle, D. Worldwide mobile data pricing. https://www.cable.co.uk/mobiles/worldwide-data-pricing/#pricing, 2022.

Huffman, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

Ivkin, N., Rothchild, D., Ullah, E., Stoica, I., Arora, R., et al. Communication-efficient distributed SGD with sketching. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode,

G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pp. 3252–3261. PMLR, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khaled, A. and Richtárik, P. Better theory for SGD in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

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

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lian, X., Huang, Y., Li, Y., and Liu, J. Asynchronous parallel stochastic gradient for nonconvex optimization. *Advances in neural information processing systems*, 28, 2015.

Liu, X., Li, Y., Tang, J., and Yan, M. A double residual compression algorithm for efficient distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 133–143. PMLR, 2020.

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

McMahan, B. and Ramage, D. Federated learning: Collaborative machine learning without centralized training data. `https://ai.googleblog.com/2017/04/federated-learning-collaborative.html`, 2017.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

McMahan, H. B., Thakurta, A., Andrew, G., Balle, B., Kairouz, P., Ramage, D., Song, S., Steinke, T., Terzis, A., Thakkar, O., et al. Federated learning with formal differential privacy guarantees. *Google AI Blog*, 2022.

Ng, D., Lan, X., Yao, M. M.-S., Chan, W. P., and Feng, M. Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets. *Quantitative Imaging in Medicine and Surgery*, 11(2):852, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

Philippenko, C. and Dieuleveut, A. Preserved central model for faster bidirectional compression in distributed settings. *Advances in Neural Information Processing Systems*, 34: 2387–2399, 2021.

Rabbani, T., Feng, B., Yang, Y., Rajkumar, A., Varshney, A., and Huang, F. Comfetch: Federated Learning of Large Networks on Memory-Constrained Clients via Sketching. *arXiv preprint arXiv:2109.08346*, 2021.

Ramezani-Kebrya, A., Faghri, F., Markov, I., Aksenov, V., Alistarh, D., and Roy, D. M. NUQSGD: Provably Communication-efficient Data-parallel SGD via Nonuniform Quantization. *J. Mach. Learn. Res.*, 22:114–1, 2021.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=LkFG3lB13U5`.

Richtárik, P., Sokolov, I., and Fatkhullin, I. EF21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34:4384–4396, 2021.

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., and Arora, R. FetchSGD: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pp. 8253–8265. PMLR, 2020.

Safaryan, M., Shulgin, E., and Richtárik, P. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA*, 11(2):557–580, 2022.

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

Shah, S. M. and Lau, V. K. Model compression for communication efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Shlezinger, N., Chen, M., Eldar, Y. C., Poor, H. V., and Cui, S. UVeQFed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 69: 500–514, 2020.

Stewart, J. *Calculus*. Cengage Learning, 2015.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Sumra, H. Best and Worst Countries for Wi-Fi Access. `https://www.ooma.com/blog/best-worst-wifi-countries/`, July 2020.

Suresh, A. T., Felix, X. Y., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *International conference on machine learning*, pp. 3329–3337. PMLR, 2017.

Tai, C., Xiao, T., Wang, X., and E, W. Convolutional neural networks with low-rank regularization. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1511.06067`.

Tang, H., Yu, C., Lian, X., Zhang, T., and Liu, J. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pp. 6155–6165. PMLR, 2019.

Vargaftik, S., Ben-Basat, R., Portnoy, A., Mendelson, G., Ben-Itzhak, Y., and Mitzenmacher, M. DRIVE: one-bit distributed mean estimation. *Advances in Neural Information Processing Systems*, 34:362–377, 2021.

Vargaftik, S., Basat, R. B., Portnoy, A., Mendelson, G., Itzhak, Y. B., and Mitzenmacher, M. EDEN: Communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning*, pp. 21984–22014. PMLR, 2022.

Vogels, T., Karimireddy, S. P., and Jaggi, M. PowerSGD: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *Advances in neural information processing systems*, 30, 2017.

Yang, W., Zhang, Y., Ye, K., Li, L., and Xu, C.-Z. FFD: A federated learning based method for credit card fraud detection. In *International conference on big data*, pp. 18–32. Springer, 2019.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zheng, S., Huang, Z., and Kwok, J. Communication-efficient distributed blockwise momentum SGD with error-feedback. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Suboptimality of Gradient Descent with Weights Compression

In this section, we give an example of a (strongly) convex function on $\mathbb{R}$, for which we show that running gradient descent with gradients computed at estimated (i.e., lossily compressed and then decompressed) iterates (rather than at the exact iterates) does not converge to the global minimum. Instead, it converges to a suboptimal solution.

Let $f : \mathbb{R} \to \mathbb{R}$ be the following convex and smooth function:

$$f(w) = \frac{1}{2}(w-1)^2 + \frac{1}{2}\left[w-1\right]_+^2 \ ,$$

where $\left[w\right]_+ = \max\left(0, w\right)$. Note that $w^* = 1$ is the global minimizer of f. We analyze the following update rule:

$$\begin{aligned} w_{t+1} &= w_t - \eta \nabla f(\hat{w}_t) \\ \hat{w}_t &= \mathcal{C}_w(w_t) \ , \end{aligned}$$

where $\eta > 0$ is the step size, and $\mathcal{C} : \mathbb{R} \to \mathbb{R}$ is a randomized, unbiased compression operator with bounded NMSE, i.e.,

$$\mathbb{E}[\mathcal{C}_w(w)] = w, \quad \mathbb{E}|\mathcal{C}_w(w) - w|^2 \leq \omega_w^2 |w|^2, \quad \forall w \in \mathbb{R} \ .$$

We can alternatively write: $\hat{w}_t = w_t + \epsilon_t |w_t|$, where $\mathbb{E}[\epsilon_t] = 0,\ \mathbb{E}[\epsilon_t^2] \leq \omega_w^2$, and $\{\epsilon_t\}_t$ are independent. Thus, we can rewrite the above update rule as

$$w_{t+1} = w_t - \eta \nabla f(w_t + \epsilon_t |w_t|) \ . \tag{9}$$

In Eq. (9), we repeatedly apply the stochastic mapping: $w \mapsto w - \eta \nabla f(w + \epsilon |w|)$. If this process converges in expectation, it converges to a point $\tilde{w}$ for which $\mathbb{E}[\nabla f(\tilde{w} + \epsilon |\tilde{w}|)] = 0$. We show that $w^* = 1$ does not satisfy this condition, which will imply that this process does not converge to w^*. First, note that f is differentiable, and $\nabla f(w) = (w-1) + \left[w-1\right]_+$. Thus,

$$\mathbb{E}\left[\nabla f(w^* + \epsilon |w^*|)\right] = \mathbb{E}\left[\nabla f(1+\epsilon)\right] = \mathbb{E}\left[\epsilon + \left[\epsilon\right]_+\right] = \mathbb{E}\left[\epsilon\right]_+ \ ,$$

where the last equality follows from the linearity of expectation, $\mathbb{E}[\epsilon] = 0$. Now, note that unless $\epsilon = 0$ almost surely, we necessarily have $\mathbb{E}[\max\left\{0, \epsilon\right\}] > 0$, which implies that the iterative update in Eq. (9) does not converge in expectation to $w^* = 1$.

B. Why DoCoFL with Anchor Compression Works

To support Theorem 3.4, in which we establish the convergence of DoCoFL when the anchor compression $\mathcal{C}_w$ is identity, in this section we give theoretical intuition and numerical results that convey as for why DoCoFL works when $\mathcal{C}_w$ is not the identity mapping.

Consider the framework we analyze in Appendix C, namely the generalization of DoCoFL given by Algorithm 3. Adding an anchor compressor (i.e., $\mathcal{C}_w$) implies that each client now obtains a **compressed** outdated model $\hat{y}_t^i = \mathcal{C}_w(y_t^i)$ and a corresponding correction $\hat{\Delta}_t^i = \mathcal{C}_c(w_t - \hat{y}_t^i)$, and constructs $\hat{w}_t^i = \hat{y}_t^i + \hat{\Delta}_t^i$. Thus, adding an anchor compression affects the client's model estimation error $\mathbb{E}\|\hat{w}_t^i - w_t\|^2$, which we bound in Eq. (23).

Denote by $e_{t,i}^2 \coloneqq \|y_t^i + \hat{\Delta}_t^i - w_t\|^2$ the client's squared estimation error when not using anchor compression (i.e., when $\mathcal{C}_w$ is identity), and by $\hat{e}_{t,i}^2 \coloneqq \|\hat{y}_t^i + \hat{\Delta}_t^i - w_t\|^2$ the squared estimation error when using anchor compression. If one could show that the following condition holds:

$$\mathbb{E}[\hat{e}_{t,i}^2] \leq C^2 \mathbb{E}[e_{t,i}^2] \ , \tag{10}$$

for some moderate $C > 0$, then we can simply bound $\mathbb{E}[\hat{e}_{t,i}^2]$ in the left-hand side of Eq. (23), and the rest of our analysis holds. However, we know that, in general, this condition does not hold (recall the counter-example in Appendix A).

Nevertheless, we empirically show that it holds in our evaluation where we have non-convex and noisy optimization. More generally, in cross-device FL, client sampling and stochastic gradient estimation add natural noise to the optimization process, and we empirically show that the additional estimation error due to anchor compression with ECUQ is sufficiently low and allows convergence, as conveyed above.

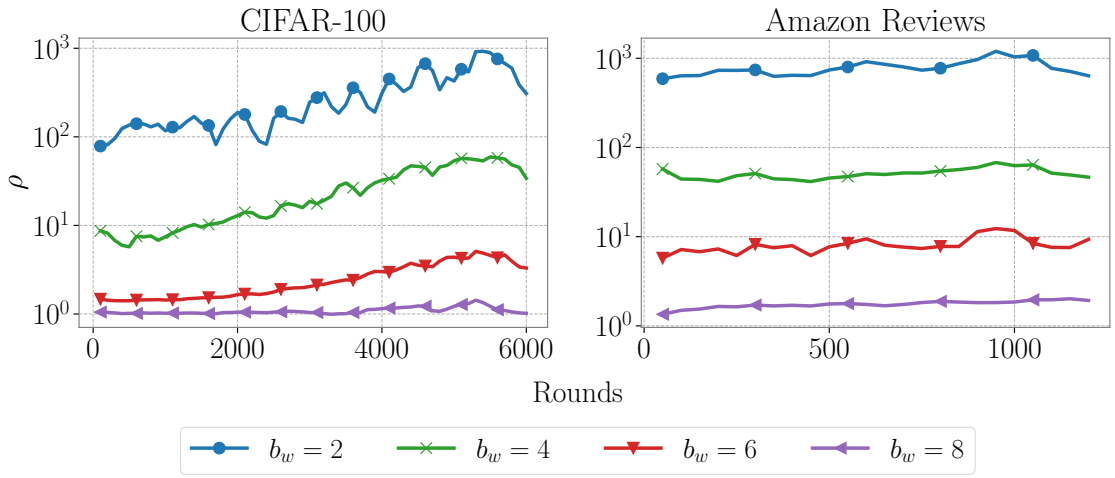

Figure 4. Client estimation error ratio ρ_t on the CIFAR-100 and Amazon Reviews tasks for different anchor compression budgets. For both tasks, we used 2 bits per coordinate for the correction and gradients (uplink) compression.

In Figure 4 we present the ratio $\rho_t \coloneqq \sum_{i\in\mathcal{S}_t} \hat{e}_{t,i}^2 / \sum_{i\in\mathcal{S}_t} e_{t,i}^2$ for different anchor compression budgets in CIFAR-100 and Amazon Reviews experiments. First, note that the ratio is mostly stable throughout the entire training. Additionally, when we increase the bandwidth for anchor compression, the ratio decreases, to the extent that for 8 bits per coordinate, the ratio is ≈ 1; this is when the error induced by the correction compression dominates the estimation error.

We note that our intuition gives rise to using an adaptive budget for anchor compression; since ρ_t can be measured by the PS (i.e., it has access to $y_t^i, \hat{y}_t^i, w_t$ and the correction compressor $\mathcal{C}_c$), we can keep track of it, and increase the anchor compression budget if ρ_t is too large. We leave such investigation to future work.

C. Proof of Theorem 3.4

In this section we prove Theorem 3.4, which we restate here for convenience,

Theorem 3.4. *Let $\tilde{\sigma}^2 \coloneqq \sigma^2 + 4\left(1-\frac{S}{N}\right)G^2, \gamma \coloneqq 1 + \left(1-\frac{S}{N}\right)\frac{B^2}{S}$, and $M \coloneqq f(w_0) - f^*$. Then, running DoCoFL with $\mathcal{C}_w$ and $\mathcal{C}_g$ as the identity mappings (and with appropriately selected η) guarantees*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(w_t)\|^2\right] \in \mathcal{O}\left(\sqrt{\frac{M\beta\tilde{\sigma}^2}{TS}} + \frac{(M^2\beta^2\omega^2 K\mathcal{V}\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}} + \frac{\gamma M\beta(\omega K\mathcal{V}+1)}{T}\right).$$

Proof. To simplify mathematical notations and computations, we analyze a more general framework than DoCoFL, where at each round, each client can download *any* model from up to $\mathcal{T}$ rounds prior to their participation round as anchor. This generalized policy is described in Algorithm 3.

Using Theorem 1, that proves the convergence of Algorithm 3 (see Appendix C.1), we prove Theorem 3.4. Namely, DoCoFL with $\mathcal{C}_w$ and $\mathcal{C}_g$ as identity mappings is a private case of Algorithm 3 where $\mathcal{T} = K\mathcal{V}$ and clients can only download models from specific prior rounds (multiplications of K). Thus, plugging-in $\mathcal{T} = K\mathcal{V}$ to Theorem 1 concludes the proof. □

C.1. Proof of Theorem 1

Theorem 1. *Suppose Assumptions 3.1-3.3 are satisfied. Let $\tilde{\sigma}^2 \coloneqq \sigma^2 + 4\left(1-\frac{S}{N}\right)G^2, \gamma \coloneqq 1+\left(1-\frac{S}{N}\right)\frac{B^2}{S}, \theta \coloneqq \omega\mathcal{T}+1$, and $M \coloneqq f(w_0) - f^*$. Then, running Algorithm 3 with $\eta = \min\left\{\frac{1}{30\gamma\beta\theta}, \sqrt{\frac{2MS}{\beta\tilde{\sigma}^2 T}}, \left(\frac{MS}{12\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2 T}\right)^{1/3}\right\}$ guarantees*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(w_t)\|^2\right] \leq 4\sqrt{\frac{2M\beta\tilde{\sigma}^2}{TS}} + 8\frac{(12M^2\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}} + \frac{120\gamma M\beta\theta}{T}. \tag{11}$$

Algorithm 3 Meta-Algorithm (generalization of DoCoFL)

Input: Initial weights $w_0 \in \mathbb{R}^d$, learning rate η, correction compression $\mathcal{C}_c$, client participation process $\mathcal{P}(\cdot)$
for $t = 0, \ldots, T-1$ **do**
  Obtain participating clients, $\mathcal{S}_t \leftarrow \mathcal{P}(t)$ ▷ $|\mathcal{S}_t| = S$
  for client $i \in \mathcal{S}_t$ in parallel **do**
    Obtain model weights (anchor), $y_t^i = w_{t-\tau_t^i}$ ▷ $\tau_t^i \in [0, \mathcal{T}]$
    Obtain compressed correction, $\hat{\Delta}_t^i = \mathcal{C}_c(w_t - y_t^i)$
    Construct model estimate, $\hat{w}_t^i = y_t^i + \hat{\Delta}_t^i$
    Compute local gradient, $\hat{g}_t^i = g_t^i(\hat{w}_t^i)$
    Communicate $\hat{g}_t^i$ back to server
  end for
  Aggregate local gradients, $\hat{g}_t \coloneqq \frac{1}{S}\sum_{i\in\mathcal{S}_t} \hat{g}_t^i$
  Update weights, $w_{t+1} = w_t - \eta \hat{g}_t$
end for

Proof. For the ease of notation, let $\nabla_t \coloneqq \nabla f(w_t)$ and $\tilde{\sigma}_S^2 \coloneqq \tilde{\sigma}^2/S$. Throughout our analysis, we sometimes use $\hat{w}_t^i$ even when $i \notin \mathcal{S}_t$, which is not well-defined. To resolve this, one can think about the following mathematically equivalent process, where at each round, all clients $i \in [N]$ obtain some previous model (anchor) y_t^i and the corresponding correction $\hat{\Delta}_t^i$, but only $i \in \mathcal{S}_t$ actually participate in the optimization. In that sense, for all $i \notin \mathcal{S}_t$, $\hat{w}_t^i$ is the estimated model of client i if it were to participate in round t.

Let $\hat{\nabla}_t \coloneqq \frac{1}{N}\sum_{i\in[N]} \nabla f_i(\hat{w}_t^i) = \mathbb{E}[\hat{g}_t]$. From the β-smoothness of the objective,

$$\begin{aligned}
\mathbb{E}[f(w_{t+1}) - f(w_t)] &\le -\eta \mathbb{E}[\hat{g}_t^\top \nabla_t] + \frac{\beta\eta^2}{2}\mathbb{E}\|\hat{g}_t\|^2 \\
&= -\eta \mathbb{E}[\hat{\nabla}_t^\top \nabla_t] + \frac{\beta\eta^2}{2}\mathbb{E}\|\hat{g}_t\|^2 \\
&= -\eta \mathbb{E}\|\nabla_t\|^2 + \underbrace{\eta \mathbb{E}[\nabla_t^\top(\nabla_t - \hat{\nabla}_t)]}_{=(A)} + \frac{\beta\eta^2}{2}\mathbb{E}\|\hat{g}_t\|^2 \,, \qquad (12)
\end{aligned}$$

where the first equality follows from the law of total expectation, and the second equality from the linearity of expectation.

Bounding (A)**:** Using the inequality $a^\top b \le \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$, we get that

$$\eta \mathbb{E}[\nabla_t^\top(\nabla_t - \hat{\nabla}_t)] \le \frac{\eta}{2}\mathbb{E}\|\nabla_t\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla_t - \hat{\nabla}_t\|^2 \,.$$

Focusing on the second term in the right-hand side, we have:

$$\begin{aligned}
\mathbb{E}\|\nabla_t - \hat{\nabla}_t\|^2 &= \mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N \left(\nabla f_i(w_t) - \nabla f_i(\hat{w}_t^i)\right)\right\|^2 \\
&\le \frac{1}{N}\sum_{i=1}^N \mathbb{E}\|\nabla f_i(w_t) - \nabla f_i(\hat{w}_t^i)\|^2 \\
&\le \frac{\beta^2}{N}\sum_{i=1}^N \mathbb{E}\|\hat{w}_t^i - w_t\|^2 \,, \qquad (13)
\end{aligned}$$

where in the first inequality we used Lemma C.5, and the second inequality follows from the β-smoothness of each f_i. Plugging back this bound, we get:

$$\eta \mathbb{E}[\nabla_t^\top(\nabla_t - \hat{\nabla}_t)] \le \frac{\eta}{2}\mathbb{E}\|\nabla_t\|^2 + \frac{\beta^2\eta}{2N}\sum_{i=1}^N \mathbb{E}\|\hat{w}_t^i - w_t\|^2 \,.$$

Using Lemma C.1 to bound $\mathbb{E}\|\hat{g}_t\|^2$ and the bound on (A), we get from Eq. (12) that

$$\begin{aligned}\mathbb{E}[f(w_{t+1}) - f(w_t)] \leq & -\frac{\eta}{2}\mathbb{E}\|\nabla_t\|^2 + \frac{\beta^2\eta}{2N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \\ & + \frac{\beta\eta^2}{2}\left(\tilde{\sigma}_S^2 + 4\gamma\mathbb{E}\|\nabla_t\|^2 + \frac{2\beta^2}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2\right) \\ = & \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right)\mathbb{E}\|\nabla_t\|^2 + \frac{\beta\eta^2\tilde{\sigma}_S^2}{2} \\ & + \left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\cdot\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \ .\end{aligned}$$

Applying Lemma C.2, we can bound $\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2$ to get that

$$\begin{aligned}\mathbb{E}[f(w_{t+1}) - f(w_t)] \leq & \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right)\mathbb{E}\|\nabla_t\|^2 + \frac{\beta\eta^2\tilde{\sigma}_S^2}{2} \\ & + \left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\left[\alpha\left(1 + \sum_{k=1}^{t}k(\rho\mathcal{T})^k\right)\tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}}\sum_{k=1}^{t}(\rho\mathcal{T})^k\sum_{\ell=t-k\mathcal{T}}^{t-k}\mathbb{E}\|\nabla_\ell\|^2\right] \\ = & \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right)\mathbb{E}\|\nabla_t\|^2 + \left[\frac{\beta\eta^2}{2} + \alpha\left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\left(1 + \sum_{k=1}^{t}k(\rho\mathcal{T})^k\right)\right]\tilde{\sigma}_S^2 \\ & + \left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\frac{2\gamma}{\beta^2\mathcal{T}}\underbrace{\sum_{k=1}^{t}(\rho\mathcal{T})^k\sum_{\ell=t-k\mathcal{T}}^{t-k}\mathbb{E}\|\nabla_\ell\|^2}_{=(B)}\ ,\end{aligned} \tag{14}$$

where $\alpha = 4\omega^2\eta^2\mathcal{T}$ and $\rho = 20\beta^2\omega^2\eta^2\mathcal{T}$. Since $\eta \leq \frac{1}{30\gamma\beta(\omega\mathcal{T}+1)} \leq \frac{1}{\sqrt{40}\beta\omega\mathcal{T}}$, it holds that $\rho\mathcal{T} = 20\beta^2\omega^2\mathcal{T}^2\eta^2 \leq 1/2 < 1$, and thus we can bound the coefficient of $\tilde{\sigma}_S^2$ using Lemma C.7 as

$$\sum_{k=1}^{t}k(\rho\mathcal{T})^k \leq \sum_{k=1}^{\infty}k(\rho\mathcal{T})^k = \frac{\rho\mathcal{T}}{(1-\rho\mathcal{T})^2} \leq 4\rho\mathcal{T} \leq 2\ , \tag{15}$$

where we used $\frac{1}{(1-\rho\mathcal{T})^2} \leq 4$, and $\rho\mathcal{T} \leq 1/2$.

Bounding (B): To bound (B), we change the summation order. Consider a fixed $\ell \in \mathbb{N}$. Note that $(\rho\mathcal{T})^k$ appears as a coefficient of $\mathbb{E}\|\nabla_\ell\|^2$ if and only if $t - k\mathcal{T} \leq \ell \leq t - k$, which is equivalent to $\frac{t-\ell}{\mathcal{T}} \leq k \leq t - \ell$. Therefore, we have

$$\begin{aligned}\sum_{k=1}^{t}(\rho\mathcal{T})^k\sum_{\ell=t-k\mathcal{T}}^{t-k}\mathbb{E}\|\nabla_\ell\|^2 &= \sum_{\ell=0}^{t-1}\left(\sum_{k=\lceil\frac{t-\ell}{\mathcal{T}}\rceil}^{t-\ell}(\rho\mathcal{T})^k\right)\mathbb{E}\|\nabla_\ell\|^2 \\ &\leq \sum_{\ell=0}^{t-1}\left(\sum_{k=\lceil\frac{t-\ell}{\mathcal{T}}\rceil}^{\infty}(\rho\mathcal{T})^k\right)\mathbb{E}\|\nabla_\ell\|^2 \\ &= \frac{1}{1-\rho\mathcal{T}}\sum_{\ell=0}^{t-1}(\rho\mathcal{T})^{\lceil\frac{t-\ell}{\mathcal{T}}\rceil}\mathbb{E}\|\nabla_\ell\|^2\end{aligned}$$

Plugging this bound and Eq. (15) back to Eq. (14) gives

$$\begin{aligned}\mathbb{E}[f(w_{t+1}) - f(w_t)] \leq & \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right)\mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 3\alpha\left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\right)\tilde{\sigma}_S^2 \\ & + \left(\eta + 2\beta\eta^2\right)\frac{\gamma}{(1-\rho\mathcal{T})\mathcal{T}}\sum_{k=0}^{t-1}(\rho\mathcal{T})^{\lceil\frac{t-k}{\mathcal{T}}\rceil}\mathbb{E}\|\nabla_k\|^2\ .\end{aligned}$$

Summing over $t = 0, \dots, T-1$, we obtain

$$\begin{aligned}\mathbb{E}[f(w_T) - f(w_0)] &= \sum_{t=0}^{T-1} \mathbb{E}[f(w_{t+1}) - f(w_t)] \\ &\leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 3\alpha\left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\right) T\tilde{\sigma}_S^2 \\ &\quad + \left(\eta + 2\beta\eta^2\right) \frac{\gamma}{(1-\rho\mathcal{T})\mathcal{T}} \underbrace{\sum_{t=0}^{T-1} \sum_{k=0}^{t-1} (\rho\mathcal{T})^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \mathbb{E}\|\nabla_k\|^2}_{=(C)} . \end{aligned} \tag{16}$$

Focusing on (C), we can change the outer summation bounds as

$$\sum_{t=0}^{T-1} \sum_{k=0}^{t-1} (\rho\mathcal{T})^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \mathbb{E}\|\nabla_k\|^2 = \sum_{t=1}^{T-1} \sum_{k=0}^{t-1} (\rho\mathcal{T})^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \mathbb{E}\|\nabla_k\|^2 \leq \sum_{t=1}^{T} \sum_{k=0}^{t-1} (\rho\mathcal{T})^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \mathbb{E}\|\nabla_k\|^2 . \tag{17}$$

Now, we can bound the right-hand side using Lemma C.8 with $a = \rho\mathcal{T} < 1$ and $x_k = \mathbb{E}\|\nabla_k\|^2 \geq 0$ to get that

$$\sum_{t=1}^{T} \sum_{k=0}^{t-1} (\rho\mathcal{T})^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \mathbb{E}\|\nabla_k\|^2 \leq \mathcal{T} \frac{\rho\mathcal{T}}{1-\rho\mathcal{T}} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 .$$

Plugging this bound back to Eq. (16) and using $\frac{1}{(1-\rho\mathcal{T})^2} \leq 4$ gives

$$\begin{aligned}\mathbb{E}[f(w_T) - f(w_0)] &\leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2\right) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 3\alpha\left(\frac{\beta^2\eta}{2} + \beta^3\eta^2\right)\right) T\tilde{\sigma}_S^2 \\ &\quad + \left(\eta + 2\beta\eta^2\right) \frac{\gamma\rho\mathcal{T}}{(1-\rho\mathcal{T})^2} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 \\ &\leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2 + 4\gamma\rho\mathcal{T}\left(\eta + 2\beta\eta^2\right)\right) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 \\ &\quad + \left(\frac{\beta\eta^2}{2} + \frac{3\alpha\beta^2}{2}\left(\eta + 2\beta\eta^2\right)\right) T\tilde{\sigma}_S^2 \\ &\leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2 + 8\gamma\rho\mathcal{T}\eta\right) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 3\alpha\beta^2\eta\right) T\tilde{\sigma}_S^2 , \end{aligned}$$

where in the last inequality we used the fact that $\eta \leq \frac{1}{30\gamma\beta\theta} \leq \frac{1}{30\beta}$ to bound $2\beta\eta^2 \leq \eta$.

Substituting α and ρ, we obtain:

$$\begin{aligned}\mathbb{E}[f(w_T) - f(w_0)] &\leq \left(-\frac{\eta}{2} + 2\gamma\beta\eta^2 + 160\gamma\beta^2\omega^2\mathcal{T}^2\eta^3\right) \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 \\ &\quad + \left(\frac{\beta\eta^2}{2} + 12\beta^2\omega^2\mathcal{T}\eta^3\right) T\tilde{\sigma}_S^2 . \end{aligned}$$

Since $\eta \leq \frac{1}{30\gamma\beta\theta}$, we can bound the coefficient of $\sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2$ using Lemma C.9. We get:

$$\mathbb{E}[f(w_T) - f(w_0)] \leq -\frac{\eta}{4} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla_t\|^2 + \left(\frac{\beta\eta^2}{2} + 12\beta^2\omega^2\mathcal{T}\eta^3\right) T\tilde{\sigma}_S^2 . \tag{18}$$

Rearranging terms, multiplying by $4/\eta T$, and plugging $\tilde{\sigma}_S^2 = \tilde{\sigma}^2/S$ then gives

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla_t\|^2\right] \leq \frac{4M}{\eta T} + \frac{2\beta\tilde{\sigma}^2}{S}\eta + \frac{48\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2}{S}\eta^2 \,,$$

where we also used $\mathbb{E}[f(w_0) - f(w_T)] \leq f(w_0) - f^* \leq M$. Applying Lemma C.10 with our learning rate η, we finally obtain:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla_t\|^2\right] \leq& \frac{4M}{T}\left(30\gamma\beta\theta + \sqrt{\frac{\beta\tilde{\sigma}^2 T}{2MS}} + \left(\frac{12\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2 T}{MS}\right)^{1/3}\right) + \frac{2\beta\tilde{\sigma}^2}{S}\cdot\sqrt{\frac{2MS}{\beta\tilde{\sigma}^2 T}} \\ &+ \frac{48\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2}{S}\cdot\left(\frac{MS}{12\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2 T}\right)^{2/3} \\ =& 4\sqrt{\frac{2M\beta\tilde{\sigma}^2}{TS}} + 8\frac{(12M^2\beta^2\omega^2\mathcal{T}\tilde{\sigma}^2)^{1/3}}{T^{2/3}S^{1/3}} + \frac{120\gamma\beta\theta}{T} \,,\end{aligned}$$

which concludes the proof.

□

C.2. Technical Lemmata

In this section, we introduce some technical results used throughout our analysis. We start with the following lemma, yielding a bound on the second moment of the aggregated gradients that our PS uses to update its model.

Lemma C.1. *Consider the notations of Theorem 1. For every $t \in [T]$, it holds that*

$$\mathbb{E}\|\hat{g}_t\|^2 \leq \tilde{\sigma}_S^2 + 4\gamma\mathbb{E}\|\nabla_t\|^2 + 2\beta^2\cdot\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \,. \tag{19}$$

Proof. Since $\hat{g}_t$ is an aggregation of local gradient, we can write,

$$\begin{aligned}\mathbb{E}\|\hat{g}_t\|^2 &= \mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\hat{g}_t^i\right\|^2 = \mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\hat{g}_t^i - \nabla f_i(\hat{w}_t^i) + \nabla f_i(\hat{w}_t^i)\right)\right\|^2 \\ &= \mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\hat{g}_t^i - \nabla f_i(\hat{w}_t^i)\right)\right\|^2 + \mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(\hat{w}_t^i)\right\|^2 \,,\end{aligned}$$

where the last equality follows from Assumption 3.1 as $\mathbb{E}[(\hat{g}_t^i(\hat{w}_t^i) - \nabla f_i(\hat{w}_t^i))^\top \nabla f_i(\hat{w}_t^i)] = 0$. Note that the first term in the right-hand side is the variance of the average of S independent random variables with zero mean and variance bounded by σ^2; therefore, it is bounded by σ^2/S. Thus, we get that

$$\mathbb{E}\|\hat{g}_t\|^2 \leq \frac{\sigma^2}{S} + \mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(\hat{w}_t^i)\right\|^2 \,. \tag{20}$$

Focusing on the second term in the right-hand side, we have that

$$\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(\hat{w}_t^i)\right\|^2 \leq \underbrace{2\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\nabla f_i(\hat{w}_t^i) - \nabla f_i(w_t)\right)\right\|^2}_{(A)} + \underbrace{2\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(w_t)\right\|^2}_{=(B)} \,, \tag{21}$$

where we used the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.

Bounding (A)**:** Using Lemma C.5 and the β-smoothness of the objective, we get that

$$\begin{aligned}
2\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\nabla f_i(\hat{w}_t^i)-\nabla f_i(w_t)\right)\right\|^2 &\leq \frac{2}{S}\mathbb{E}\left[\sum_{i\in\mathcal{S}_t}\|\nabla f_i(\hat{w}_t^i)-\nabla f_i(w_t)\|^2\right] \\
&\leq \frac{2\beta^2}{S}\mathbb{E}\left[\sum_{i\in\mathcal{S}_t}\|\hat{w}_t^i-w_t\|^2\right] \\
&= \frac{2\beta^2}{S}\mathbb{E}\left[\sum_{i=1}^{N}\|\hat{w}_t^i-w_t\|^2\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}\right] \\
&= \frac{2\beta^2}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i-w_t\|^2 \ ,
\end{aligned}$$

where the last equality follows from our assumption about the client participation process $\mathcal{P}(\cdot)$, which guarantees that $\mathbb{P}(i\in\mathcal{S}_t)=S/N$, independently of the optimization process.

Bounding (B)**:** By the law of total expectation, (B) can be written as follows,

$$2\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(w_t)\right\|^2 = 2\mathbb{E}\left\|\sum_{i=1}^{N}\left(\frac{1}{S}\nabla f_i(w_t)\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}\right)\right\|^2 = 2\mathbb{E}\left[\mathbb{E}\left[\left\|\sum_{i=1}^{N}\left(\frac{1}{S}\nabla f_i(w_t)\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}\right)\right\|^2\ \middle|\ w_t\right]\right] \ . \tag{22}$$

Thus, we can use Lemma C.3 with $X_i=\frac{1}{S}\nabla f_i(w_t)\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}},\ i\in[N]$ to bound the inner expectation. Using $\mathbb{P}(i\in\mathcal{S}_t)=S/N$, we have that

$$\mathbb{E}[X_i|w_t]=\frac{1}{S}\nabla f_i(w_t)\cdot\frac{S}{N}=\frac{1}{N}\nabla f_i(w_t),$$

and,

$$\begin{aligned}
\mathbb{E}[\|X_i-\mathbb{E}[X_i|w_t]\|^2|w_t] &= \|\nabla f_i(w_t)\|^2\cdot\mathbb{E}\left[\left(\frac{1}{S}\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}-\frac{1}{N}\right)^2\right] = \|\nabla f(w_t)\|^2\cdot\mathrm{Var}\left(\frac{1}{S}\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}\right) \\
&= \frac{\|\nabla f(w_t)\|^2}{S^2}\cdot\frac{S}{N}\left(1-\frac{S}{N}\right) = \frac{\|\nabla f_i(w_t)\|^2}{SN}\left(1-\frac{S}{N}\right) \ ,
\end{aligned}$$

where we used the fact that for any event $\mathcal{A}$, the following holds: $\mathrm{Var}(\mathbb{1}_{\mathcal{A}})=\mathbb{P}(\mathcal{A})\cdot(1-\mathbb{P}(\mathcal{A}))$. Therefore, using Lemma C.3, we obtain that

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{i=1}^{N}\left(\frac{1}{S}\nabla f_i(w_t)\cdot\mathbb{1}_{\{i\in\mathcal{S}_t\}}\right)\right\|^2\ \middle|\ w_t\right] &\leq 2\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(w_t)\right\|^2+\frac{2}{SN}\left(1-\frac{S}{N}\right)\sum_{i=1}^{N}\|\nabla f_i(w_t)\|^2 \\
&\leq 2\|\nabla_t\|^2+\frac{2}{S}\left(1-\frac{S}{N}\right)\left(G^2+B^2\|\nabla_t\|^2\right) \\
&= \left(1-\frac{S}{N}\right)\frac{2G^2}{S}+2\underbrace{\left(1+\left(1-\frac{S}{N}\right)\frac{B^2}{S}\right)}_{:=\gamma}\|\nabla_t\|^2 \ ,
\end{aligned}$$

where in the second inequality we used the bounded gradient dissimilarity assumption (Assumption 3.2). Plugging back to Eq. (22), we get the following bound on (B):

$$2\mathbb{E}\left\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\nabla f_i(w_t)\right\|^2\leq\left(1-\frac{S}{N}\right)\frac{4G^2}{S}+4\gamma\mathbb{E}\|\nabla_t\|^2 \ .$$

Plugging the bounds on (A) and (B) in Eq. (20) finally gives

$$\mathbb{E}\|\hat{g}_t\|^2 \leq \underbrace{\frac{\sigma^2}{S} + \left(1 - \frac{S}{N}\right)\frac{4G^2}{S}}_{=\tilde{\sigma}^2/S} + 4\gamma\mathbb{E}\|\nabla_t\|^2 + \frac{2\beta^2}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2$$

$$= \tilde{\sigma}_S^2 + 4\gamma\mathbb{E}\|\nabla_t\|^2 + \frac{2\beta^2}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \,,$$

which concludes the proof. □

The next result establishes a bound on the model estimation error of the clients.

Lemma C.2. *Consider the notations of Theorem 1. Let $\alpha \coloneqq 4\omega^2\eta^2\mathcal{T}$, $\rho \coloneqq 20\beta^2\omega^2\eta^2\mathcal{T}$, and $\nabla_{-\ell} \coloneqq 0, \ \forall \ell \in \mathbb{N}$. Then, the following result holds:*

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \leq \alpha\left(1 + \sum_{k=1}^{t}k(\rho\mathcal{T})^k\right)\tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}}\sum_{k=1}^{t}(\rho\mathcal{T})^k\sum_{\ell=t-k\mathcal{T}}^{t-k}\mathbb{E}\|\nabla_\ell\|^2$$

Proof. We use strong induction. Particularly, to prove the result holds at round t, we rely on its correctness over the $\mathcal{T}$ prior rounds, i.e., for every $s = t-\mathcal{T}, \ldots, t-1$. Thus, in our base case, we show that the result holds up to round $\mathcal{T}$.
We start with some general observations that hold for any t. Recall that $\hat{w}_t^i = y_t^i + \mathcal{C}(w_t - y_t^i)$. From Assumption 3.3, we have

$$\mathbb{E}\|\hat{w}_t^i - w_t\|^2 = \mathbb{E}\|\mathcal{C}_c(w_t - y_t^i) - (w_t - y_t^i)\|^2 \leq \omega^2\mathbb{E}\|w_t - y_t^i\|^2 \,. \tag{23}$$

Unrolling the update rule for w_t, we have for all $i \in [N]$ that

$$w_t = w_{t-\tau_t^i} - \eta\sum_{k=t-\tau_t^i}^{t-1}\hat{g}_k = y_t^i - \eta\sum_{k=t-\tau_t^i}^{t-1}\hat{g}_k \,.$$

Let $\hat{g}_{-k} \coloneqq 0$ for all $k \in \mathbb{N}$. Additionally, let $\hat{\xi}_k = \hat{g}_k - \hat{\nabla}_k$ for all k, where $\hat{\nabla}_k = \mathbb{E}[\hat{g}_k]$, as defined in the proof of Theorem 1. Plugging back to Eq. (23), we get that

$$\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \leq \omega^2\eta^2\mathbb{E}\|\sum_{k=t-\tau_t^i}^{t-1}\hat{g}_k\|^2 \leq 2\omega^2\eta^2\mathbb{E}\|\sum_{k=t-\tau_t^i}^{t-1}\hat{\nabla}_k\|^2 + 2\omega^2\eta^2\mathbb{E}\|\sum_{k=t-\tau_t^i}^{t-1}\hat{\xi}_k\|^2 \,, \tag{24}$$

where the last inequality follows from $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$. Using Lemma C.5, we can bound the first term in the right-hand side, as

$$\mathbb{E}\|\sum_{k=t-\tau_t^i}^{t-1}\hat{\nabla}_k\|^2 \leq \tau_t^i\sum_{k=t-\tau_t^i}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 \leq \mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 \,,$$

where the last inequality follows from $\tau_t^i \leq \mathcal{T}$. Since $\mathbb{E}[\hat{\xi}_k] = 0$, and $\mathbb{E}[\hat{\xi}_k^\top\hat{\xi}_\ell] = 0$ for all k, ℓ, we can apply Lemma C.4 to bound the second term in the right-hand side as follows:

$$\mathbb{E}\|\sum_{k=t-\tau_t^i}^{t-1}\hat{\xi}_k\|^2 = \sum_{k=t-\tau_t^i}^{t-1}\mathbb{E}\|\hat{\xi}_k\|^2 \leq \sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{\xi}_k\|^2 \leq 2\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 + 2\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{g}_k\|^2 \,,$$

where we used $\tau_t^i \leq \mathcal{T}$, and $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.

Plugging-in both bounds to Eq. (24), we obtain:

$$\mathbb{E}\|\hat{w}_t^i - w_t\|^2 \leq (2\omega^2\eta^2\mathcal{T} + 4\omega^2\eta^2)\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 + 4\omega^2\eta^2\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{g}_k\|^2$$

$$\leq 6\omega^2\eta^2\mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{\nabla}_k\|^2 + 4\omega^2\eta^2\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{g}_k\|^2 \,. \tag{25}$$

Note that we can bound $\mathbb{E}\|\hat{\nabla}_k\|^2$ as:

$$\mathbb{E}\|\hat{\nabla}_k\|^2 \leq 2\mathbb{E}\|\hat{\nabla}_k - \nabla_k\|^2 + 2\mathbb{E}\|\nabla_k\|^2 \leq \frac{2\beta^2}{N}\sum_{i=1}^{N}\mathbb{E}\|\hat{w}_k^i - w_k\|^2 + 2\mathbb{E}\|\nabla_k\|^2 \ ,$$

where in the last inequality we used Eq. (13) to bound $\mathbb{E}\|\hat{\nabla}_k - \nabla_k\|^2$.

For the ease of notation, denote: $e_t^i \coloneqq \mathbb{E}\|\hat{w}_t^i - w_t\|^2$, and $e_t \coloneqq \frac{1}{N}\sum_{i=1}^{N} e_t^i$. Therefore, we obtain from Eq. (25) that

$$e_t^i \leq 12\beta^2\omega^2\eta^2\mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1} e_k + 12\omega^2\eta^2\mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\nabla_k\|^2 + 4\omega^2\eta^2\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\hat{g}_k\|^2 \ . \tag{26}$$

Base Case: For $t = 0$, each client obtains the exact model weights, i.e., $\hat{w}_0^i = w_0$, which trivially implies result. For every $t = 1, \ldots, \mathcal{T}$ and $i \in [N]$, we have from Eq. (26) that

$$e_t^i \leq 12\beta^2\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1} e_k + 12\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + 4\omega^2\eta^2\sum_{k=0}^{t-1}\mathbb{E}\|\hat{g}_k\|^2 \ . \tag{27}$$

Using Lemma C.1 to bound $\mathbb{E}\|\hat{g}_k\|^2$, we get:

$$\begin{aligned}
e_t^i &\leq 12\beta^2\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1} e_k + 12\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + 4\omega^2\eta^2\sum_{k=0}^{t-1}\left(\tilde{\sigma}_S^2 + 4\gamma\mathbb{E}\|\nabla_k\|^2 + 2\beta^2 e_k\right) \\
&\leq 4\omega^2\eta^2\mathcal{T}\tilde{\sigma}_S^2 + \left(12\omega^2\eta^2\mathcal{T} + 16\gamma\omega^2\eta^2\right)\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \left(12\beta^2\omega^2\eta^2\mathcal{T} + 8\beta^2\omega^2\eta^2\right)\sum_{k=0}^{t-1} e_k \\
&\leq 4\omega^2\eta^2\mathcal{T}\tilde{\sigma}_S^2 + 28\gamma\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + 20\beta^2\omega^2\eta^2\mathcal{T}\sum_{k=0}^{t-1} e_k \ .
\end{aligned}$$

Note that this bound on e_t^i is independent of i, and thus, it holds for the average of e_t^i over $i \in [N]$, namely, e_t. Therefore, Eq. (26) implies a recursive bound on e_t; for every $t = 1, \ldots, \mathcal{T}$:

$$e_t \leq \alpha\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \rho\sum_{k=0}^{t-1} e_k \ , \tag{28}$$

where we denoted $\nu \coloneqq 28\gamma\omega^2\eta^2\mathcal{T}$. Plugging-in this bound instead of e_k in the right-hand side, we obtain:

$$\begin{aligned}
e_t &\leq \alpha\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \rho\sum_{k=0}^{t-1}\left(\alpha\tilde{\sigma}_S^2 + \nu\sum_{\ell=0}^{k-1}\mathbb{E}\|\nabla_\ell\|^2 + \rho\sum_{\ell=0}^{k-1} e_\ell\right) \\
&\leq \alpha\left(1 + \rho\mathcal{T}\right)\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \nu\rho\sum_{k=0}^{t-1}\sum_{\ell=0}^{k-1}\mathbb{E}\|\nabla_\ell\|^2 + \rho^2\sum_{k=0}^{t-1}\sum_{\ell=0}^{k-1} e_\ell \ ,
\end{aligned}$$

where we used $t \leq \mathcal{T}$. Note that we can bound the double sums in right-hand side using Lemma C.6 as

$$\sum_{k=0}^{t-1}\sum_{\ell=0}^{k-1}\mathbb{E}\|\nabla_\ell\|^2 \leq t\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_k\|^2 \leq \mathcal{T}\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_k\|^2 \ ,$$

and similarly,

$$\sum_{k=0}^{t-1}\sum_{\ell=0}^{k-1} e_\ell \leq \mathcal{T}\sum_{k=0}^{t-2} e_k \ .$$

Plugging-back, we get:

$$e_t \leq \alpha\left(1+\rho\mathcal{T}\right)\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \nu\rho\mathcal{T}\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_\ell\|^2 + \rho^2\mathcal{T}\sum_{k=0}^{t-2}e_k \ .$$

We can once again apply Eq. (28) to bound e_k, and obtain:

$$\begin{aligned}
e_t \leq& \alpha\left(1+\rho\mathcal{T}\right)\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \nu\rho\mathcal{T}\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_\ell\|^2 + \rho^2\mathcal{T}\sum_{k=0}^{t-2}\left(\alpha\tilde{\sigma}_S^2 + \nu\sum_{\ell=0}^{k-1}\mathbb{E}\|\nabla_\ell\|^2 + \rho\sum_{\ell=0}^{k-1}e_\ell\right)\\
\leq& \alpha\left(1+\rho\mathcal{T}+\rho^2\mathcal{T}^2\right)\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \nu\rho\mathcal{T}\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_\ell\|^2 + \nu\rho^2\mathcal{T}\sum_{k=0}^{t-2}\sum_{\ell=0}^{k-1}\mathbb{E}\|\nabla_\ell\|^2 + \rho^3\mathcal{T}\sum_{k=0}^{t-2}\sum_{\ell=0}^{k-1}e_\ell\\
\leq& \alpha\left(1+\rho\mathcal{T}+\rho^2\mathcal{T}^2\right)\tilde{\sigma}_S^2 + \nu\sum_{k=0}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \nu\rho\mathcal{T}\sum_{k=0}^{t-2}\mathbb{E}\|\nabla_\ell\|^2 + \nu\rho^2\mathcal{T}^2\sum_{k=0}^{t-3}\mathbb{E}\|\nabla_\ell\|^2 + \rho^3\mathcal{T}^2\sum_{k=0}^{t-3}e_k \ ,
\end{aligned}$$

where in the last inequality we used Lemma C.6. Repeating this process of alternately applying Eq. (28) to bound e_k and Lemma C.6, finally gives:

$$e_t \leq \alpha\left(1+\sum_{k=1}^{t}\left(\rho\mathcal{T}\right)^k\right)\tilde{\sigma}_S^2 + \frac{\nu}{\rho\mathcal{T}}\sum_{k=1}^{t}\left(\rho\mathcal{T}\right)^k\sum_{\ell=0}^{t-k}\mathbb{E}\|\nabla_\ell\|^2 \ .$$

Plugging ν and ρ, we can bound the coefficient $\nu/\rho\mathcal{T}$ as:

$$\frac{\nu}{\rho\mathcal{T}} = \frac{28\gamma\omega^2\eta^2\mathcal{T}}{20\beta^2\omega^2\eta^2\mathcal{T}^2} \leq \frac{2\gamma}{\beta^2\mathcal{T}} \ .$$

Using $(\rho\mathcal{T})^k \leq k(\rho\mathcal{T})^k$, which holds for any $k \geq 1$, we then obtain:

$$e_t \leq \alpha\left(1+\sum_{k=1}^{t}k\left(\rho\mathcal{T}\right)^k\right)\tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}}\sum_{k=1}^{t}\left(\rho\mathcal{T}\right)^k\sum_{\ell=0}^{t-k}\mathbb{E}\|\nabla_\ell\|^2 \ .$$

Note that for all $t \leq \mathcal{T}$ and $k \geq 1$, we have $t - k\mathcal{T} \leq 0$. Therefore, since for $\nabla_{-\ell} = 0$ for all $\ell \in \mathbb{N}$, we can equivalently write:

$$e_t \leq \alpha\left(1+\sum_{k=1}^{t}k\left(\rho\mathcal{T}\right)^k\right)\tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}}\sum_{k=1}^{t}\left(\rho\mathcal{T}\right)^k\sum_{\ell=t-k\mathcal{T}}^{t-k}\mathbb{E}\|\nabla_\ell\|^2 \ ,$$

which establishes the result for the base case.

Induction step: The induction hypothesis is that the following holds:

$$e_s \leq \alpha\left(1+\sum_{k=1}^{s}k\left(\rho\mathcal{T}\right)^k\right)\tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}}\sum_{k=1}^{s}\left(\rho\mathcal{T}\right)^k\sum_{\ell=s-k\mathcal{T}}^{s-k}\mathbb{E}\|\nabla_\ell\|^2, \quad \forall s = t-\mathcal{T},\ldots,t-1 \ . \tag{29}$$

We focus on Eq. (26). Using Lemma C.1 to bound $\mathbb{E}\|\hat{g}_k\|^2$ and following similar steps to those used to derive Eq. (28), we get:

$$\begin{aligned}
e_t^i \leq& 12\beta^2\omega^2\eta^2\mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1}e_k + 12\omega^2\eta^2\mathcal{T}\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\nabla_k\|^2 + 4\omega^2\eta^2\sum_{k=t-\mathcal{T}}^{t-1}\left(\tilde{\sigma}_S^2 + 4\gamma\mathbb{E}\|\nabla_k\|^2 + 2\beta^2e_k\right)\\
\leq& \alpha\tilde{\sigma}_S^2 + \nu\sum_{k=t-\mathcal{T}}^{t-1}\mathbb{E}\|\nabla_k\|^2 + \rho\underbrace{\sum_{k=t-\mathcal{T}}^{t-1}e_k}_{=(\dagger)} \ .
\end{aligned} \tag{30}$$

From the induction hypothesis (29), we can bound e_k for every $k \in [t-\mathcal{T}, t-1]$ as follows:

$$e_k \leq \alpha \left(1 + \sum_{\ell=1}^{k} \ell(\rho\mathcal{T})^\ell\right) \tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{\ell=1}^{k} (\rho\mathcal{T})^\ell \sum_{m=k-\ell\mathcal{T}}^{k-\ell} \mathbb{E}\|\nabla_m\|^2 \,.$$

Denote this bound by $B(k) \coloneqq \alpha \left(1 + \sum_{\ell=1}^{k} \ell(\rho\mathcal{T})^\ell\right) \tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{\ell=1}^{k} (\rho\mathcal{T})^\ell \sum_{m=k-\ell\mathcal{T}}^{k-\ell} \mathbb{E}\|\nabla_m\|^2$; that is, $e_k \leq B(k)$. We can therefore bound $(\dagger)$ as

$$\sum_{k=t-\mathcal{T}}^{t-1} e_k \leq \sum_{k=t-\mathcal{T}}^{t-1} B(k) \leq \mathcal{T} B(t-1) \,,$$

where the last inequality holds because $B(k)$ is monotonically increasing. Plugging back to Eq. (30) and substituting $B(t-1)$ gives

$$\begin{aligned}
e_t^i \leq & \alpha\tilde{\sigma}_S^2 + \nu \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 + \rho \cdot \mathcal{T} B(t-1) \\
= & \alpha\tilde{\sigma}_S^2 + \nu \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 + \rho\mathcal{T} \left(\alpha \left(1 + \sum_{k=1}^{t-1} k(\rho\mathcal{T})^k\right) \tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=1}^{t-1} (\rho\mathcal{T})^k \sum_{\ell=t-1-k\mathcal{T}}^{t-1-k} \mathbb{E}\|\nabla_\ell\|^2\right) \\
= & \underbrace{\alpha \left(1 + \rho\mathcal{T} + \sum_{k=1}^{t-1} k(\rho\mathcal{T})^{k+1}\right)}_{=(A)} \tilde{\sigma}_S^2 + \underbrace{\nu \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=1}^{t-1} (\rho\mathcal{T})^{k+1} \sum_{\ell=t-1-k\mathcal{T}}^{t-1-k} \mathbb{E}\|\nabla_\ell\|^2}_{=(B)} \,.
\end{aligned} \tag{31}$$

Bounding (A): Using simple algebra, we have that

$$\rho\mathcal{T} + \sum_{k=1}^{t-1} k(\rho\mathcal{T})^{k+1} = \rho\mathcal{T} + \sum_{k=2}^{t} (k-1)(\rho\mathcal{T})^k \leq \rho\mathcal{T} + \sum_{k=2}^{t} k(\rho\mathcal{T})^k = \sum_{k=1}^{t} k(\rho\mathcal{T})^k \,. \tag{32}$$

This implies that (A) is bounded by $\alpha \left(1 + \sum_{k=1}^{t} k(\rho\mathcal{T})^k\right)$.

Bounding (B): Focusing on the first term in (B), we can bound:

$$\nu \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 = \frac{\nu}{\rho\mathcal{T}} \cdot \rho\mathcal{T} \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 \leq \frac{2\gamma}{\beta^2\mathcal{T}} \cdot \rho\mathcal{T} \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 \,. \tag{33}$$

Focusing on the second sum in (B), we can bound

$$\begin{aligned}
\frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=1}^{t-1} (\rho\mathcal{T})^{k+1} \sum_{\ell=t-1-k\mathcal{T}}^{t-1-k} \mathbb{E}\|\nabla_\ell\|^2 &= \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=2}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-1-(k-1)\mathcal{T}}^{t-1-(k-1)} \mathbb{E}\|\nabla_\ell\|^2 \\
&= \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=2}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-k\mathcal{T}+\mathcal{T}-1}^{t-k} \mathbb{E}\|\nabla_\ell\|^2 \\
&\leq \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=2}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-k\mathcal{T}}^{t-k} \mathbb{E}\|\nabla_\ell\|^2 \,,
\end{aligned} \tag{34}$$

where the last inequality holds since $\mathcal{T} - 1 \geq 0$ and $\mathbb{E}\|\nabla_\ell\|^2 \geq 0$ for all ℓ. Combining the bounds in Eq. (33) and (34), we can then bound (B) as

$$\begin{aligned}
\nu \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=1}^{t-1} (\rho\mathcal{T})^{k+1} \sum_{\ell=t-1-k\mathcal{T}}^{t-1-k} \mathbb{E}\|\nabla_\ell\|^2 &\leq \frac{2\gamma}{\beta^2\mathcal{T}} \cdot \rho\mathcal{T} \sum_{k=t-\mathcal{T}}^{t-1} \mathbb{E}\|\nabla_k\|^2 + \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=2}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-k\mathcal{T}}^{t-k} \mathbb{E}\|\nabla_\ell\|^2 \\
&= \frac{2\gamma}{\beta^2\mathcal{T}} \sum_{k=1}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-k\mathcal{T}}^{t-k} \mathbb{E}\|\nabla_\ell\|^2 \,.
\end{aligned} \tag{35}$$

Plugging back to Eq. (31) the bounds on (A) and (B) (from Eq. (32) and (35), respectively), we get:

$$e_t^i \leq \alpha \left(1 + \sum_{k=1}^{t} k(\rho\mathcal{T})^k\right) \tilde{\sigma}_S^2 + \frac{2\gamma}{\beta^2 \mathcal{T}} \sum_{k=1}^{t} (\rho\mathcal{T})^k \sum_{\ell=t-k\mathcal{T}}^{t-k} \mathbb{E}\|\nabla_\ell\|^2 \ .$$

Since this bound is independent of i, it also holds for the average $e_t = \frac{1}{N}\sum_{i=1}^N e_t^i$, establishing the result. □

In the following two lemmas, we characterize the second moment of the sum of independent random variables.

Lemma C.3 (Lemma 4, Karimireddy et al., 2020)**.** *Let $X_1, \ldots, X_N \in \mathbb{R}^d$ be N independent random variables. Suppose that $\mathbb{E}[X_i] = \mu_i$ and $\mathbb{E}\|X_i - \mu_i\|^2 \leq \sigma_i^2$. Then, the following holds*

$$\mathbb{E}\left\|\sum_{i=1}^N X_i\right\|^2 \leq 2\left\|\sum_{i=1}^N \mu_i\right\|^2 + 2\sum_{i=1}^N \sigma_i^2 \ .$$

Lemma C.4. *Let $X_1, \ldots, X_N \in \mathbb{R}^d$ be N orthogonal, zero mean random variables, i.e., $\mathbb{E}[X_i] = 0$ for all $i \in [N]$, and $\mathbb{E}[X_i^\top X_j] = 0$ for all $i \neq j$. Then, the following holds:*

$$\mathbb{E}\left\|\sum_{i=1}^N X_i\right\|^2 = \sum_{i=1}^N \mathbb{E}\|X_i\|^2 \ .$$

Proof. By the linearity of expectation, and the following property: $\mathbb{E}[X_i^\top X_j] = 0, \ \forall i \neq j$, we immediately get that $\mathbb{E}\|\sum_{i=1}^N X_i\|^2 = \mathbb{E}\left[\sum_{i=1}^N \sum_{j=1}^N X_i^\top X_j\right] = \sum_{i=1}^N \mathbb{E}\|X_i\|^2$. □

Next, we state a simple result about the squared norm of the sum of vectors.

Lemma C.5. *For any $u_1, \ldots, u_N \in \mathbb{R}^d$, it holds that $\|\sum_{i=1}^N u_i\|^2 \leq N \sum_{i=1}^N \|u_i\|^2$.*

Proof. By the convexity of $\|\cdot\|^2$ and Jensen's inequality: $\|\frac{1}{N}\sum_{i=1}^N u_i\|^2 \leq \frac{1}{N}\sum_{i=1}^N \|u_i\|^2$, which implies the result. □

The next result is a simple bound on a double sum of non-negative numbers.

Lemma C.6. *Let $t, \tau \in \mathbb{N}$ such that $t \geq \tau + 1$. For any sequence of non-negative numbers $x_0, x_1, \ldots, x_{t-\tau-1}$, the following holds*

$$\sum_{k=0}^{t-\tau} \sum_{\ell=0}^{k-1} x_\ell \leq t \cdot \sum_{k=0}^{t-\tau-1} x_k \ .$$

Proof. Immediately: $\sum_{k=0}^{t-\tau} \sum_{\ell=0}^{k-1} x_\ell = \sum_{k=0}^{t-\tau-1} (t-\tau-k) x_k \leq t \cdot \sum_{k=0}^{t-\tau-1} x_k$. □

The following lemma gives a bound on the derivative of a power series.

Lemma C.7. *Let $a < 1$. Then,*

$$\sum_{k=1}^{\infty} k a^k = \frac{a}{(1-a)^2} \ .$$

Proof. Let $f_k(a) = a^k$.

$$\sum_{k=1}^{\infty} k a^k = a \sum_{k=1}^{\infty} k a^{k-1} = a \sum_{k=1}^{\infty} f_k'(a) \ .$$

Using term-by-term differentiation (Stewart, 2015), we have that

$$\sum_{k=1}^{\infty} f_k'(a) = \left(\sum_{k=1}^{\infty} f_k(a)\right)' = \left(\sum_{k=1}^{\infty} a^k\right)' = \left(\frac{a}{1-a}\right)' = \frac{1}{(1-a)^2} \ .$$

Multiplying by a gives the result. □

Next, we state a non-trivial inequality to bound the double sum that appears on the right-hand side of Eq. (17).

Lemma C.8. *Let $a \in (0,1)$ and $T, \mathcal{T} \in \mathbb{N}$. Moreover, let $x_0, \ldots, x_{T-1}$ be a sequence of non-negative numbers. Then,*

$$\sum_{t=1}^{T} \sum_{k=0}^{t-1} a^{\lceil \frac{t-k}{\mathcal{T}} \rceil} x_k \leq \mathcal{T} \frac{a}{1-a} \sum_{k=0}^{T-1} x_k \ .$$

Proof. We start with changing the order of summation in the left-hand side. Note that for any fixed k, the element x_k appears in the inner sum if $k \leq t-1$, or equivalently, $t \geq k+1$. Therefore,

$$\sum_{t=1}^{T} \sum_{k=0}^{t-1} a^{\lceil \frac{t-k}{\mathcal{T}} \rceil} x_k = \sum_{k=0}^{T-1} \left(\sum_{t=k+1}^{T} a^{\lceil \frac{t-k}{\mathcal{T}} \rceil} \right) x_k = \sum_{k=0}^{T-1} \left(\sum_{t=1}^{T-k} a^{\lceil \frac{t}{\mathcal{T}} \rceil} \right) x_k \ . \tag{36}$$

Focusing on the inner sum in the right-hand side, $\sum_{t=1}^{T-k} a^{\lceil t/\mathcal{T} \rceil}$, we can divide the interval of integers from 1 to $T-k$ into non-overlapping intervals of length $\mathcal{T}$ (and possibly a small residual) and get that

$$\sum_{t=1}^{T-k} a^{\lceil \frac{t}{\mathcal{T}} \rceil} \leq \sum_{m=1}^{\lceil \frac{T-k}{\mathcal{T}} \rceil} \sum_{\ell=1}^{\mathcal{T}} a^{\lceil \frac{(m-1)\mathcal{T}+\ell}{\mathcal{T}} \rceil} \overset{(\dagger)}{=} \sum_{m=1}^{\lceil \frac{T-k}{\mathcal{T}} \rceil} \sum_{\ell=1}^{\mathcal{T}} a^m = \mathcal{T} \sum_{m=1}^{\lceil \frac{T-k}{\mathcal{T}} \rceil} a^m \leq \mathcal{T} \frac{a}{1-a} \ .$$

where $(\dagger)$ holds because for every $\ell = 1, \ldots, \mathcal{T}$ we have $\lceil \frac{(m-1)\mathcal{T}+\ell}{\mathcal{T}} \rceil = m$, and the last inequality follows from $\sum_{m=1}^{\lceil \frac{T-k}{\mathcal{T}} \rceil} a^m \leq \sum_{m=1}^{\infty} a^m = \frac{a}{1-a}$ as $a < 1$. Plugging back to Eq. (36) concludes the proof. □

The next lemma establishes that for small enough η, we have $-\eta/2 + \mathcal{O}(\eta^2) \leq -\eta/4$.

Lemma C.9. *Let $\gamma, \theta \geq 1$. For every $\eta \leq \frac{1}{30\gamma\beta\theta}$, it holds that*

$$-\frac{\eta}{2} + 2\gamma\beta\eta^2 + 160\gamma\beta^2\theta^2\eta^3 \leq -\frac{\eta}{4} \ .$$

Proof. We equivalently prove that

$$2\gamma\beta\eta^2 + 160\gamma\beta^2\theta^2\eta^3 \leq \frac{\eta}{4} \ .$$

Since both $\gamma \geq 1$ and $\theta \geq 1$, we have

$$\begin{aligned} 2\gamma\beta\eta^2 + 160\gamma\beta^2\theta^2\eta^3 &\leq 2\gamma\beta\theta\eta^2 + 160\gamma^2\beta^2\theta^2\eta^3 \\ &= \frac{\eta}{4} \left(8\gamma\beta\theta\eta + 640\gamma^2\beta^2\theta^2\eta^2 \right) \\ &\leq \frac{\eta}{4} \left(\frac{8\gamma\beta\theta}{30\gamma\beta\theta} + \frac{640\gamma^2\beta^2\theta^2}{900\gamma^2\beta^2\theta^2} \right) = \frac{\eta}{4} \cdot \frac{44}{45} \leq \frac{\eta}{4} \ , \end{aligned}$$

where the second inequality follows from the upper bound on η. □

We also make use of the following result, which we prove using simple algebra.

Lemma C.10. *Suppose $\eta = \min\{\eta_1, \eta_2, \eta_3\}$ for some $\eta_1, \eta_2, \eta_3 > 0$, and let $A, B, C > 0$. Then, the following holds:*

$$\frac{A}{\eta} + B\eta + C\eta^2 \leq A \left(\frac{1}{\eta_1} + \frac{1}{\eta_2} + \frac{1}{\eta_3} \right) + B\eta_2 + C\eta_3^2 \ .$$

Proof. Since η is the minimum of three terms, $1/\eta$ is the maximum of their inverses. Thus, we can bound $1/\eta$ by the sum of the inverses as follows:

$$\frac{A}{\eta} = A \max \left\{ \frac{1}{\eta_1}, \frac{1}{\eta_2}, \frac{1}{\eta_3} \right\} \leq A \left(\frac{1}{\eta_1} + \frac{1}{\eta_2} + \frac{1}{\eta_3} \right) \ .$$

The terms $B\eta$ and $C\eta^2$ are monotonically increasing with η. We can therefore bound η by η_2 and η^2 by η_3^2. □

D. Entropy-Constrained Uniform Quantization

In this section, we describe a new compression technique entitled Entropy-Constrained Uniform Quantization (ECUQ), which we developed for anchor compression, although it can be of independent interest. ECUQ is described in Algorithm 4. Let $x = (x(1), \ldots, x(d)) \in \mathbb{R}^d$ be some input vector we wish to compress using ECUQ. Denote: $x_{\min} \coloneqq \min_i x(i)$, $x_{\max} \coloneqq \max_i x(i)$. Given some bandwidth budget of b bits/coordinate, ECUQ initially divided the interval $[x_{\min}, x_{\max}]$ into $K = 2^b$ non-overlapping bins of equal size $\Delta = (x_{\max} - x_{\min})/K$. Then, it sets the quantization values, which we denote by $\mathcal{Q}$, to be the centers of these bins. Afterwards, the vector x is quantized into elements of $\mathcal{Q}$, that is, each element $x(i)$ is assigned to its closest quantization value $q \in \mathcal{Q}$ to generate the quantized vector $\hat{x}_{\mathcal{Q}}$, whose elements are all in $\mathcal{Q}$. We then compute the empirical distribution of the quantized vector by counting for every $q \in \mathcal{Q}$ the number of times it appears in $\hat{x}_{\mathcal{Q}}$, and the entropy of the resulting distribution. Note that the entropy is upper bounded by $\log K = b$. Finally, for some small tolerance parameter ϵ (we use $\epsilon = 0.1$), we check whether the entropy is within ϵ distance from the budget b: if it is not the case, then we perform a double binary search, repeating the above procedure with increased number of quantization values K, to find the maximal number of uniformly spaced quantization values such that the entropy of the empirical distribution of the resulting quantized vector is within ϵ distance from b. Only after this entropy condition is satisfied, we encode $\hat{x}_{\mathcal{Q}}$ using some entropy encoding (we use Huffman coding).

Algorithm 4 Entropy-Constrained Uniform Quantization (ECUQ)

Input: Vector $x \in \mathbb{R}^d$, bandwidth budget b (bits/coordinate), tolerance ϵ.
$x_{\max} \leftarrow \max_i x(i), \;\; x_{\min} \leftarrow \min_i x(i)$ ▷ Get max/min values of input vector
$K \leftarrow 2^b, \;\; \Delta \leftarrow (x_{\max} - x_{\min})/K$ ▷ Initialize # of quantization values and bin length
$\mathcal{Q} \leftarrow \left\{x_{\min} + \left(k + \frac{1}{2}\right) \cdot \Delta : k = 0, \ldots, K-1\right\}$ ▷ Set uniformly spaced quantization values
$\hat{x}_{\mathcal{Q}} \leftarrow \texttt{Quantize}(x, \mathcal{Q})$ ▷ $\hat{x}_{\mathcal{Q}}(i) = \arg\min_{q \in \mathcal{Q}} \|x(i) - q\|$
$p_{\mathcal{Q}} \leftarrow \texttt{Empirical\_Density}(\hat{x}_{\mathcal{Q}})$ ▷ $p_{\mathcal{Q}}(q) = \frac{1}{N} \sum_{i \in [N]} \mathbb{1}\{\hat{x}_{\mathcal{Q}}(i) = q\}, \;\; \forall q \in \mathcal{Q}$
$\mathcal{H}(p_{\mathcal{Q}}) \leftarrow \texttt{Entropy}(p_{\mathcal{Q}})$ ▷ $\mathcal{H}(p_{\mathcal{Q}}) = -\sum_{q \in \mathcal{Q}} p_{\mathcal{Q}}(q) \log p_{\mathcal{Q}}(q)$
if $\mathcal{H}(p_{\mathcal{Q}}) < b - \epsilon$ **then**
  $\hat{x}_{\mathcal{Q}} \leftarrow$ DOUBLE_BINARY_SEARCH_NUM_QUANTIZATION_LEVELS(x, b)
end if
$\hat{x}_e \leftarrow \texttt{Huffman\_Coding}(\hat{x}_{\mathcal{Q}})$ ▷ Entropy encoding of the quantized vector
Return: $\hat{x}_e$

Procedure DOUBLE_BINARY_SEARCH_NUM_QUANTIZATION_LEVELS(x, b)
  Initialize: low $\leftarrow 2^b$, high $\leftarrow \infty$, $p \leftarrow -1$
  while low $\leq$ high **do**
    if high $== \infty$ **then**
      $p \leftarrow p + 1$
      mid $\leftarrow 2^b + 2^p$ ▷ Increase # of levels exponentially
    else
      mid $\leftarrow$ (low + high)/2
    end if
    $K \leftarrow$ mid, $\Delta \leftarrow (x_{\max} - x_{\min})/$mid
    $\mathcal{Q} \leftarrow \left\{x_{\min} + \left(k + \frac{1}{2}\right) \cdot \Delta : k = 0, \ldots, K-1\right\}$
    $\hat{x}_{\mathcal{Q}} \leftarrow \texttt{Quantize}(x, \mathcal{Q})$
    $p_{\mathcal{Q}} \leftarrow \texttt{Empirical\_Density}(\hat{x}_{\mathcal{Q}})$
    $\mathcal{H}(p_{\mathcal{Q}}) \leftarrow \texttt{Entropy}(p_{\mathcal{Q}})$
    if $\mathcal{H}(p_{\mathcal{Q}}) > b$ **then**
      high $\leftarrow$ mid $- 1$
    else if $\mathcal{H}(p_{\mathcal{Q}}) < b - \epsilon$ **then**
      low $\leftarrow$ mid $+ 1$
    else
      return $\hat{x}_{\mathcal{Q}}$
    end if

Figure 5 illustrates ECUQ's encoder, as described in the text above. The corresponding decoder is fairly simple as it only performs entropy decoding in linear time (Huffman decoding).

While devising ECUQ, we also considered an additional method to approximate ECQ. It is similar to ECUQ, but instead of using uniformly spaced quantization values, it uses K-Means clustering to find the quantization values that minimize the overall squared error. We used a double binary search to find the largest number of levels K such that, after entropy encoding, the bandwidth constraint is satisfied. We termed this method *Entropy-Constrained K-Means (ECK-Means)*.

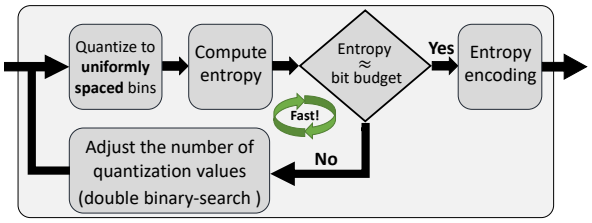

Figure 5. ECUQ encoder's illustration.

We compare the performance of ECUQ with ECQ and ECK-Means in terms of their NMSE, and we also measure their encoding time. As we mentioned in the Section 4, ECQ is sensitive to hyperparameters; thus, we implemented it using a grid search over its hyperparameters to guarantee near-optimal performance.[6] We evaluate the three methods on vectors drawn from three different synthetic distributions: **(1)** *LogNormal*$(0,1)$; **(2)** *Normal*$(0,1)$; and **(3)** *Normal*$(1,0.1)$. In Figure 6 (**top**) we show the NMSE and encoding time for different sizes of input vectors when the budget constraint is $b = 2$ bits/coordinate. As a complementary result, in Figure 6 (**bottom**) we fix the dimension of the input vectors to $d = 2^{12} = 4096$ and vary the bandwidth budget constraint from 2 to 5 bits/coordinate. The results imply that ECUQ exhibits a good speed-accuracy trade-off: it consistently outperforms ECK-Means while being an order of magnitude faster, and it is competitive with ECQ but about three orders of magnitude faster. Note additionally that it takes ≈ 20 minutes for ECQ to encode even a small vectors of size 2^{12} with budget constraint of 4 bits/coordinate; this means that ECQ without some acceleration is not suitable for compressing neural networks with millions and even billions of parameters.

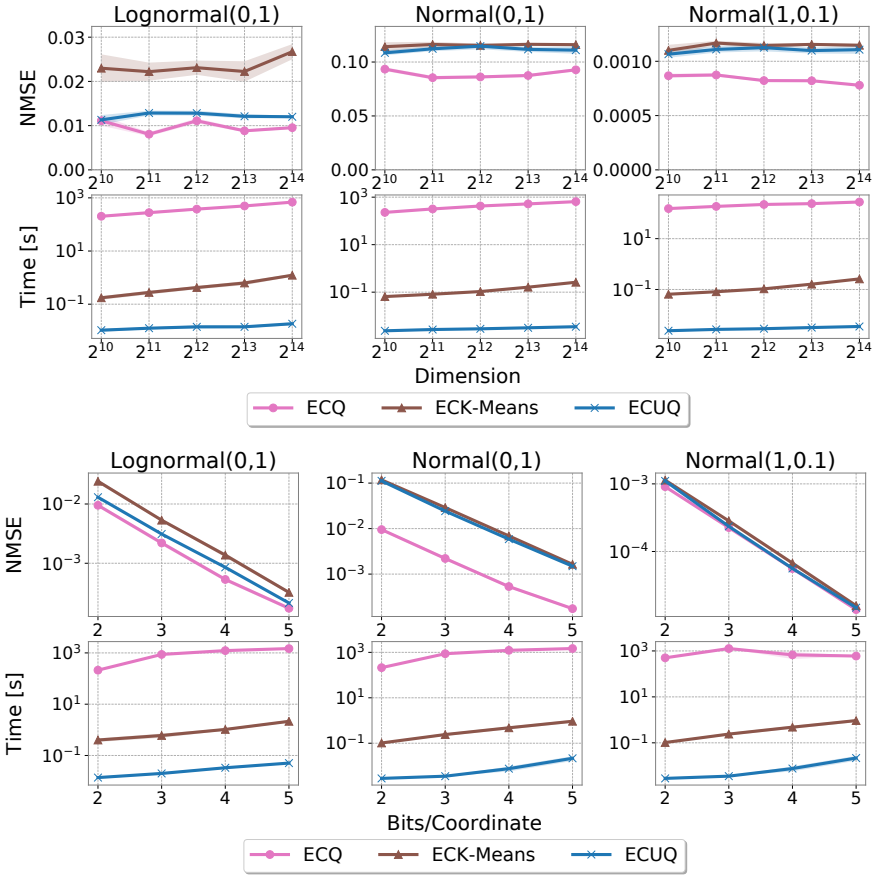

Figure 6. ECQ vs. ECK-Means vs. ECUQ: NMSE and encoding time for different input distributions: **(top)** as a function of the bandwith budget for fixed input dimension of $d = 2^{12}$; and **(bottom)** as a function of the input dimension for fixed bandwidth budget of 2 bits/coordinate.

[6]While such implementation may increase the encoding time, we are not aware of any other approach to guarantee an optimal performance. ECQ aims at solving a hard non-convex problem, and different hyperparameters may result in different local minima.

E. Additional ECUQ Evaluations

Since ECUQ is a quantization-based method, in Section 4 and Appendix D we compare it with quantization-based techniques. In Figure 7, we give a complementary result comparing it with sparsification methods (Rand-K, Top-K) and sketching (Count-Sketch), where similar trends are observed. Note, however, that such techniques are mostly orthogonal to quantization-based methods and they can be used in conjunction.

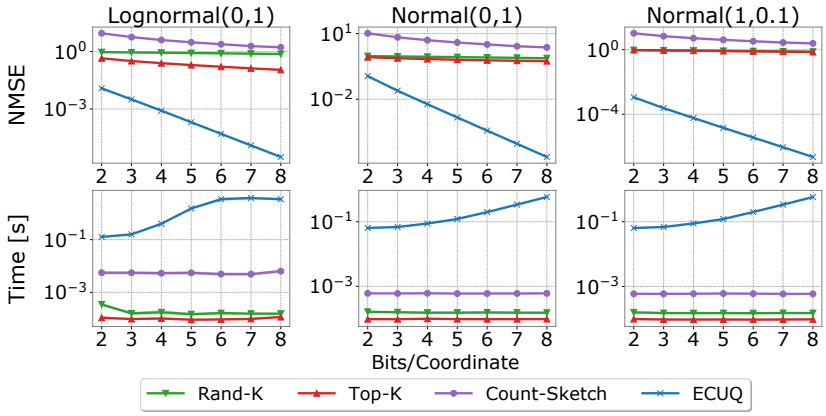

Figure 7. ECUQ vs. sparsification and sketching: NMSE (**top**) and encoding time (**bottom**) as a function of the bandwidth budget for different input distributions and a fixed dimension of $d = 2^{20}$.

F. Experimental Details

We implemented DoCoFL in PyTorch (Paszke et al., 2019). In all experiments, the PS uses Momentum SGD as optimizer with a momentum of 0.9 and L_2 regularization (i.e., weight decay) with parameter 10^{-5}. The clients, on the other hand, use vanilla SGD for all tasks but Amazon Reviews, for which Adam provided better results. In Table 4 we report the hyperparameters used in our experiments. To ease the computational burden and long training times, in the Shakespeare task we reduced the amount of train and validation data for each speaker (i.e., client) by a factor of 10 by using only the first 10% of train and validation data, but no less than 2 samples per speaker.

Table 4. Hyperparameters for our experiments.

Task	Batch size	Client optimizer	Client lr	Server lr
EMNIST	64	SGD	0.05	1
CIFAR-100	128	SGD	0.05	1
Amazon Review	64	Adam	0.005	0.1
Shakespeare	4	SGD	0.5	1

G. Additional Results

In this section we present additional results that were deferred from the main text.

G.1. Learning Curves

We next provide the learning curves for the experiments we conducted in § 5. In Figures 8 and 9 we show the validation and train accuracy throughout training, respectively. We measure train and validation accuracy every 50 rounds for EMNIST and Amazon Reviews, every 500 rounds for CIFAR-100, and every 1000 rounds for Shakespeare.

Following the discussion in § 5, Figure 8 demonstrates that using less bandwidth may improve the generalization ability as it can serve as a form of regularization. For example, consider the EMNIST task, where DoCoFL$(2, 2, 3)$ (i.e., 2 bits per coordinate for anchor and correction compression, and 3 bits per coordinate for uplink compression) outperforms both FedAvg and DoCoFL$(4, 4, 3)$. Unsurprisingly, examining Figure 9 reveals a reverse image – less bandwidth implies lower train accuracy. This suggests that in some settings using less bandwidth (but not too little) may help to prevent overfitting.

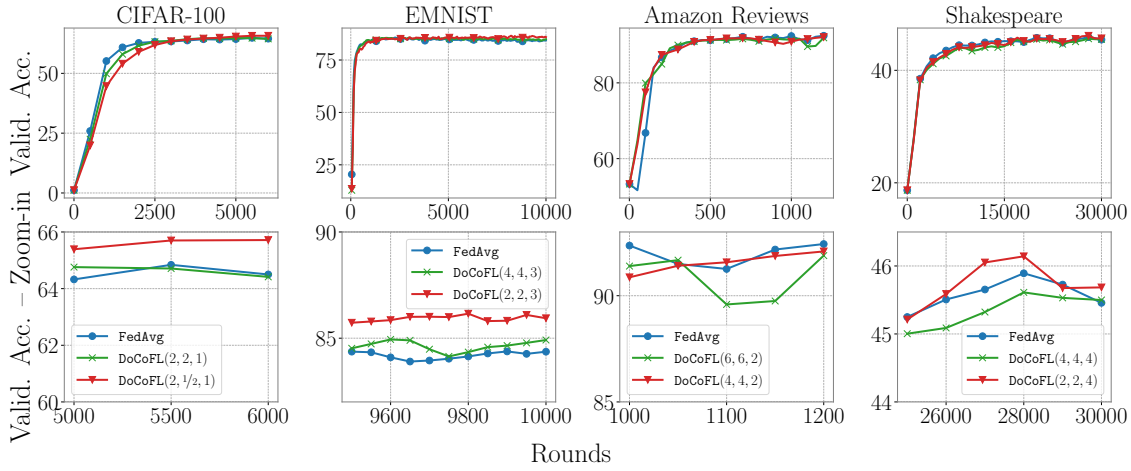

Figure 8. Validation accuracy for different tasks.

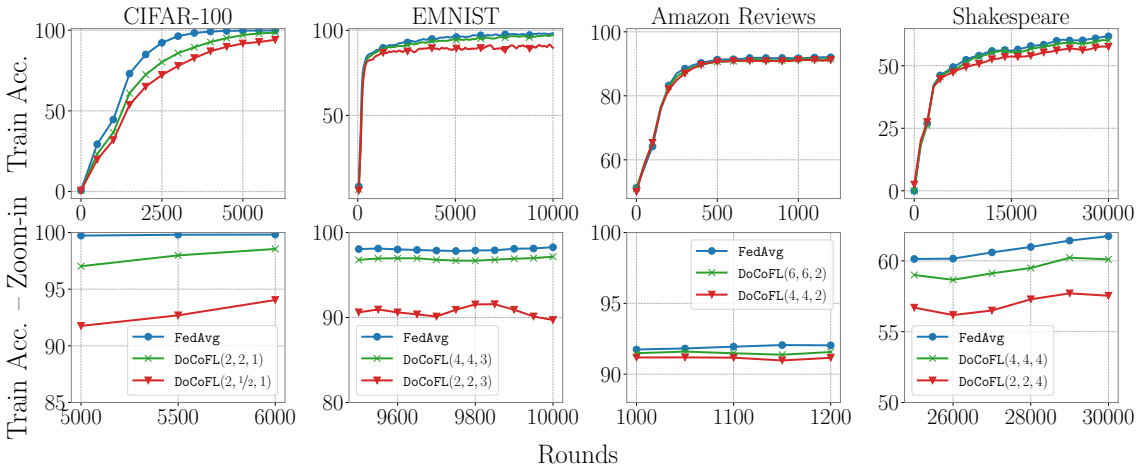

Figure 9. Train accuracy for different tasks.

G.2. Bandwidth Budget Ablation

Next, we provide numerical results that demonstrate the effect of the downlink (anchor and correction) bandwidth budget on DoCoFL's performance. We consider the CIFAR-100 with ResNet-9 experiment with anchor deployment rate $K = 10$ and anchor queue capacity $\mathcal{V} = 3$.

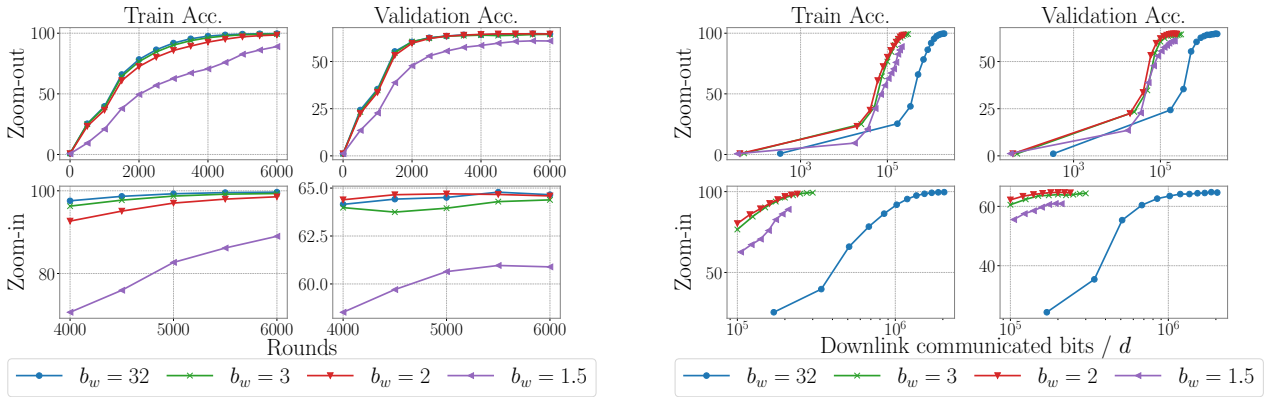

Figure 10. Train and validation accuracy for different anchor bandwidth budgets (32, 3, 2, and 1.5 bits per coordinate) as a function of the number of rounds (**left**) and the number of communicated bits in the downlink direction (**right**).

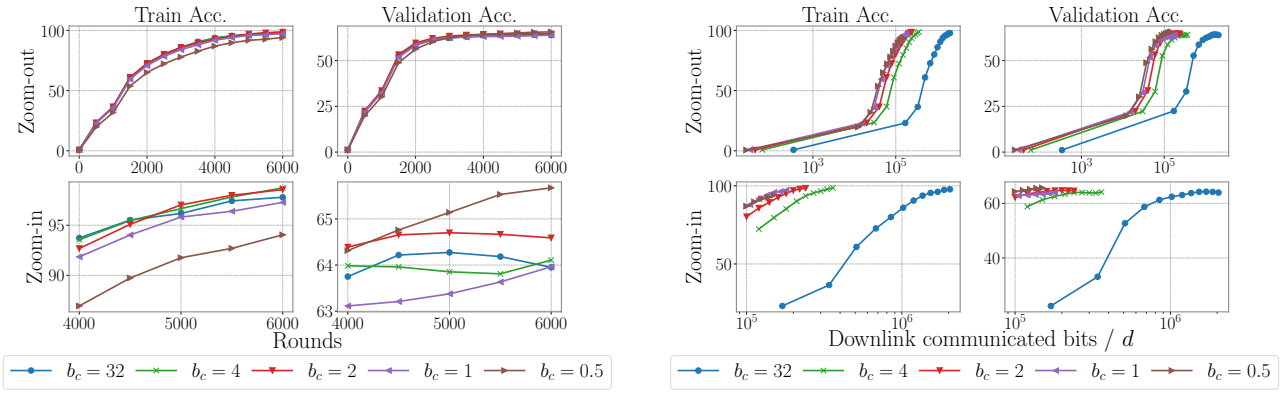

Figure 11. Train and validation accuracy for different correction bandwidth budgets (32, 4, 2, 1, and 0.5 bits per coordinate) as a function of the number of rounds (**left**) and the number of communicated bits in the downlink direction (**right**).

In Figure 10 we show the train and validation accuracy for different anchor bandwidth budgets b_w, namely, 32 (full-precision), 3, 2 and 1.5 bits per coordinate, while the correction budget is fixed and equals $b_c = 2$ bits per coordinate, as a function of both number of rounds and number of communicated bits in the downlink direction. The results indicate that one can significantly reduce the bandwidth used for communicating the anchors and use as low as $b_w = 2$ bits per coordinate for anchor compression ($16\times$ reduction), without degrading validation accuracy. Again, similarly to evidence from the previous section, less bandwidth typically results in lower train accuracy.

In Figure 11 we show the train and validation accuracy for different correction bandwidth budgets b_c (32, 4, 2, 1 and 0.5 bits per coordinate), while the anchor budget is fixed and equals $b_w = 2$ bits per coordinate. We observe similar trends, where less bandwidth leads to lower train accuracy but possibly higher validation accuracy. Additionally, we see that one may even use a sub-bit compression ratio for the correction term, allowing for significant *online* bandwidth reduction, which is especially important in our context.

G.3. The Value of the Correction term

In this section, we discuss the effect of ignoring the correction term on DoCoFL's performance, namely, we consider the case where clients only obtain an anchor (i.e., a previous model) and use it perform local optimization. As mentioned in §5, ignoring the correction may resemble other frameworks such as delayed gradients. Delayed SGD (DSGD, Arjevani et al. (2020)) is well-studied in the literature both theoretically and empirically. Indeed, theory supports that optimization with delay can work, e.g., Stich & Karimireddy (2019) showed that as long as the maximal delay is bounded by $\mathcal{O}(\sqrt{T})$, DSGD enjoys the same asymptotic convergence rate as SGD; Cohen et al. (2021) later improved the dependence on the maximal delay to average delay with a variant of DSGD, allowing for arbitrary delays. However, it has also been observed that, in practice, introducing delay can slow down and even destabilize convergence, and as a result hyperparameters should be chosen with great care to ensure stability (Giladi et al., 2020). We thus convey that sending the correction is crucial and allows for improved performance.

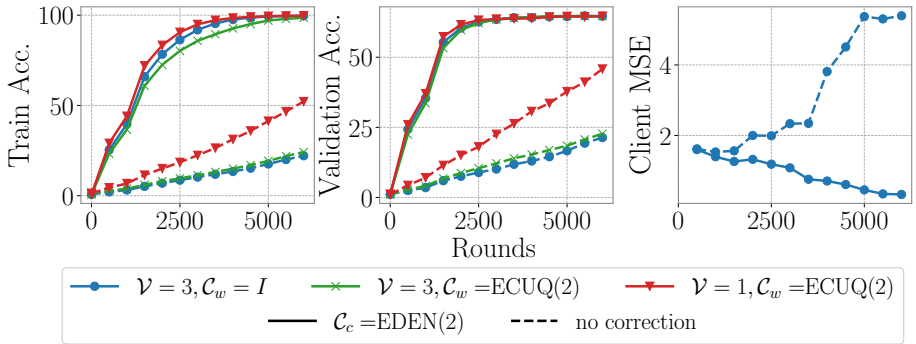

Figure 12. The effect of ignoring the correction term. Train (**left**) and validation (**middle**) accuracy for different configurations, and average client's model estimation error (**right**).

To reinforce this, we conducted an experiment to numerically evaluate the effect of ignoring the correction. We consider the CIFAR-100 with ResNet-9 experiment. We test DoCoFL with and without sending the correction to the clients for three different configurations: **(1)** $\mathcal{V} = 3$ and no anchor compression (i.e., 32 bits per coordinate); **(2)** $\mathcal{V} = 3$ and $\mathcal{C}_w$ is ECUQ with 2 bits per coordinate; and **(3)** $\mathcal{V} = 1$ with $\mathcal{C}_w$ as in the second configuration. For all configurations, we use an anchor deployment rate of $K = 10$. In Figure 12 we present the train and validation accuracy, and also the average client squared model estimation error, i.e., $\frac{1}{S}\sum_{i\in\mathcal{S}_t}\|w_t - \hat{w}_t^i\|^2$, for the first configuration. The results clearly indicate that accounting for the correction term results in faster convergence. While ignoring the correction may eventually still result in similar performance, it is expected to take significantly more communication rounds; this is evident even when the anchor is sent with full precision. Examining the rightmost plot, we observe that ignoring the correction leads to larger model estimation error, which provides insight into why the performance deteriorates when the correction is ignored.

G.4. DoCoFL and EF21

While our focus in on setups where a client may participate in training only once or a few times, in some setups, partial but repeated participation can be expected. For such setups, we consider some additional related work. Specifically, we focus on EF21 (Richtárik et al., 2021) and some of its extensions. To assess the value of DoCoFL in this context, we attempted to extend EF21-BC (Algorithm 5 of Fatkhullin et al. (2021)) to the partial participation setting, but were not able to achieve convergence. We suspect that it is attributed to an accumulated discrepancy between the models of the clients and the server, and thus a more sophisticated extension is required, which is out of scope. Instead, we extended EF21-PP (Algorithm 4 of Fatkhullin et al. (2021)) to support downlink compression in two different ways: **(1)** direct compression of the model parameters using EDEN; and **(2)** using DoCoFL. We compare these approaches with a baseline that sends the exact model to the clients (i.e., no downlink compression).

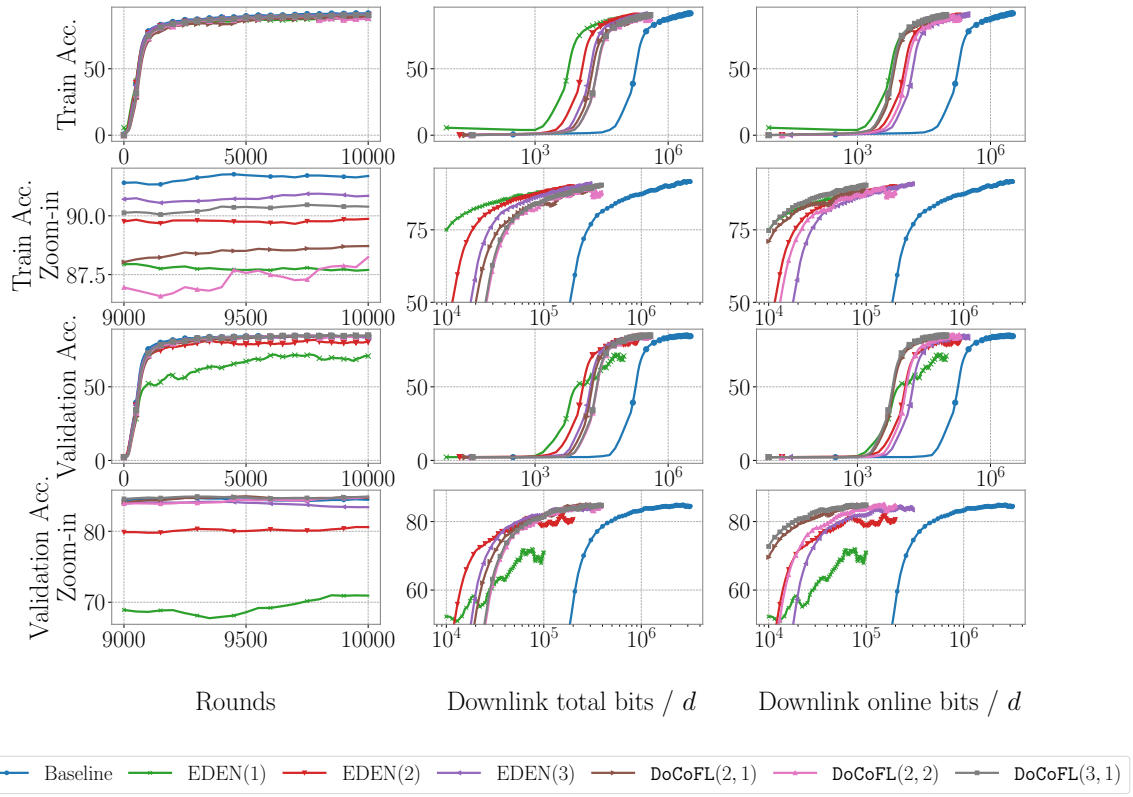

Figure 13. Train and validation accuracy for EF21-PP (Fatkhullin et al., 2021) (baseline) and its extensions supporting downlink compression, either directly with EDEN, or with DoCoFL, over EMNIST. Results are displayed against the number of communication rounds (**left**), the total number of downlink communicated bits (**middle**), and the number of online downlink communicated bits (**right**).

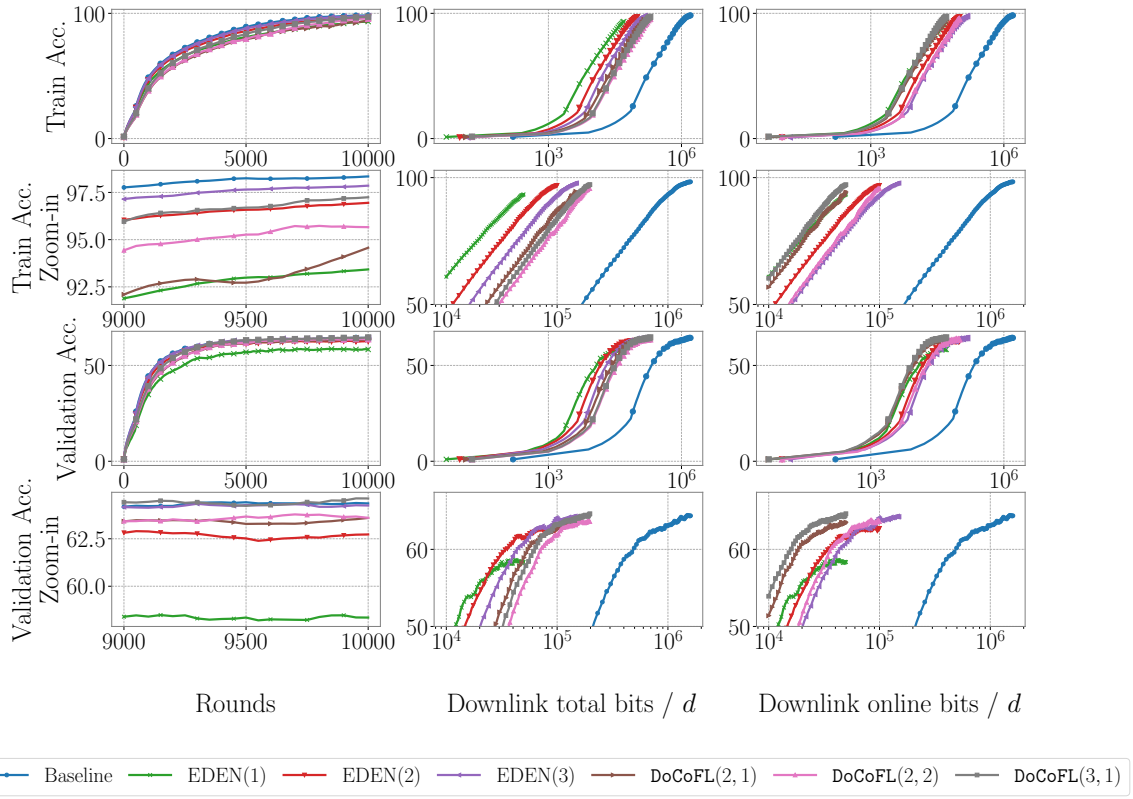

Figure 14. Repeating the experiments from Figure 13 for CIFAR-100.

We consider two tasks: **(1)** EMNIST + LeNet with $N = 200$ clients and $S = 10$ participating clients per-round; and **(2)** CIFAR-100 + ResNet-9 with $N = 25$ clients and $S = 5$ participating clients per-round. We used less clients here compared to the experiments in the main text due to GPU memory limitations (EF21 requires keeping all N clients persistent). In both experiments we use EDEN with 1 bit/coordinate for uplink compression. Figures 13 and 14 depict the train and validation accuracy as a function of the number of communication rounds, the total number of communicated bits in the downlink direction, and the number of communicated bits in the downlink direction required *online* (i.e., at the clients' participation round) for EMNIST and CIFAR, respectively. We note that using a direct compression of the model with 1 or 2 bits per coordinate results in a notable drop in validation accuracy compared to the baseline. Indeed, using EDEN with 3 bits per coordinate performs similarly to the baseline. Examining DoCoFL with 2 and 1 bits/coordinate for anchor and correction, respectively, reveals that it performs similarly to direct downlink compression with 3 bits/coordinate, i.e., when using the same overall downlink bandwidth; however, it requires $3\times$ less online bandwidth. This is especially important in our context since online bandwidth demand directly translates to client delays; indeed, this is a main design goal of DoCoFL. Additionally, one may improve the results even further by increasing the anchor budget to 3 bits/coordinate, while keeping the online bandwidth usage the same or even lower (e.g., see Figure 11).

Another important point of comparison is the EF21-P + DIANA method (Gruntkowska et al., 2022), which supports bi-directional compression. In particular, their server compression mechanism is similar to ours in the following sense: their server and clients hold control variates that track the global model; these control variates can be seen as an anchor that is being updated in each round and the server sends to the clients a compressed correction with respect to the control variates. However, their approach requires client-side memory with full participation (i.e., updated control variates). The authors propose to study an extension of their framework to partial participation as future work. It is interesting to investigate whether DoCoFL can be used in conjunction with this framework to achieve this.

H. Future Work

We point out several directions for future research: **(1)** an interesting avenue would be to investigate how to combine DoCoFL with the delayed gradients framework (Stich & Karimireddy, 2019). While delayed gradients do not reduce downlink bandwidth, they are especially useful for clients that may require a long time to perform local updates and communicate them back to the PS. Thus, accounting for delayed gradients may enhance DoCoFL's versatility and robustness in real FL deployments; **(2)** our theoretical framework focuses on the SGD optimizer. Exploring the implications of using adaptive optimizers, such as Adam, on the theoretical analysis and guarantees would be of great interest; **(3)** as we convey in Appendix B, an intriguing extension of DoCoFL involves the incorporation of adaptive bandwidth budget for anchor and correction compression; although it introduces a significant theoretical challenge due to the coupling between optimization and compression, it may yield a convergence guarantee for DoCoFL with anchor compression and achieve even larger bandwidth savings; **(4)** while we employ extensive simulations and account for various overheads of DoCoFL, it is desired to further strengthen our conclusions through real deployments.