BRIDGING COMPRESSED COMMUNICATION AND LOCAL STEPS VIA SPECTRAL METHOD IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) enables collaborative training of complex models across large-scale networks of edge devices while keeping data localized. However, data heterogeneity and high communication overhead remain fundamental bottlenecks. Recently, the spectral optimizer Muon has demonstrated remarkable advantages over Adam-like methods in centralized training. While recent efforts have attempted to transition Muon to FL, naive integrations into traditional architectures fail short. Crucially, existing approaches do not support the the communication compression and local steps — the two foundational techniques of efficient FL. To address this limitation, we introduce a novel spectral FL method that utilizes both local updates and compression. We establish rigorous convergence guarantees under mild assumptions, notably eliminating the restrictive bounded heterogeneity condition. Empirical evaluation on standard image classification benchmarks validates the efficiency and robustness of our approach.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017) is a paradigm of training a shared model across a set of edge devices (often called workers or clients). The clients collect data locally and use it to perform updates, which are aggregated by the server. Formally, the problem statement is as follows:

$$\min_{x\in\mathbb{R}^{d_1\times d_2}} \left[f(x) = \frac{1}{M}\sum_{i=1}^{M} f_i(x) \right],$$

where x denotes the model weights, f_i is the empirical risk (Shalev-Shwartz & Ben-David, 2014) of client i computed on its local dataset, and M is the total number of clients.

The communication bottleneck remains a central challenge in federated learning (Li et al., 2020). The source of this issue is two-fold. Firstly, due to the number of devices in the network, coordinating them becomes a substantial asynchronous workload (Le et al., 2024). Thus, reducing communication frequency is important in practice. Algorithmically, the problem can be addressed by local steps. Here, each client performs several local update steps between aggregations (Gorbunov et al., 2021b). Secondly, an individual model may contain up to billions of parameters (Zhou et al., 2025), and its full transmission severely slows down the aggregation process. Communication compression aims to reduce the size of each message by removing redundant or low-importance information. In practice, model size can be reduced by orders of magnitude with essentially no performance loss (Haddadpour et al., 2021; Li & Li, 2023).

Modern machine learning problems are growing in complexity and demand advanced optimization techniques. Currently, the most widely used solutions rely on adaptive algorithms like AdaGrad (Duchi et al., 2011), Adam (Kingma & Ba, 2014), AdamW (Loshchilov & Hutter, 2017), and Shampoo (Gupta et al., 2018). These methods adapt the update to the local geometry of the problem using current optimization statistics. The paradigm has become the gold standard since its development (Liu et al., 2025b), and it was successfully applied to the FL setting (Deng et al., 2020; Reddi et al., 2020; Wu et al., 2023).

A different strategy emerged recently: an adaptation to the spectrum of matrix updates instead of the geometry of the parameter space (Jordan et al., 2024; Riabinin et al., 2025; Pethick et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Subsequent work has empirically demonstrated its advantages over classic optimizers (Liu et al., 2025a; Semenov et al., 2025), and Muon (Jordan et al., 2024) has attracted significant attention as the most efficient. Notably, Muon can be viewed as a Linear Minimization Oracle (LMO) under the spectral matrix norm, and this perspective yields theoretical improvements over classic approaches (Bernstein & Newhouse, 2024; Pethick et al., 2025). Hence, transferring this paradigm to FL can yield methods that surpass adaptive approaches.

2 Related Works

2.1 Methods for Communication Compression

Compression aims to reduce the size of the transmitted message while retaining nearly all essential information. Two common approaches are quantization – reducing the bit precision of each parameter (Alistarh et al., 2017; Bernstein et al., 2018), and sparsification – removing a subset of the vector's values (Lin et al., 2017; Stich et al., 2018).

Theoretically, compressors fall into two commonly studied classes, unbiased and contractive (Beznosikov et al., 2023). For unbiased compressors, the expectation of the output is equal to the original vector, while contractive compressors do not have this property. Unbiased compression is conceptually similar to the stochasticity of gradients in the finite-sum setting. In both cases, the distortion is bounded but does not vanish as the iterate approaches the optimum. Hence, techniques developed for the stochastic setting carry over. Namely, the majority of works in this line are driven by Variance Reduction (VR) (Johnson & Zhang, 2013; Schmidt et al., 2017) mechanisms that were initially designed for stochastic gradients (Gorbunov et al., 2021a; Beznosikov et al., 2022; Tyurin & Richtárik, 2022; Mishchenko et al., 2025). However, unbiased compressors tend to perform poorly in practice (Beznosikov et al., 2023, Figure 5). Thus, we focus on contractive compression.

Notably, simple SGD with an arbitrary contractive compressor does not converge. To overcome the obstacle, special techniques are required, the most widely used of which being Error Feedback (EF). First appeared as a heuristic (Seide et al., 2014), EF later received a theoretical foundation with arbitrary compression (Karimireddy et al., 2019; Stich & Karimireddy, 2020; Beznosikov et al., 2023). The core concept of EF is combining the current message with the error from the previous compression, which compensates for inaccuracies at previous steps. Crucially, EF allows one to establish convergence. EF21 is a noteworthy advancement that improves convergence rates in the non-convex setup by utilizing a distinct error accounting procedure (Richtárik et al., 2021). The idea is to compress the residuals instead of the full message. Both EF and EF21 have found numerous applications in FL (Karimireddy et al., 2019; Li & Li, 2023; Bylinkin et al., 2025).

2.2 Local Steps

Another approach to tackling the communication bottleneck is local steps. Instead of calculating a single gradient, each client performs several steps over its local dataset and communicates the accumulated model update. This way, the costly server-side aggregation is performed less often. The technique is extensively studied (Woodworth et al., 2020; Khaled et al., 2020; Koloskova et al., 2020; Gorbunov et al., 2021b).

Local steps have a crucial downside: clients lose the alignment of their training trajectories between aggregations. As a result, under strong data heterogeneity, the local models diverge, and the training process stalls. Early work introduced similarity assumptions across clients, such as identical data or bounded gradients (Koloskova et al., 2020; Gorbunov et al., 2021b). In practice, most of these assumptions are infeasible (Karimireddy et al., 2020, Table 3). The issue was tackled by the SCAFFOLD algorithm (Karimireddy et al., 2020). The authors introduced control variates that allow clients to maintain the correct update direction during local steps. This method was later extended with compression in SCAFCOM (Huang et al., 2024).

2.3 SPECTRAL METHODS IN FL

Jordan et al. (2024) introduced `Muon`, a matrix-structure-aware method. In a nutshell, the algorithm orthogonalizes the updates via SVD, normalizing the gradients and thereby stabilizing training. Notably, this orthogonalization can be viewed as a Linear Minimization Oracle operator over the gradients in the spectral matrix norm (Bernstein & Newhouse, 2024; Pethick et al., 2025). Moreover, Kim & Oh (2026) established the convergence of `Muon` under inexact SVD computed by the Newton-Schulz iteration, further motivating its practical adoption. These theoretical insights and its practical performance have drawn considerable attention, motivating a growing body of work that studies and extends the method (Liu et al., 2025a; Qian et al., 2025; Ahn et al., 2025; Si et al., 2025; Liu et al., 2026; Zhang et al., 2026).

Some recent papers employed spectral methods in a distributed manner (Ahn et al., 2025; Thérien et al., 2025), but only in the homogeneous setting. As discussed above, this assumption is unrealistic. The first work to fully address this was (Takezawa et al., 2025). The authors use a general scheme of `SCAFFOLD` (Karimireddy et al., 2020). Their `FedMuon` replaces the plain gradients in the local steps of `SCAFFOLD` with LMO updates. However, the scheme assumes the transmission of full model updates and control variates, which incurs communication overhead. Subsequently, compression was applied to `Muon` updates via `EF21` (Gruntkowska et al., 2025) and the VR technique of `VR-MARINA` (Qian et al., 2026). Crucially, both works forgo local steps, accessing the LMO once per round at the server – yet local steps are a pillar of communication-efficient FL. The communication bottleneck for `Muon`-like methods, therefore, remains unresolved.

3 CONTRIBUTION

We summarize the contribution of our work in this section.

- We propose `FedCALM-D` (**Fed**erated **C**ompressed **A**nd **L**ocal **M**uon — **D**eterministic), a novel scheme that integrates both compression and local steps into `Muon`-like LMO updates in the deterministic FL setting. This requires more than a direct application of existing tools: we adapt the compression mechanism to the structure of the LMO update.
- We extend the scheme to the stochastic setting and obtain our main algorithm, `FedCALM`.
- We establish rigorous convergence guarantees for both methods in the non-convex setting under mild assumptions. Crucially, our analysis is heterogeneity-free.
- We validate our method on an image-classification benchmark (*ResNet-18* on *CIFAR-10*) and further on a language task (*LoRA* fine-tuning of *RoBERTa* on the *GLUE* benchmark), confirming its practical advantages across modalities. In terms of communicated bits, `FedCALM` outperforms all baselines in both convergence speed and final accuracy.

4 MAIN RESULTS

4.1 PRELIMINARIES

First, we introduce the relevant notation for the equations in the following sections.

Matrix norms. We consider $\|\cdot\|$ as the spectral matrix norm in $\mathbb{R}^{d_1\times d_2}$, $\|\cdot\|_*$ denotes its conjugate nuclear norm, and $\|\cdot\|_F$ is Frobenius norm. The following inequalities hold for introduced norms with $\rho = \sqrt{\min(d_1, d_2)}$ (Kim & Oh, 2026, Proposition 2):

$$\|x\| \le \|x\|_F \le \|x\|_* \le \rho\|x\|_F, \tag{1}$$

Compression. Since the cornerstone of our paper is compression, we define the corresponding operator.

Definition 4.1. *Operator $C : \mathbb{R}^{d_1\times d_2} \to \mathbb{R}^{d_1\times d_2}$ is a contractive compressor if there exists a constant $\beta_1 > 1$ such that*

$$\mathbb{E}[\|C(x) - x\|] \le \left(1 - \frac{1}{\beta_1}\right)\|x\|. \tag{2}$$

Remark 4.2. *We also assume that C is a contractive compressor in the conjugate norm if there exists a constant $\beta_2 > 1$ such that*

$$\mathbb{E}[\|C(x) - x\|_*] \leq \left(1 - \frac{1}{\beta_2}\right) \|x\|_*.$$

We denote $\beta = \max(\beta_1, \beta_2)$. An example of a contractive compressor in matrix norm is truncated SVD:

$$C(X) = U^T \Sigma_k V,$$

where $X = U^T \Sigma V$ is a full SVD decomposition, and Σ_k retains k largest singular values $\sigma_1 \dots \sigma_k$ of X. The compressor satisfies the introduced definitions with $\beta_1 = \frac{\sigma_1}{\sigma_1 - \sigma_{k+1}}$, and $\beta_2 = \frac{\sum_{j=1}^r \sigma_j}{\sum_{j=1}^k \sigma_j}$.

LMO. The spectral optimization is performed via the Linear Minimization Oracle (LMO). It refers to the solution of the following subproblem:

$$\mathrm{lmo}(x) = \arg \min_{y \in \mathbb{R}^{d_1 \times d_2}, \|y\| \leq 1} \langle x, y \rangle.$$

By definition, for any vector u we have $\|\mathrm{lmo}(u)\| \leq 1$. We refer to this equation as the unit-length of LMO. The other distinctive feature of LMO is described in Lemma B.1:

$$\langle x, \mathrm{lmo}(x) \rangle = -\|x\|_*.$$

The proof can be found in the appendix.

Newton-Schultz iterations. Since the exact calculation of LMO requires costly SVD, in practice, an approximation of the operator is calculated via the Newton-Schulz iterations (Kim & Oh, 2026).

Algorithmic notation. Next, we explain the notation of the algorithmic variables. $x^r \in \mathbb{R}^{d_1 \times d_2}$ denotes the global model weights at the start of round r, M is the total number of clients. η is the stepsize, and y_i^r is the full update for client i after K local steps. The local and global control variates originating from `SCAFFOLD` are refered to as c_i^r and c^r, respectively.

Assumptions. In our proofs, we assume the L-smoothness of the objective, following the LMO analysis in prior work (Pethick et al., 2025; Riabinin et al., 2025).

Assumption 4.3. *The objective f_i for each client i in L_*-smooth: for any $x, y \in \mathbb{R}^{d_1 \times d_2}$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\|_* \leq L_* \|x - y\|.$$

The loss landscapes of neural networks are typically highly non-convex (Cybenko, 1989). As we target practical, real-world scenarios, our analysis focuses on establishing convergence in the non-convex setting.

Assumption 4.4. *The function f is non-convex with at least one global minimum: there exists may be not unique x^* such that*

$$f(x^*) = \inf_{x \in \mathbb{R}^d} f(x) > -\infty.$$

Communication. Finally, we introduce two notions of communication efficiency.

Definition 4.5. *The number of communication rounds (round complexity) is the number of times the server communicates with the clients.*

This notion disregards which clients take part in a round and the sizes of the transmitted messages.

Definition 4.6. *Communication time (time complexity) is the total time of communication with all clients, assuming their channels are of equal quality.*

Unlike the first, this notion accounts for the sizes of the messages. Unless stated otherwise, we adopt Definition 4.6 as our measure of communication complexity, since it captures the practical advantages of compression: compression reduces the transmission time of each round, which in turn accelerates convergence. Since we assume all channels are equal, communication time is proportional to the total number of transmitted bits; we therefore report communicated bits in our experiments as a direct proxy for time complexity.

4.2 MOTIVATION

This section explains the design choices of our algorithm. First, we consider the deterministic setting.

Local drift. As discussed in Section 2, current federated methods with LMO updates lack one of two FL techniques: compression (Takezawa et al., 2025) or local steps (Gruntkowska et al., 2025; Qian et al., 2026). We aim to implement both in the same scheme. Since we assume no heterogeneity bound, local steps create client drift and spoil the convergence (Karimireddy et al., 2020, Figure 1). To overcome this, we consider control variates of SCAFFOLD.

Two compressions. With compression, the main challenge arises: **how to properly implement compressors in the scheme?** We need to communicate both the local updates $y_i^r = x_i^{r,K} - x_i^{r,0}$ and control variates c_i^r. Huang et al. (2024) faced a similar challenge in the design of SCAFCOM. The authors managed to combine the two messages into one and apply EF21. The update for the control variate Δc_i^r can be expicitly derived from the update y_i^r for the model.

However, this unification is impossible in LMO setting. Client drift has an LMO structure and *provides no information about the scale of the gradient* due to the unit length of LMO, which is essential for the control variate update. *Thus, the client drift and the control-variate update cannot be encoded in a single message and call for two distinct compression steps.*

Model update. Let us discuss the compression for model update y_i^r. For SCAFCOM, EF21 was fitting due to the diminishing gradients. As the model approaches the optimal point, $\|\nabla f(x^r)\| \to 0$ and $\|y_i^r\| \to 0$ lead to smaller errors. This is not the case in the spectral setting. Since the update has LMO nature, it therefore always has the size $\mathcal{O}(\eta K)$. The direction is what changes across rounds. Even near the optimal point, $\|y_i^{r+1} - y_i^r\| = \mathcal{O}(\eta K)$. EF21 scheme would introduce bias at each round, and of the same magnitude as the full vector y_i^r. This creates non-vanishing error and prevents convergence.

Control variates. The motivation for the implementation of control variates instantly suggests the error compensation scheme to use – EF21. Contrary to the model update, the gradient changes slowly across the rounds due to nature of the model update, and EF21 is designed to track these small changes. Moreover, for c_i^r, the natural transition to the virtual sequence of EF is impossible.

This yields FedCALM-D (**Fed**erated **C**ompressed **A**nd **L**ocal **Muon-DeterministicD**) given in Algorithm 1.

Algorithm 1 FedCALM-D

Input: initial point $x^0 \in \mathbb{R}^d$, $c^0 = \nabla f(x^0)$, $e_i^0 = 0, c_i^0 = \nabla f_i(x^0)$ for every client $i = 1, \dots M$, number of rounds R, round length K

Hyperparameters: stepsize $\eta > 0$, number of Newton-Schulz steps q

for $r = 0, 1, \dots, R-1$ **do**
  for clients $i = 1, \dots, M$ **do**
    $x_i^{r,0} = x^r$
    for local step $k = 0, 1, \dots, K-1$ **do**
      $x_i^{r,k+1} = x_i^{r,k} + \eta \mathrm{NS}^q(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)$
    end for
    $y_i^{r+1} = x_i^{r,K} - x_i^{r,0}$
    $v_i^{r+1} = C(y_i^{r+1} + e_i^r)$
    $e_i^{r+1} = e_i^r + y_i^{r+1} - v_i^{r+1}$
    Clients send v_i^{r+1} to the server
    $\delta_i^{r+1} = C(\nabla f_i(x_i^{r,K}) - c_i^r)$
    $c_i^{r+1} = c_i^r + \delta_i^{r+1}$
    Clients send δ_i^{r+1} to the server
  end for
  Server updates $x^{r+1} = x^r + \frac{1}{M}\sum_{i=1}^M v_i^{r+1}$
  Server updates $c^{r+1} = c^r + \frac{1}{M}\sum_{i=1}^M \delta_i^{r+1}$
  Server sends x^{r+1}, c^{r+1} to all clients
end for
Output: $\frac{1}{R}\sum_{r=0}^{R-1} x^r$

4.3 DETERMINISTIC ALGORITHM

We describe the FedCALM-D algorithm in detail. At the start, the control variates are initialized to the initial gradients. A full, uncompressed gradient communication is required at this stage, but not at any other point; this one-time transmission does not affect the convergence rate. Moreover, a similar heavy initialization is typical for methods with compression (Richtárik et al., 2021; Huang et al., 2024). At each round, all clients perform local training in parallel, starting from the shared point x^r

received from the server (Line 3). The local steps follow the discussed structure of SCAFCOM (Line 5). Here, $\mathrm{NS}^q(\cdot)$ denotes q steps of the Newton-Schulz algorithm (Kim & Oh, 2026). As discussed in Section 4.2, the clients maintain $c_i^r \approx \nabla f_i(x^r)$, $c^r \approx \nabla f(x^r)$, which provides $\nabla f_i(x_i^{r,k}) - c_i^r + c^r \approx \nabla f(x^r)$. For sufficiently small steps, this estimate is expected to hold. After K local steps, the total client update y_i^{r+1} is compressed and sent to the server, and the clients' errors are corrected via EF (Lines 7–10). The control variates are in turn updated via EF21 (Lines 11–13). The server collects the compressed updates v_i^{r+1} and δ_i^{r+1} and utilizes those to compute the new point (Line 15) and the new global control variate c^{r+1} (Line 16). The vectors are then and broadcasted to all clients at the end of the round (Line 17).

Note that the setting is deterministic: we assume access to full gradients $\nabla f_i(x)$. In real-world applications, this makes FedCALM-D impractical, and that is by design: it serves as a theoretical milestone that converges provably under both compression and local steps. We now state the convergence results for Algorithm 1. The proof of Theorem 4.7 can be found in the appendix.

Theorem 4.7. *Suppose Assumptions 4.3, 4.4 hold. Then Algorithm 1 to achieve ε-accuracy needs*

$$R = \mathcal{O}\left(\frac{\beta^2 \Delta L_*}{\varepsilon^2 (1-\varepsilon_q)^2}\right) \quad \textit{communication rounds,}$$

where $\varepsilon = \mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|_*\right]$, $\Delta = f(x^0) - f(x^*)$, *and ε_q is the Newton-Schulz steps error.*

Discussion. Our asymptotic rate coincides with that of SCAFCOM when only the non-stochastic terms are considered (Huang et al., 2024, Theorem 3):

$$R = \mathcal{O}\left(\frac{\beta \Delta L_F}{\varepsilon^2}\right), \quad \text{where} \quad \varepsilon = \mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|_F\right].$$

Thus, in the deterministic setting, Algorithm 1 attains the same asymptotic rate in ε under both compression and local steps. Our bound nonetheless differs from that of SCAFCOM.

First, we state convergence in the nuclear norm, which upper-bounds the Frobenius norm equation 1. Consequently, driving $\|\nabla f(x^r)\|_*$ below ε automatically drives $\|\nabla f(x^r)\|_F$ below the same ε, so our stationarity criterion is strictly stronger. Empirically, the Hessian spectrum of deep networks is approximately sparse, concentrated on a few large eigenvalues (Papyan, 2018), which keeps the two smoothness constants comparable, $L_* \approx L_F$. In this regime the two bounds carry essentially the same constants, and Algorithm 1 therefore certifies a stronger criterion at the same asymptotic cost — mirroring the advantage of Muon over Euclidean methods.

Inexact LMO analysis also introduces the Newton-Schulz error ε_q, which is thoroughly analysed by Kim & Oh (2026). The authors state it exponentially converges to zero, making it negligible.

Further, our bound carries a higher power of β. This stems from our compression procedure: the output of the first compressor (Line 8) enters, indirectly, the argument of the second (Line 11). Such nested compression is known to introduce an additional factor of β (Gao et al., 2025):

$$c_i^{r+1} = c_i^r + C\left[\nabla f_i\left(x^r + \frac{1}{M}\sum_{i=1}^{M} C(y_i^{r+1} + e_i^r)\right) - c_i^r\right].$$

4.4 Stochastic Algorithm

We now turn to the stochastic setting, where computing full gradients is infeasible and only unbiased estimates are available. We denote a stochastic gradient by $g_i(x, \xi)$, where i indexes the client and ξ is a random sample from its local dataset. We adopt the standard assumption of bounded variance.

Assumption 4.8. *An unbiased stochastic gradient is available for client i with bounded variance:*

$$\mathbb{E}[g_i(x,\xi)] = \nabla f_i(x),$$
$$\mathbb{E}\|g_i(x,\xi) - \nabla f_i(x)\|_2^2 \le \sigma^2.$$

Algorithm 2 `FedCALM`

Input: initial point $x^0 \in \mathbb{R}^d$, $c^0 = \nabla f(x^0)$, $e_i^0 = 0, m_i^{0,0} = c_i^0 = \nabla f_i(x^0)$ for every client $i = 1, \dots M$, number of rounds R, round length K

Hyperparameters: stepsize $\eta > 0$, momentum $\alpha \in [0; 1]$, number of Newton-Schulz steps q

1: **for** $r = 0, 1, \dots, R-1$ **do**
2: **for** clients $i = 1, \dots, M$ **do**
3: $x_i^{r,0} = x^r, m_i^{r,0} = m_i^{r-1,K}$
4: **for** local step $k = 0, 1, \dots, K-1$ **do**
5: $m_i^{r,k+1} = (1-\alpha)m_i^{r,k} + \alpha g_i(x_i^{r,k}, \xi_i^{r,k})$
6: $x_i^{r,k+1} = x_i^{r,k} + \eta \mathrm{NS}^q(m_i^{r,k} - c_i^r + c^r)$
7: **end for**
8: $y_i^{r+1} = x_i^{r,K} - x_i^{r,0}$
9: $v_i^{r+1} = C(y_i^{r+1} + e_i^r)$
10: $e_i^{r+1} = e_i^r + y_i^{r+1} - v_i^{r+1}$
11: Clients send v_i^{r+1} to the server
12: $\delta_i^{r+1} = C(m_i^{r,K} - c_i^r)$
13: $c_i^{r+1} = c_i^r + \delta_i^{r+1}$
14: Clients send δ_i^{r+1} to the server
15: **end for**
16: Server updates $x^{r+1} = x^r + \frac{1}{M}\sum_{i=1}^M v_i^{r+1}$
17: Server updates $c^{r+1} = c^r + \frac{1}{M}\sum_{i=1}^M \delta_i^{r+1}$
18: Server sends x^{r+1}, c^{r+1} to all clients
19: **end for**
20: **Output:** $\frac{1}{R}\sum_{r=0}^{R-1} x^r$.

We modify Algorithm 1 by adding a momentum, which leads to our main algorithm `FedCALM` (**Fed**erated **C**ompressed **A**nd **L**ocal **M**uon), given in Algorithm 2. `FedCALM` is a generalization of `FedCALM-D` with the momentum. The momentum in the local steps (Line 5) mitigates the variance within a round; this update rule coincides with that of `FedMuon`. The introduced momentum becomes the target for control variate update (Line 12). Notably, if one sets $\alpha = 1$ in Algorithm 2, that would directly yield Algorithm 1.

Theorem 4.9. *Suppose Assumptions 4.3, 4.4, and 4.8 hold. Then Algorithm 2 to achieve ε-accuracy needs*

$$R = \mathcal{O}\left(\frac{\Delta L_* \beta^2}{\varepsilon^2(1-\varepsilon_q)^2} + \frac{\Delta L_* \rho^2 \sigma^2 \beta}{M\varepsilon^4(1-\varepsilon_q)^4}\right) \text{ communication rounds,}$$

where $\varepsilon = \mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|_*\right]$, $\Delta = f(x^0) - f(x^*)$, *and ε_q is of Newton-Schulz steps error.*

The proof of Theorem 4.9 can be found in the appendix.

Discussion. In the stochastic setting, Algorithm 2 attains the same asymptotic rate as `SCAFCOM`, showing that it successfully assimilates both compression and local steps into the spectral FL. Our bound matches that of `SCAFCOM` in the two leading terms (Huang et al., 2024, Theorem 3):

$$R = \mathcal{O}\left(\frac{\beta \Delta L_F}{\varepsilon^2} + \frac{\Delta L_F \sigma^2}{MK\varepsilon^4}\right).$$

As in the deterministic case, our criterion is stronger: since the analysis is carried out in the spectral norm, we control a larger ε over the same asymptotic number of rounds. As before, both terms carry an additional factor of β due to nested compression. The stochastic term, in turn, carries an extra factor of ρ^2, which stems from the norm inequalities (1) and reflects the conversion between the Frobenius and nuclear norms. Up to this factor, the stochastic term coincides with that of `SCAFCOM`.

5 EXPERIMENTS

Setup. The task at hand is image classification on the *CIFAR-10* dataset (Krizhevsky et al., 2009). The chosen architecture is *ResNet-18* (He et al., 2016) – the standard benchmark for optimization problems (Meng et al., 2019). We examine the following federated setting. The training data is split among $M = 10$ clients with two different distributions. For more details, refer to the Appendix A.

Baselines. We compare our approach to the competitors discussed in Section 2: `SCAFCOM` (Huang et al., 2024), `FedMuon` (Takezawa et al., 2025), and `EF21-Muon` (Gruntkowska et al., 2025). We also implement `LocalSGD` (Stich, 2018) as a simple FL benchmark without compression.

Compression. We choose *TopK%* sparsification as a compression operator. This means the output vector retains a fixed fraction of entries with the largest absolute values. *Top1%* and *Top10%* compression rates are explored in this study. *Top100%* compression means communicating the full message, and is implmented for methods that do not account for compression (`FedMuon`, `LocalSGD`).

Communication details. To demonstrate the practical advantages of compression, we present the plots in terms of the communicated bits. Under assumption of a stable chanell, this is equivalent to the total communication time (Definition 4.6). In Table 1 we state the total bits transmitted in a single round for each method per client. Notably, `FedCALM` communicates two separate messages over the round and thus doubles the transmission size compared to other schemes with compression. Hence, `FedCALM` achives twice as much total communicated bits as `SCAFCOM` and `EF21-Muon`, while being trained for the same number of rounds (200 rounds).

5.1 EMPIRICAL RESULTS

Table 1: Comparison of the methods performance. Model size designates the amount of transmitted information compared to the size of a single model. Last columns report best accuracy across training for all methods in three setups: *CIFAR-10* with *Heterogeneous* and *Pathological* distributions, and *SST-2* with *IID* distribution.

Method	Compression	Model size	Accuracy (*CIFAR-10, Het.*)	Accuracy (*CIFAR-10, Path.*)	Accuracy (*SST-2*)
`FedCALM`	*Top1%*	$\times 0.02$	0.846	0.958	0.930
	Top10%	$\times 0.2$	0.816	0.959	0.928
`SCAFCOM`	*Top1%*	$\times 0.01$	0.460	0.904	0.933
	Top10%	$\times 0.1$	0.801	0.923	0.927
`EF21-Muon`	*Top1%*	$\times 0.01$	0.374	0.556	0.900
	Top10%	$\times 0.1$	0.601	0.746	0.912
`FedMuon`	*Top100%*	$\times 1$	0.826	0.903	0.930
`LocalSGD`	*Top100%*	$\times 1$	0.811	0.886	0.927

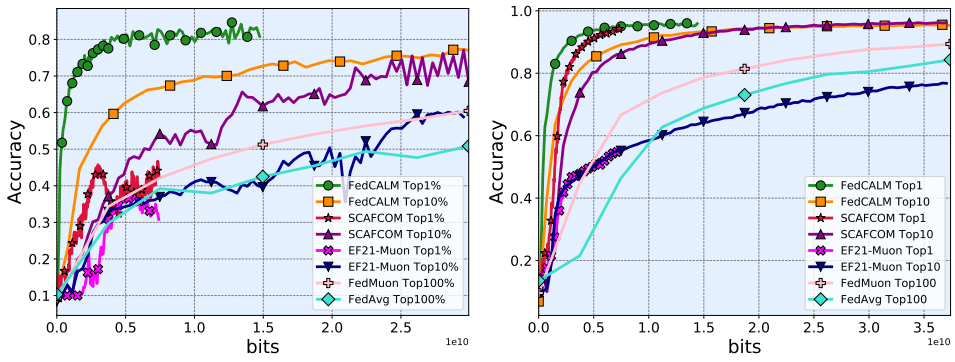

Figure 1: The test accuracy accross the training on *Heterogeneous distibution* (left) and *Pathological distribution* (right) for all methods. The horizontal axis designates the amount of communicated bits.

***CIFAR-10* experiments.** As demonstrated by the plots (see Figure 5.1), proposed FedCALM method consistently outperforms all of the competitors in terms of communicated bits. Across two data distributions, FedCALM and SCAFCOM show the best performance by a large margin, evidently demonstrating the superiority of methods with both local steps and compression. Moreover, stronger compression almost directly translates to faster convergence. However, in *Heterogeneous distribution*, SCAFCOM with *Top1%* compression performs worse than with the moderate compression *Top10%*. The reason for that may be the shear complexity of the problem: severe heterogeneity is combined with strong sparsification.

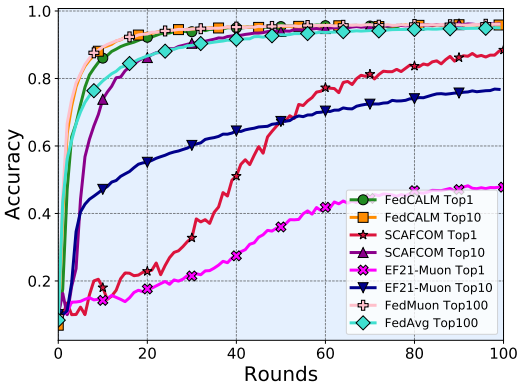

Figure 2: The test accuracy accross the training on *Pathological distibution* for all methods over the communication rounds. That is, compression rates are not taken into account.

Figure 2 clearly demonstrates the advantages of compression. The plot shows that the performance of most methods is similar in terms of rounds, and there is almost no slowdown with compression applied.

Moreover, the domination of spectral methods – FedMuon, FedCALM – is apparent compared to their euclidean counterparts – LocalSGD, SCAFCOM. Notably, the EF21-Muon stays left behind. The poor performance is likely rooted in the absence of local steps.

LLM fine-tuning. To further extend the scope of our validation, we conduct experiments with a LLM fine-tuning on *GLUE* benchmark (Wang et al., 2018). We report the best accuracy across training in Table 5.1. The results demonstrate the strong performance of proposed FedCALM in the different modality, which matches our previous experiments. The details of the setup are presented in Appendix A.1.

5.2 Network load

Moreover, we perform a careful study of network load and observe the theoretical advantages of Algorithm 2 in terms of communication uplink. The results are described in Appendix A.2.

6 Limitations

First, we rely on smoothness of the objective. In practice, it is fulfilled only locally, however, it is classic in the field. Second, the scope of our empirical validation is limited, and the future work can extend the scale of the experiments. Lastly, we consider all clients to be available at every communication round, while this may not be the case in real-world scenarios.

7 Conclusion

In this paper, we address the communication bottleneck of FL by proposing a novel spectral method, FedCALM-D , which utilizes two key techniques of FL: local steps and communication compression. Crucially, we manage to perform two separate communications with proper error compensation mechanisms for both. The scheme provably converges with arbitrary heterogeneity in the deterministic case.

Furthermore, we expand our approach with FedCALM – the stochastic version of Algorithm 1. We obtain heterogeneity-free convergence bounds for FedCALM, asymptotically matching the classic rates of SCAFCOM in the spectral setting.

We perform experimental validation on computer vision and language processing tasks and demonstrate the empirical advantages of FedCALM over standard baselines.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work by proof-reading the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our theoretical analysis is presented in Appendix B. The proposed method is described in detail in Section 4. The details of experimental validation are listed in Appendix A.4, including the hardware and hyperparameter choices. The dataset and the benchmark task are open-sourced.

REFERENCES

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International conference on machine learning*, pp. 560–569. PMLR, 2018.

Aleksandr Beznosikov, Peter Richtárik, Michael Diskin, Max Ryabinin, and Alexander Gasnikov. Distributed methods with compressed communication for solving variational inequalities, with theoretical guarantees. *Advances in Neural Information Processing Systems*, 35:14013–14029, 2022.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

Dmitry Bylinkin, Daniil Medyakov, Vladimir Aletov, Nail Bashirov, Michael Diskin, and Aleksandr Beznosikov. A robust training method for federated learning with partial participation. 2025.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Yuan Gao, Anton Rodomanov, Jeremy Rack, and Sebastian U Stich. Accelerated distributed optimization with compression and error feedback. *arXiv preprint arXiv:2503.08427*, 2025.

Eduard Gorbunov, Konstantin P Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster nonconvex distributed learning with compression. In *International Conference on Machine Learning*, pp. 3788–3798. PMLR, 2021a.

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 3556–3564. PMLR, 2021b.

Kaja Gruntkowska, Alexander Gaponov, Zhirayr Tovmasyan, and Peter Richtárik. Error feedback for muon and friends. *arXiv preprint arXiv:2510.00643*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pp. 2350–2358. PMLR, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Xinmeng Huang, Ping Li, and Xiaoyun Li. Stochastic controlled averaging for federated learning with communication compression. In *International Conference on Learning Representations*, volume 2024, pp. 27956–27994, 2024.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International conference on machine learning*, pp. 3252–3261. PMLR, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International conference on artificial intelligence and statistics*, pp. 4519–4529. PMLR, 2020.

Gyu Yeol Kim and Min-hwan Oh. Convergence of muon with newton-schulz. *arXiv preprint arXiv:2601.19156*, 2026.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International conference on machine learning*, pp. 5381–5393. PMLR, 2020.

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

Khiem Le, Nhan Luong-Ha, Manh Nguyen-Duc, Danh Le-Phuoc, Cuong Do, and Kok-Seng Wong. Exploring the practicality of federated learning: A survey towards the communication perspective. *arXiv preprint arXiv:2405.20431*, 2024.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Xiaoyun Li and Ping Li. Analysis of error feedback in federated non-convex optimization with biased compression: Fast convergence and partial participation. In *International Conference on Machine Learning*, pp. 19638–19688. PMLR, 2023.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025a.

Xiaodong Liu, Huaizhou Qi, Suisui Jia, Yongjing Guo, and Yang Liu. Recent advances in optimization methods for machine learning: a systematic review. *Mathematics*, 13(13):2210, 2025b.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Yequan Zhao, Yupeng Su, Zi Yang, and Zheng Zhang. Muon ^{2}: Boosting muon via adaptive second-moment preconditioning. *arXiv preprint arXiv:2604.09967*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Debin Meng, Xiaojiang Peng, Kai Wang, and Yu Qiao. Frame attention networks for facial expression recognition in videos. In *2019 IEEE international conference on image processing (ICIP)*, pp. 3866–3870. IEEE, 2019.

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, 40(5):1181–1196, 2025.

Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025.

Xun Qian, Hussein Rammal, Dmitry Kovalev, and Peter Richtarik. Muon is provably faster with momentum variance reduction. *arXiv preprint arXiv:2512.16598*, 2025.

Xun Qian, Alexander Gaponov, Grigory Malinovsky, and Peter Richtárik. Communication-efficient gluon in federated learning. *arXiv preprint arXiv:2604.10689*, 2026.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again!(bridging theory and practice of lmo-based optimizers for llms). *arXiv preprint arXiv:2505.13416*, 2025.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, volume 2014, pp. 1058–1062. Singapore, 2014.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Chongjie Si, Debing Zhang, and Wei Shen. Adamuon: Adaptive muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Sgd with delayed gradients. *Journal of Machine Learning Research*, 21(237):1–36, 2020.

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in neural information processing systems*, 31, 2018.

Yuki Takezawa, Anastasia Koloskova, Xiaowen Jiang, and Sebastian U Stich. Fedmuon: Federated learning with bias-corrected lmo-based optimization. *arXiv preprint arXiv:2509.26337*, 2025.

Benjamin Thérien, Xiaolong Huang, Aaron Defazio, Irina Rish, and Eugene Belilovsky. Muloco: Muon is a practical inner optimizer for diloco. *arXiv preprint arXiv:2505.23725*, 2025.

Alexander Tyurin and Peter Richtárik. Dasha: Distributed nonconvex optimization with communication compression, optimal oracle complexity, and no client synchronization. *arXiv preprint arXiv:2202.01268*, 2022.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020.

Xidong Wu, Feihu Huang, Zhengmian Hu, and Heng Huang. Faster adaptive federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10379–10387, 2023.

Ruijie Zhang, Yequan Zhao, Ziyue Liu, Zhengyang Wang, Dongyang Li, Yupeng Su, Sijia Liu, and Zheng Zhang. Teon: Tensorized orthonormalization beyond layer-wise muon for large language model pre-training. *arXiv preprint arXiv:2601.23261*, 2026.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 16(12):9851–9915, 2025.

APPENDIX

A ADDITIONAL EXPERIMENTS

A.1 LANGUAGE MODEL FINE-TUNING ON SST-2

Setup. We extend our experiemnts to a bigger scale with a NLP task. We use SST-2 problem from the GLUE benchmark (Wang et al., 2018): binary sentiment classification of single sentences taken from movie reviews. Sentences are tokenized with the RoBERTa tokenizer and truncated or padded to 128 tokens. The rest of the setting is described in details in section A.4. The results of the experiment are presented in Table 2. Each of the methods achieved roughly 93% accuracy in the totla runtime of 30 communication rounds.

Method	Compression	Accuracy
FedCALM	*Top1%*	0.930
	Top10%	0.928
SCAFCOM	*Top1%*	0.933
	Top10%	0.927
EF21-Muon	*Top1%*	0.900
	Top10%	0.912
FedMuon	*Top100%*	0.930
LocalSGD	*Top100%*	0.927

Table 2: The test accuracy accross the training on *IID distibution* for all methods over 30 communication rounds in *SST-2* experiments.

A.2 NETWORK LOAD

In the main part we described the communication efficiency of FedCALM in terms of accuracy over the rounds and transmitted bits. However, we left out the network load during the runs and did not compare the methods based on the communication time. In this section, we report the corresponding metrics. In particular, Table 3 shows the following statistics: the peak client GPU usage during local training, total uplink memory and aggregation time at the server. The table shows that while GPU is utilized similarly across methods, the uplink and aggregation time benefit strongly from the compression. FedCALM requires almost twice as much resources as SCAFCOM and EF21-Muon due to the two-stage structure of the method. However, this is alleviated by the faster convergence. Table 4 reports the total uplink time and memory untill the methods reach 90% accuracy. The results demonstrate the acceleration provided by FedCALM in terms of both of the metrics. Another statistic that we provide is the average client memory usage over the training. Our method demonstrates almost the lowest costs, only rivaled by FedMuon. EF21-Muon is absent in Table 4 since it did not achieve target accuracy.

Table 3: Network statistics for *CIFAR-10* experiments.

Method	Client GPU	Compression	Uplink	Aggr. time
FedCALM	208.9 MB	*Top1%*	8.94 MB	171 ms
		Top10%	89.4 MB	815 ms
SCAFCOM	235.9 MB	*Top1%*	4.47 MB	85 ms
		Top10%	44.7 MB	407 ms
EF21-Muon	181.3 MB	*Top1%*	4.47 MB	85 ms
		Top10%	44.7 MB	407 ms
FedMuon	225.1 MB	*Top100%*	894 MB	3.62 s
LocalSGD	181.6 MB	*Top100%*	447 MB	3.62 s

Table 4: Memory usage and total aggregation time to reach 90% accuracy for *CIFAR-10* experiments.

Method	Compression	Uplink	Time	Average memory
FedCALM	*Top1%*	143 MB	2.73 s	7.87 MB
	Top10%	1072 MB	9.78 s	5.90 MB
SCAFCOM	*Top1%*	522 MB	9.94 s	57.6 MB
	Top10%	11803 MB	11.8 s	14.2 MB
FedMuon	*Top100%*	4917 MB	39.2 s	4.39 MB
LocalSGD	*Top100%*	13857 MB	110 s	13.8 MB

Hardware. We consider cross-silo setup for M=10 clients with uplink time of 100 Mb/s and 50 ms setup latency. Each client is leveraging a single CPU (Intel Xeon 2.20 GHz) and a single GPU (NVIDIA A100) for computation, and the server operates with one CPU. The experiments are implemented in Python using the PyTorch library.

Network load for SST-2. Similarly to the experiments on *CIFAR-10*, the advantages of compression are clearly visible when the costs are properly considered. In Table 5, we report client peak GPU usage, uplink memory and time. FedCALM continues to showcase low resource demand with an overhead compared to SCAFCOM, EF21-Muon. Table 6 demonstrates the gains of our method over the competitors when the total costs are considered untill 90% accuracy is achieved. This highlights the dominance of FedCALM among the methods: it requires the least uplink time and memory to solve the task, which signifies strong practical performance.

Table 5: Network statistics for *SST-2* experiments.

Method	Client memory	Compression	Uplink	Aggr. time
FedCALM	2952.98 MB	*Top1%*	0.74 MB	109 ms
		Top10%	7.39 MB	641 ms
SCAFCOM	2947.30 MB	*Top1%*	0.37 MB	80 ms
		Top10%	3.70 MB	346 ms
EF21-Muon	2937.51 MB	*Top1%*	0.37 MB	80 ms
		Top10%	3.70 MB	346 ms
FedMuon	2948.33 MB	*Top100%*	73.91 MB	5.96 s
LocalSGD	2937.95 MB	*Top100%*	36.96 MB	3.01 s

Table 6: Memory usage and total aggregation time to reach 91% accuracy for *SST-2* experiments.

Method	Compression	Uplink	Time	Average memory
FedCALM	*Top1%*	4 MB	546 ms	615.20 MB
	Top10%	15 MB	1.28 s	250.48 MB
SCAFCOM	*Top1%*	7 MB	1.43 s	2210.48 MB
	Top10%	22 MB	2.07 s	736.83 MB
EF21-Muon	*Top1%*	–	–	–
	Top10%	89 MB	8.30 s	2937.51 MB
FedMuon	*Top100%*	222 MB	17.89 s	368.54 MB
LocalSGD	*Top100%*	111 MB	9.02 s	367.24 MB

A.3 Ablation Study

To validate the stability of proposed FedCALM, we provide ablation over number of Newton-Schulz steps q. Figure 3 and Table 7 demonstrate the robustness of our method to the choice of q. The performance gets worse only slightly if a single Newton-Schulz step is applied. This means that FedCALM is robust to the choice of q, which makes it more easily applicable in practice.

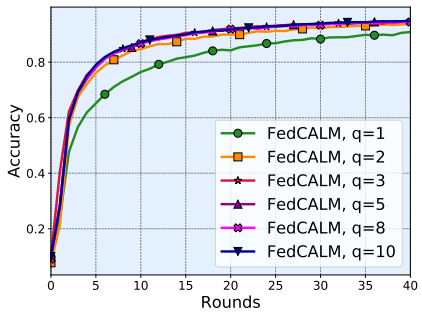

Figure 3: The ablation of `FedCALM` over the number of Newton-Schulz steps q with *CIFAR-10* experiments. The setup is the *Pathological distribution* with *Top10%* compression rate.

q	Accuracy
1	0.912
2	0.926
3	0.927
5	0.928
8	0.928
10	0.929

Table 7: The ablation of `FedCALM` over the number of Newton-Schulz steps q with *SST-2* experiments. The setup is the *IID distribution* with *Top10%* compression rate during 20 rounds.

A.4 Experimental Details

In this section we describe the specifics of our experimental setup.

GLUE setup. The backbone is RoBERTa-base (125M parameters) with LoRA adapters Hu et al. (2022) of rank $r = 4$ and scaling $\alpha = 2r = 8$, attached to the query, key and value projections and to every dense projection of each of the 12 transformer blocks. Per round, each client performs at most 20 local optimizer steps on minibatches of 32 sentences, i.e. it sees 640 samples between two communications. The local optimizer is Muon with momentum 0.95 and 5 Newton–Schulz iterations, applied to the matrix-shaped LoRA factors. Every run lasts 30 communication rounds.

Hyperparameters. We present the hyperparameters of all methods for final runs. **CIFAR-10 experiments.** The following parameters are shared across all methods: batch size 256, 5 local steps. The stepsizes were chosen for the methods with the grid search on a following grid: (0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1). Final stepsizes are listed in Table 8. For spectral methods (`FedCALM`, `EF21-Muon`, `FedMuon`) `Muon` optimizer was implemented with 5 Newton-Schulz steps and momentum 0.95. For the rest of the methods we implemented `Adam` optimizer with momentums (β_1, β_2)=(0.9, 0.999) and weight decay λ=0.001.

Table 8: Stepsizes for the final runs of *CIFAR-10* experiments.

Method	Compression	*Heter. distr.*	*Path. distr.*
`FedCALM`	*Top1%*	0.01	0.01
	Top10%	0.01	0.01
`SCAFCOM`	*Top1%*	0.0005	0.0005
	Top10%	0.0005	0.0005
`EF21-Muon`	*Top1%*	0.0005	0.0001
	Top10%	0.0005	0.0001
`FedMuon`	*Top100%*	0.005	0.01
`LocalSGD`	*Top100%*	0.0001	0.0001

GLUE experiments. For every method the learning rate is tuned over $0.0005, 0.001, 0.025, 0.005, 0.02$. Because Muon normalizes its update direction, its step size does not live on the same scale as the SGD or AdamW steps of the reference works, so the grid is treated as a starting range rather than as a transferred setting. The tuning is carried out at *Top10%* and *Top1%* indendently, so that a loss of accuracy under aggressive sparsification cannot be an artifact of a step size inherited from a milder compression level.

Table 9: Stepsizes for the final runs of *SST-2* experiments.

Method	Compression	Stepsize
FedCALM	*Top1%*	0.005
	Top10%	0.005
SCAFCOM	*Top1%*	0.0025
	Top10%	0.0025
EF21-Muon	*Top1%*	0.0005
	Top10%	0.0005
FedMuon	*Top100%*	0.005
LocalSGD	*Top100%*	0.0025

Distributions. For the *GLUE* experiments, the training data is split among $M = 10$ clients, all of which participate in every round. The split is IID: within each class the sentences are shuffled. Strong heterogeneity is not explored to ensure the stability of the training.

CIFAR-10 experiments. For *Pathological distribution* each client was assigned a number of objects from every class sampled uniformly from $U[0.9,1.1]\times$ 475. For *Heterogeneous distribution*, 4 clients were evenly assigned with objects from first 4 classes, and the rest 6 classes were evenly split among last 6 clients. However, each client has at least 2 objects from every class for stability of cross-enthropy loss. Tables 12, 13, 10, 11 report exact amount of object of each class in clients datasets.

Table 10: Number of objects in client datasets for *Pathological distibution* (clients 1-5).

	Client 1	Client 2	Client 3	Client 4	Client 5
Class 0	1184	1184	1184	1184	2
Class 1	1185	1185	1185	1185	2
Class 2	1185	1185	1185	1185	2
Class 3	1184	1184	1184	1184	2
Class 4	2	2	2	2	790
Class 5	2	2	2	2	791
Class 6	2	2	2	2	791
Class 7	2	2	2	2	790
Class 8	2	2	2	2	790
Class 9	2	2	2	2	790

Table 11: Number of objects in client datasets for *Pathological distibution* (clients 6-10).

	Client 6	Client 7	Client 8	Client 9	Client 10
Class 0	2	2	2	2	2
Class 1	2	2	2	2	2
Class 2	2	2	2	2	2
Class 3	2	2	2	2	2
Class 4	790	790	790	790	790
Class 5	791	791	791	791	791
Class 6	791	791	791	791	791
Class 7	790	790	790	790	790
Class 8	790	790	790	790	790
Class 9	790	790	790	790	790

Table 12: Number of objects in client datasets for *Heterogeneous distibution*.

	Client 1	Client 2	Client 3	Client 4	Client 5
Class 0	440	448	400	428	450
Class 1	501	481	488	488	502
Class 2	465	479	509	518	513
Class 3	470	428	476	495	437
Class 4	449	482	477	486	478
Class 5	495	512	547	507	524
Class 6	452	428	398	438	426
Class 7	471	495	534	481	519
Class 8	471	468	439	439	440
Class 9	536	529	482	470	461

Table 13: Number of objects in client datasets for *Pathological distibution*.

	Client 6	Client 7	Client 8	Client 9	Client 10
Class 0	421	448	412	430	482
Class 1	510	479	523	535	503
Class 2	469	496	446	504	497
Class 3	447	527	490	435	486
Class 4	443	506	480	460	425
Class 5	497	501	500	525	482
Class 6	469	450	441	419	435
Class 7	516	447	514	483	477
Class 8	456	437	427	452	437
Class 9	522	459	517	507	526

B PROOFS

First, we list some useful inequalities. Let u, v be vectors in $\mathbb{R}^d$, $\{a_j\}_{j=1}^n$ be a sequence of vectors, and $\varphi : \mathbb{R}^d \to \mathbb{R}$ be a scalar L_*-smooth function.

$$\langle u, v\rangle \leq \|u\| \cdot \|v\|_*, \tag{Conj}$$

$$\|u + v\| \leq \|u\| + \|v\|, \tag{Norm}$$

$$\|\frac{1}{n}\sum_{j=1}^{n} a_j\|_* \leq \frac{1}{n}\sum_{j=1}^{n} \|a_j\|_*, \tag{Jen}$$

$$\varphi(v) \leq \varphi(u) + \langle \nabla\varphi(u), v-u\rangle + \frac{L}{2}\|u-v\|^2, \tag{Smooth}$$

$$\left\|\sum_{i=1}^n a_i\right\|^2 \leq n\sum_{i=1}^n \|a_i\|^2. \tag{CS}$$

Then, we prove a few auxiliary lemmas.

Lemma B.1. *For any vector u, we have*

$$\langle u, \mathrm{lmo}(u)\rangle = -\|u\|_*.$$

Proof. The proof follows from the definition of conjugate norm.

$$-\|u\|_* = -\max_{\|v\|\leq 1}\langle u, v\rangle = \min_{\|v\|\leq 1}\langle u, v\rangle = \langle u, \arg\min_{\|v\|\leq 1}(\langle u, v\rangle)\rangle = \langle u, \mathrm{lmo}(u)\rangle.$$

□

Following Kim & Oh (2026), we introduce the error factor of Newton-Schultz steps ε_q. Formally,

$$\varepsilon_q = \max_{r=1\ldots R}\max_{k=1\ldots K}\max_{i=1\ldots M}\|\mathrm{lmo}(p_i^{r,k}) - \mathtt{NS}^q(p_i^{r,k})\|, \tag{3}$$

where $p_i^{r,k}$ denotes $\nabla f_i(x_i^{r,k}) - c_i^r + c^r$ in the context of the algorithm 1 and $p_i^{r,k} = m_i^{r,k} - c_i^r + c^r$ for the algorithm 2. Kim & Oh (2026) showed, that $\varepsilon_q < 1$ and approaches zero at a doubly exponential rate over q.

Corollary B.2. *During the run of Algorithm 1 or 2, the following holds for any $p_i^{r,k}$ (as defined earlier):*

$$\|NS(p_i^{r,k})\| \leq 1 + \varepsilon_q.$$

Proof. We utilize the unit length of LMO and introduced error of Newton-Schultz steps:

$$\|\mathtt{NS}^q(p_i^{r,k})\| \overset{Norm}{\leq} \|\mathrm{lmo}(p_i^{r,k})\| + \|\mathrm{lmo}(p_i^{r,k}) - \mathtt{NS}^q(p_i^{r,k})\| \overset{(3)}{\leq} \|\mathrm{lmo}(p_i^{r,k})\| + \varepsilon_q \leq 1 + \varepsilon_q.$$

□

Lemma B.3. *During the run of Algorithm 1 or 2, the following inequality holds for each $i = 1, \ldots M$ and $r = 1, \ldots R+1$:*

$$\|x_i^{r,k} - x_r\| \leq \eta k(1+\varepsilon_q).$$

Proof. The proof relies on the definition of LMO. Since the $\arg\min$ is taken over the $\|\cdot\|$-norm ball, the LMO output has the norm of 1 or less. Thus,

$$\|x_i^{r,k} - x^r\| = \|x_i^{r,0} - x_i^{r,0}\| = \|\sum_{l=0}^{k-1}(x_i^{r,l} - x_i^{r,l+1})\| \overset{Norm}{\leq} \sum_{l=0}^{k-1}\|x_i^{r,l} - x_i^{r,l+1}\|.$$

As we noted, each update $x_i^{r,l} - x_i^{r,l+1}$ is an output of $\mathtt{NS}$ steps scaled by η. Corollary B.2 implies that the norm of every update equals to or less than $\eta(1+\varepsilon_q)$, which concludes the proof:

$$\|x_i^{r,k} - x^r\| \leq \sum_{l=0}^{k-1}\|x_i^{r,l} - x_i^{r,l+1}\| \leq \sum_{l=0}^{k}\eta(1+\varepsilon_q) = \eta k(1+\varepsilon_q).$$

□

Corollary B.4. *From the definition of y_i^r it is easy to observe that $\|y_i^r\| \leq \eta K(1+\varepsilon_q)$ for any client i and round r.*

Lemma B.5. *During the run of Algorithm 1 or 2, the following inequality holds for each $i = 1, \ldots M$ and $r = 1, \ldots R+1$:*

$$\|e_i^r\| \leq \eta K\beta(1+\varepsilon_q).$$

Proof. This proof resembles classic error analysis for EF.

$$\begin{aligned}\|e_i^r\| &= \|e_i^{r-1} + y_i^r - v_i^r\| \\ &= \|e_i^{r-1} + y_i^r - C(y_i^r + e_i^{r-1})\| \\ &\overset{(Def.2)}{\leq} (1 - \frac{1}{\beta})\|y_i^r + e_i^{r-1}\| \\ &\overset{Norm}{\leq} (1 - \frac{1}{\beta})\|y_i^r\| + (1 - \frac{1}{\beta})\|e_i^{r-1}\|.\end{aligned}$$

Now, we unroll a recurcion over $\|e_i^r\|$ to $r = 0$:

$$\begin{aligned}\|e_i^r\| &\leq (1 - \frac{1}{\beta})\|y_i^r\| + (1 - \frac{1}{\beta})\|e_i^{r-1}\| \\ &\dots \\ &\leq \sum_{t=0}^{r}(1 - \frac{1}{\beta})^{r+1-t}\|y_i^t\| + (1 - \frac{1}{\beta})^r\|e_i^0\| \\ &\overset{(Cor.B.4)}{\leq} \sum_{t=0}^{r}(1 - \frac{1}{\beta})^{r+1-t}\eta K(1 + \varepsilon_q) + (1 - \frac{1}{\beta})^r\|e_i^0\| \\ &\leq \eta K(1 + \varepsilon_q)\sum_{t=0}^{\infty}(1 - \frac{1}{\beta})^t + (1 - \frac{1}{\beta})^r\|e_i^0\| \\ &= \eta K\beta(1 + \varepsilon_q) + (1 - \frac{1}{\beta})^r\|e_i^0\|.\end{aligned}$$

Finally, we note that $\|e_i^0\| = 0$ due to initialization.

□

Lemma B.6. *During the run of Algorithm 1 or 2, the following inequality holds for each round $r = 1, \dots R + 1$:*

$$\|x^r - x^{r-1}\| \leq 6\eta K\beta.$$

Proof. We start with the definition of the step:

$$\begin{aligned}\|x^r - x^{r-1}\| &= \|\frac{1}{M}\sum_{i=1}^{M} v_i^r\| \\ &\overset{(Norm)}{\leq} \frac{1}{M}\sum_{i=1}^{M}\|C(y_i^r + e_i^{r-1})\| \\ &\overset{(Norm)}{\leq} \frac{1}{M}\sum_{i=1}^{M}\|C(y_i^r + e_i^{r-1}) - y_i^r - e_i^{r-1}\| + \frac{1}{M}\sum_{i=1}^{M}\|y_i^r + e_i^{r-1}\| \\ &\overset{(Def.2)}{\leq} \frac{1}{M}\sum_{i=1}^{M}(1 - \frac{1}{\beta})\|y_i^r + e_i^{r-1}\| + \frac{1}{M}\sum_{i=1}^{M}\|y_i^r + e_i^{r-1}\| \\ &= \frac{2 - \frac{1}{\beta}}{M}\sum_{i=1}^{M}\|y_i^r + e_i^{r-1}\| \\ &= \frac{2 - \frac{1}{\beta}}{M}\sum_{i=1}^{M}(\|y_i^r\| + \|e_i^{r-1}\|).\end{aligned}$$

Then, we apply Lemma B.5 and Corollary B.4 to bound $\|e_i^{r-1}\|$ and $\|y_i^r\|$ individually:

$$\|x^r - x^{r-1}\| \leq \frac{2 - \frac{1}{\beta}}{M}\sum_{i=1}^{M}(\|y_i^r\| + \|e_i^{r-1}\|)$$

$$
\begin{aligned}
&\leq \frac{2-\frac{1}{\beta}}{M}\sum_{i=1}^{M}(\eta K(1+\varepsilon_q)+\eta K\beta(1+\varepsilon_q))\\
&= \frac{(2-\frac{1}{\beta})(\beta+1)}{M}(1+\varepsilon_q)\eta K\sum_{i=1}^{M}1\\
&= (1+2\beta-\frac{1}{\beta})\eta K(1+\varepsilon_q)\\
&\leq 6\beta\eta K,
\end{aligned}
$$

where the last inequality was achived via trivial bounds $1 \leq \beta, \frac{1}{\beta} > 0$ and $\varepsilon_q \leq 1$.

□

B.1 Proof of Theorem 4.7

Now, we are ready to prove the convergence for Algorithm 1.

Theorem B.7 (Theorem 4.7)**.** *Suppose Assumption 4.3 holds. Then Algorithm 1 to achieve ε-accuracy needs*

$$R = \mathcal{O}\left(\frac{\beta^2\Delta(L_* + L_2)}{\varepsilon^2(1-\varepsilon_q)^2}\right) \text{ communication rounds,}$$

where $\varepsilon = \mathbb{E}\left[\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_*\right]$, and $\Delta = f(x^0) - f(x^R)$, and ε_q is the error of Newton-Schultz iterations.

Proof. Classicly for the EF, we start with defining a virtual sequence.

$$\tilde{x}^r \overset{\text{def}}{\equiv} x^r - \frac{1}{M}\sum_i e_i^r.$$

The vectors $\tilde{x}^r$ are useful for the proof but never explicitly calculated in the algorithm (hence the name *virtual*). The main feature of the sequence is its update rule:

$$
\begin{aligned}
\tilde{x}^{r+1} =& x^{r+1} - \frac{1}{M}\sum_i e_i^{r+1}\\
=& (x^r - \frac{1}{M}\sum_i v_i^{r+1}) - \frac{1}{M}\sum_i (e_i^r + y_i^{r+1} - v_i^{r+1})\\
=& x^r - \frac{1}{M}\sum_i \left(v_i^{r+1} + e_i^r + y_i^{r+1} - v_i^{r+1}\right)\\
=& (x^r - \frac{1}{M}\sum_i e_i^{r+1}) - \frac{1}{M}\sum_i y_i^{r+1}\\
=& \tilde{x}^r - \frac{1}{M}\sum_i y_i^{r+1}.
\end{aligned}
$$

Now, we write the Smoothness inequality substituting $v = \tilde{x}^{r+1}$ and $u = \tilde{x}^r$:

$$
\begin{aligned}
f(\tilde{x}^{r+1}) &\overset{(Smooth)}{\leq} f(\tilde{x}^r) - \langle\nabla f(\tilde{x}^r), \frac{1}{M}\sum_{i=1}^{M} y_i^{r+1}\rangle + \frac{L_*}{2}\|\frac{1}{M}\sum_{i=1}^{M} y_i^{r+1}\|^2\\
&\overset{(CS)}{\leq} f(\tilde{x}^r) - \frac{1}{M}\sum_{i=1}^{M}\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle + \frac{L_*}{2M}\sum_{i=1}^{M}\|y_i^{r+1}\|^2.
\end{aligned}
$$

Let us analyse scalar product separately.

$$-\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle \quad = \quad -\langle\nabla f(\tilde{x}^r), x^r - x_i^{r,K}\rangle$$

$$
\begin{aligned}
&= \langle \nabla f(\tilde{x}^r), \sum_{k=0}^{K-1} \eta \mathtt{NS}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&= \langle \nabla f(\tilde{x}^r), \sum_{k=0}^{K-1} \eta \,\mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \langle \nabla f(\tilde{x}^r), \sum_{k=0}^{K-1} \eta [\mathrm{lmo} - \mathtt{NS}^q](\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&= \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), [\mathrm{lmo} - \mathtt{NS}^q](\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\overset{(Conj)}{\leq} \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \|[\mathrm{lmo} - \mathtt{NS}^q](\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\| \\
&\overset{(3)}{\leq} \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \varepsilon_q.
\end{aligned}
$$

Here, we transitioned from inexact NS to LMO, gaining the error term proportonal to ε_q. Next, we operate over the first term to utilize Lemma B.1.

$$
\begin{aligned}
-\langle \nabla f(\tilde{x}^r), y_i^{r+1}\rangle &\leq \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \varepsilon_q \\
&= \eta \sum_{k=0}^{K-1} \langle \nabla f_i(x_i^{r,k}) - c_i^r + c^r, \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r, \mathrm{lmo}(\nabla f_i(x_i^{r,k}) - c_i^r + c^r)\rangle \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \varepsilon_q.
\end{aligned}
$$

We apply Lemma B.1 to the first term, retrieving the conjugate norm. For the second term, we use Conj inequality keeping in mind the unit norm of LMO.

$$
\begin{aligned}
-\langle \nabla f(\tilde{x}^r), y_i^{r+1}\rangle &\leq -\eta \sum_{k=0}^{K-1} \|\nabla f_i(x_i^{r,k}) - c_i^r + c^r\|_* \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r\|_* \cdot 1 \\
&\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \varepsilon_q
\end{aligned}
$$

$$
\begin{aligned}
&\overset{Norm}{\leq} && -\eta \sum_{k=0}^{K-1} \|\nabla f(x^r)\|_* \\
&&& +2\eta \sum_{k=0}^{K-1} \|\nabla f(x^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r\|_* \\
&&& +\eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* \\
&&& +\eta \sum_{k=0}^{K-1} (\|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* + \|\nabla f(x^r)\|_*)\varepsilon_q \\
&\overset{Norm}{\leq} && -\eta(1-\varepsilon_q) \sum_{k=0}^{K-1} \|\nabla f(x^r)\|_* \\
&&& +2\eta \sum_{k=0}^{K-1} \|\nabla f(x^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r\|_* \\
&&& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_*.
\end{aligned}
$$

The last term can be easilty bounded via Lemma B.5:

$$
\begin{aligned}
\|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* &\overset{(Ass.4.3)}{\leq} L_* \|\tilde{x}^r - x^r\| \\
&= L_* \|\frac{1}{M} \sum_{i=1}^{M} e_i^r\| \\
&\overset{(Norm)}{\leq} L_* \frac{1}{M} \sum_{i=1}^{M} \|e_i^r\| \\
&\overset{(Lem.B.5)}{\leq} \frac{L_*}{M} \sum_{i=1}^{M} \eta K \beta (1+\varepsilon_q) \\
&= L_* \eta K \beta (1+\varepsilon_q).
\end{aligned}
$$

Now, the second term deserves much attention. First, we remove the dependance on k:

$$
\begin{aligned}
\|\nabla f(x^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r\|_* &\overset{Norm}{\leq} \|\nabla f(x^r) - c^r\|_* + \|\nabla f_i(x^r) - c_i^r\|_* + \|\nabla f_i(x^r) - \nabla f_i(x_i^{r,k})\|_* \\
&\overset{4.3}{\leq} \|\nabla f(x^r) - c^r\|_* + \|\nabla f_i(x^r) - c_i^r\|_* + L_* \|x^r - x_i^{r,k}\| \\
&\overset{(Lem.B.3)}{\leq} \|\nabla f(x^r) - c^r\|_* + \|\nabla f_i(x^r) - c_i^r\|_* + L_* \eta k (1+\varepsilon_q) \\
&\leq \|\nabla f(x^r) - c^r\|_* + \|\nabla f_i(x^r) - c_i^r\|_* + L_* \eta K (1+\varepsilon_q).
\end{aligned}
$$

Next, we bound the difference of control variate and respective gradient.

$$
\begin{aligned}
\|\nabla f_i(x^r) - c_i^r\|_* &= \|\nabla f_i(x^r) - c_i^{r-1} - \delta_i^r\|_* \\
&= \|\nabla f_i(x^r) - c_i^{r-1} - C(\nabla f_i(x_i^{r-1,K}) - c_i^{r-1})\|_* \\
&\overset{(Norm)}{\leq} \|\nabla f_i(x_i^{r-1,K}) - c_i^{r-1} - C(\nabla f_i(x_i^{r-1,K}) - c_i^{r-1})\|_* \\
&\quad + \|\nabla f_i(x_i^{r-1,K}) - \nabla f_i(x^r)\|_* \\
&\overset{(Def.2)}{\leq} (1 - \frac{1}{\beta}) \|\nabla f_i(x_i^{r-1,K}) - c_i^{r-1}\|_* + \|\nabla f_i(x_i^{r-1,K}) - \nabla f_i(x^r)\|_* \\
&\overset{(Norm)}{\leq} (1 - \frac{1}{\beta}) \|\nabla f_i(x^{r-1}) - c_i^{r-1}\|_* + 2\|\nabla f_i(x_i^{r-1,K}) - \nabla f_i(x^{r-1})\|_*
\end{aligned}
$$

$$
\begin{aligned}
& +\|\nabla f_i(x_i^{r-1}) - \nabla f_i(x^r)\|_* \\
\overset{(Ass.4.3)}{\leq} \quad & (1-\frac{1}{\beta})\|\nabla f_i(x^{r-1}) - c_i^{r-1}\|_* + 2L_*\|x^{r-1,K} - x^{r-1}\| \\
& +L_*\|x^{r-1} - x^r\| \\
\overset{(Lem.B.6)}{\leq} \quad & (1-\frac{1}{\beta})\|\nabla f_i(x^{r-1}) - c_i^{r-1}\|_* + 18L_*\eta K\beta.
\end{aligned}
$$

Then, we unroll a recursion over r to the first round.

$$
\begin{aligned}
\|\nabla f_i(x^r) - c_i^r\|_* &\leq (1-\frac{1}{\beta})\|\nabla f_i(x^{r-1}) - c_i^{r-1}\|_* + 18L_*\eta K\beta \\
&\cdots \\
&\leq (1-\frac{1}{\beta})^r\|\nabla f_i(x^0) - c_i^0\|_* + \sum_{t=0}^{r-1} 18(1-\frac{1}{\beta})^t L_*\eta K\beta \\
&\leq (1-\frac{1}{\beta})^r\|\nabla f_i(x^0) - c_i^0\|_* + 18L_*\eta K\beta\sum_{t=0}^{\infty}(1-\frac{1}{\beta})^t \\
&= (1-\frac{1}{\beta})^r\|\nabla f_i(x^0) - c_i^0\|_* + 18L_*\eta K\beta^2.
\end{aligned}
$$

Again, due to initialization the first term is zero, resulting in the following bound:

$$
\|\nabla f_i(x^r) - c_i^r\|_* \leq 18L_*\eta K\beta^2. \tag{4}
$$

This gives an easy estimate for the average control varible:

$$
\|\nabla f(x^r) - c^r\|_* = \|\frac{1}{M}\sum_{i=1}^{M}(\nabla f_i(x^r) - c_i^r)\|_* \overset{(Norm)}{\leq} \frac{1}{M}\sum_{i=1}^{M}\|\nabla f_i(x^r) - c_i^r\|_*.
$$

Now we plug in equation 4 and gain

$$
\|\nabla f(x^r) - c^r\|_* \leq \frac{1}{M}\sum_{i=1}^{M}\|\nabla f_i(x^r) - c_i^r\|_* \leq \frac{1}{M}\sum_{i=1}^{M}18L_*\eta K\beta^2 = 18L_*\eta K\beta^2.
$$

This allows us to continue with the scalar product analysis. We restate the descent inequality for convinience:

$$
\begin{aligned}
-\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle \leq \quad & -\eta(1-\varepsilon_q)\sum_{k=0}^{K-1}\|\nabla f(x^r)\|_* \\
& +2\eta\sum_{k=0}^{K-1}\|\nabla f(x^r) - \nabla f_i(x_i^{r,k}) + c_i^r - c^r\|_* \\
& +\eta(1+\varepsilon_q)\sum_{k=0}^{K-1}\|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_*.
\end{aligned}
$$

All acquired bounds bring us the following:

$$
\begin{aligned}
-\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle \leq \quad & -\eta(1-\varepsilon_q)K\|\nabla f(x^r)\|_* \\
& +2\eta\sum_{k=0}^{K-1}\big(\|\nabla f(x^r) - c^r\|_* + \|\nabla f_i(x^r) - c_i^r\|_* \\
& +\|\nabla f_i(x_i^{r,k}) - \nabla f_i(x^r)\|_*\big) \\
& +\eta(1+\varepsilon_q)\sum_{k=0}^{K-1}\|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* \\
\overset{(4)}{\leq} \quad & -\eta(1-\varepsilon_q)K\|\nabla f(x^r)\|_*
\end{aligned}
$$

$$
\begin{aligned}
& +2\eta \sum_{k=0}^{K-1} \left(36 L_* \eta K \beta^2 + \|\nabla f_i(x_i^{r,k}) - \nabla f_i(x^r)\|_*\right) \\
& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* \\
\overset{(Ass.4.3)}{\leq} \quad & -\eta(1-\varepsilon_q) K \|\nabla f(x^r)\|_* \\
& +2\eta \sum_{k=0}^{K-1} \left(36 L_* \eta K \beta^2 + \|x_i^{r,k} - x^r\|\right) \\
& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} \|\tilde{x}^r - x^r\| \\
\overset{(Lem.B.3,B.5)}{\leq} \quad & -\eta(1-\varepsilon_q) K \|\nabla f(x^r)\|_* \\
& +2\eta \sum_{k=0}^{K-1} \left(36 L_* \eta K \beta^2 + L_* \eta K (1+\varepsilon_q)\right) \\
& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} \eta K \beta (1+\varepsilon_q) \\
\leq \quad & -\eta(1-\varepsilon_q) K \|\nabla f(x^r)\|_* + 76 L_* \eta^2 K^2 \beta^2 + 4\eta^2 K^2 L_* \beta \\
\leq \quad & -\eta(1-\varepsilon_q) K \|\nabla f(x^r)\|_* + 80 L_* \eta^2 K^2 \beta^2,
\end{aligned}
$$

where we trivially estimate $1 + \varepsilon_q \leq 2 \leq 2\beta$.

Finally, we return to the descent equation.

$$
f(\tilde{x}^{r+1}) \quad \leq \quad f(\tilde{x}^r) - \frac{1}{M} \sum_{i=1}^{M} \langle \nabla f(\tilde{x}^r), y_i^{r+1} \rangle + \frac{L_*}{2M} \sum_{i=1}^{M} \|y_i^{r+1}\|^2.
$$

For $\|y_i^{r+1}\|^2$, we apply Corollary B.4.

$$
\frac{L_*}{2M} \sum_{i=1}^{M} \|y_i^{r+1}\|^2 \overset{(Cor.B.4)}{\leq} \frac{L_*}{2M} \sum_{i=1}^{M} (\eta K (1+\varepsilon_q))^2 = 2 L_* \eta^2 K^2.
$$

This plug in the obtained result and take the expectation.

$$
\mathbb{E} f(\tilde{x}^{r+1}) \quad \leq \quad \mathbb{E} f(\tilde{x}^r) - \eta(1-\varepsilon_q) K \mathbb{E} \|\nabla f(x^r)\|_* + 80 L_* \eta^2 K^2 \beta^2 + 2 L_* \eta^2 K^2.
$$

We rearrange the terms and take the average over $r = 0, \ldots R-1$:

$$
\begin{aligned}
\mathbb{E}\left[\frac{1}{R} \sum_{r=0}^{R-1} \eta(1-\varepsilon_q) K \|\nabla f(x^r)\|_*\right] \quad & \leq \quad \frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E}\left[f(\tilde{x}^r) - f(\tilde{x}^{r+1})\right] + 82 L_* \eta^2 K^2 \beta^2 \\
& \leq \quad \frac{\Delta}{R} + 82 L_* \beta^2 \eta^2 K^2,
\end{aligned}
$$

where $f(x^0) - f(x^R) \leq f(x^0) - f(x^*) = \Delta$. Now, we divide the equation by $\eta K (1-\varepsilon_q)$.

$$
\mathbb{E}\left[\frac{1}{R} \sum_{r=0}^{R-1} \|\nabla f(x^r)\|_*\right] \quad \leq \quad \frac{\Delta}{\eta(1-\varepsilon_q) K R} + \frac{34 L_* \beta^2 \eta K}{1-\varepsilon_q},
$$

Now we choose the stepsize $\eta = \Theta\left(\frac{\sqrt{\Delta}}{K\beta\sqrt{R L_*}}\right)$ and acquire:

$$
\mathbb{E}\left[\frac{1}{R} \sum_{r=0}^{R-1} \|\nabla f(x^r)\|_*\right] = \mathcal{O}\left(\frac{\beta\sqrt{\Delta L_*}}{(1-\varepsilon_q)\sqrt{R}}\right).
$$

Lastly, we denote $\varepsilon = \mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_*\right]$ and rearrange the terms to obtain round complexity.

$$R = \mathcal{O}\left(\frac{\beta^2 \Delta L_*}{\varepsilon^2(1-\varepsilon_q)^2}\right).$$

□

B.2 Proof of Theorem 4.9

Remark B.8. *We recall that Lemmas B.1, B.3, B.5, B.6 all are applicable to Algorithm 2, as well as Corollaries B.2, B.4.*

Theorem B.9. *Suppose Assumptions 4.3, 4.4, and 4.8 hold. Then Algorithm 2 to achieve ε-accuracy needs*

$$R = \mathcal{O}\left(\frac{\Delta L_* \beta^2}{\varepsilon^2(1-\varepsilon_q)^2} + \frac{\Delta L_* \rho^2 \sigma^2 \beta}{M\varepsilon^4(1-\varepsilon_q)^4}\right)$$

communication rounds, where $\varepsilon = \mathbb{E}\left[\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_*\right]$, and $\Delta = f(x^0) - f(x^R)$, τ is a unit of time required to send one uncompressed model to the server, ε_q is the error of Newton-Schulz iterations.

We divide the lengthy proof into lemmas for clarity. First one deals with the difference of the gradient and momentum:

Lemma B.10. *Suppose Assumptions 4.3, 4.4, and 4.8 hold. Then during the run of Algorithm 2 the following holds:*

$$\mathbb{E}\|m^{r,K} - \nabla f(x^r)\|_* \leq \rho\sigma\sqrt{\alpha} + 2L_*\eta K + \frac{6L_*\eta K\beta}{1-(1-\alpha)^K},$$

where $m^{r,k} = \frac{1}{M}\sum_{i=1}^M m_i^{r,k}$.

Proof. First, we bound the discrepancy of momentum and gradient throughout the round. We denote $g_i^{r,k} = g_i(x_i^{r,k}, \xi_i^{r,k})$, $g^{r,k} = \frac{1}{M}\sum_{i=1}^M g_i^{r,k}$ and $\nabla f_i^{r,k} = \nabla f_i(x_i^{r,k})$, $\nabla f^{r,k} = \frac{1}{M}\sum_{i=1}^M \nabla f_i^{r,k}$ for the clarity of notation.

$$\begin{aligned}
m^{r,K} - \nabla f(x^r) &= (1-\alpha)m^{r,K-1} + \alpha g^{r,K-1} - \nabla f(x^r) \\
&= (1-\alpha)(m^{r,K-1} - \nabla f(x^r)) + \alpha(g^{r,K-1} - \nabla f^{r,K-1}) \\
&\quad +\alpha(\nabla f^{r,K-1} - \nabla f(x^r)) \\
&= \dots \\
&= (1-\alpha)^K(m^{r,0} - \nabla f(x^r)) + \alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-1-l}(g^{r,l} - \nabla f^{r,l}) \\
&\quad +\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-1-l}(\nabla f^{r,l} - \nabla f(x^r)).
\end{aligned}$$

Next, we continue the equality to the previous round.

$$\begin{aligned}
m^{r,K} - \nabla f(x^r) &= (1-\alpha)^K(m^{r,0} - \nabla f(x^r)) + \alpha\sum_{l=0}^{K-1}(1-\alpha)^l(g^{r,K-1-l} - \nabla f^{r,K-1-l}) \\
&\quad +\alpha\sum_{l=0}^{K-1}(1-\alpha)^l(\nabla f^{r,K-1-l} - \nabla f(x^r)) \\
&= (1-\alpha)^K(m^{r-1,K} - \nabla f(x^{r-1})) + (1-\alpha)^K(\nabla f(x^{r-1} - \nabla f(x^r)) \\
&\quad +\alpha\sum_{l=0}^{K-1}(1-\alpha)^l(g^{r,K-1-l} - \nabla f^{r,K-1-l})
\end{aligned}$$

$$+\alpha \sum_{l=0}^{K-1}(1-\alpha)^l(\nabla f^{r,K-1-l} - \nabla f(x^r)).$$

Now, we unroll the recursion to the $r = 0$:

$$\begin{aligned}
m^{r,K} - \nabla f(x^r) &= (1-\alpha)^K(m^{r-1,K} - \nabla f(x^{r-1})) + (1-\alpha)^K(\nabla f(x^{r-1}) - \nabla f(x^r)) \\
&\quad +\alpha \sum_{l=0}^{K-1}(1-\alpha)^l(g^{r,K-1-l} - \nabla f^{r,K-1-l}) \\
&\quad +\alpha \sum_{l=0}^{K-1}(1-\alpha)^l(\nabla f^{r,K-1-l} - \nabla f(x^r)) \\
&= \dots \\
&= (1-\alpha)^{(r-1)K}(m^{0,K} - \nabla f(x^0)) \\
&\quad +\alpha \sum_{t=1}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t-1)K-l-1}(g^{t,l} - \nabla f^{t,l}) \\
&\quad +\alpha \sum_{t=1}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t-1)K-l-1}(\nabla f^{t,l} - \nabla f(x^t)) \\
&\quad +\sum_{t=0}^{r-1}(1-\alpha)^{K(r-t)}(\nabla f(x^t) - \nabla f(x^{t+1})) \\
&= (1-\alpha)^{rK}(m^{0,0} - \nabla f(x^0)) \\
&\quad +\alpha \sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}(g^{t,l} - \nabla f^{t,l}) \\
&\quad +\alpha \sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}(\nabla f^{t,l} - \nabla f(x^t)) \\
&\quad +\sum_{t=0}^{r-1}(1-\alpha)^{K(r-t)}(\nabla f(x^t) - \nabla f(x^{t+1})).
\end{aligned}$$

We will separately bound the expected norm of the terms. The first one equlas to 0 due to proper initialization. The last two terms are bounded by the Assumption 4.3 and Lemmas B.3,B.6:

$$\begin{aligned}
&\mathbb{E}\Big\|\alpha \sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}(\nabla f^{t,l} - \nabla f(x^t)) \\
&\quad +\sum_{t=0}^{r-1}(1-\alpha)^{K(r-t)}(\nabla f(x^t - \nabla f(x^{t+1}))\Big\|_* \\
\overset{(Norm)}{\leq} &\quad \frac{\alpha}{M}\sum_{i=1}^{M}\sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}\mathbb{E}\|\nabla f_i^{t,l} - \nabla f_i(x^t)\|_* \\
&\quad +\sum_{t=0}^{r-1}(1-\alpha)^{K(r-t)}\|\nabla f_i(x^t) - \nabla f_i(x^{t+1})\|_* \\
\overset{(Ass.4.3)}{\leq} &\quad \frac{\alpha}{M}\sum_{i=1}^{M}\sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}L_*\mathbb{E}\|x_i^{t,l} - x^t\| \\
&\quad +\sum_{t=0}^{r-1}(1-\alpha)^{K(r-t)}L_*\mathbb{E}\|x^t - x^{t+1})\| \\
\overset{(Lem.B.3mB.6)}{\leq} &\quad \frac{\alpha}{M}\sum_{i=1}^{M}\sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}L_*\eta K(1+\varepsilon_q)
\end{aligned}$$

$$
\begin{aligned}
& + \sum_{t=0}^{r-1} (1-\alpha)^{K(r-t)} \cdot 6L_* \eta K \beta \\
\leq \quad & 2\alpha L_* \eta K \sum_{n=0}^{\infty} (1-\alpha)^n \\
& + 6L_* \eta K \beta \sum_{n=0}^{\infty} (1-\alpha)^{Kn} \\
= \quad & 2L_* \eta K + \frac{6L_* \eta K \beta}{1-(1-\alpha)^K}.
\end{aligned}
$$

The second term can be bounded via transition to the Frobenius norm.

$$
\begin{aligned}
& \mathbb{E} \left\| \alpha \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{(r-t)K-l-1} (g^{t,l} - \nabla f^{t,l}) \right\|_* \\
\leq \quad & \rho \mathbb{E} \left\| \alpha \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{(r-t)K-l-1} (g^{t,l} - \nabla f^{t,l}) \right\|_F.
\end{aligned}
$$

Next, we apply Jensen inequality to the expectation:

$$
\begin{aligned}
& \rho \mathbb{E} \left\| \alpha \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{(r-t)K-l-1} (g^{t,l} - \nabla f^{t,l}) \right\|_F \\
\leq \quad & \rho \sqrt{\mathbb{E} \left\| \alpha \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{(r-t)K-l-1} (g^{t,l} - \nabla f^{t,l}) \right\|_F^2}.
\end{aligned}
$$

The Frobenius norm can be presented as a scalar product:

$$
\begin{aligned}
& \mathbb{E} \left\| \alpha \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{(r-t)K-l-1} (g^{t,l} - \nabla f^{t,l}) \right\|_F^2 \\
= \quad & \alpha^2 \sum_{t,t'=0}^{r-1} \sum_{l,l'=0}^{K-1} (1-\alpha)^{(2r-t-t')K-l-l'-1} \mathbb{E} \left\langle g^{t,l} - \nabla f^{t,l}, g^{t',l'} - \nabla f^{t',l'} \right\rangle \\
= \quad & \frac{\alpha^2}{M^2} \sum_{i,i'=1}^{M} \sum_{t,t'=0}^{r-1} \sum_{l,l'=0}^{K-1} (1-\alpha)^{(2r-t-t')K-l-l'-1} \mathbb{E} \left\langle g_i^{t,l} - \nabla f_i^{t,l}, g_{i'}^{t',l'} - \nabla f_{i'}^{t',l'} \right\rangle.
\end{aligned}
$$

We separate the sum in two cases: S_1 denotes all scalar products, where $i = i', t = t', l = l'$, while S_2 denotes the rest of the sum. Since the stochastic gradients are indenendent, all terms in S_2 equal to 0:

$$
\mathbb{E} \left\langle g_i^{t,l} - \nabla f_i^{t,l}, g_{i'}^{t',l'} - \nabla f_{i'}^{t',l'} \right\rangle = \left\langle \mathbb{E}[g_i^{t,l} - \nabla f_i^{t,l}], \mathbb{E}[g_{i'}^{t',l'} - \nabla f_{i'}^{t',l'}] \right\rangle = \langle 0, 0 \rangle = 0.
$$

S_2 can be bounded via Assumption 4.3.

$$
\begin{aligned}
S_2 \quad = \quad & \frac{\alpha^2}{M^2} \sum_{i=1}^{M} \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{2(r-t)K-2l-2} \mathbb{E} \| g_i^{t,l} - \nabla f_i^{t,l} \|_F^2 \\
\leq \quad & \frac{\alpha^2}{M^2} \sum_{i=1}^{M} \sum_{t=0}^{r-1} \sum_{l=0}^{K-1} (1-\alpha)^{2(r-t)K-2l-2} \sigma^2 \\
\leq \quad & \frac{\alpha^2 \sigma^2}{M} \sum_{t=0}^{\infty} \sum_{l=0}^{K-1} (1-\alpha)^{2(r-t)K-2l-2} \\
= \quad & \frac{\alpha \sigma^2}{M}.
\end{aligned}
$$

Thus, returning to the conjugate norm, we obtain the following.

$$\mathbb{E}\Big\|\alpha\sum_{t=0}^{r-1}\sum_{l=0}^{K-1}(1-\alpha)^{(r-t)K-l-1}(g^{t,l}-\nabla f^{t,l})\Big\|_* \leq \rho\sqrt{\frac{\alpha\sigma^2}{M}} = \rho\sigma\sqrt{\frac{\alpha}{M}}.$$

Combining the results yields the Lemma statement. □

Corollary B.11. *The same bound holds for a single client's momentum without the $\frac{1}{\sqrt{M}}$ factor:*

$$\mathbb{E}\|m^{r,K}-\nabla f(x^r)\|_* \leq \rho\sigma\sqrt{\alpha} + 2L_*\eta K + \frac{6L_*\eta K\beta}{1-(1-\alpha)^K}.$$

Lemma B.12. *Suppose Assumptions 4.3, 4.4, and 4.8 hold. Then durin the run of Algorithm 2 the following holds:*

$$\mathbb{E}\|c^r-\nabla f(x^r)\|_* \leq 3\rho\sigma\beta\sqrt{\alpha} + 6L_*\eta K\beta + \frac{18L_*\eta K\beta^2}{\alpha}.$$

Proof. The cornerstone of the theorem is a proper bound for $\|c^r-\nabla f(x^r)\|_*$. We start applying Lemma B.10:

$$\begin{aligned}
\mathbb{E}\|c^r-\nabla f(x^r)\|_* &\overset{(Norm)}{\leq} \mathbb{E}\|c^r-m^{r,0}\|_* + \mathbb{E}\|m^{r,0}-\nabla f(x^r)\|_* &(5)\\
&\leq \mathbb{E}\|c^r-m^{r,0}\|_* + \rho\sigma\sqrt{\frac{\alpha}{M}} + 2L_*\eta K + \frac{6L_*\eta K\beta}{1-(1-\alpha)^K}. &(6)
\end{aligned}$$

In order to estimate $\mathbb{E}\|c^r-m^{r,0}\|_*$ in (5), we separate the clients with (Jen) and plug in the definition of c_i^r.

$$\begin{aligned}
\mathbb{E}\|c_i^r-m_i^{r,0}\|_* &= \mathbb{E}\left\|c_i^{r-1}+C\left(m_i^{r-1,K}-c_i^r\right)-m_i^{r-1,K}\right\|_*\\
&\overset{(Def.2)}{\leq} (1-\frac{1}{\beta})\mathbb{E}\left\|c_i^{r-1}-m_i^{r-1,K}\right\|_*\\
&\overset{(Norm)}{\leq} (1-\frac{1}{\beta})\mathbb{E}\left\|c_i^{r-1}-m_i^{r-1,0}\right\|_* + (1-\frac{1}{\beta})\mathbb{E}\|m_i^{r-1,K}-m_i^{r-1,0}\|_*.
\end{aligned}$$

The difference of the momentum through the round deserves special attention.

$$\begin{aligned}
m_i^{r,K}-m_i^{r,0} &= (1-\alpha)m_i^{r,K-1}+\alpha(g_i^{r,K-1}-\nabla f_i^{r,K-1})+\alpha\nabla f_i^{r,K-1}-m_i^{r,0}\\
&= \dots\\
&= (1-\alpha)^K m_i^{r,0}+\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(g_i^{r,l}-\nabla f_i^{r,l})\\
&\quad +\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}\nabla f_i^{r,l}-m_i^{r,0}\\
&= -(1-(1-\alpha)^K)m_i^{r,0}+\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(g_i^{r,l}-\nabla f_i^{r,l})\\
&\quad +(1-(1-\alpha))\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}\nabla f_i^{r,l}\\
&= (1-(1-\alpha)^K)(\nabla f_i^{r,0}-m_i^{r,0})+\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(g_i^{r,l}-\nabla f_i^{r,l})\\
&\quad +(\nabla f_i^{r,K-1}-\nabla f_i^{r,0})+\sum_{l=0}^{K-2}(1-\alpha)^{K-l-1}(\nabla f_i^{r,l}-\nabla f_i^{r,l+1}).
\end{aligned}$$

The first term is bounded via Corrolary B.11 and Bernoulli inequality $(1-\alpha)^K \geq 1-\alpha K$, which leads to $1-(1-\alpha)^K \leq \min\{\alpha K; 1\}$.

$$\begin{aligned}(1-(1-\alpha)^K)(\nabla f_i^{r,0} - m_i^{r,0}) &\leq (1-(1-\alpha)^K)(\rho\sigma\sqrt{\alpha} + 2L_*\eta K + \frac{6L_*\eta K\beta}{1-(1-\alpha)^K}) \\ &\leq \rho\sigma K\alpha^{3/2} + 2L_*\eta K + 6L_*\eta K\beta.\end{aligned}$$

Similarly to Lemma B.10, we can bound the remaining terms.

$$\begin{aligned}&\mathbb{E}\Bigg\|\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(g_i^{r,l} - \nabla f_i^{r,l}) \\ &+\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(\nabla f_i^{r,l} - \nabla f_i^{r,0}) + \nabla f_i^{r,K-1} - \nabla f_i^{r,0}\Bigg\|_* \\ \overset{(Ass.4.3)}{\leq}\ &\alpha\sqrt{\mathbb{E}\left\|\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}(g_i^{r,l} - \nabla f_i^{r,l})\right\|_F^2} \\ &+\alpha\sum_{l=0}^{K-1}(1-\alpha)^{K-l-1}\|\nabla f_i^{r,l} - \nabla f_i^{r,0}\|_* + \|\nabla f_i^{r,K-1} - \nabla f_i^{r,0}\|_* \\ \overset{(Lem.B.3)}{\leq}\ &\alpha\rho\sigma\sqrt{K} + \alpha\sum_{l=0}^{\infty}(1-\alpha)^l L_*\eta(1+\varepsilon_q) + L_*\eta K(1+\varepsilon_q) \\ \leq\ &\alpha\rho\sigma\sqrt{K} + 4L_*\eta K.\end{aligned}$$

Combining together the bounds, we have

$$\begin{aligned}\mathbb{E}\|m_i^{r,K} - m_i^{r,0}\|_* &\leq \rho\sigma K\alpha^{3/2} + 2L_*\eta K + 6L_*\eta K\beta + \alpha\rho\sigma\sqrt{K} + 4L_*\eta K \\ &\leq \rho\sigma\alpha\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta.\end{aligned}$$

The obtained result gives a recursive equation for the difference of the momentum and control variate:

$$\begin{aligned}\mathbb{E}\|c_i^r - m_i^{r,0}\|_* &\leq (1-\frac{1}{\beta})\mathbb{E}\left\|c_i^{r-1} - m_i^{r-1,0}\right\|_* + \rho\sigma\alpha\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta \\ &\leq \dots \\ &\leq (1-\frac{1}{\beta})^r\mathbb{E}\left\|c_i^0 - m_i^{0,0}\right\|_* \\ &\quad +2\sum_{t=1}^{r}(1-\frac{1}{\beta})^t(\rho\sigma\alpha\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta) \\ &\leq (1-\frac{1}{\beta})^r\mathbb{E}\left\|c_i^0 - m_i^{0,0}\right\|_* \\ &\quad +(\rho\sigma\alpha\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta)\sum_{t=0}^{\infty}(1-\frac{1}{\beta})^t \\ &\leq (1-\frac{1}{\beta})^r\mathbb{E}\left\|c_i^0 - m_i^{0,0}\right\|_* + \rho\sigma\alpha\beta\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta^2.\end{aligned}$$

Then, we note that the first term is zero and plug in the result into (5):

$$\begin{aligned}\mathbb{E}\|c^r - \nabla f(x^r)\|_* &\leq (\rho\sigma\alpha\beta\sqrt{K}(1+\sqrt{\alpha K}) + 12L_*\eta K\beta^2) \\ &\quad +\rho\sigma\sqrt{\frac{\alpha}{M}} + 2L_*\eta K + \frac{6L_*\eta K\beta}{1-(1-\alpha)^K} \\ &\leq \rho\sigma\sqrt{\alpha}\left(\sqrt{\frac{1}{M}} + \beta\sqrt{\alpha K} + \beta\alpha K\right) + 14L_*\eta K\beta^2 + \frac{6L_*\eta K\beta}{\alpha}.\end{aligned}$$

□

Finally, we return to the original theorem.

Theorem B.13. *Suppose Assumptions 4.3, 4.4, and 4.8 hold. Then Algorithm 2 to achieve ε-accuracy needs*

$$R = \mathcal{O}\left(\frac{\beta^2 \Delta L_*}{\varepsilon^2(1-\varepsilon_q)^2} + \frac{\rho^4 \Delta L_* \sigma^2}{\varepsilon^4 (1-\varepsilon_q)^4 M}\right) \text{ communication rounds,}$$

where $\varepsilon = \mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x^r)\|_*\right]$, *and* $\Delta = f(x^0) - f(x^*)$, *and* ε_q *is the error of Newton-Schulz iterations.*

Proof. We start similarly to Theorem 4.7, with smoothness inequality.

$$\begin{aligned} f(\tilde{x}^{r+1}) &\overset{(Ass.4.3)}{\leq} f(\tilde{x}^r) + \langle \nabla f(\tilde{x}^r), \tilde{x}^{r+1} - \tilde{x}^r \rangle + \frac{L_*}{2}\|\tilde{x}^{r+1} - \tilde{x}^r\|^2 \\ &\overset{(Norm)}{\leq} f(\tilde{x}^r) - \sum_{i=1}^M \langle \nabla f(\tilde{x}^r), y_i^{r+1} \rangle + \frac{L_*}{2M}\sum_{i=1}^M \|y_i^{r+1}\|^2. \end{aligned}$$

We pay special attention to the scalar product.

$$\begin{aligned} -\langle \nabla f(\tilde{x}^r), y_i^{r+1} \rangle &= \langle \nabla f(\tilde{x}^r), \sum_{k=0}^{K-1} \eta \,\mathrm{NS}^q(m_i^{r,k} - c_i^r + c^r) \rangle \\ &= \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{NS}^q(m_i^{r,k} - c_i^r + c^r) \rangle \\ &= \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle \\ &\quad + \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), [\mathrm{NS}^q - \mathrm{lmo}](m_i^{r,k} - c_i^r + c^r) \rangle \\ &\overset{(Conj)}{\leq} \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle \\ &\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \left\|[\mathrm{NS}^q - \mathrm{lmo}](m_i^{r,k} - c_i^r + c^r)\right\| \\ &\overset{(B.2)}{\leq} \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \end{aligned}$$

With inexact Newton-Schulz out of the way, we can apply main LMO property equation B.1.

$$\begin{aligned} -\langle \nabla f(\tilde{x}^r), y_i^{r+1} \rangle &\leq \eta \sum_{k=0}^{K-1} \langle \nabla f(\tilde{x}^r), \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \\ &= \eta \sum_{k=0}^{K-1} \langle m_i^{r,k} - c_i^r + c^r, \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle \\ &\quad - \eta \sum_{k=0}^{K-1} \langle -\nabla f(\tilde{x}^r) + m_i^{r,k} - c_i^r + c^r, \mathrm{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle \\ &\quad + \eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \\ &\overset{(Lem.B.1)}{=} -\eta \sum_{k=0}^{K-1} \|m_i^{r,k} - c_i^r + c^r\|_* \end{aligned}$$

$$
\begin{aligned}
& -\eta \sum_{k=0}^{K-1} \langle -\nabla f(\tilde{x}^r) + m_i^{r,k} - c_i^r + c^r, \operatorname{lmo}(m_i^{r,k} - c_i^r + c^r) \rangle \\
& +\eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \\
\overset{(Conj)}{\leq} \quad & -\eta \sum_{k=0}^{K-1} \|m_i^{r,k} - c_i^r + c^r\|_* \\
& +\eta \sum_{k=0}^{K-1} \| - \nabla f(\tilde{x}^r) + m_i^{r,k} - c_i^r + c^r\|_* \cdot \| \operatorname{lmo}(m_i^{r,k} - c_i^r + c^r)\| \\
& +\eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \\
\leq \quad & -\eta \sum_{k=0}^{K-1} \|m_i^{r,k} - c_i^r + c^r\|_* \\
& +\eta \sum_{k=0}^{K-1} \| - \nabla f(\tilde{x}^r) + m_i^{r,k} - c_i^r + c^r\|_* \cdot 1 \\
& +\eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r)\|_* \cdot \varepsilon_q \\
= \quad & -\eta \sum_{k=0}^{K-1} \|m_i^{r,k} - c_i^r + c^r \pm \nabla f(x^r)\|_* \\
& +\eta \sum_{k=0}^{K-1} \| - \nabla f(\tilde{x}^r) + m_i^{r,k} - c_i^r + c^r \pm \nabla f(x^r)\|_* \cdot 1 \\
& +\eta \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) \pm \nabla f(x^r)\|_* \cdot \varepsilon_q \\
\overset{(Norm)}{\leq} \quad & -\eta \sum_{k=0}^{K-1} \|\nabla f(x^r)\|_* (1 - \varepsilon_q) \\
& +2\eta \sum_{k=0}^{K-1} \| - \nabla f(x^r) + m_i^{r,k} - c_i^r + c^r\|_* \\
& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} \|\nabla f(\tilde{x}^r) - \nabla f(x^r)\|_* \\
\overset{(Ass.4.3)}{\leq} \quad & -\eta \sum_{k=0}^{K-1} \|\nabla f(x^r)\|_* (1 - \varepsilon_q) + 2\eta \sum_{k=0}^{K-1} \| - \nabla f(x^r) + m_i^{r,k} - c_i^r + c^r\|_* \\
& +\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} L_* \|\tilde{x}^r - x^r\|.
\end{aligned}
$$

The last term can be easily bounded via Lemma B.5:

$$
\begin{aligned}
\eta(1+\varepsilon_q) \sum_{k=0}^{K-1} L_* \|\tilde{x}^r - x^r\| \quad &= \quad \eta(1+\varepsilon_q) K L_* \left\| \frac{1}{M} \sum_{i=1}^{M} e_i^r \right\| \\
&\overset{(Norm)}{\leq} \quad \eta(1+\varepsilon_q) K L_* \frac{1}{M} \sum_{i=1}^{M} \|e_i^r\|
\end{aligned}
$$

$$\begin{aligned} &\overset{(Lem.B.5)}{\leq} && \eta(1+\varepsilon_q)KL_*\frac{1}{M}\sum_{i=1}^{M}\eta K\beta(1+\varepsilon_q) \\ &= && \eta^2K^2\beta L_*(1+\varepsilon_q)^2. \end{aligned}$$

Then, we can return all the way to the scalar product:

$$\begin{aligned} -\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle \quad &\leq \quad -\eta\sum_{k=0}^{K-1}\|\nabla f(x^r)\|_*(1-\varepsilon_q) \\ &\quad +2\eta\sum_{k=0}^{K-1}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_*+4\eta^2K^2\beta L_*. \end{aligned}$$

Then, we plug in the bound for the scalar product into the first smoothness inequality:

$$\begin{aligned} f(\tilde{x}^{r+1}) \quad &\leq \quad f(\tilde{x}^r)-\frac{1}{M}\sum_{i=1}^{M}\langle\nabla f(\tilde{x}^r), y_i^{r+1}\rangle+\frac{L_*}{2M}\sum_{i=1}^{M}\|y_i^{r+1}\|^2 \\ &\leq \quad f(\tilde{x}^r)+\frac{1}{M}\sum_{i=1}^{M}\Bigg(-\eta\sum_{k=0}^{K-1}\|\nabla f(x^r)\|_*(1-\varepsilon_q)+4L_*\eta^2K^2\beta \\ &\quad +2\eta\sum_{k=0}^{K-1}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_*\Bigg)+\frac{L_*}{2M}\sum_{i=1}^{M}\|y_i^{r+1}\|^2 \\ &\leq \quad f(\tilde{x}^r)-\eta K\|\nabla f(x^r)\|_*(1-\varepsilon_q)+4L_*\eta^2K^2\beta \\ &\quad +\frac{2\eta}{M}\sum_{i=1}^{M}\sum_{k=0}^{K-1}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_*+\frac{L_*}{2M}\sum_{i=1}^{M}\|y_i^{r+1}\|^2. \end{aligned}$$

Further, Corollary B.4 bounds the last term:

$$\frac{L_*}{2M}\sum_{i=1}^{M}\|y_i^{r+1}\|^2\leq\frac{L_*}{2M}\sum_{i=1}^{M}(\eta K(1+\varepsilon_q))^2\leq 2L_*\eta^2K^2.$$

We regroup the terms and take the average over the rounds $r=0\dots R-1$.

$$\begin{aligned} \eta K(1-\varepsilon_q)\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_* \quad &\leq \quad \frac{1}{R}\sum_{r=0}^{R-1}(f(\tilde{x}^r)-f(\tilde{x}^{r+1})) \\ &\quad +4L_*\eta^2K^2\beta+2L_*\eta^2K^2 \\ &\quad +\frac{2\eta}{MR}\sum_{r=0}^{R-1}\sum_{i=1}^{M}\sum_{k=0}^{K-1}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_*. \end{aligned}$$

Next, we define $f(\tilde{x}^0)-f(\tilde{x}^R)\leq f(\tilde{x}^0)-f(x^*)=:\Delta$ and divide the equation by $\eta K(1-\varepsilon_q)$:

$$\begin{aligned} \frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_* \quad &\leq \quad \frac{\Delta}{\eta KR(1-\varepsilon_q)}+\frac{6L_*\eta K\beta}{1-\varepsilon_q} \\ &\quad +\frac{2}{MRK(1-\varepsilon_q)}\sum_{r=0}^{R-1}\sum_{i=1}^{M}\sum_{k=0}^{K-1}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_*. \end{aligned}$$

We take the expectation and utilize the Lemmas B.12,B.10 to bound the last term. We analyse the norm:

$$\begin{aligned} &&& \frac{2}{MRK(1-\varepsilon_q)}\sum_{r=0}^{R-1}\sum_{i=1}^{M}\sum_{k=0}^{K-1}\mathbb{E}\|-\nabla f(x^r)+m_i^{r,k}-c_i^r+c^r\|_* \\ &\overset{(Norm)}{\leq} &&\frac{2}{MRK(1-\varepsilon_q)}\sum_{r=0}^{R-1}\sum_{i=1}^{M}\sum_{k=0}^{K-1}(\mathbb{E}\|m_i^{r,k}-c_i^r\|_*+\mathbb{E}\|c^r-\nabla f(x^r)\|_*) \end{aligned}$$

$$\overset{(Lem.B.12)}{\leq} \quad \frac{2}{1-\varepsilon_q}\left(\rho\sigma\sqrt{\alpha}\left(\sqrt{\frac{1}{M}}+2\beta\sqrt{\alpha K}+2\beta\alpha K\right)+28L_*\eta K\beta^2+\frac{6L_*\eta K\beta}{\alpha}\right).$$

Now, we choose the parameters.

$$\eta=\min\left\{\frac{\sqrt{\Delta}}{K\sqrt{RL_*\beta}},\sqrt{\frac{\Delta\alpha}{K^2\beta^2RL_*}}\right\},\alpha=\sqrt{\frac{\Delta L_*\beta}{RM\rho^2\sigma^2}}.$$

We note that since α decreases over R, in the bracket $\left(\sqrt{\frac{1}{M}}+2\beta\sqrt{\alpha K}+2\beta\alpha K\right)$ the first term is leading. Thus, we obtain the bounds as follows.

$$\begin{aligned}\mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_*\right] &\leq \frac{\Delta}{\eta KR(1-\varepsilon_q)}+\frac{6L_*\eta K\beta}{1-\varepsilon_q}\\ &\quad+\frac{2}{1-\varepsilon_q}\left(\rho\sigma\sqrt{\frac{\alpha}{M}}+28L_*\eta K\beta^2+\frac{6L_*\eta K\beta}{\alpha}\right)\\ &\leq \frac{\Delta}{\eta KR(1-\varepsilon_q)}+\frac{62L_*\eta K\beta^2}{1-\varepsilon_q}\\ &\quad+\frac{2\rho\sigma}{1-\varepsilon_q}\sqrt{\frac{\alpha}{M}}+\frac{6L_*\eta K\beta}{\alpha(1-\varepsilon_q)}.\end{aligned}$$

To analyse the round complexity, we plug in the minima:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x^r)\|_*\right] &\leq \frac{\Delta}{\eta KR(1-\varepsilon_q)}+\frac{62L_*\eta K\beta^2}{1-\varepsilon_q}\\ &\quad+\frac{2\rho\sigma}{1-\varepsilon_q}\sqrt{\frac{\alpha}{M}}+\frac{6L_*\eta K\beta}{\alpha(1-\varepsilon_q)}\\ &\leq \frac{\Delta}{KR(1-\varepsilon_q)}\left(\frac{K\sqrt{RL_*\beta^2}}{\sqrt{\Delta}}+\sqrt{\frac{K^2\beta^2RL_*}{\Delta\alpha}}\right)\\ &\quad+\frac{62L_*K\beta^2}{1-\varepsilon_q}\frac{\sqrt{\Delta}}{K\sqrt{RL_*\beta^2}}\\ &\quad+\frac{2\rho\sigma}{1-\varepsilon_q}\sqrt{\frac{\alpha}{M}}+\frac{6L_*K\beta}{\alpha(1-\varepsilon_q)}\cdot\sqrt{\frac{\Delta\alpha}{K^2\beta^2RL_*}}\\ &= \frac{63\beta}{(1-\varepsilon_q)}\sqrt{\frac{\Delta L_*}{R}}+\frac{2\rho\sigma}{1-\varepsilon_q}\sqrt{\frac{\alpha}{M}}+\frac{6\sqrt{\Delta L_*\beta}}{\sqrt{R\alpha}(1-\varepsilon_q)}\\ &= \frac{63\beta}{(1-\varepsilon_q)}\sqrt{\frac{\Delta L_*}{R}}+\frac{8}{1-\varepsilon_q}\left(\frac{\rho^2\sigma^2\beta}{RM}\right)^{1/4}.\end{aligned}$$

Rearranging the terms, we arrive at the Theorem statement.

$$R \quad = \quad \mathcal{O}\left(\frac{\Delta L_*\beta^2}{\varepsilon^2(1-\varepsilon_q)^2}+\frac{\Delta L_*\rho^2\sigma^2\beta}{M\varepsilon^4(1-\varepsilon_q)^4}\right).$$

□