On Communication-Efficient Training of Ensembles in Federated Learning

Valery Parfenov[1,2,3]*, **Mikhail Aleksandrov**[1,2]
Daniil Medyakov[4,1,2], **Dmitry Bylinkin**[4,1,2], **Aleksandr Beznosikov**[4,1,2,5]

[1]Basic Research of Artificial Intelligence Laboratory (BRAIn Lab)
[2]Moscow Independent Research Institute of Artificial Intelligence
[3]University of Tyumen [4]Federated Learning Problems Laboratory
[5]Innopolis University

Abstract

In recent years, ensemble ideas have found parallels in federated learning (FL). Specific ensemble learning formulations under the federated setup involve information exchange between clients during the training process, making the classic FL communication bottleneck a critical issue. Importantly, the resulting communication patterns differ from those studied in traditional communication-efficient FL methods, placing ensembling outside their analytical scope. In this work, we close this gap by extending classic communication-efficient techniques to the ensembling setting and introducing a new mechanism specifically tailored to it. We instantiate these ideas in four algorithms and provide convergence guarantees for each under the smooth non-convex setup. We further support our theoretical findings with experimental results on image classification tasks.

1 Introduction

Ensemble learning refers to the paradigm of combining multiple predictive models to produce a single, often more robust estimator [Dietterich, 2000, Hansen and Salamon, 1990, Zhou, 2021]. Over the past decades, ensembles have become part of the toolkit for achieving state-of-the-art results across a wide range of machine learning tasks [Zheng et al., 2018, Cao et al., 2020, Tanveer et al., 2023]. The core intuition behind this approach lies in leveraging the collective intelligence of multiple models: by aggregating their predictions, ensembles tend to outperform individual learners, particularly regarding generalization and robustness.

The ensembling paradigm offers several universal advantages. Firstly, we can evaluate it from the perspectives of model dimensionality/sample size. When the training set size is insufficient compared to classifier complexity, multiple different learners can achieve similar accuracy yet vary considerably in their generalization behavior. Averaging such predictions effectively reduces the risk of selecting a poorly generalizing classifier [Dietterich, 2000]. In essence, the ensemble behaves like a form of model regularization by pooling independent sources of stochasticity. Secondly, Dietterich notes that the weighted combination of models increases representational flexibility, enabling the ensemble to approximate target functions more accurately than its individual members. This observation, in turn, leads to a modification of the widely recognized bias–variance decomposition [Geman et al., 1992] in the context of ensembles: a third component of *diversity* is introduced, opening new opportunities for trade-offs that enable more effective problem solving [Hansen and Salamon, 1990, Wood et al., 2023].

*Corresponding author: parfenov.vr@gmail.com

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In recent years, ensemble ideas have found a parallel [Jeong et al., 2018, Li and Wang, 2019, Lin et al., 2020] in federated learning (FL) [Konečnỳ et al., 2016, McMahan et al., 2017, Smith et al., 2017, Li et al., 2020, Kairouz et al., 2021], where a central server coordinates training across a set of clients that each store local data and share model updates rather than raw samples. The FL paradigm naturally induces heterogeneity [Qu et al., 2022, Ye et al., 2023]: clients often possess data that follow distinct distributions, which leads to divergence among locally trained models. While such heterogeneity poses a significant challenge for conventional FL algorithms, it can be a desirable property from the ensemble perspective, as it increases the diversity among local models, thereby enhancing joint system performance [Wood et al., 2023, Breiman, 1996]. Furthermore, federated systems frequently exhibit heterogeneity not only in data distributions but also in computational capabilities. Some clients may have constrained resources, while others can afford to train larger or deeper models [Pfeiffer et al., 2023]. Ensemble-based approaches naturally accommodate this client-side network variability, as individual members do not need to share identical architectures Dudek [2016], allowing flexible adaptation to varying client capacities. Moreover, even when identical model classes are used across all clients, the resulting individual learners, trained on distinct local data subsets, serve as naturally personalized ensemble components, thereby enabling improved adaptation to client-specific distributions [M Ghari and Shen, 2024].

From the formal point of view, ensemble training can be introduced as $\min_x F(x)$, where $x = (x_1^\top, x_2^\top, \ldots, x_M^\top)^\top$ and x_m represent the weights of the m-th ensemble member, F denotes the loss function. Currently, the primary approach is independent training. In this setting, $F(x)$ decomposes into the sum $F(x) = F_{local}(x) = \sum_{m=1}^{M} F_{local}^m(x_m) = \sum_{m=1}^{M} \frac{1}{N_m} \sum_{i=1}^{N_m} F_i^m(x_m)$, where $m \in \{1, \ldots, M\}$ indexes clients (and their models), while $i \in \{1, \ldots, N_m\}$ indexes observations in client m's local dataset. Thus $F_i^m(x_m)$ is the loss of model m on its ith local observation. We recall that the ensembling technique relies on the diversity effect, while full ensembling at the inference stage poses challenges in a federated setting. By limiting the bias/variance/diversity trade-off [Hansen and Salamon, 1990, Wood et al., 2023], this increases the importance of local model quality, which is also a crucial consideration in federated learning formulations [Tan et al., 2022]. At the same time, $F_{\text{local}}^m(x_m)$ does not depend on x_n for $n \neq m$. Hence, an independent formulation that excludes interactions between clients allows individual nodes to overfit to their local data.

An alternative is joint training [Lee et al., 2015, Dutt et al., 2020] on N observations shared by all clients. For shared observation $i \in \{1, \ldots, N\}$, let $\hat{f}_i(x_m) \in \mathbb{R}^a$ denote model m's logit vector, $\overline{f}_i(x) = M^{-1} \sum_{m=1}^{M} \hat{f}_i(x_m)$ their mean, and $\mathcal{L} : \mathbb{R}^a \to \mathbb{R}$ the loss function applied to it. The joint objective in this setup is $F(x) = F_{joint}(x) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m)\right)$. It makes the training procedure consistent with the inference stage. In essence, logit averaging acts as a form of regularization. However, this approach does not correspond to the federated setting, where devices possess their own local datasets.

In this work, we consider a flexible formulation that combines the strengths of both aforementioned techniques. Specifically, we assume that (i) each client possesses its own local dataset, ensuring learner diversity and aligning with the federated setting. Additionally, (ii) all clients share a set of common observations, enabling the use of regularization through a joint component. Hence, the objective function is defined as:

$$F(x) = \lambda F_{joint}(x) + F_{local}(x) = \frac{\lambda}{N} \sum_{i=1}^{N} \mathcal{L}\left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m)\right) + \sum_{m=1}^{M} \frac{1}{N_m} \sum_{i=1}^{N_m} \mathcal{L}\left(f_i(x_m)\right), \quad (1)$$

where F_{joint} is computed over the shared observations by averaging the logits of ensemble members, and F_{local} involves local samples. The parameter λ controls the balance between the *individual* and *joint* components.

Shared dataset assumption. The assumption of an additional shared dataset is established in federated learning, both in settings that communicate logits [Chen and Chao, 2020, Lin et al., 2020, Tang et al., 2022] and gradients [Xie et al., 2019, Park et al., 2021]. We note that this assumption may seem restrictive. However, in practice it can be instantiated via public benchmarks, which exist for many tasks. Synthetic data can further relax it: federated methods have constructed privacy-preserving generators on the server to approximate clients' data distributions [Zhang et al., 2022, Wang et al., 2023, Zhang et al., 2024]. As shown by Kotelnikov et al. [2023], synthetic data can be sufficiently anonymized (Table 6) and effective for machine learning tasks (Table 4).

Incorporating F_{joint} into formulation (1) necessitates aggregating information from ensemble participants during training. In the federated setting, this requirement – referred to as the *communication bottleneck* [McMahan et al., 2017] – can substantially hinder the learning process. Therefore, the performance of algorithms under the federated setup depends on their communication efficiency. To address this issue, the research community has developed a wide range of approaches [Seide et al., 2014, Alistarh et al., 2017, Luo et al., 2025] aimed at mitigating the overhead caused by gradient transmission in standard FL formulations. However, in our setup, logits rather than gradients are transmitted, and communication efficiency in this regime remains largely unexplored. Our work aims to fill this gap. The same question is relevant to related settings, such as federated ensemble distillation [Lin et al., 2020, Li and Wang, 2019].

2 Brief Related Work

A detailed overview of related literature is provided in Appendix B.

Variance reduction. Variance reduction (VR) methods address the non-vanishing variance of stochastic gradients by leveraging past information. Two main paradigms are widely used. The first maintains gradient estimates across samples, as in `SAG` [Roux et al., 2012] and `SAGA` [Defazio et al., 2014], yielding improved convergence at the cost of additional memory. The second periodically computes reference gradients to construct low-variance estimators, as in `SVRG` [Johnson and Zhang, 2013], `SARAH` [Nguyen et al., 2017], and `PAGE` [Li et al., 2021].

Communication efficiency in federated learning. Communication-efficient FL primarily relies on *compression* and *local steps* techniques. Compression methods reduce communication cost via quantization [Alistarh et al., 2017] or sparsification [Beznosikov et al., 2023]. More advanced error feedback techniques [Richtárik et al., 2021] focus on stability enhancement. Local-step methods, such as `LocalSGD` [Stich, 2018] and `SCAFFOLD` [Karimireddy et al., 2020], reduce communication frequency by performing multiple client-side updates.

3 Classic Techniques in the New Learning Setup

3.1 Compression preliminaries

First, we highlight a key conceptual difference in the communication pattern of the considered setup. In particular, for each client's step of the gradient method, the derivatives of $F_{\text{joint}}(x)$ and $F_{\text{local}}(x)$ are required. From formulation (1), we note that only a single term in the local part depends on x_m, which implies that the computation of $\nabla_{x_m} F_{\text{local}}(x)$ is performed entirely locally. In contrast, introducing $\overline{f}_i(x) = \frac{1}{M}\sum_{m=1}^{M} \hat{f}_i(x_m)$, we derive

$$\nabla_{x_n} F_{\text{joint}}(x) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} \nabla_{x_n} \mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M} \hat{f}_i(x_m)\right) = \frac{1}{NM}\sum_{i=1}^{N} \frac{\partial \mathcal{L}}{\partial \overline{f}}\Big|_{\overline{f}_i = \overline{f}_i(x)} \cdot \frac{\partial \overline{f}_i(x_n)}{\partial x_n}. \quad (2)$$

While the second factor of each term in (2) is computed locally, the derivative $\frac{\partial \mathcal{L}}{\partial \overline{f}}$ of the first factor is evaluated at $\overline{f}_i(x) = \frac{1}{M}\sum_{m=1}^{M} \hat{f}_i(x_m)$, which necessitates the aggregation of $f_i(x_m)$. Consequently, in the considered setting, it is the logits of each sample from the joint dataset $f_i(x_m)$ that are communicated. This stands in contrast to the standard FL paradigm, in which clients exchange gradient estimates or model states. In this light, transferring compression techniques to the considered federated ensembling setup is nontrivial and requires novel modifications in both the algorithmic design and its theoretical analysis.

For further convenience, we introduce the notation $\nabla \mathcal{L}(a, y) = \frac{\partial \mathcal{L}}{\partial \overline{f}}\Big|_{\overline{f}=a} \frac{\partial \overline{f}}{\partial x}\Big|_{x=y}$.

Thus, the gradient $\nabla \mathcal{L}(\overline{f}(x))$ can be rewritten as $\nabla \mathcal{L}(\overline{f}(x)) = \nabla \mathcal{L}(\overline{f}(x), x)$, and the compression of the communicated information in the considered setup can essentially be viewed as a perturbation in the first argument of $\nabla \mathcal{L}(\overline{f}(x), x)$. For compression in this work, we choose contractive operators.

Definition 3.1. We say that a map $\mathcal{Q} : \mathbb{R}^d \to \mathbb{R}^d$ is a *contractive compression operator*, or simply a *contractive compressor*, if there exists a constant $0 < \alpha \leq 1$ such that the following holds:

$$\mathbb{E}[\|Q(x) - x\|^2] \leq (1 - \alpha)\|x\|^2 \text{ for all } x \in \mathbb{R}^d.$$

This choice is motivated by the widespread practical use of this specific compression technique, driven by its demonstrated empirical success [Beznosikov et al., 2023].

3.2 Algorithm

We introduce **J**oint **EF21** - `JEF21` (Algorithm 1). It inherently incorporates a local steps technique, reflected in Lines 2 and 12. Specifically, with probability $1 - p$, in Lines 14 and 15, no new computations or communications are performed. In contrast, Lines 3-11, executed with probability p, implement the `EF21` [Richtárik et al., 2021] scheme. Each client maintains a local state $h_{i,m}^k$, while the server holds h_i^k. Updates of the form $h_{i,m}^k = h_{i,m}^{k-1} + \Delta_{i,m}^k$ allow the transmission of only compressed increments $\Delta_{i,m}^k = \mathcal{Q}\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right)$ to preserve consistency between the local and global states $h_i^k = \frac{1}{M}\sum_{m=1}^M h_{i,m}^k$. For `JEF21` and further algorithms, we use initialization $h_{i,m}^0 = \hat{f}_i(x_m^0)$, $\hat{g}_m^0 = \frac{\lambda}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(\overline{f}_i(x^0))$. Finally, in Lines 17 and 18, the gradient is computed on the local dataset, and the model parameters are updated, respectively.

Algorithm 1 JEF21

1: **For** $k = 1, 2, \ldots, K$ **do:**
2: **With probability p:**
3: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
4: $\Delta_{i,m}^k = \mathcal{Q}\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right)$
5: $h_{i,m}^k = h_{i,m}^{k-1} + \Delta_{i,m}^k$
6: Send $\Delta_{i,m}^k$ to the server
7: **For** $i = 1, 2, \ldots, N$ **server do:**
8: $h_i^k = h_i^{k-1} + \frac{1}{M}\sum_{m=1}^M \Delta_{i,m}^k$
9: Broadcast h_i^k to nodes
10: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
11: $\hat{g}_m^k = \lambda\frac{1}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(h_i^k, x_m^k)$
12: **With probability** $1 - \mathbf{p}$**:**
13: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
14: $\hat{g}_m^k = \hat{g}_m^{k-1}$
15: $h_{i,m}^k = h_{i,m}^{k-1}$
16: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
17: $g_m^k = \nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
18: $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

3.3 Theoretical analysis

We conduct an analysis in a non-convex setting. The classic Assumption 3.1 of objective L-smoothness, in the context of a federated ensembling setup, needs to be complemented with Assumptions 3.2-3.4, which are intuitively similar in spirit and play an analogous role in the analysis.

Assumption 3.1. *Let F be L-smooth, i.e., $\|\nabla F(x_1) - \nabla F(x_2)\| \leq L\|x_1 - x_2\|$ holds for all $x_1, x_2 \in \mathbb{R}^{dM}$.*

Assumption 3.2. *Let $\hat{f}_i$ be $\tilde{L}_{\hat{f}}$-Lipschitz, i.e., $\|\hat{f}_i(x_1) - \hat{f}_i(x_2)\| \leq \tilde{L}_{\hat{f}}\|x_1 - x_2\|$ holds for all $x_1, x_2 \in \mathbb{R}^d$ and $i = 1, 2, \ldots, N$.*

Assumption 3.3. *Let $\mathcal{L}$ be $L_{\mathcal{L}}$-smooth, i.e., $\|\nabla\mathcal{L}(a_1) - \nabla\mathcal{L}(a_2)\| \leq L_{\mathcal{L}}\|a_1 - a_2\|$ holds for all $a_1, a_2 \in \mathbb{R}^a$.*

Assumption 3.4. *For every shared observation $i \in \{1, \ldots, N\}$, let the composition $x \mapsto \mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M \hat{f}_i(x_m)\right)$ be L_{joint}-smooth, i.e., $\|\nabla\mathcal{L}(\overline{f}_i(x_1)) - \nabla\mathcal{L}(\overline{f}_i(x_2))\| \leq L_{joint}\|x_1 - x_2\|$ holds for all $x_1, x_2 \in \mathbb{R}^{dM}$ and every $i \in \{1, \ldots, N\}$.*

Assumption 3.5. *Let exist finite F^* such that $F^* \leq F(x)$, for all $x \in \mathbb{R}^{dM}$.*

As previously noted, the perturbation of the logits induced by compression corresponds to a perturbation in the first argument of $\nabla\mathcal{L}(\cdot, \cdot)$. To establish theoretical guarantees, it is necessary to control the associated error

$$\|\nabla\mathcal{L}(a, y) - \nabla\mathcal{L}(b, y)\| = \left\|\frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=a}\frac{\partial\overline{f}}{\partial x}\Big|_{x=y} - \frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=b}\frac{\partial\overline{f}}{\partial x}\Big|_{x=y}\right\| \leq \left\|\frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=a} - \frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=b}\right\| \cdot \left\|\frac{\partial\overline{f}}{\partial x}\Big|_{x=y}\right\|.$$

From this representation, one can note that the required bound can be obtained by assuming $L_{\mathcal{L}}$ smoothness of $\mathcal{L}$ together with a bounded gradient norm $\frac{\partial\overline{f}}{\partial x}$, which follows from the Lipschitz continuity of f. This reasoning highlights the strong coupling between the compression analysis of

logits and Assumptions 3.2, 3.3, and is formalized in Lemma D.1. It guarantees $\frac{\tilde{L}_{\hat{f}} L_{\mathcal{L}}}{\sqrt{M}}$-Lipschitz continuity of $\nabla \mathcal{L}(a, y)$ with respect to its first argument.

We provide a theoretical analysis of the generalized Algorithm 5 in Theorems D.1-D.2, where all effects induced by `EF21` are fully isolated by setting $q = 1$. However, for clarity of exposition in the main text, we follow a more incremental approach and present a high-level overview below. To initiate the analysis of the iterations, we rely on the classic descent Lemma D.2. This immediately reveals that the key ingredient is a relation on $\left\|\hat{g}^{k+1} - \nabla_x \mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|$, i.e., the oracle distortion bound. This bound is derived in Theorem D.1 ($q = 1$) in inequalities (6)-(8). In more detail, in (6), under the expectation, the probabilistic approach to the local steps naturally separates the analysis into cases with and without local states $h_{i,m}^k$ updates. In the former case (7), the Cauchy-Schwarz inequality (CS) and L_{joint}-smoothness of $F_{joint}(x)$ (Assumption 3.4) yield a recursive term. The latter case (6) is founded on Lemma D.1 and bounds the oracle distortion in terms of $\sum_{i=1}^N \sum_{m=1}^M \left\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\right\|^2$. Subsequently, in Theorem D.2, we derive recursive relations for the perturbations of the logits $\left\|h_{i,m}^k - \hat{f}_i(x_m^k)\right\|$. This is achieved by explicitly leveraging their update rules together with Definition 3.1 of a contractive compressor. By combining the resulting estimates with the oracle distortion bound in Theorem D.1 and descent Lemma D.2, we obtain the convergence guarantees presented in Corollary 3.1 below.

Corollary 3.1. *Let Assumptions 3.1-3.5 hold and* $\gamma = \min\left\{\frac{1}{4L}; \frac{p}{2\sqrt{6}\lambda L_{joint}}; \frac{Mp\alpha_1}{30\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}\right\}$. *Then,* $\mathcal{O}\left(\frac{1}{\varepsilon} \cdot \left[L + \frac{\lambda L_{joint}}{p} + \frac{\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}{Mp\alpha_1}\right]\right)$ *iterations and* $\mathcal{O}\left(\frac{1}{\varepsilon} \cdot N\left[p\alpha_1 L + \alpha_1 \lambda L_{joint} + \frac{\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}{M}\right]\right)$ *bits communicated from nodes to the server are required for Algorithm 1 to reach* ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$, *where* $\overline{x}^K \sim U\{x^0, x^1, \ldots, x^{K-1}\}$.

Discussion. The bound decomposes into three terms with distinct origins. The first term follows directly from the descent lemma and depends only on the smoothness constant L of the full objective. The second term arises from the analysis of local steps and is therefore governed by the L_{joint} (as local steps performed w.r.t. joint part of the objective) and ensembling probability p. The third term is induced by compression and depends on M, p, and α_1. It additionally scales with $L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2$, reflecting how the effect of logit perturbations is controlled. In addition, as the weight λ of the joint loss tends to zero, choosing $p \sim \lambda$ keeps the iteration amount in Corollary 3.1 bounded while its per-client uplink vanishes. This reflects the diminishing need to communicate for a vanishing joint objective.

It is worth noting that contractive compression and local-step techniques typically do not improve communication complexity [Richtárik et al., 2021, Beznosikov et al., 2023, Luo et al., 2025]. A distinctive feature of the derived guarantees is the cost additionally scales with the joint dataset size N, since logits are transmitted per sample, yielding an average cost of $p\alpha_1 Na$ bits per iteration, where a denotes the number of classes.

4 Novel Perspective on Communication Efficiency

4.1 Motivation

As follows from the discussion at the end of the previous section, in the new setup, the finite-sum structure (1) has a direct impact on communication efficiency. In contrast to classic federated learning, where a single gradient estimator or model parameters are communicated, logits are transmitted for each sample in the joint dataset. This induces a clear analogy with stochastic optimization and enables the direct use of its techniques to reduce communication costs.

For instance, when employing an `SGD` oracle $\nabla_\xi F_{joint}(x) = \frac{\lambda}{|I|} \sum_{i \in I} \mathcal{L}\left(\frac{1}{M} \sum_{m=1}^M \hat{f}_i(x_m)\right)$, logits need to be communicated only for $i \in I$, i.e. the sampled observations. As in standard stochastic optimization, this also eliminates the need to perform model evaluation and backpropagation on the remaining samples. Consequently, in the considered setup, stochasticity and communication efficiency can be naturally aligned: introducing a single shared source of randomness simultaneously

reduces both the number of oracle calls and communicated bits. Further we refer to this mechanism of communication efficiency as *stochasticity-aligned compression*.

4.2 JPAGE: Stochasticity-Aligned Compression Method

Similarly to naive compression, SGD oracle suffers from non-vanishing variance. A standard remedy is the use of variance reduction techniques, which are also applicable in the federated ensembling setup. We focus on the PAGE [Li et al., 2021] method. Its probabilistic structure allows us to incorporate and analyze a variant with local steps, presented in **J**oint **PAGE** - JPAGE (Algorithm 2).

The design of JPAGE follows the same principles as JEF21. Particularly, Lines 15-17 correspond to updates performed without computing or communicating the joint component, while in Lines 2-11 the EF21 scheme is replaced by a PAGE estimator (Line 14). It employs a probabilistic mechanism: with probability q, a full gradient is computed, whereas in the remaining case, the previous estimate is refined using only a subset of observations, reducing both the number of oracle evaluations and aggregated logits.

Algorithm 2 JPAGE

1: **For** $k = 1, 2, \ldots, K$ **do:**
2: **With probability** $\mathbf{p}$**:**
3: **With probability** $\mathbf{q}$**:**
4: (1) $I_k = I$
5: **With probability** $\mathbf{1-q}$**:**
6: (2) $I_k \subset I$, $|I_k| = b$
7: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
8: Compute $\hat{f}_i(x_m^k)$ and, in case (2), $\hat{f}_i(x_m^{k-1})$ for $i \in I_k$ and send it to the server
9: **For** $i \in I_k$ **server do:**
10: $\overline{f}_i(x^k) = \frac{1}{M}\sum_{m=1}^M \hat{f}_i(x_m^k)$
11: (2) $\overline{f}_i(x^{k-1}) = \frac{1}{M}\sum_{m=1}^M \hat{f}_i(x_m^{k-1})$
12: Broadcast logits to nodes
13: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
14: $\begin{bmatrix} (1)\hat{g}_m^k = \lambda\frac{1}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(\overline{f}_i(x^k)) \\ (2)\hat{g}_m^k = \hat{g}_m^{k-1} + \frac{\lambda}{b}\sum_{i\in I_k}\nabla_{x_m}\Big(\mathcal{L}(\overline{f}_i(x^k)) \\ -\mathcal{L}(\overline{f}_i(x^{k-1}))\Big) \end{bmatrix}$
15: **With probability** $\mathbf{1-p}$**:**
16: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
17: $\hat{g}_m^k = \hat{g}_m^{k-1}$
18: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
19: $g_m^k = \nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
20: $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

The analysis of JPAGE is provided in Appendix D.4. It largely follows the analysis of JEF21: the probabilistic formulation of local steps allows us to decompose the contributions of updates with and without communication under expectation. The latter contribution is analyzed in the same way as in JEF21 using the Cauchy-Schwarz (CS) inequality, while the former is controlled via the classic PAGE [Li et al., 2021] oracle distortion bound. Here, we report only the final convergence guarantees in Corollary 4.1.

Corollary 4.1. *Let Assumptions 3.1, 3.4 and 3.5 hold. Then Algorithm 2 with* $\gamma = \min\left\{\frac{1}{2L}; \frac{pq}{4\lambda L_{joint}}\right\}$ *requires* $\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{pq}\right]\right)$ *iterations and* $\mathcal{O}\left(\frac{1}{\varepsilon}\cdot p\left(qN + (1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq}\right]\right)$ *joint objective oracle calls and bits communicated from nodes to the server to reach* ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$*, where* $\overline{x}^K \sim U\{x^0, x^1, \ldots, x^{K-1}\}$.

Discussion. The communication pattern follows a three-regime mixture: no communication with probability $1-p$, full evaluation with probability pq, and mini-batch evaluation with probability $p(1-q)$. This induces a multiplicative factor $p(qN+(1-q)b)$ in both oracle and communication complexity, aligning stochastic gradient computation with communication reduction. Finally, incorporating local steps requires the use of the Cauchy-Schwarz inequality, which prevents recovering the classic mini-batch speedup [Li et al., 2021] for $b > 1$. Appendix E restores this mini-batch gain by changing the correction's reference point.

4.3 CJSAGA: Bridging Stochasticity-Aligned and classic Compression

In federated optimization, the communication bottleneck may pose a more severe limitation on algorithmic performance than computational constraints. This naturally raises the question: *Can additional compression be incorporated on top of the stochasticity-aligned one*?

$$\hat{g}_m^k = \frac{\lambda}{N}\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(\overline{f}_i(x^{k-1}),x^k) + \frac{\lambda}{|I_k|}\sum_{i\in I_k}\nabla_{x_m}\left(\mathcal{L}(\overline{f}_i(x^k),x^k) - \mathcal{L}(\overline{f}_i(x^{k-1}),x^k)\right). \quad (3)$$

To answer this, we focus on advanced techniques that combine EF21 with variance reduction approaches. We first observe that the EF21 scheme naturally maintains local states. A similar local memory mechanism forms the backbone of SAG-type variance reduction methods, which motivates us to begin with this class of algorithms. Moreover, in the analysis of the compressed communications, we rely on Lemma D.1, where the coincidence of the second arguments of the function $\nabla\mathcal{L}(\cdot,\cdot)$ plays a crucial role. This observation further supports the use of a SAG-type oracle of the form (3).

We then introduce Algorithm 3 **C**ompressed **J**oint **SAGA** - CJSAGA. Lines 2 and 14-17 implement the local steps mechanism, which remains unchanged from the previous algorithms. Lines 3-11 describe the error feedback scheme that enables communication-efficient updates of local and global states. The corresponding oracle is provided in Line 13. Due to this choice, only the logits associated with indexes $i \in I_k$ need to be transmitted. In contrast to (3), the oracle employed here operates on the states h_i^k rather than the ground-truth logits $\hat{f}_i(x^k)$.

Algorithm 3 CJSAGA

1: **For** $k = 1, 2, \ldots, K$ **do:**
2: **With probability p:**
3: Sample $I_k \subset I$, $|I_k| = b$
4: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
5: $\Delta_{i,m}^k = \mathcal{Q}\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right),\ i \in I_k$
6: $h_{i,m}^k = h_{i,m}^{k-1} + \Delta_{i,m}^k,\ i \in I_k$
7: $h_{i,m}^k = h_{i,m}^{k-1},\ i \in I \setminus I_k$
8: Send $\Delta_{i,m}^k$ to the server
9: **For** $i \in I_k$ **server do:**
10: $h_i^k = h_i^{k-1} + \frac{1}{M}\sum_{m=1}^{M}\Delta_{i,m}^k$
11: Broadcast h_i^k to nodes
12: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
13: $\hat{g}_m^k = \frac{\lambda}{N}\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(h_i^{k-1},x^k) + \frac{\lambda}{b}\sum_{i\in I_k}\nabla_{x_m}\left(\mathcal{L}(h_i^k,x^k) - \mathcal{L}(h_i^{k-1},x^k)\right)$
14: **With probability 1 − p:**
15: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
16: $\hat{g}_m^k = \hat{g}_m^{k-1}$
17: $h_{i,m}^k = h_{i,m}^{k-1}$
18: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
19: $g_m^k = \nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
20: $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

The analysis of CJSAGA follows the same general structure as that of JEF21. The only substantial difference lies in the treatment of the oracle distortion bound, presented in Theorem D.1 for the case $q = 0$ (inequalities (9)-(15)). In this analysis, we introduce a virtual sequence $\tilde{h}_{i,m}^k = h_{i,m}^{k-1} + \mathcal{Q}_2\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right)$ and exploit the key fact that $\frac{\lambda}{b}\sum_{i\in I_{k+1}}\left(\nabla\mathcal{L}\left(h_i^k,x^{k+1}\right) - \nabla\mathcal{L}\left(h_i^{k+1},x^{k+1}\right)\right)$ constitutes an unbiased estimator of $\frac{\lambda}{N}\sum_{i=1}^{N}\left(\nabla\mathcal{L}\left(h_i^k,x^{k+1}\right) - \nabla\mathcal{L}\left(\tilde{h}_i^{k+1},x^{k+1}\right)\right)$. Applying Lemma D.1 together with the assumptions and definitions of $h_i^k, \tilde{h}_i^k, \mathcal{Q}_2$ yields a recursive upper bound on the oracle distortion, with additional terms of the form $\left\|\hat{f}_i(x_m^k) - h_{i,m}^k\right\|$, $\left\|\hat{f}_i(x_m^k) - \tilde{h}_{i,m}^k\right\|$ and $\|x^{k+1} - x^k\|$. The remainder of the proof is fully analogous to the derivations for JEF21. We state the final guarantees in Corollary 4.2.

Corollary 4.2. *Let Assumptions 3.1- 3.5 hold and* $\gamma = \min\left\{\frac{1}{4L}; \frac{p}{2\sqrt{6}\lambda L_{joint}}; \frac{Mp\frac{\alpha_2 b}{N}}{30\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}$. *Then,* $\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{p} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp\frac{\alpha_2 b}{N}}\right]\right)$ *iterations,* $\mathcal{O}\left(\frac{1}{\varepsilon}\cdot pN\left[L + \frac{\lambda L_{joint}}{p} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp\frac{\alpha_2 b}{N}}\right]\right)$ *joint objective oracle calls, and* $\mathcal{O}\left(\frac{1}{\varepsilon}\cdot N\left[p\frac{\alpha_2 b}{N}L + \frac{\alpha_2 b}{N}\lambda L_{joint} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{M}\right]\right)$ *bits communicated from nodes to the server are required for Algorithm 3 to reach* ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$, *where* $\overline{x}^K \sim U\{x^0, x^1, \ldots, x^{K-1}\}$.

Discussion. As in the previous bounds, the effects of local steps and compression are isolated into separate terms. A distinctive feature is the factor b/N before the compression rate α_2. This fully aligns with the intuition behind the employed communication mechanism: at each iteration with ensembling, only the logits corresponding to b out of N joint observations are transmitted and compressed using the compressor Q_2, resulting in an effective compression rate of $\alpha_{\text{eff}} = \alpha_2\frac{b}{N}$. Finally, we note that

computing the oracle at Line 13 of Algorithm 3, due to the term $\frac{\lambda}{N}\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(h_i^{k-1},x^k)$, requires computation over the entire joint dataset. This prevents any reduction in the number of oracle calls.

4.4 CJPAGE: Restoring the Advantage in Oracle Calls

We now aim to achieve an advantage in the number of oracle calls alongside the additional compression. To this end, we revisit the CJSAGA estimator (Line 13) and examine its mechanism in more detail. The second argument of the function $\nabla\mathcal{L}(\cdot,\cdot)$ in all terms corresponds to the current point x^k, reflecting the absence of stochasticity with respect to this argument and necessitating the computation of $\frac{\partial\overline{f}_i}{\partial x_m}\big|_{x_m=x_m^k}$ over the entire joint dataset. In contrast, the first argument of $\nabla\mathcal{L}(\cdot,\cdot)$ under the full sum $\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(h_i^{k-1},x^k)$ represents the local states from the previous iteration, while the correction sum $\sum_{i\in I_k}\nabla_{x_m}\left(\mathcal{L}(h_i^k,x^k)-\mathcal{L}(h_i^{k-1},x^k)\right)$ additionally uses b updated states h_i^k to complete the oracle into an unbiased estimate with a reduced number of compressed communications. To handle additional stochasticity, we should extend this mechanism to the weights x.

To construct a variance reduction scheme with respect to x, we relax the requirement of identical second arguments of the function $\nabla\mathcal{L}(\cdot,\cdot)$ in the estimator and replace the correction sum with $\frac{\lambda}{b}\sum_{i\in I^k}\nabla_{x_m}\left(\mathcal{L}(h_i^k,x^k)-\mathcal{L}(h_i^{k-1},x^{k-1})\right)$. Then, we note that replacing the full sum with $\frac{\lambda}{N}\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(h_i^{k-1},x^{k-1})$ does not remove the need to process the entire joint dataset. To eliminate this limitation, two design choices arise. The first is to store a reference point w^k together with the corresponding values $\frac{\partial\overline{f}_i}{\partial x_m}\big|_{x_m=\omega_m^k}$, leading to a SAG-type estimator. However, this approach increases the memory utilization. To avoid this overhead, we adopt an alternative approach: replacing the full sum in the oracle with the estimator from the previous iteration g_m^{k-1}, leading to a PAGE-type approximation $\hat{g}_m^k=\hat{g}_m^{k-1}+\frac{\lambda}{b}\sum_{i\in I^k}\nabla_{x_m}\left(\mathcal{L}(h_i^k,x^k)-\mathcal{L}(h_i^{k-1},x^{k-1})\right)$.

We are now ready to present the **C**ompressed **J**oint **PAGE** - CJPAGE (Algorithm 4). Lines 2, 17-20 implement the local step mechanism. Lines 8-14 incorporate the error feedback scheme, analogous to JEF21 and CJSAGA. Finally, Lines 16, 3-6 implement the PAGE-type estimator introduced above, which requires periodic processing of the full dataset. Notably, we employ different compressors, $\mathcal{Q}_1$ and $\mathcal{Q}_2$, for transmitting logits in iterations involving full or partial joint observation processing. This provides additional flexibility and enables stronger compression of logits during the iterations with full evaluations.

Algorithm 4 CJPAGE

1: **For** $k=1,2,\ldots,K$ **do:**
2: **With probability p:**
3: **With probability q:**
4: (1) $I^k=I$
5: **With probability $1-$q:**
6: (2) $I^k\subset I$, $|I^k|=b$
7: **For** $m=1,2,\ldots,M$ **in parallel do:**
8: $\Delta_{i,m}^k=\mathcal{Q}_{(\cdot)}\left(\hat{f}_i(x_m^k)-h_{i,m}^{k-1}\right), i\in I^k$
9: $h_{i,m}^k=h_{i,m}^{k-1}+\Delta_{i,m}^k,\ i\in I^k$
10: $h_{i,m}^k=h_{i,m}^{k-1},\ i\in I\setminus I^k$
11: Send $\Delta_{i,m}^k$ to the server
12: **For** $i\in I^k$ **server do:**
13: $h_i^k=h_i^{k-1}+\frac{1}{M}\sum_{m=1}^{M}\Delta_{i,m}^k$
14: Broadcast h_i^k to nodes
15: **For** $m=1,2,\ldots,M$ **in parallel do:**
16: $\begin{bmatrix}(1)\hat{g}_m^k=\lambda\frac{1}{N}\sum_{i=1}^{N}\nabla_{x_m}\mathcal{L}(h_i^k,x^k)\\(2)\hat{g}_m^k=\hat{g}_m^{k-1}+\frac{\lambda}{b}\sum_{i\in I^k}\nabla_{x_m}\big(\mathcal{L}(h_i^k,x^k)\\\qquad-\mathcal{L}(h_i^{k-1},x^{k-1})\big)\end{bmatrix}$
17: **With probability $1-$p:**
18: **For** $m=1,2,\ldots,M$ **in parallel do:**
19: $\hat{g}_m^k=\hat{g}_m^{k-1}$
20: $h_{i,m}^k=h_{i,m}^{k-1}$
21: **For** $m=1,2,\ldots,M$ **in parallel do:**
22: $g_m^k=\nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
23: $x_m^{k+1}=x_m^k-\gamma(\hat{g}_m^k+g_m^k)$

The theoretical analysis is presented in Theorem D.4 and follows the same structure as for the preceding algorithms, with a key difference in the treatment of the oracle distortion bound. In more detail, using points x from different iterations in the oracle (Line 16) complicates the application of Lemma D.1, which is central to the compression analysis. We overcome this challenge by separating the effects of compression and stochasticity via a carefully chosen add-and-subtract argument, applying the Cauchy-Schwarz (CS) inequality in the derivation (35). Essentially, it retains only the differences of terms with identical points x in the compression-induced part. We provide the final guarantees in Corollary 4.3 below.

Corollary 4.3. *Let Assumptions 3.1- 3.5 hold and* $\gamma = \min\left\{\frac{1}{4L}; \frac{pq}{10\lambda L_{joint}}; \frac{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{50\lambda L_{\mathcal{L}}\tilde{L}_f^2}\right\}$. *Then Algorithm 4 requires* $\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_f^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)$ *iterations,* $\mathcal{O}\left(\frac{1}{\varepsilon}\cdot p\left(qN + (1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_f^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)$ *joint objective oracle calls, and* $\mathcal{O}\left(\frac{1}{\varepsilon}\cdot p\left(\alpha_1 qN + \alpha_2(1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}}\tilde{L}_f^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)$ *bits communicated from nodes to the server to reach* ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.

Discussion. As in the guarantees for `CJSAGA`, the effects induced by different mechanisms can be disentangled in the resulting bounds, while joint objective oracle complexity scales with $qN+(1-q)b$, compared with $N + b$ for `CJSAGA`. The factor $\alpha_{\text{eff}} = \alpha_1 q + \alpha_2 \frac{b}{N}(1-q)$ in the denominator fully aligns with the underlying intuition and once again highlights the close analogy between stochasticity-aligned and classical compression mechanisms discussed in more detail in Appendix A.

5 Experiments

Setup. We evaluate our method on a personalized federated image classification task using ResNet-18 [He et al., 2016] trained on CIFAR-10 [Krizhevsky, 2009] across 10 clients. An additional experimental setup on the larger Food-101 task is presented in Appendix C.8. To model statistical heterogeneity, we partition the data using a Dirichlet distribution with parameter $\alpha = 0.5$. Each client is assigned 1000 local samples with client-specific label skew, along with access to a shared dataset of 3000 samples. Performance is evaluated on client-specific validation sets that follow the same distribution as each client's local data, capturing personalization quality. We focus on `CJPAGE` in ablations, as it represents the most advanced variant of our method.

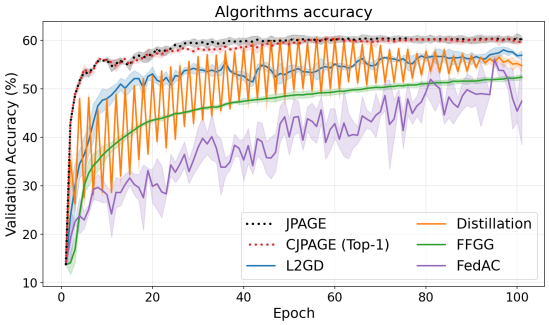


Figure 1: `JPAGE` and `CJPAGE` against baselines.

Baselines. We compare against several personalized federated learning methods: (i) `FFGG` [Mishchenko et al., 2023], which splits parameters into shared and local components; (ii) `L2GD` [Hanzely and Richtárik, 2020], which regularizes local models toward a global solution; (iii) `FedAC` [Hsu et al., 2023], a clustering-based approach for non-IID data; and (iv) `Distillation` [Lin et al., 2020], where clients align predictions via server-aggregated outputs.

Discussion. Figure 1 highlights several important trends. First, `JPAGE` demonstrates stable and monotonic convergence, in contrast to `Distillation`, which suffers from late-stage degradation. Second, consistent improvements over `L2GD` and `FedAC` indicate better adaptation to heterogeneous client distributions. Finally, unlike clustering- or regularization-based methods, `JPAGE` does not rely on sensitive hyperparameter tuning or client grouping, resulting in more robust optimization behavior.

Table 1: Per-client communication (MB) to reach 50% validation accuracy.

Method	Uplink	Downlink
`JPAGE`	0.65 ± 0.2	0.65 ± 0.2
`CJPAGE`	0.10 ± 0.03	0.65 ± 0.2
`Distillation`	0.44 ± 0.07	0.44 ± 0.07
`L2GD`	2700 ± 600	2700 ± 600
`FFGG`	2800 ± 300	2800 ± 300
`FedAC`	620 ± 50	620 ± 50

Table 1 reports per-client communication required to reach 50% validation accuracy. `CJPAGE` has the lowest uplink cost among the compared methods. Its downlink exceeds its uplink since compression is applied only to client-to-server messages. This advantage combines the small dimension of logits for selected observations ($a \ll d$, where d is the model dimension) with local steps, mini-batching, and compression. In the same ResNet-18/CIFAR-10 setting, the configuration $N = 3000$, $b = 64$, $p = q = 0.1$ with Top-1 compression and 32-bit logits yields an expected uplink payload of approximately 1.1×10^3 bits per client update. In stark contrast, an uncompressed ResNet-18 gradient requires approximately 3.5×10^8 bits.

Figure 2 shows that reducing C parameter in `Top-C` compressor (i.e., increasing compression) leads to only a minor degradation in accuracy, while preserving overall convergence behavior across different values of p. Hence, aggressive compression appears to be the best choice for the considered setup, indicating the robustness of the method to communication constraints. Additional ablations on compression, local-step probability, and communication cost, as well as full experimental details, are provided in Appendix C.

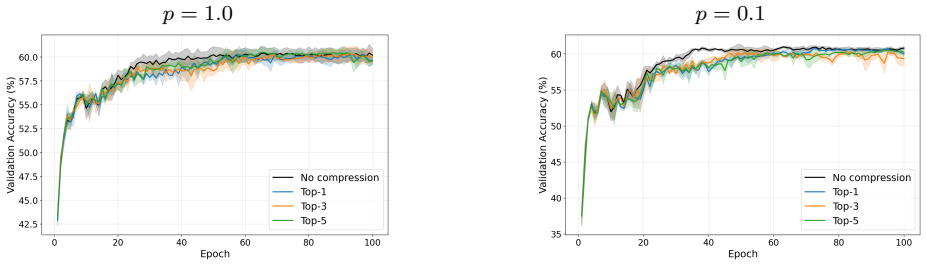


Figure 2: Acc. vs epoch across different C, p.

Summary. Overall, `CJPAGE` consistently outperforms strong personalized federated learning baselines, exhibits stable training dynamics under non-IID data, and admits communication-efficient variants with minimal loss in performance. Exchanging logits creates a distinct communication tradeoff: a full joint update scales with aN, the output dimension and the shared-set size, rather than the model dimension d of a gradient update. Standard compression and local steps reduce the payload and frequency of these updates without changing their dependence on N. Stochasticity-aligned compression changes that dependence: a mini-batch update sends logits for b instead of N observations. Theoretically, this softens the dependence of total communication on N from linear to $\sqrt{N}$ when $b \sim \sqrt{N}$ and $q = b/N$ (Appendix E).

6 Conclusion

In this work, we present a systematic view of communication efficiency in the federated ensembling setup. On the one hand, we extend classic techniques such as local steps and compression to this setting. On the other hand, we explore new opportunities enabled by its deviation from the standard FL formulation. These insights inform the design of four algorithms: `JEF21`, `JPAGE`, `CJSAGA`, and `CJPAGE`. For each of them, we establish convergence guarantees in the smooth non-convex setting. The proposed methods incorporate advanced variance reduction and error-feedback techniques while simultaneously combining multiple communication-efficiency mechanisms, thereby enhancing their practical relevance. Conducted experiments verify our findings.

7 Limitations and future work

Implementation constraints. To evaluate the joint objective, each client computes logits, retains the corresponding computation graph until the aggregated values arrive from the server, and then performs backpropagation. This can constrain gradient accumulation, where several forward and backward passes contribute to a single optimizer step. If the joint data are processed in microbatches whose logits are communicated together, the client must retain each microbatch's graph and activations until aggregation or recompute its forward pass to rebuild the graph before backpropagation. Without gradient accumulation, only one such graph is needed at a time, so this constraint introduces no additional storage or recomputation.

Cross-algorithm guidance. The four methods progressively combine local steps, variance reduction, and two forms of compression. Their effects on joint-dataset processing and logit communication suggest the following choices. When communication is the sole bottleneck, `JEF21` (Algorithm 1) offers the simplest direct reduction through compression. If further communication reduction is needed, `CJSAGA` (Algorithm 3) adds stochasticity-aligned compression. When clients cannot process the full shared set at every joint update, `JPAGE` (Algorithm 2) reduces oracle work. If communication is also limiting, `CJPAGE` (Algorithm 4) combines these mechanisms with standard compression. These recommendations require systematic validation across model scales, compute and network regimes, which we leave to future work.

Acknowledgments

The contribution of Valery Parfenov to this study was supported by the Ministry of Science and Higher Education of the Russian Federation, project FEWZ-2024-0052.

References

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.

Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

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.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

Léon Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. In *Proceedings of the symposium on learning and data science, Paris*, volume 8, pages 2624–2633, 2009.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Yue Cao, Thomas Andrew Geddes, Jean Yee Hwa Yang, and Pengyi Yang. Ensemble deep learning in bioinformatics. *Nature Machine Intelligence*, 2(9):500–508, 2020.

Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Grzegorz Dudek. Heterogeneous ensembles for short-term electricity demand forecasting. In *2016 17th International Scientific Conference on Electric Power Engineering (EPE)*, pages 1–6. IEEE, 2016.

Anuvabh Dutt, Denis Pellerin, and Georges Quénot. Coupled ensembles of neural networks. *Neurocomputing*, 396:346–357, 2020.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.

Ilyas Fatkhullin, Igor Sokolov, Eduard Gorbunov, Zhize Li, and Peter Richtárik. Ef21 with bells & whistles: Six algorithmic extensions of modern error feedback. *Journal of Machine Learning Research*, 26(189):1–50, 2025.

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

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Eduard Gorbunov, Konstantin P Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster nonconvex distributed learning with compression. In *International Conference on Machine Learning*, pages 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*, pages 3556–3564. PMLR, 2021b.

Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. In *International Conference on Learning Representations*, volume 2024, pages 29368–29391, 2024.

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*, pages 770–778, 2016.

Hua-Yang Hsu, Kay Hooi Keoy, Jun-Ru Chen, Han-Chieh Chao, and Chin-Feng Lai. Personalized federated learning algorithm with adaptive clustering for non-iid iot data incorporating multi-task learning and neural network model characteristics. *Sensors*, 23(22):9016, 2023.

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

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

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

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*, pages 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*, pages 5132–5143. PMLR, 2020.

Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34:28663–28676, 2021.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pages 17564–17579. PMLR, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *https //www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

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.

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International conference on machine learning*, pages 6286–6295. PMLR, 2021.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

Ruichen Luo, Sebastian U Stich, Samuel Horváth, and Martin Takáč. Revisiting localsgd and scaffold: Improved rates and missing analysis. *arXiv preprint arXiv:2501.04443*, 2025.

Pouya M Ghari and Yanning Shen. Personalized federated learning with mixture of models for adaptive prediction and model fine-tuning. *Advances in Neural Information Processing Systems*, 37:92155–92183, 2024.

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*, pages 1273–1282. PMLR, 2017.

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pages 15750–15769. PMLR, 2022.

Konstantin Mishchenko, Rustem Islamov, Eduard Gorbunov, and Samuel Horváth. Partially personalized federated learning: Breaking the curse of data heterogeneity. *arXiv preprint arXiv:2305.18285*, 2023.

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

Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.

Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. *arXiv preprint arXiv:2102.03198*, 2021.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. *Advances in neural information processing systems*, 34:840–851, 2021.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Kilian Pfeiffer, Martin Rapp, Ramin Khalili, and Jörg Henkel. Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Computing Surveys*, 55(14s):1–27, 2023.

Xun Qian, Zheng Qu, and Peter Richtárik. Saga with arbitrary sampling. In *International Conference on Machine Learning*, pages 5190–5199. PMLR, 2019.

Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10061–10071, 2022.

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.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Nicolas Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. *Advances in neural information processing systems*, 25, 2012.

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, pages 1058–1062. Singapore, 2014.

Yong Shi, Yuanying Zhang, Yang Xiao, and Lingfeng Niu. Optimization strategies for client drift in federated learning: A review. *Procedia Computer Science*, 214:1168–1173, 2022.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

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

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

Nikko Ström. Scalable distributed dnn training using commodity gpu cloud computing. 2015.

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.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *International Conference on Machine Learning*, pages 21111–21132. PMLR, 2022.

M Tanveer, Aryan Rastogi, Vardhan Paliwal, MA Ganaie, Ashwani Kumar Malik, Javier Del Ser, and Chin-Teng Lin. Ensemble deep learning in speech signal tasks: a review. *Neurocomputing*, 550: 126436, 2023.

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.

Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, Ming Gao, et al. Dfrd: Data-free robustness distillation for heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 36:17854–17866, 2023.

Danny Wood, Tingting Mu, Andrew M Webb, Henry WJ Reeve, Mikel Luján, and Gavin Brown. A unified theory of diversity in ensemble learning. *Journal of machine learning research*, 24(359): 1–49, 2023.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International conference on machine learning*, pages 6893–6901. PMLR, 2019.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

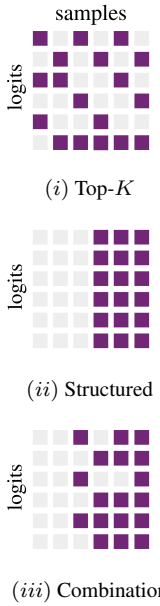
Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35:21414–21428, 2022.

Junyuan Zhang, Songhua Liu, and Xinchao Wang. One-shot federated learning via synthetic distiller-distillate communication. *Advances in neural information processing systems*, 37:102611–102633, 2024.

Jiewan Zheng, Xianbin Cao, Baochang Zhang, Xiantong Zhen, and Xiangbo Su. Deep ensemble machine for video classification. *IEEE transactions on neural networks and learning systems*, 30 (2):553–565, 2018.

Zhi-Hua Zhou. Ensemble learning. In *Machine learning*, pages 181–210. Springer, 2021.

A A Joint Perspective on Stochasticity-Aligned and Classic Compression



(i) Top-K

(ii) Structured

(iii) Combination

Figure 3: sparsification illustration

The mechanisms described in the main text, along with the derived bounds, reveal a similarity between stochasticity-aligned and classic compression. In this section, we elaborate on this connection.

One of the most widely used compression mechanisms in practice is sparsification. For contractive operators, a canonical example is the `Top-C` method, which retains only the C largest components of its input and sets the remaining ones to zero. For a-dimensional logits, its contraction rate in Definition 3.1 is $\alpha = C/a$. In our setting, the compressed quantity can be viewed as a matrix. Its columns correspond to logits for individual samples, while the rows represent logits of a given class across different data points. The output of a `Top-50` compressor is illustrated in panel (i) of Figure 3. Such a mechanism can be directly employed in `JEF21`.

Stochasticity-aligned compression operates by updating logits $\hat{f}_i(x_m)$ (or states $h^k_{i,m}$) only for a subset of all observations. For each sample, either all logits (or states) are transmitted, or none of them. This behavior is illustrated in panel (ii) of Figure 3. To ensure comparable communication across all variants, we sparsify half of the inputs in this illustration. In light of the above, stochasticity-aligned compression can be viewed as a form of structured sparsification. Importantly, the absence of updates for a subset of samples enables computations to be performed only on a fraction of the data, thereby reducing the number of oracle calls per iteration, as in `JPAGE`.

Finally, it is possible to transmit logits (states) for a fraction of the dataset and use additional sparsification, as done in `CJSAGA` and `CJPAGE`. This regime is depicted in panel (iii) of Figure 3.

B Additional Related Works

We thoroughly covered ensembles and their natural adaptability to the federated learning setting in the introduction. In this section, we focus on another relevant fields for our work: variance reduction and communication efficiency.

Variance reduction. In the context of modern large-scale problems, computing the full gradient is often computationally prohibitive [Bottou, 2010, Dean et al., 2012], making stochastic optimization particularly relevant. One of the earliest methods for working with stochastic oracles is `SGD` [Robbins and Monro, 1951], which offers computational advantages but suffers from non vanishing variance [Bottou, 2009, Moulines and Bach, 2011, Gower et al., 2020] and converges only to a neighborhood of the optimum. To address this issue, variance reduction (VR) methods [Roux et al., 2012] were developed. These algorithms share a common idea: leveraging information from previous iterations to improve the quality of gradient approximations. Currently, two orthogonal approaches have emerged in this line of research. The first relies on storing previously computed mini-batch gradients and therefore incurs additional memory overhead. The earliest algorithm in this branch is `SAG` [Roux et al., 2012], followed by a major advancement with `SAGA` [Defazio et al., 2014], which employs an unbiased gradient estimator and is adapted to proximal operators. Further extensions of the gradient history paradigm include `L-SAGA` and `SAGA-AS` [Qian et al., 2019]. The second, orthogonal approach avoids additional memory requirements but periodically necessitates computing the full gradient with a small probability. This line of work originated with `SVRG` [Johnson and Zhang, 2013] and was followed by `SARAH` [Nguyen et al., 2017]. More recently, `PAGE` [Li et al., 2021] was proposed, replacing the nested loop structure with a probabilistic update scheme. Contemporary representatives of this family include `SPIDER` [Fang et al., 2018], `STORM` [Cutkosky and Orabona, 2019], and `Katyusha` [Allen-Zhu, 2018].

Communication efficiency. One of the primary contemporary approaches in this area is *compression methods*. Their core idea is to mitigate the high costs of communication by compressing the transmitted information. Early works on compression focused on one-bit quantization [Seide et al.,

2014, Bernstein et al., 2018]. Subsequently, the community formalized this paradigm through the introduction of unbiased and contractive compression operators [Beznosikov et al., 2023].

One of the first compression algorithms was QSGD Alistarh et al. [2017], a version of SGD with quantized communications. Notably, even with exact local oracles, it converged only to a neighborhood of the optimum. Indeed, it exhibits substantial similarity to the SGD issue mentioned in the previous paragraph: as the algorithm approaches the optimum, the gradient of a smooth function approaches zero, while SGD and QSGD oracles can substantially deviate from zero, causing unstable behavior near the minima. Using a variance reduction-inspired modification – transmitting the difference of local oracles rather than their absolute values – the authors of the DIANA [Mishchenko et al., 2025] algorithm overcame this problem. Subsequent works [Gorbunov et al., 2021a, Tyurin and Richtárik, 2022] introduced variance reduction techniques for the local oracles themselves.

For practical applications, contractive compressors have proven to be the most effective [Alistarh et al., 2018, Beznosikov et al., 2023], with Top-k sparsification as their primary representative. A major breakthrough in contractive compression was the error feedback technique, initially proposed as a successful heuristic [Seide et al., 2014, Ström, 2015], and later accompanied by theoretical guarantees [Stich et al., 2018, Karimireddy et al., 2019]. Later, Richtárik et al. showed that local state difference compression can be interpreted as a variant of the error feedback, which led to the development of the EF21 [Richtárik et al., 2021] algorithm with a simplified design and convergence analysis in a nonconvex setup. The most recent improvements in this area are presented in [Fatkhullin et al., 2025, Gao et al., 2025].

An orthogonal direction for improving communication efficiency is the use of *local steps* [Stich, 2018, Gorbunov et al., 2021b]. The key insight in this area is that clients can update their models independently after computing gradients on their own data. Performing several such local updates, followed by the aggregation and averaging of the client models, allows for effectively taking a larger optimization step per communication round. Incorporating this procedure into SGD led to the first algorithm in this area, known as LocalSGD [Stich, 2018]. However, this idea is not a remedy for the communication bottleneck in federated learning. In particular, using an excessively large number of local updates may cause clients to overfit to their local data, and averaging may fail to produce a high-quality state. This phenomenon, known as *client drift* [Shi et al., 2022], prevents learning a global model within a single communication round and necessitates periodic communication [Karimireddy et al., 2020, Table 3]. A major subsequent step in this direction was the SCAFFOLD algorithm [Karimireddy et al., 2020]. Inspired by the variance reduction technique SAGA [Defazio et al., 2014], the authors introduced a variance-reduced scheme for local steps that leverages the most recently aggregated global gradient to mitigate client drift. The ideas behind LocalSGD and SCAFFOLD have demonstrated strong practical performance and have been widely developed within the research community [Murata and Suzuki, 2021, Karimireddy et al., 2021, Mishchenko et al., 2022, Takezawa et al., 2025]. Nevertheless, understanding local-step methods from the perspective of theoretical guarantees remains challenging and leaves several open questions [Luo et al., 2025]. For example, it has not yet been theoretically established that local steps provide acceleration under a general non-convex federated learning setup.

C Experiments

C.1 Additional experimental details for ResNet-18/CIFAR-10

This subsection reports the hyperparameters of the methods compared in Figure 1. All experiments ran for 100 epochs. A grid search over the `JPAGE` learning rate γ and joint-loss weight λ selected $\gamma = 10^{-3}$, linearly decayed to 10^{-5}, and $\lambda = 1$. `CJPAGE` used the same learning-rate schedule and λ. For Figure 1, we set $p = 1$ and fixed $b = 64$ and $q = 0.1$ without tuning for both methods, `CJPAGE` additionally used Top-C compression with $C = 1$.

We utilized the following configurations for the baseline methods, selected based on their respective optimal performances:

- `FedAC`: We set the clustering thresholds $\alpha = 5.0$ and $\beta = 0.0$. The number of local epochs is set to $E = 1$, and the learning rate is set to $\gamma = 0.001$.
- `L2GD`: We use probability $p = 0.75$ and $\lambda = 1.0$. The learning rate is set to $\gamma = 0.001$.
- `FFGG`: Following the partial personalization strategy, we designated the final classifier layer as local parameters and the remaining layers as global parameters. We used separate learning rates: $\gamma_{global} = 0.001$ for shared parameters and $\gamma_{local} = 0.01$ for private parameters. The method was configured with $k = 5$ and $\tau = 6$.
- `Distillation`: We perform a distillation epoch after each local epoch. We pick learning rate $\gamma = 0.001$.

The experiments are implemented in Python using the PyTorch library [Paszke et al., 2017], leveraging both a single CPU (Intel Xeon 2.20 GHz) and a single GPU (NVIDIA A100) for computation. The total runtime for all experiments is approximately 100 GPU-hours. Unless stated otherwise, subsequent experiments use these hyperparameter settings.

C.2 Experimental setup

We evaluate our theoretical findings on a classic federated learning classification task. We train the ResNet-18 [He et al., 2016] model on the subset of CIFAR-10 [Krizhevsky, 2009] dataset shared across 10 clients.

The shared part of the dataset, corresponding to the joint component of loss (1), consists of 3000 samples and is distributed across all clients. Following the characteristic features of the federated setup, each node is additionally endowed with a local dataset of 1000 observations that have a client-specific class distribution. The complete data partitioning is constructed to model data heterogeneity using a Dirichlet distribution with parameter $\alpha = 0.5$, a commonly adopted choice in federated learning, and is reported in Table 2.

Table 2: Distribution of training and shared datasets across clients.

Class	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	Shared
airplane	1	197	138	23	67	69	49	83	2	83	300
automobile	89	0	40	185	60	98	1	103	68	2	300
bird	258	0	97	110	132	36	5	33	488	19	300
cat	29	377	0	5	99	216	127	326	12	133	300
deer	155	74	159	36	453	183	512	52	52	5	300
dog	16	45	184	34	2	63	67	75	70	66	300
frog	11	41	257	365	2	33	204	293	234	456	300
horse	0	49	46	22	9	103	0	28	64	40	300
ship	410	182	78	105	21	167	20	0	8	109	300
truck	31	35	1	115	155	32	15	7	2	87	300
Total	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	3000

We evaluate models performance on personalized data distributions. To this end, each client is assigned a validation dataset with a corresponding class partition, represented in Table 3.

Table 3: Distribution of classes across clients validation datasets.

Class	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
airplane	0	99	69	11	34	34	25	42	1	42
automobile	45	0	20	93	29	49	0	52	34	1
bird	129	0	48	55	66	18	2	16	244	9
cat	14	189	0	2	50	108	64	163	6	67
deer	78	37	80	18	227	92	256	26	26	2
dog	8	22	92	16	1	31	34	37	35	32
frog	5	20	129	183	1	16	102	147	117	228
horse	0	25	23	11	4	52	0	14	32	20
ship	205	91	39	53	10	84	10	0	4	55
truck	16	17	0	58	78	16	7	3	1	44
Total	500	500	500	500	500	500	500	500	500	500

C.3 Comparison of JPAGE with the baselines

We compare Algorithm 2 with personalized federated learning approaches, namely `FFGG` [Mishchenko et al., 2023], `L2GD` [Hanzely and Richtárik, 2020], and the clustering-based method `FedAC` [Hsu et al., 2023]. `FFGG` introduces a partially personalized framework that splits variables into shared global parameters, which are used to learn data representations, and private local parameters that are fine-tuned for specific clients. `L2GD` incorporates a regularization term that penalizes deviations of local models from the global average. `FedAC` addresses non-IID data via automatic clustering, dynamically grouping clients. Finally, we include a `Distillation` [Lin et al., 2020] baseline where clients align their objectives by training to predict the server-aggregated mean of all client outputs.

As summarized in Table 4, the proposed method `JPAGE` achieves the highest performance with a validation accuracy of 60.40%, outperforming the `L2GD`, `FFGG`, and `FedAC` baselines. Additionally, Figure 1 illustrates the convergence trajectories throughout the training process. Although the `Distillation` approach reaches comparable peak accuracy, it exhibits performance degradation in the later stages, whereas `JPAGE` and `CJPAGE` maintain stability while converging.

Table 4: Comparison of JPAGE, CJPAGE with varying C and fixed $p = 1.0$, and the baselines in terms of the average of the five highest validation accuracies achieved during training.

Method	`JPAGE` **(Ours)**	`L2GD`	`Distillation`	`FedAC`	`FFGG`	`CJPAGE` ($C = 5$)	`CJPAGE` ($C = 3$)	`CJPAGE` ($C = 1$)
Accuracy (%)	**60.40**	57.56	60.31	57.51	52.19	*59.67*	*59.99*	*59.57*

The proposed methods combine compression, local steps, and joint-dataset mini-batching in different ways. The following sections examine their practical effects.

C.4 Ablation study on the compression rate

We evaluate the impact of the compression rates α_1 and α_2 on the performance of `CJPAGE` (Algorithm 4). We set $\alpha_1 = \alpha_2$ (both equal to C/a for Top-C compression) and vary $C \in \{1, 3, 5, 10\}$ accordingly for different ensembling probabilities $p \in \{0.1, 0.3, 1.0\}$. Figure 4 shows accuracy against training epochs (top) and transferred bits (bottom). As observed, introducing compression leads to mild degradation in performance. However, further increasing the compression level does not yield a statistically significant deterioration. At the same time, decreasing C substantially reduces uplink communication. In this light, aggressive compression appears to be the most favorable choice for the considered setup.

C.5 Ablation study on the local-step probability

In this section, we study the effect of the ensembling probability p on the performance of `CJPAGE`. We vary p over the set $\{0.1, 0.3, 1.0\}$ for $C \in \{10, 5, 3, 1\}$. Figure 5 shows accuracy against training epochs (left) and transferred bits (right). Notably, across all compression levels, a slowdown in accuracy improvement is observed as the ensembling probability decreases, particularly in the range

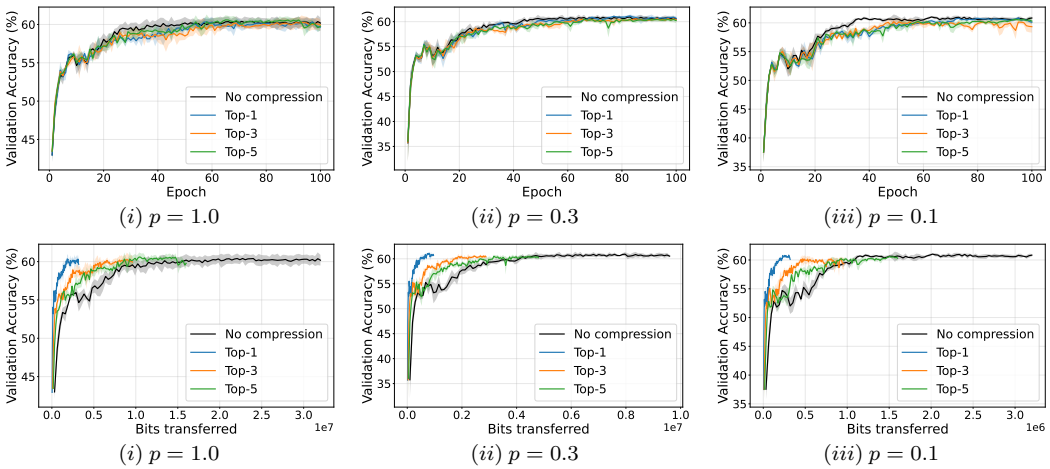


Figure 4: Compression-rate ablation for CJPAGE at $p = 1, 0.3, 0.1$ (left to right): accuracy versus epochs (top) and uplink bits (bottom).

of epochs 5-25. At the same time, the use of local steps substantially reduces communication, leading to faster progress in terms of communicated bits.

We further compare CJPAGE, CJSAGA, and JEF21 at $p \in \{1, 0.3, 0.1\}$. All methods use Top-C compression with $C = 1$ and train for 100 epochs with $\lambda = 1$ and a learning rate decaying linearly from 10^{-3} to 10^{-5}. We set $b = 64$ for CJSAGA and CJPAGE, and $q = 0.1$ for CJPAGE. Table 5 reports validation accuracy, communication volume, and joint-oracle calls.

Table 5: Ensembling-probability ablation: validation accuracy, communication, and joint-oracle calls.

Method	p	Accuracy (%)	Communication (MB)	Oracle calls (10^6)
CJPAGE	1	61.4 ± 0.4	44.9	5
CJPAGE	0.3	61.4 ± 0.4	13.1	1.5
CJPAGE	0.1	61.6 ± 0.3	4.6	0.5
CJSAGA	1	61.7 ± 0.3	44.9	50
CJSAGA	0.3	61.8 ± 0.3	13.1	15
CJSAGA	0.1	61.5 ± 0.7	4.6	5
JEF21	1	61.9 ± 0.2	2105.7	48
JEF21	0.3	62.18 ± 0.09	636.1	15
JEF21	0.1	61.9 ± 0.2	200	5

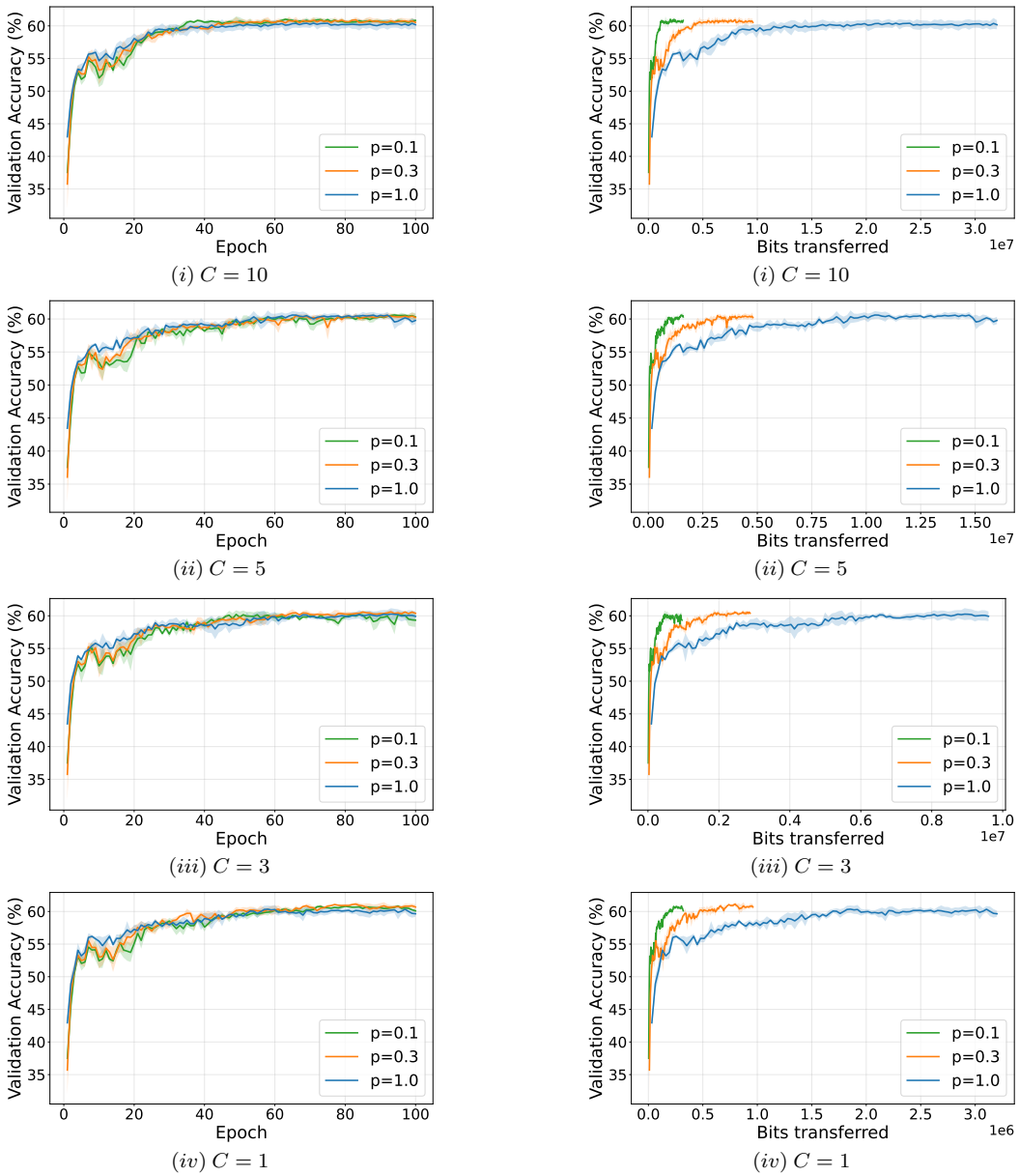
Within each method, mean accuracy varies by at most 0.32% across the tested p, while reducing p from 1 to 0.1 lowers communication and joint-oracle calls by approximately an order of magnitude. The methods thus retain accuracy under infrequent ensembling in this setting.

C.6 Ablation study on the mini-batch size

Using the same CIFAR-10 setup for CJPAGE, with $p = 1$, $q = 0.1$, and $C = 5$, we vary the joint mini-batch size b. Table 6 reports the resulting validation accuracy.

Table 6: Validation accuracy under different joint mini-batch sizes.

b	Accuracy (%)
16	59.7 ± 0.5
32	60 ± 1
64	60.0 ± 0.4



(i) $C = 10$ (i) $C = 10$

(ii) $C = 5$ (ii) $C = 5$

(iii) $C = 3$ (iii) $C = 3$

(iv) $C = 1$ (iv) $C = 1$

Figure 5: Ensembling-probability (p) ablation for `CJPAGE` at $C = 10, 5, 3, 1$ (top to bottom): accuracy versus epochs (left) and transferred bits (right).

The mean accuracies vary by at most 0.37% across the tested mini-batch sizes.

C.7 Ablation study on the shared-dataset size and label distribution

We assess sensitivity to the quality of the shared dataset using `CJPAGE` in the same ResNet-18/CIFAR-10 setting, with $p = q = 0.1$, $b = 64$, and $C = 1$. We vary the shared-set size $N \in \{500, 1000, 3000\}$ and, for $N = 3000$, its class imbalance through a Dirichlet distribution with $\alpha \in \{1, 0.5\}$. Table 7 reports validation accuracy over three seeds.

Table 7: Validation accuracy across shared-set sizes and label distributions.

Shared set	Accuracy (%)
Balanced, $N = 500$	60.8 ± 0.7
Balanced, $N = 1000$	61.0 ± 0.7
Balanced, $N = 3000$	60.4 ± 0.4
Dirichlet $\alpha = 1$, $N = 3000$	60.8 ± 0.8
Dirichlet $\alpha = 0.5$, $N = 3000$	60.5 ± 0.4

Mean accuracy remains between 60.4% and 61.0% across these settings, showing robustness to the tested shared-set sizes and class imbalance.

C.8 Additional experimental setup

We test the methods on Food-101 [Bossard et al., 2014], which has 101 classes, using ImageNet-1K-pretrained FasterViT-0 [Hatamizadeh et al., 2024] with a new classification head and LoRA. Each of 20 clients has 1,000 training and 500 validation images with client-specific label distributions drawn from a Dirichlet partition with $\alpha = 0.5$. A disjoint, class-balanced set of $N = 3{,}000$ images is shared across clients. We compare `JPAGE` and `CJPAGE` with `L2GD`, `FedAC`, and `FFGG` over three seeds.

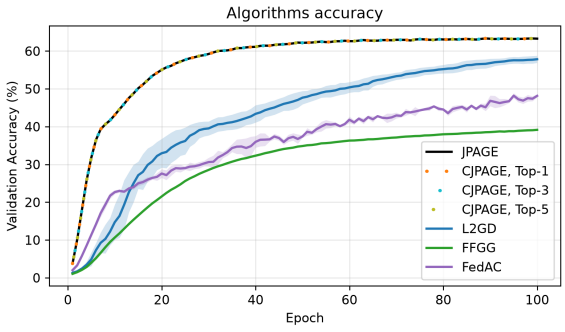


Figure 6: Mean client validation accuracy on Food-101 over three seeds.

Figure 6 shows `JPAGE` and `CJPAGE` converging to approximately 63.3% accuracy, compared with 57.8%, 48.2%, and 39.2% for `L2GD`, `FedAC`, and `FFGG`. Notably, all three `CJPAGE` variants track `JPAGE`, and even the most aggressive Top-1 ($C = 1$) compression causes no visible convergence loss.

Table 8 reports epochs and per-client uplink and downlink communication to reach three validation-accuracy targets. At 40%, `JPAGE` and `CJPAGE` require 8 epochs, about 4.0 and 6.7 times fewer than `L2GD` and `FedAC`, respectively. With $C = 1$, `CJPAGE` uses about 970 and 53 times less uplink than these baselines. Its downlink is about 14 times smaller than `L2GD`'s but 1.4 times larger than `FedAC`'s. `FedAC` does not reach 50%. At 55%, the `PAGE`-type methods need about 4 times fewer epochs than `L2GD`, with uplink savings from 14 times for `JPAGE` to 940 times for `CJPAGE` at $C = 1$. Only the logit-communication methods reach 60%, where $C = 1$ reduces uplink about 65 times relative to `JPAGE`. For parameter-sharing baselines, communication counts only LoRA adapters.

Table 8: Epochs and per-client communication (MB) to reach each mean validation-accuracy target (mean $\pm$ standard deviation over three seeds). A dash indicates that no seed reached the target within 100 epochs.

	40%			55%			60%		
Method	Epochs	Uplink	Downlink	Epochs	Uplink	Downlink	Epochs	Uplink	Downlink
CJPAGE, $C = 1$	8	0.14 ± 0.02	10 ± 2	20.3 ± 0.5	0.35 ± 0.01	23.6 ± 0.5	33.0 ± 0.8	0.54 ± 0.02	37 ± 1
CJPAGE, $C = 3$	8	0.42 ± 0.07	10 ± 2	20.3 ± 0.5	1.05 ± 0.02	23.6 ± 0.5	32.3 ± 0.5	1.58 ± 0.08	35 ± 2
CJPAGE, $C = 5$	8	0.7 ± 0.1	10 ± 2	20.3 ± 0.5	1.75 ± 0.04	23.6 ± 0.5	32.3 ± 0.5	2.6 ± 0.1	35 ± 2
JPAGE	8	10 ± 2	10 ± 2	20.3 ± 0.5	23.6 ± 0.5	23.6 ± 0.5	32.3 ± 0.5	35 ± 2	35 ± 2
L2GD	32 ± 7	136 ± 4	136 ± 4	79 ± 7	329 ± 5	329 ± 5	–	–	–
FedAC	54 ± 2	7.4 ± 0.3	7.4 ± 0.3	–	–	–	–	–	–
FFGG	–	–	–	–	–	–	–	–	–

We use the following configurations for our methods and baselines:

- JPAGE and CJPAGE: We set $\lambda = p = 1$, $q = 0.1$, $b = 64$, and the learning rate to 3×10^{-4}. CJPAGE uses Top-C compression with $C \in \{1, 3, 5\}$.
- L2GD: We set $\lambda = 1$, $p = 0.75$, and the learning rate to 3×10^{-4}.
- FedAC: We set $\alpha = 5$, $\beta = 0$, one local epoch, and the learning rate to 3×10^{-4}.
- FFGG: We use $K = 5$, $\tau = 6$, a local head, and separate learning rates of 3×10^{-4} for global parameters and 3×10^{-3} for local parameters.

All methods use AdamW-style updates, batch size 128, weight decay 0.01, dropout 0.1, and 100 epochs without a learning-rate schedule. Images are resized to 224×224 and normalized using Food-101 statistics. Random cropping with padding 4 and horizontal flipping are applied only to local training data.

D Missing proofs

As already outlined in the main part of the paper, we analyze the generalized Algorithm 5 in Appendix D.2. The results for JEF21 and CJSAGA are obtained in Appendix D.3 by setting $q = 1$ and $q = 0$, respectively. This derivation further underpins the guarantees for CJPAGE in Appendix D.5, which differs only in the oracle used with probability $1 - q$ at Line 15.

Algorithm 5 Generalized algorithm

1: **With probability p:**
2: **With probability q:**
3: (1) $I_k = I$
4: **With probability $1 - q$:**
5: (2) $I_k \subset I$, $|I_k| = b$
6: **For** $m = 1, 2, \ldots, M$ **nodes in parallel do:**
7: $\Delta_{i,m}^k = \mathcal{Q}_{(\cdot)}\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right),\ i \in I_k$
8: $h_{i,m}^k = h_{i,m}^{k-1} + \Delta_{i,m}^k,\ i \in I_k$
9: $h_{i,m}^k = h_{i,m}^{k-1},\ i \in I \setminus I_k$
10: Send $\Delta_{i,m}^k$ to the server
11: **For** $i \in I_k$ **server do:**
12: $h_i^k = \frac{1}{M}\sum_{m=1}^M h_{i,m}^k$
13: Broadcast h_i^k to nodes
14: **For** $m = 1, 2, \ldots, M$ **nodes in parallel do:**
15: $\left[\begin{array}{l} (1)\hat{g}_m^k = \lambda\frac{1}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(h_i^k, x^k) \\ (2)\hat{g}_m^k = \frac{\lambda}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k) + \frac{\lambda}{b}\sum_{i\in I_k}\nabla_{x_m}\left(\mathcal{L}(h_i^k, x^k) - \mathcal{L}(h_i^{k-1}, x_m^k)\right)\end{array}\right.$
16: **With probability $1 - p$:**
17: **For** $m = 1, 2, \ldots, M; i \in I_k$ **nodes in parallel do:**
18: $\hat{g}_m^k = \hat{g}_m^{k-1}$
19: $h_{i,m}^k = h_{i,m}^{k-1}$
20: **For** $m = 1, 2, \ldots, M$ **nodes in parallel do:**
21: $g_m^k = \nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
22: $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

D.1 General inequalities and additional notation

Suppose $x, y, x_i \in \mathbb{R}^d$. Then, following inequality holds:

$$\langle x, y\rangle \leq \frac{\beta}{2}\|x\|^2 + \frac{1}{2\beta}\|y\|^2, \tag{Fen}$$

$$\begin{array}{ll}\|x+y\|^2 \leq (1+\alpha)\|x\|^2 + \left(1+\alpha^{-1}\right)\|y\|^2, & \\ \left\|\sum_{i=1}^n x_i\right\|^2 \quad \leqslant \quad n\sum_{i=1}^n \|x_i\|^2. & \end{array} \tag{CS}$$

We recall $x = (x_1^\top, x_2^\top, \ldots, x_M^\top)^\top$ - vector, which combines all local model states x_m. Hence, we can conclude $\|x\|^2 = \sum_{m=1}^M \|x_m\|^2$. Similarly $g = (g_1^\top, g_2^\top, \ldots, g_M^\top)^\top$. Then, we define $\overline{f}_i(x) = \frac{1}{M}\sum_{m=1}^M \hat{f}_i^m(x_m)$ and $\mathcal{L}\left(\overline{f}(x^{k+1})\right) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left(\overline{f}_i(x^{k+1})\right)$. Additionally, we will need the event indicators from Algorithm 5, with probabilities pq, $p(1-q)$ and $(1-p)$ from Lines 2, 5 and 16. Indicators of these events on the iteration k we denote as $\mathbb{I}_{pq}^k$, $\mathbb{I}_{p(1-q)}^k$ and $\mathbb{I}_{(1-p)}^k$ respectively, $I_p = \mathbb{I}_{p(1-q)}^k + \mathbb{I}_{pq}^k = 1 - \mathbb{I}_{1-p}^k$. We denote a σ-algebra generated by x^k and all previous variables (variables with iteration index less than k) as $\mathcal{F}^k$.

Definition D.1. Let $\|z\|_{bits}$ represents the amount of bits required to encode the vector $z \in \mathbb{R}^d$, c denotes the number of bits per floating point value, and d is the dimensionality of the problem (i.e., $cd = \|z\|_{bits}$). Then for compression operator $\mathcal{Q}$ we define the expected density of compressed

vector

$$q_\alpha = \frac{\mathbb{E}\|\mathcal{Q}(z)\|_{bits}}{cd}.$$

We proceed by proving Lemma D.1, which is central to the compression analysis in the federated ensembling setup.

Lemma D.1. *Let Assumptions 3.2 and 3.3 hold. Then function* $\nabla\mathcal{L}(a,y) = \frac{\partial\mathcal{L}}{\partial\overline{f}}\big|_{\overline{f}=a}\frac{\partial\overline{f}}{\partial x}\big|_{x=y}$ *is* $\frac{\tilde{L}_{\hat{f}}L_{\mathcal{L}}}{\sqrt{M}}$*-Lipschitz with respect to the first variable, i.e.*

$$\|\nabla\mathcal{L}(a,y) - \nabla\mathcal{L}(b,y)\| \leq \frac{\tilde{L}_{\hat{f}}L_{\mathcal{L}}}{\sqrt{M}}\|a-b\|.$$

Proof. We use explicitly the definition of $\nabla\mathcal{L}(\cdot,\cdot)$ and Assumption 3.3. Additionally we utilize Assumption 3.2 to bound $\left\|\frac{\partial\overline{f}}{\partial x}\Big|_{x=y}\right\|^2 \leq \sum_{m=1}^M \left\|\frac{1}{M}\frac{\partial\hat{f}_i}{\partial x_m}\Big|_{x_m=y_m}\right\|^2 \leq \frac{\tilde{L}_{\hat{f}}^2}{M}$.

$$\begin{aligned}
\|\nabla\mathcal{L}(a,y) - \nabla\mathcal{L}(b,y)\|^2 &= \left\|\frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=a}\frac{\partial\overline{f}}{\partial x}\Big|_{x=y} - \frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=b}\frac{\partial\overline{f}}{\partial x}\Big|_{x=y}\right\|^2 \\
&\leq \left\|\frac{\partial\overline{f}}{\partial x}\Big|_{x=y}\right\|^2\left\|\frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=a} - \frac{\partial\mathcal{L}}{\partial\overline{f}}\Big|_{\overline{f}=b}\right\|^2 \\
&\overset{\substack{3.2\\3.3}}{\leq} \frac{\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M}\|a-b\|^2.
\end{aligned}$$

□

D.2 Analysis of a generalized iteration

To initiate the analysis, we rely on the classic descent Lemma D.2.

Lemma D.2 (Lemma 2 [Li et al., 2021])**.** *Let function* F *is* L*-smooth and let* $x^{k+1} = x^k - \gamma G^k$*. Then for any* $G^t \in \mathbb{R}^d$ *and* $\gamma > 0$*, we have*

$$F(x^{k+1}) \leq F(x^k) - \frac{\gamma}{2}\|\nabla F(x^k)\|^2 - \left(\frac{1}{2\gamma} - \frac{L}{2}\right)\|x^k - x^{k+1}\|^2 + \frac{\gamma}{2}\|G^k - \nabla F(x^k)\|^2.$$

As shown in Lemma D.2, the key quantity in the analysis is the oracle distortion bound $\|G^k - \nabla F(x^k)\|^2$. In Theorem D.1 below, we establish a recursive relation for it.

Theorem D.1 (Oracle distortion bound)**.** *Let Assumptions 3.1–3.4 be satisfied. Then for Algorithm 5 the following oracle distortion bound holds:*

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{g}^{k+1} - \nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\Big|\mathcal{F}^{k+1}\right] \\
&\leq \left(1+\frac{p}{2}\right)(1-p)\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 \\
&\quad + \left(\left(1+\frac{2}{p}\right)(1-p)\lambda^2L_{joint}^2 + \frac{4p(1-q)\lambda^2L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4}{M^2b}(1+c_1^{-1})\right)\left\|x^{k+1}-x^k\right\|^2 \\
&\quad + \frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2bN}(1+c_1)\sum_{m=1}^M\sum_{i=1}^N\left\|\hat{f}_i(x_m^k) - h_{i,m}^k\right\|^2 \\
&\quad + \frac{\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}p(1-q)\sum_{i=1}^N\sum_{m=1}^M\mathbb{E}\left[\left\|\tilde{h}_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\right\|^2\Big|\mathcal{F}^{k+1}\right] \\
&\quad + \frac{\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^N\sum_{m=1}^M\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}\left\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\right\|^2\Big|\mathcal{F}^{k+1}\right].
\end{aligned} \tag{4}$$

Proof. We start evaluation of the oracle distortion term by the substitution of the explicit target function formulation and Line 21 of the Algorithm 5.

$$\sum_{m=1}^{M}\left\|\left(\hat{g}_m^{k+1}+g_m^{k+1}\right)-\nabla_{x_m}F(x^{k+1})\right\|^2=\sum_{m=1}^{M}\left\|\hat{g}_m^{k+1}-\nabla_{x_m}\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2$$
$$=\left\|\hat{g}^{k+1}-\nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2. \tag{5}$$

Then we consider the conditional expectation on $\mathcal{F}^{k+1}$.

$$\mathbb{E}\left[\left\|\hat{g}^{k+1}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\middle|\mathcal{F}^{k+1}\right]$$
$$=\mathbb{E}\left[\mathbb{I}_{1-p}^{k+1}\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\middle|\mathcal{F}^{k+1}\right]$$
$$+\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\middle|\mathcal{F}^{k+1}\right]$$
$$+\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^k,x^{k+1})+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right.\right.$$
$$\left.\left.-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\middle|\mathcal{F}^{k+1}\right]. \tag{6}$$

Let us examine (6) terms separately. Firstly, we note that g^k and x^{k+1} are $\mathcal{F}^{k+1}$ measurable and $\mathbb{I}_{1-p}^{k+1}$ is independent to it.

$$\mathbb{E}\left[\mathbb{I}_{1-p}^{k+1}\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2|\mathcal{F}^{k+1}\right]$$
$$=(1-p)\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2$$
$$\overset{\text{(CS)}}{\leq}(1+c_0)(1-p)\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k})\right)\right\|^2$$
$$+\left(1+c_0^{-1}\right)(1-p)\left\|\nabla\frac{\lambda}{N}\sum_{i=1}^{N}\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)-\nabla\frac{\lambda}{N}\sum_{i=1}^{N}\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k})\right)\right\|^2$$
$$\overset{3.4}{\leq}(1+c_0)(1-p)\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k})\right)\right\|^2+\left(1+c_0^{-1}\right)(1-p)\lambda^2L_{joint}^2\left\|x^{k+1}-x^k\right\|^2. \tag{7}$$

At the next step we work with the second term of (6).

$$\mathbb{I}_{pq}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)-\nabla\frac{\lambda}{N}\sum_{i=1}^{N}\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2$$
$$\overset{\text{(CS)}}{\leq}\frac{\mathbb{I}_{pq}^{k+1}\lambda^2}{N}\sum_{i=1}^{N}\left\|\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)-\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2$$
$$\overset{\text{(D.1)}}{\leq}\frac{\mathbb{I}_{pq}^{k+1}\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{NM}\sum_{i=1}^{N}\left\|\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1}-\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right\|^2$$
$$\overset{\text{(CS)}}{\leq}\frac{\mathbb{I}_{pq}^{k+1}\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\left\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2. \tag{8}$$

Now we continue with the last term of (6). We introduce the virtual sequence $\tilde{h}_{i,m}^k=h_{i,m}^{k-1}+\mathcal{Q}_2\left(\hat{f}_i(x_m^k)-h_{i,m}^{k-1}\right)$. Then, we add and subtract $\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)$ inside the

last term of (6).

$$
\begin{aligned}
&\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\\
&\quad-\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right]\\
&\quad\left.+\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\\
&=\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\\
&\quad\left.-\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right]\right\|^2\\
&\quad+\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\\
&\quad+\mathbb{I}_{p(1-q)}^{k+1}\left\langle\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\\
&\quad-\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right],\\
&\quad\left.\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\rangle. \qquad (9)
\end{aligned}
$$

Then we take expectation over batch I_{k+1}. We use that $\tilde{h}_{i,m}^{k+1}$ is independent of I_{k+1} as well as $\hat{f}_i(x_m^{k+1})$. It allows us to move the expectation insede the dot product and eliminate it in the (9) RHS. Hence, after conditioning on $\mathcal{F}^{k+1}$ we derive

$$
\begin{aligned}
&\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)+\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right.\right.\\
&\quad\left.\left.-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)\right]-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&=\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\right.\\
&\quad\left.-\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right]\right\|^2\\
&\qquad+\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\\
&\qquad\left.\left.-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]. \qquad (10)
\end{aligned}
$$

We continue by considering the first term of (10) RHS.

$$
\begin{aligned}
&\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right)\right.\right.\\
&\quad\left.\left.-\left(\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right)\right\|^{2}\Big|\mathcal{F}^{k+1}\right]\\
&\leq\mathbb{E}\left[\frac{\mathbb{I}_{p(1-q)}^{k+1}}{b^{2}}\sum_{j=1}^{b}\left\|\left(\lambda\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{j,m}^{k},x^{k+1}\right)-\lambda\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{j,m}^{k+1},x^{k+1}\right)\right)\right.\right.\\
&\quad\left.\left.-\left(\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right)\right\|^{2}\Big|\mathcal{F}^{k+1}\right]\\
&\leq\frac{1}{b}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\lambda\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{j,m}^{k},x^{k+1}\right)-\lambda\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{j,m}^{k+1},x^{k+1}\right)\right\|^{2}\Big|\mathcal{F}^{k+1}\right]\\
&\overset{\text{(D.1)}}{\leq}\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{Mb}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\frac{1}{M}\sum_{m=1}^{M}h_{j,m}^{k}-\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{j,m}^{k+1}\right\|^{2}\Big|\mathcal{F}^{k+1}\right]\\
&\overset{\text{(CS)}}{\leq}\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|h_{j,m}^{k}-\tilde{h}_{j,m}^{k+1}\right\|^{2}|\mathcal{F}^{k+1}\right].
\end{aligned}
\tag{11}
$$

We proceed by using the definition of virtual sequence $\tilde{h}_{i,m}^{k}=h_{i,m}^{k-1}+\mathcal{Q}_{2}\left(\hat{f}_{i}(x_{m}^{k})-h_{i,m}^{k-1}\right)$.

$$
\begin{aligned}
&\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|h_{j,m}^{k}-\tilde{h}_{j,m}^{k+1}\right\|^{2}|\mathcal{F}^{k+1}\right]\\
=&\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\mathcal{Q}_{2}\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}|\mathcal{F}^{k+1}\right]\\
=&\frac{\mathbb{I}_{p(1-q)}^{k+1}\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\mathcal{Q}_{2}\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\pm\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}|\mathcal{F}^{k+1}\right]\\
\overset{\text{(CS)}}{\leq}&\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\Bigg[2\mathbb{I}_{p(1-q)}^{k+1}\left\|\mathcal{Q}_{2}\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)-\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}\\
&+2\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}|\mathcal{F}^{k+1}\Bigg].
\end{aligned}
\tag{12}
$$

We note, that $\mathcal{Q}_2$ on iteration $k+1$ is independent to the $\mathcal{F}^{k+1}$. It enables us to take expectation over $\mathcal{Q}_2$ under the $\mathbb{E}[\cdot|\mathcal{F}^{k+1}]$ and use the Definition 3.1 of the contractive compressor.

$$
\begin{aligned}
&\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\Bigg[2\mathbb{I}_{p(1-q)}^{k+1}\left\|\mathcal{Q}_{2}\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)-\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}\\
&\quad+2\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}|\mathcal{F}^{k+1}\Bigg]\\
\leq&\frac{\lambda^{2}\tilde{L}_{\hat{f}}^{2}L_{\mathcal{L}}^{2}}{M^{2}b}\sum_{m=1}^{M}\mathbb{E}\Bigg[2\mathbb{I}_{p(1-q)}^{k+1}(1-\alpha)\left\|\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}\\
&\quad+2\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\hat{f}_{j}(x_{m}^{k+1})-h_{j,m}^{k}\right)\right\|^{2}|\mathcal{F}^{k+1}\Bigg]
\end{aligned}
$$

$$
\begin{aligned}
&\leq \frac{4\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 b}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\hat{f}_j(x_m^{k+1})-h_{j,m}^k\right)\right\|^2\left|\mathcal{F}^{k+1}\right.\right]\\
&=\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left\|\left(\hat{f}_i(x_m^{k+1})-h_{i,m}^k\right)\right\|^2\\
&=\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left\|\left(\hat{f}_i(x_m^{k+1})-h_{i,m}^k\pm\hat{f}_i(x_m^k)\right)\right\|^2\\
&\overset{\text{(CS)}}{\leq}\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left((1+c_1)\left\|\hat{f}_i(x_m^k)-h_{i,m}^k\right\|^2\right.\\
&\quad\left.+(1+c_1^{-1})\left\|\hat{f}_i(x_m^{k+1})-\hat{f}_i(x_m^k)\right\|^2\right)\\
&\overset{(3.2)}{\leq}\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left((1+c_1)\left\|\hat{f}_i(x_m^k)-h_{i,m}^k\right\|^2\right.\\
&\quad\left.+(1+c_1^{-1})\tilde{L}_{\hat{f}}^2\left\|x_m^{k+1}-x_m^k\right\|^2\right).
\end{aligned}\tag{13}
$$

Finally, (11), (12), and (13) provides the further bound

$$
\begin{aligned}
&\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\left(\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^k,x^{k+1}\right)-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right)\right.\right.\\
&\quad\left.\left.-\left(\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^k,x^{k+1}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)\right)\right\|^2\left|\mathcal{F}^{k+1}\right.\right]\\
&\leq\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left((1+c_1)\left\|\hat{f}_i(x_m^k)-h_{i,m}^k\right\|^2\right.\\
&\quad\left.+(1+c_1^{-1})\tilde{L}_{\hat{f}}^2\left\|x_m^{k+1}-x_m^k\right\|^2\right).
\end{aligned}\tag{14}
$$

The evaluation of the second term in (10) coincides with (8) up to indicator function. Using it along with (14) we transform (10) into

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^k,x^{k+1}\right)+\left[\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{h}_{i,m}^{k+1},x^{k+1}\right)\right.\right.\right.\\
&\quad\left.\left.\left.-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^k\right)\right]-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\right\|^2\left|\mathcal{F}^{k+1}\right.\right]\\
&\leq\frac{4p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left((1+c_1)\left\|\hat{f}_i(x_m^k)-h_{i,m}^k\right\|^2\right.\\
&\quad\left.+(1+c_1^{-1})\tilde{L}_{\hat{f}}^2\left\|x_m^{k+1}-x_m^k\right\|^2\right)\\
&\quad+\frac{\lambda^2\tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2\left|\mathcal{F}^{k+1}\right.\right].
\end{aligned}\tag{15}
$$

Since $\mathbb{I}^{k+1}_{p(1-q)}$ independent to $\tilde{h}^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)$ and $\mathcal{F}^{k+1}$ expectation in last term can be transformed into $p(1-q)\mathbb{E}\left[\left\|\tilde{h}^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\right\|^2 \Big| \mathcal{F}^{k+1}\right]$.

Finally, we substitute (7) with $c_0 = \frac{p}{2}$, (8) and (15) into (6).

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\hat{g}^{k+1} - \nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2 \Big| \mathcal{F}^{k+1}\right] \\
&\leq \left(1 + \frac{p}{2}\right)(1-p)\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 + \left(1 + \frac{2}{p}\right)(1-p)\lambda^2 L^2_{joint}\left\|x^{k+1} - x^k\right\|^2 \\
&\quad + \frac{\lambda^2 \tilde{L}^2_{\hat{f}} L^2_{\mathcal{L}}}{M^2 N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}^{k+1}_{pq}\left\|h^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\right\|^2 \Big| \mathcal{F}^{k+1}\right] \\
&\quad + \frac{4p(1-q)\lambda^2 \tilde{L}^2_{\hat{f}} L^2_{\mathcal{L}}}{M^2 bN}\sum_{m=1}^{M}\sum_{i=1}^{N}\Bigg((1+c_1)\left\|\hat{f}_i(x^k_m) - h^k_{i,m}\right\|^2 \\
&\quad + (1 + c_1^{-1})\tilde{L}^2_{\hat{f}}\left\|x^{k+1}_m - x^k_m\right\|^2\Bigg) \\
&\quad + \frac{\lambda^2 \tilde{L}^2_{\hat{f}} L^2_{\mathcal{L}}}{M^2 N} p(1-q)\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\tilde{h}^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\right\|^2 \Big| \mathcal{F}^{k+1}\right].
\end{aligned} \tag{16}
$$

By rearranging the terms in (16) and applying (5) we get the required result.

□

We now observe that, to complete the iteration analysis using Lemma D.2 and Theorem D.1, it remains to establish bounds on the perturbations of the logits $\left\|\hat{f}_i(x^k_m) - h^k_{i,m}\right\|^2$ induced by the compression scheme. In Theorem D.2 below, we derive recursive relations for these quantities, and completing the analysis of an iteration of Algorithm 5.

Theorem D.2. *Let Assumptions 3.1 - 3.5 be satisfied and*

$$
\gamma \leq \min\left\{\frac{1}{4L}; \frac{p}{2\sqrt{6}\lambda L_{joint}}; \frac{pM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{30\lambda L_{\mathcal{L}}\tilde{L}^2_{\hat{f}}}\right\}.
$$

Then for Algorithm 5 the following bound holds

$$
\begin{aligned}
\frac{\gamma}{2}\mathbb{E}\|\nabla F(x^k)\|^2 \leq \mathbb{E}\Bigg[&F(x^k) - F(x^{k+1}) + \frac{\gamma}{p}\Bigg(\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 \\
&- \left\|\hat{g}^{k+1} - \nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\Bigg) \\
&+ \frac{32\gamma\lambda^2 L^2_{\mathcal{L}}\tilde{L}^2_{\hat{f}}}{Mp\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)}\frac{1}{MN}\sum_{m=1}^{M}\sum_{i=1}^{N}\Bigg(\left\|h^k_{i,m} - \hat{f}_i(x^k_m)\right\|^2 \\
&- \|h^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\|^2\Bigg)\Bigg].
\end{aligned} \tag{17}
$$

Proof. In order to work further with the results of Theorem D.1, we need to obtain recursive relations for expressions of the form $\|h^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\|^2$ and $\|h^{k+1}_{i,m} - \hat{f}_i(x^{k+1}_m)\|^2$. We will use the definition of indicators and relation $\mathbb{I}^k_{p(1-q)} + \mathbb{I}^k_{pq} + \mathbb{I}^k_{1-p} = 1$.

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right]\\
&\overset{7,8}{=}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}\|h_{i,m}^{k}+\mathcal{Q}_1(\hat{f}_i(x_m^{k+1})-h_{i,m}^{k})-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right]\\
&\overset{3.1}{\leq}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}(1-\alpha_1)\|h_{i,m}^{k}-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right]\\
&=\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{pq}^{k+1}(1-\alpha_1)\|h_{i,m}^{k}-\hat{f}_i(x_m^{k+1})\pm\hat{f}_i(x_m^{k})\|^2|\mathcal{F}^{k+1}\right]\\
&\overset{\text{(CS)}}{\leq}\frac{(1-\alpha_1)}{N}\sum_{i=1}^{N}\mathbb{E}\Big[\mathbb{I}_{pq}^{k+1}\big\{(1+c_2)\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2\\
&\quad+(1+c_2^{-1})\|\hat{f}_i(x_m^{k+1})-\hat{f}_i(x_m^{k})\|^2\big\}\Big|\mathcal{F}^{k+1}\Big]\\
&\overset{3.2}{\leq}(1-\alpha_1)pq\sum_{i=1}^{N}\left(\frac{(1+c_2)}{N}\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2+(1+c_2^{-1})\tilde{L}_{\hat{f}}^2\|x_m^{k+1}-x_m^{k}\|^2\right). \quad (18)
\end{aligned}
$$

Then, we note $\mathbb{I}_{(1-p)}^{k+1}\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2=\mathbb{I}_{(1-p)}^{k+1}\|h_{i,m}^{k}-\hat{f}_i(x_m^{k+1})\|^2$. With the same usage of (CS) as in (18) it yields

$$
\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{(1-p)}^{k+1}\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right] \quad &\leq \quad (1-p)\sum_{i=1}^{N}\left(\frac{(1+c_3)}{N}\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2\right.\\
&\left.\qquad+(1+c_3^{-1})\tilde{L}_{\hat{f}}^2\|x_m^{k+1}-x_m^{k}\|^2\right). \quad (19)
\end{aligned}
$$

Finally, under the $\mathbb{I}_{p(1-q)}^{k+1}$ term $\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2$ equals to $\|h_{i,m}^{k}-\hat{f}_i(x_m^{k+1})\|^2$ with probability $\frac{|I\setminus I_{k+1}|}{|I|}=\frac{N-b}{N}$ and to $\|h_{i,m}^{k}+\mathcal{Q}(h_{i,m}^{k}-\hat{f}_i(x_m^{k+1}))-\hat{f}_i(x_m^{k+1})\|^2$ with probability $\frac{|I_{k+1}|}{|I|}=\frac{b}{N}$. It provides

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right]\\
&\leq p(1-q)\sum_{i=1}^{N}\left(\frac{N-b}{N^2}\Big[(1+c_5)\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2+N(1+c_5^{-1})\tilde{L}_{\hat{f}}^2\|x_m^{k+1}-x_m^{k}\|^2\Big]\right.\\
&\quad\left.+\frac{(1-\alpha_2)b}{N^2}\Big[(1+c_4)\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2+N(1+c_4^{-1})\tilde{L}_{\hat{f}}^2\|x_m^{k+1}-x_m^{k}\|^2\Big]\right). \quad (20)
\end{aligned}
$$

Reasoning in (18) also provides the relation on $\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2$.

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2|\mathcal{F}^{k+1}\right]\\
&\leq(1-\alpha_2)\sum_{i=1}^{N}\left(\frac{(1+c_6)}{N}\|\hat{f}_i(x_m^{k})-h_{i,m}^{k}\|^2+(1+c_6^{-1})\tilde{L}_{\hat{f}}^2\|x_m^{k+1}-x_m^{k}\|^2\right). \quad (21)
\end{aligned}
$$

Lemma D.2 in our notation along with equation (5) provides

$$F(x^{k+1}) \quad \leq \quad F(x^k) - \frac{\gamma}{2}\|\nabla F(x^k)\|^2 - \left(\frac{1}{2\gamma} - \frac{L}{2}\right)\|x^k - x^{k+1}\|^2 + \frac{\gamma}{2}\left\|\hat{g}^k - \nabla_x \mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2. \tag{22}$$

At the next step we summing $\mathbb{E}\big[(22) + \gamma A(4) + B(\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2)\frac{1}{M}\sum_{m=1}^{M}((18)+(19)+(20)) + \gamma A(\lambda^2 \tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2)\frac{1}{M^2}\sum_{m=1}^{M}(p(1-q)(21)+(18))\big]$:

$$\begin{aligned}
&\mathbb{E}\Bigg[\frac{\gamma}{2}\|\nabla F(x^k)\|^2 + F(x^{k+1}) - F(x^k) + \gamma A\left\|\hat{g}^{k+1} - \nabla_x \mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2 \\
&\quad + B(\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2)\frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\|^2\Bigg] \\
&\leq \mathbb{E}\Bigg[\gamma\left(A - \frac{Ap}{2} + \frac{1}{2}\right)\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 \\
&\quad + \Bigg(-\frac{1}{2\gamma} + \frac{L}{2} + \gamma A\Bigg(\left(1+\frac{2}{p}\right)(1-p)\lambda^2 L_{joint}^2 + \frac{4p(1-q)\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^4}{M^2 b}(1+c_1^{-1})\Bigg) \\
&\quad + \gamma\frac{A}{M^2}\lambda^2 \tilde{L}_{\hat{f}}^4 L_{\mathcal{L}}^2\left[p(1-q)(1-\alpha_2)(1+c_6^{-1}) + pq(1-\alpha_1)(1+c_2^{-1})\right] \\
&\quad + \frac{B}{M}(\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2)\Bigg((1-\alpha_1)pq(1+c_2^{-1})\tilde{L}_{\hat{f}}^2 + (1-p)(1+c_3^{-1})\tilde{L}_{\hat{f}}^2 \\
&\quad + p(1-q)\left(\frac{N-b}{N}(1+c_5^{-1})\tilde{L}_{\hat{f}}^2 + \frac{(1-\alpha_2)b}{N}(1+c_4^{-1})\tilde{L}_{\hat{f}}^2\right)\Bigg)\Bigg)\|x^k - x^{k+1}\|^2 \\
&\quad + \Bigg(\gamma A\frac{4p(1-q)\lambda^2 \tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 bN}(1+c_1) \\
&\quad + \gamma A\frac{\lambda^2 \tilde{L}_{\hat{f}}^2 L_{\mathcal{L}}^2}{M^2 N}\left[p(1-q)(1-\alpha_2)(1+c_6) + pq(1-\alpha_1)(1+c_2)\right] \\
&\quad + \frac{B}{M}(\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2)\Bigg(\frac{(1-\alpha_1)pq}{N}(1+c_2) + \frac{(1-p)}{N}(1+c_3) + \frac{p(1-q)}{N}\Bigg(\frac{N-b}{N}(1+c_5) \\
&\quad + \frac{(1-\alpha_2)b}{N}(1+c_4)\Bigg)\Bigg)\Bigg)\sum_{m=1}^{M}\sum_{i=1}^{N}\left\|\hat{f}_i(x_m^k) - h_{i,m}^k\right\|^2\Bigg].
\end{aligned} \tag{23}$$

Considering terms $\left\|\hat{g}^{k+1} - \nabla_x \mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2$ we choose $A = \frac{1}{p}$. To provide the recursion on $\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\|^2$ we pick $c_1 = c_6 = 1$, $c_2 = c_3 = c_4 = c_5 = \frac{p}{2}\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)$ and $B = \frac{32}{Mp\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)}$. To omit the $\|x^{k+1} - x^k\|$ term we bound

$$\gamma \leq \min\left\{\frac{1}{4L}; \frac{p}{2\sqrt{6}\lambda L_{joint}}; \frac{pM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{30\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}.$$

Hence, (23) transforms into

$$\begin{aligned}
&\frac{\gamma}{2}\mathbb{E}\|\nabla F(x^k)\|^2 \\
&\leq \mathbb{E}\Bigg[F(x^k) - F(x^{k+1}) + \frac{\gamma}{p}\Bigg(\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 - \left\|\hat{g}^{k+1} - \nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\Bigg)
\end{aligned}$$

$$+\frac{32\gamma\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{Mp\left(\alpha_1 q+\frac{\alpha_2 b}{N}(1-q)\right)}\frac{1}{MN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left(\left\|h_{i,m}^k-\hat{f}_i(x_m^k)\right\|^2-\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\|^2\right)\Bigg] \quad (24)$$

□

D.3 Guarantees for JEF21 and CJSAGA

Given Theorem D.2 we obtain bounds for JEF21 (Algorithm 1) and CJSAGA (Algorithm 3) by the substitution of $q=1$ and $q=0$ respectively.

Corollary D.1. *In setting of Theorem D.2. After K iterations of Algorithm 1 ($q=1$) and Algorithm 3 ($q=0$) the following bound holds:*

$$\mathbb{E}\|\nabla F(\overline{x}^K)\|^2 \quad \leq \quad \frac{2}{K\gamma}(F(x^0)-F^*), \quad (25)$$

Where $\overline{x}^K \sim U\{x^0,x^1,\ldots,x^{K-1}\}$.

Proof. We proceed by summing the result of Theorem D.2 - inequality (17) over k from 0 to $K-1$:

$$\frac{\gamma}{2}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla F(x^k)\|^2 \quad \leq \quad F(x^0)-F(x^{K-1})+\frac{\gamma}{p}\left\|\hat{g}^0-\nabla\mathcal{L}\left(\overline{f}(x^0)\right)\right\|^2$$
$$+\frac{32\gamma\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{Mp\left(\alpha_1 q+\frac{\alpha_2 b}{N}(1-q)\right)}\frac{1}{MN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left\|h_{i,m}^0-\hat{f}_i(x_m^0)\right\|^2. \quad (26)$$

Division by $\frac{\gamma K}{2}$, using assumption 3.5 and accounting $\left\|\hat{g}^0-\nabla\mathcal{L}\left(\overline{f}(x^0)\right)\right\|=\left\|h_{i,m}^0-\hat{f}_i(x_m^0)\right\|=0$ finishes the proof. □

Finally, we present the final bounds for JEF21 and CJSAGA in Corollaries 3.1 and 4.2 respectively.

Corollary 3.1 *Let Assumptions 3.1- 3.5 hold $\gamma=\min\left\{\frac{1}{4L};\frac{p}{2\sqrt{6}\lambda L_{joint}};\frac{Mp\alpha_1}{30\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}$. Then Algorithm 1 requires*

$$\mathcal{O}\left(\frac{1}{\varepsilon}\left[L+\frac{\lambda L_{joint}}{p}+\frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp\alpha_1}\right]\right)$$

iterations and

$$\mathcal{O}\left(\frac{1}{\varepsilon}\cdot N\left[p\alpha_1 L+\alpha_1\lambda L_{joint}+\frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{M}\right]\right)$$

bits communicated from nodes to the server to reach ε accuracy with respect to $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.

Proof. The result of Corollary D.1 directly provides the first bound. Then, we note that the design of JEF21 leads to transmission of a logits (states) compressed via operator with compression rate α_1 for joint dataset of the size N with probability p, and no communications otherwise. Additionally, for practically relevant compressors [Beznosikov et al., 2023] it holds $q_\alpha \leq \alpha$. Therefore, the second bound can be obtained from the first one by multiplying on additional factor of $apN\alpha_1$. We omit a constant a under the $\mathcal{O}$ notation. □

Corollary 4.2 *Let Assumptions 3.1- 3.5 hold and $\gamma=\min\left\{\frac{1}{4L};\frac{p}{2\sqrt{6}\lambda L_{joint}};\frac{Mp\frac{\alpha_2 b}{N}}{30\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}$. Then Algorithm 3 requires*

$$\mathcal{O}\left(\frac{1}{\varepsilon}\left[L+\frac{\lambda L_{joint}}{p}+\frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp\frac{\alpha_2 b}{N}}\right]\right)$$

iterations,

$$\mathcal{O}\left(\frac{1}{\varepsilon}\cdot pN\left[L+\frac{\lambda L_{joint}}{p}+\frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp\frac{\alpha_2 b}{N}}\right]\right)$$

joint objective oracle calls, and

$$\mathcal{O}\left(\frac{1}{\varepsilon}\cdot N\left[p\frac{\alpha_2 b}{N}L+\frac{\alpha_2 b}{N}\lambda L_{joint}+\frac{\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{M}\right]\right)$$

bits communicated from nodes to the server to reach ε accuracy with respect to $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.

D.4 Guarantees for JPAGE

In this section, we present the analysis of the JPAGE (Algorithm 2). On the one hand, it follows the high-level logic of Section D.2, with compression effects absent. On the other hand, it mirrors the analysis of PAGE [Li et al., 2021], while additionally accounting for the effects of local steps.

Theorem D.3. *Let Assumptions 3.1, 3.4 and 3.5 be satisfied. Then after K iterations of Algorithm 2 with $\gamma \leq \min\left\{\frac{1}{2L};\frac{pq}{4\lambda L_{joint}}\right\}$ the following bound holds:*

$$\mathbb{E}\|\nabla F(\overline{x}^K)\|^2 \quad \leq \quad \frac{2}{K\gamma}(F(x^0)-F^*), \tag{27}$$

Where $\overline{x}^K \sim U\{x^0,x^1,\ldots,x^{K-1}\}$.

Proof. We proceed with the oracle distortion bound - an analogue of the Theorem D.1. Equation (6) under the update rules of Algorithm 2 transforms into

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{g}^{k+1}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\right]\\
&=\mathbb{E}\left[\mathbb{I}_{1-p}^{k+1}\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\right]\\
&\quad+\mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla_{x_m}\left(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k})\right)\right)\right.\right.\\
&\qquad\left.\left.-\frac{\lambda}{N}\sum_{i=1}^N\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)\right\|^2\right]\\
&=(1-p)\mathbb{E}\left\|\hat{g}^{k}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\\
&\quad+p(1-q)\mathbb{E}\left\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla_{x_m}\left(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k})\right)\right)\right.\\
&\qquad\left.-\frac{\lambda}{N}\sum_{i=1}^N\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)\right\|^2.
\end{aligned}\tag{28}$$

Estimation (7) for the first term of (28) stays valid and the last term of (28) can be evaluated in the classic PAGE [Li et al., 2021] spirit.

$$\begin{aligned}
&\mathbb{E}_{I_{k+1}}\left\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\left(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k})\right)\right)\right.\\
&\quad\left.-\frac{\lambda}{N}\sum_{i=1}^N\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x_m^{k+1})\right)\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{I_{k+1}} \left\| \hat{g}^k - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right. \\
&\quad + \frac{\lambda}{b} \sum_{i \in I_{k+1}} \nabla \left(\mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^{k+1}) \right) - \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right) \\
&\quad \left. + \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^{k+1}) \right) \right\|^2 \\
&= \left\| \hat{g}^k - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right\|^2 \\
&\quad + \mathbb{E}_{I_{k+1}} \left\| \frac{\lambda}{b} \sum_{i \in I_{k+1}} \nabla \left(\mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^{k+1}) \right) - \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right) \right. \\
&\quad \left. + \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^{k+1}) \right) \right\|^2 \\
&\leq \left\| \hat{g}^k - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right\|^2 \\
&\quad + \frac{\lambda^2}{b} \mathbb{E}_i \left\| \nabla_x \mathcal{L}(\overline{f}_i(x^{k+1})) - \nabla_x \mathcal{L}(\overline{f}_i(x^k)) \right\|^2 \\
&\overset{3.4}{\leq} \left\| \hat{g}^k - \frac{\lambda}{N} \sum_{i=1}^{N} \nabla \mathcal{L} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{f}_i(x_m^k) \right) \right\|^2 + \frac{\lambda^2 L_{joint}^2}{b} \|x^{k+1} - x^k\|^2. \qquad (29)
\end{aligned}
$$

Using (7) with $c_0 = \frac{pq}{2}$ and (29) by taking full expectation we continue (28).

$$
\begin{aligned}
&\mathbb{E} \left[\left\| \hat{g}^{k+1} - \nabla \mathcal{L} \left(\overline{f}(x^{k+1}) \right) \right\|^2 \right] \\
&\leq \left[\left(1 + \frac{pq}{2} \right) (1-p) + p(1-q) \right] \mathbb{E} \left\| \hat{g}^k - \nabla \mathcal{L} \left(\overline{f}(x^k) \right) \right\|^2 \\
&\quad + \left[\left(1 + \frac{2}{pq} \right) (1-p) \lambda^2 L_{joint}^2 + \frac{\lambda^2 L_{joint}^2}{b} \right] \mathbb{E} \left\| x^{k+1} - x^k \right\|^2 \\
&\leq (1 - \frac{pq}{2}) \mathbb{E} \left\| \hat{g}^k - \nabla \mathcal{L} \left(\overline{f}(x^k) \right) \right\|^2 \\
&\quad + \left[\frac{4}{pq} \lambda^2 L_{joint}^2 \right] \mathbb{E} \left\| x^{k+1} - x^k \right\|^2. \qquad (30)
\end{aligned}
$$

Then we consider the sum $\mathbb{E}(22) + A\gamma(30)$:

$$
\begin{aligned}
&\frac{\gamma}{2} \mathbb{E} \|\nabla F(x^k)\|^2 + A\gamma \mathbb{E} \left\| \hat{g}^{k+1} - \nabla \mathcal{L} \left(\overline{f}(x^{k+1}) \right) \right\|^2 \leq \mathbb{E}(F(x^k) - F(x^{k+1})) \\
&\quad - \left(\frac{1}{2\gamma} - \frac{L}{2} - \frac{4A\gamma}{pq} \lambda^2 L_{joint}^2 \right) \mathbb{E} \|x^k - x^{k+1}\|^2 \\
&\quad + \left(\frac{\gamma}{2} + A\gamma(1 - \frac{pq}{2}) \right) \mathbb{E} \left\| \hat{g}^k - \nabla \mathcal{L} \left(\overline{f}(x^k) \right) \right\|^2. \qquad (31)
\end{aligned}
$$

We pick $A = \frac{1}{pq}$. Then, the choice $\gamma \leq \min \left\{ \frac{1}{2L}; \frac{pq}{4\lambda L_{joint}} \right\}$ justifies elimination of the $\|x^{k+1} - x^k\|$ term. Summing (31) over $k = 0, 1, \ldots, K-1$ and rearranging the terms we get:

$$\sum_{k=0}^{K-1} \mathbb{E}\|\nabla F(x^k)\|^2 \quad \leq \quad \frac{2}{\gamma}\left(F(x^0) - \mathbb{E}F(x^K)\right) + \frac{2}{\gamma}\left(\frac{\gamma}{2} + \frac{\gamma(1-pq/2)}{pq}\right)\mathbb{E}\left\|\hat{g}^0 - \nabla\mathcal{L}\left(\overline{f}(x^0)\right)\right\|^2. \tag{32}$$

Division by K and accounting $g^0 = \nabla\mathcal{L}\left(\overline{f}(x^0)\right)$ finishes the proof. □

Combining the results of Theorem D.3 with the design of the `JPAGE` algorithm directly yields Corollary 4.1.

Corollary 4.1 *Let Assumptions 3.1, 3.4 and 3.5 hold. Then Algorithm 2 with* $\gamma = \min\left\{\frac{1}{2L}; \frac{pq}{4\lambda L_{joint}}\right\}$ *requires*

$$\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{pq}\right]\right)$$

iterations and

$$\mathcal{O}\left(\frac{1}{\varepsilon}\cdot p\left(qN + (1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq}\right]\right)$$

joint objective oracle calls and bits communicated from nodes to the server to reach ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.

D.5 Analysis extension to CJPAGE

The substantial similarity between the design of `CJPAGE` and the generalized Algorithm 5 allows us to extend the result of Section D.2 to `CJPAGE` setup. More specifically, it suffices to revisit the oracle distortion bound due to the change in the oracle.

Theorem D.4. *Let Assumptions 3.1–3.5 be satisfied. Then after* K *iterations of Algorithm 4 with* $\gamma \leq \min\left\{\frac{1}{4L}; \frac{pq}{10\lambda L_{joint}}; \frac{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{50\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}$ *the following bound holds:*

$$\mathbb{E}\|\nabla F(\overline{x}^K)\|^2 \quad \leq \quad \frac{2}{K\gamma}(F(x^0) - F^*), \tag{33}$$

Where $\overline{x}^K \sim U\{x^0, x^1, \ldots, x^{K-1}\}$.

Proof. Algorithms 4 and 5 exhibit substantial similarity. In particular, they differ only in Lines 16 and 16 respectively. Consequently, the first part of the proof - deriving the oracle distortion bound - closely follows the proof of Theorem D.2 and requires only a modification of the analysis of the perturbation term, which arises with probability pq, corresponding to the last term in the analogue of (6):

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{g}^{k+1} - \nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2 \middle| \mathcal{F}^{k+1}\right] \\
&= \mathbb{E}\left[\mathbb{I}_{1-p}^{k+1}\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2 \middle| \mathcal{F}^{k+1}\right] \\
&\quad + \mathbb{E}\left[\mathbb{I}_{pq}^{k+1}\left\|\frac{\lambda}{N}\sum_{i=1}^N \nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h_{i,m}^{k+1}, x^{k+1}\right) - \frac{\lambda}{N}\sum_{i=1}^N \nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M \hat{f}_i(x_m^{k+1})\right)\right\|^2 \middle| \mathcal{F}^{k+1}\right] \\
&\quad + \mathbb{E}\left[\mathbb{I}_{p(1-q)}^{k+1}\left\|\hat{g}^k + \frac{\lambda}{b}\sum_{i\in I_{k+1}} \nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h_{i,m}^{k+1}, x^{k+1}\right)\right.\right. \\
&\quad \left.\left. - \frac{\lambda}{b}\sum_{i\in I_{k+1}} \nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h_{i,m}^{k}, x^{k}\right) - \frac{\lambda}{N}\sum_{i=1}^N \nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M \hat{f}_i(x_m^{k+1})\right)\right\|^2 \middle| \mathcal{F}^{k+1}\right].
\end{aligned} \tag{34}$$

We concentrate on the last term of (34) and begin with addition and subtraction of $\frac{\lambda}{b}\sum_{i\in I_{k+1}}\left[\nabla_x\mathcal{L}(\overline{f}_i(x^{k+1}))-\nabla_x\mathcal{L}(\overline{f}_i(x^k))\right]$ along with utilization of (CS):

$$
\begin{aligned}
&\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\Bigg\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k+1}_{i,m},x^{k+1}\right)\\
&\quad-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k}_{i,m},x^{k}\right)-\frac{\lambda}{N}\sum_{i=1}^N\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)\Bigg\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&\leq(1+c_8)p(1-q)\mathbb{E}\Bigg[\Bigg\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k}_m)\right)\Bigg)\\
&\quad-\frac{\lambda}{N}\sum_{i=1}^N\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)\Bigg\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&\quad+(1+c_8^{-1})\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\frac{\lambda^2}{b^2}\Bigg\|\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k+1}_{i,m},x^{k+1}\right)\\
&\quad+\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k}_{i,m},x^{k}\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k}_m)\right)\Bigg)\Bigg\|^2\Bigg|\mathcal{F}^{k+1}\Bigg].
\end{aligned}
\tag{35}
$$

We note that the first term was analyzed in Theorem D.3 and corresponds to the classic distortion term for PAGE-type [Li et al., 2021] algorithms. While the second term arises due to compression. We concentrate on it and note that most of the necessary analysis has already been conducted in Theorem D.1.

$$
\begin{aligned}
&\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\frac{\lambda^2}{b^2}\Bigg\|\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k+1}_{i,m},x^{k+1}\right)\\
&\quad+\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k}_{i,m},x^{k}\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k}_m)\right)\Bigg)\Bigg\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&\overset{\text{(CS)}}{\leq}2\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\frac{\lambda^2}{b^2}\Bigg\|\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k+1}_m)\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k+1}_{i,m},x^{k+1}\right)\Bigg)\Bigg\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&\quad+2p(1-q)\mathbb{E}_{I_{k+1}}\Bigg[\frac{\lambda^2}{b^2}\Bigg\|\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M h^{k}_{i,m},x^{k}\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^M\hat{f}_i(x^{k}_m)\right)\Bigg)\Bigg\|^2\Bigg]\\
&\overset{(8)}{\leq}2\mathbb{E}\Bigg[\frac{\mathbb{I}^{k+1}_{p(1-q)}\lambda^2\tilde{L}^2_{\hat{f}}L^2_{\mathcal{L}}}{M^2b}\sum_{i\in I_{k+1}}\sum_{m=1}^M\left\|h^{k+1}_{i,m}-\hat{f}_i(x^{k+1}_m)\right\|^2\Bigg|\mathcal{F}^{k+1}\Bigg]\\
&\quad+2p(1-q)\mathbb{E}_{I_{k+1}}\Bigg[\frac{\lambda^2\tilde{L}^2_{\hat{f}}L^2_{\mathcal{L}}}{M^2b}\sum_{i\in I_{k+1}}\sum_{m=1}^M\left\|h^{k}_{i,m}-\hat{f}_i(x^{k}_m)\right\|^2\Bigg].
\end{aligned}
\tag{36}
$$

Then, we note $\mathbb{E}_{I_{k+1}}\left[\frac{1}{b}\sum_{i\in I_{k+1}}\left\|h^k_{i,m}-\hat{f}_i(x^k_m)\right\|^2\right]=\frac{1}{N}\sum_{i=1}^N\left\|h^k_{i,m}-\hat{f}_i(x^k_m)\right\|^2$. Similarly, recalling the definition of the virtual sequence $\tilde{h}^k_{i,m}=h^{k-1}_{i,m}+\mathcal{Q}_2\left(\hat{f}_i(x^k_m)-h^{k-1}_{i,m}\right)$, we conclude $\mathbb{E}_{I_{k+1}}\left[\frac{\mathbb{I}^{k+1}_{p(1-q)}}{b}\sum_{i\in I_{k+1}}\left\|h^{k+1}_{i,m}-\hat{f}_i(x^{k+1}_m)\right\|^2\right]=\frac{p(1-q)}{N}\sum_{i=1}^N\left\|\tilde{h}^{k+1}_{i,m}-\hat{f}_i(x^{k+1}_m)\right\|^2$. Utilizing it and taking full expectation over (36) we derive:

$$\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\frac{\lambda^2}{b^2}\Bigg\|\sum_{i\in I_{k+1}}\nabla\Bigg(\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)$$
$$+\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k}\right)-\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k})\right)\Bigg)\Bigg\|^2\Bigg]$$
$$\leq\frac{2p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2+\left\|h_{i,m}^{k}-\hat{f}_i(x_m^{k})\right\|^2\right]. \quad (37)$$

Using (29) and (37) we transform (35) into:

$$\mathbb{E}\Bigg[\mathbb{I}^{k+1}_{p(1-q)}\Bigg\|\hat{g}^k+\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k+1},x^{k+1}\right)$$
$$-\frac{\lambda}{b}\sum_{i\in I_{k+1}}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}h_{i,m}^{k},x^{k}\right)-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k+1})\right)\Bigg\|^2\Bigg]$$
$$\leq(1+c_8)p(1-q)\mathbb{E}\left[\left\|\hat{g}^k-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{f}_i(x_m^{k})\right)\right\|^2+\frac{\lambda^2L_{joint}^2}{b}\|x^{k+1}-x^k\|^2\right]$$
$$+(1+c_8^{-1})\frac{2p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2+\left\|h_{i,m}^{k}-\hat{f}_i(x_m^{k})\right\|^2\right]. \quad (38)$$

In order to obtain oracle distortion bound we substitute (7) with $c_0=pq/4$, (8) and (38) with $c_8=\frac{q}{2}$ into (34) and rearrange the terms.

$$\mathbb{E}\left[\left\|\hat{g}^{k+1}-\nabla\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\right]$$
$$\leq\left[\left(1+\frac{pq}{4}\right)(1-p)+\left(1+\frac{q}{2}\right)p(1-q)\right]\mathbb{E}\left\|\hat{g}^k-\nabla\mathcal{L}\left(\overline{f}(x^{k})\right)\right\|^2$$
$$+\left[\left(1+\frac{4}{pq}\right)(1-p)\lambda^2L_{joint}^2+\left(1+\frac{q}{2}\right)p(1-q)\frac{\lambda^2L_{joint}^2}{b}\right]\mathbb{E}\left\|x^{k+1}-x^k\right\|^2$$
$$+\frac{\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}^{k+1}_{pq}\left\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2\right]$$
$$+\left(1+\frac{2}{q}\right)\frac{2p(1-q)\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2+\left\|h_{i,m}^{k}-\hat{f}_i(x_m^{k})\right\|^2\right]$$
$$\leq\left(1-\frac{pq}{4}\right)\mathbb{E}\left\|\hat{g}^k-\nabla\mathcal{L}\left(\overline{f}(x^{k})\right)\right\|^2+\frac{5}{pq}\lambda^2L_{joint}^2\mathbb{E}\left\|x^{k+1}-x^k\right\|^2$$
$$+\frac{\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\mathbb{I}^{k+1}_{pq}\left\|h_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2\right]$$
$$+\frac{6p}{q}\frac{\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\left[\left\|\tilde{h}_{i,m}^{k+1}-\hat{f}_i(x_m^{k+1})\right\|^2+\left\|h_{i,m}^{k}-\hat{f}_i(x_m^{k})\right\|^2\right]. \quad (39)$$

Once the oracle distortion bound established we note, that inequalities (22), (18), (19), (20) and (21) remain valid. Finally, we consider the following sum $\mathbb{E}\big[(22)+\gamma A(39)+B(\gamma\lambda^2L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2)\frac{1}{M}\sum_{m=1}^{M}((18)+(19)+(20))+\gamma A(\lambda^2\tilde{L}_{\hat{f}}^2L_{\mathcal{L}}^2)\frac{1}{M^2}\sum_{m=1}^{M}\left(\frac{6p}{q}(21)+(18)'\right)\big]$, where

(18)$'$ denotes (18) with c_2 replaced by independent constant c_7.

$$\begin{aligned}
&\mathbb{E}\Bigg[\frac{\gamma}{2}\|\nabla F(x^k)\|^2 + F(x^{k+1}) - F(x^k) + \gamma A\left\|\hat{g}^{k+1} - \nabla_x \mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2 \\
&\quad + B(\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2)\frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\|^2\Bigg] \\
&\leq \mathbb{E}\Bigg[\gamma\left(\frac{1}{2} + A - A\frac{pq}{4}\right)\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 \\
&\quad + \Bigg(-\frac{1}{2\gamma} + \frac{L}{2} + \gamma A\frac{5}{pq}\lambda^2 L_{joint}^2 \\
&\quad + \gamma\frac{A}{M^2}\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4\left[\frac{6p}{q}(1-\alpha_2)(1+c_6^{-1}) + pq(1-\alpha_1)(1+c_2^{-1})\right] \\
&\quad + \gamma\frac{B}{M}\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4\Bigg((1-\alpha_1)pq(1+c_2^{-1}) + (1-p)(1+c_3^{-1}) \\
&\quad + p(1-q)\Bigg(\frac{N-b}{N}(1+c_5^{-1}) + \frac{(1-\alpha_2)b}{N}(1+c_4^{-1})\Bigg)\Bigg)\Bigg)\|x^k - x^{k+1}\|^2 \\
&\quad + \Bigg(\gamma A\frac{6p}{q}\frac{\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{M^2N} + \gamma\frac{A}{M^2N}\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2\left[\frac{6p}{q}(1-\alpha_2)(1+c_6) + pq(1-\alpha_1)(1+c_2)\right] \\
&\quad + \gamma\frac{B}{M}\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2\Bigg(\frac{(1-\alpha_1)pq}{N}(1+c_2) + \frac{(1-p)}{N}(1+c_3) \\
&\quad + \frac{p(1-q)}{N}\left(\frac{N-b}{N}(1+c_5) + \frac{(1-\alpha_2)b}{N}(1+c_4)\right)\Bigg)\Bigg)\sum_{m=1}^{M}\sum_{i=1}^{N}\left\|\hat{f}_i(x_m^k) - h_{i,m}^k\right\|^2\Bigg]. \quad (40)
\end{aligned}$$

Considering terms $\left\|\hat{g}^{k+1} - \nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2$ we choose $A = \frac{2}{pq}$. To provide the recursion on $\|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\|^2$ we pick $c_6 = c_7 = 1$, $c_2 = c_3 = c_4 = c_5 = \frac{p}{2}\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)$ and $B = \frac{128}{Mpq^2\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)}$. To omit the $\|x^{k+1} - x^k\|$ term we bound

$$\gamma \leq \min\left\{\frac{1}{4L}; \frac{pq}{10\lambda L_{joint}}; \frac{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{50\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}\right\}.$$

Hence, (40) transforms into

$$\begin{aligned}
\frac{\gamma}{2}\mathbb{E}\|\nabla F(x^k)\|^2 \leq \mathbb{E}\Bigg[&F(x^k) - F(x^{k+1}) + \frac{2\gamma}{pq}\Bigg(\left\|\hat{g}^k - \nabla\mathcal{L}\left(\overline{f}(x^k)\right)\right\|^2 \\
&\qquad - \left\|\hat{g}^{k+1} - \nabla_x\mathcal{L}\left(\overline{f}(x^{k+1})\right)\right\|^2\Bigg) \\
&+ \frac{128\gamma\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{Mpq^2\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)}\frac{1}{MN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left(\left\|h_{i,m}^k - \hat{f}_i(x_m^k)\right\|^2 - \|h_{i,m}^{k+1} - \hat{f}_i(x_m^{k+1})\|^2\right)\Bigg]. \quad (41)
\end{aligned}$$

Summing (41) over $k = 0, 1, \ldots, K-1$ and using $\left\|\hat{g}^0 - \nabla\mathcal{L}\left(\overline{f}(x^0)\right)\right\| = \left\|h_{i,m}^0 - \hat{f}_i(x_m^0)\right\| = 0$ yields:

$$\frac{\gamma}{2}\mathbb{E}\sum_{k=0}^{K-1}\|\nabla F(x^k)\|^2 \quad \leq \quad \mathbb{E}\Bigg[F(x^0) - F(x^{K-1}) - \frac{2\gamma}{pq}\left\|\hat{g}^K - \nabla\mathcal{L}\left(\overline{f}(x^K)\right)\right\|^2$$

$$
\begin{aligned}
&- \frac{128\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2}{Mpq^2\left(\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right)} \frac{1}{MN} \sum_{m=1}^{M} \sum_{i=1}^{N} \left\| h_{i,m}^K - \hat{f}_i(x_m^K) \right\|^2 \Bigg] \\
&\overset{3.5}{\leq} \quad F(x^0) - F^*.
\end{aligned}
$$

Division by $\frac{\gamma K}{2}$ finishes the proof. □

Corollary 4.3 *Let Assumptions 3.1- 3.5 hold and* $\gamma = \min\left\{\frac{1}{4L}; \frac{pq}{10\lambda L_{joint}}; \frac{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}{50\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}\right\}$. *Then Algorithm 4 requires*

$$
\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)
$$

iterations,

$$
\mathcal{O}\left(\frac{1}{\varepsilon} \cdot p\left(qN + (1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)
$$

joint objective oracle calls, and

$$
\mathcal{O}\left(\frac{1}{\varepsilon} \cdot p\left(\alpha_1 qN + \alpha_2(1-q)b\right)\left[L + \frac{\lambda L_{joint}}{pq} + \frac{\lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2}{pqM\left[\alpha_1 q + \frac{\alpha_2 b}{N}(1-q)\right]}\right]\right)
$$

bits communicated from nodes to the server to reach ε *accuracy with respect to* $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.

Proof. The first bound follows directly from Theorem D.4. The second estimation is obtained by accounting for the probability p of computing and communicating the joint component, where the full dataset is evaluated with additional probability q and a mini-batch of size b with probability $1-q$. Finally, observing that we employ a compression operator with rate α_1 when transmitting all logits (states) and rate α_2 for partial transmissions yields the final bit rate. □

E Recovering PAGE-Style Oracle Complexity with Local Steps

Section 4 introduced JPAGE (Algorithm 2), which combines PAGE updates with local steps. Its oracle complexity in Corollary 4.1 contains a term linear in N. By contrast, in standard PAGE, the choice $b \sim \sqrt{N}$ and $q \sim \frac{1}{\sqrt{N}}$ yields the optimal $\sqrt{N}$ dependence [Li et al., 2021].

The obstruction is visible in (30): applying Young's inequality with $c_0 = \frac{pq}{2}$ to the drift accumulated during steps without a joint update contributes $\frac{4\lambda^2 L_{joint}^2}{pq}$ to the coefficient of $\mathbb{E}\|x^{k+1} - x^k\|^2$, independently of b. This extra step is absent from the original PAGE analysis. It forces $\gamma = O\Big(\frac{pq}{\lambda L_{joint}}\Big)$, so increasing the mini-batch does not restore the PAGE rate.

To see this lag directly, suppose that iteration k takes a local step. The joint estimator remains $\hat{g}_m^k = \hat{g}_m^{k-1}$. If the mini-batch event (2) on Line 6 of Algorithm 2 occurs at $k+1$, its update is

$$\hat{g}_m^{k+1} = \hat{g}_m^k + \frac{\lambda}{b} \sum_{i \in I_{k+1}} \big[\nabla_{x_m} \mathcal{L}(\overline{f}_i(x^{k+1})) - \nabla_{x_m} \mathcal{L}(\overline{f}_i(x^k))\big].$$

If $\hat{g}_m^{k-1}$ was last refreshed at x^{k-1}, the subtraction at x^k misses the intervening local step. In this one-step case, the correction should change

$$\frac{\lambda}{b} \sum_{i \in I_{k+1}} \big[\nabla_{x_m} \mathcal{L}(\overline{f}_i(x^{k+1})) - \nabla_{x_m} \mathcal{L}(\overline{f}_i(x^k))\big] \longrightarrow \frac{\lambda}{b} \sum_{i \in I_{k+1}} \big[\nabla_{x_m} \mathcal{L}(\overline{f}_i(x^{k+1})) - \nabla_{x_m} \mathcal{L}(\overline{f}_i(x^{k-1}))\big].$$

For any number of consecutive local steps, the relevant reference is the state ω^k of the most recent joint update. We therefore modify the reference state in the mini-batch correction on Line 14 of Algorithm 2, corresponding to Line 14 of Algorithm 6. The changes relative to Algorithm 2 are shown in green. As in JPAGE, initialize $\hat{g}_m^0 = \frac{\lambda}{N} \sum_{i=1}^N \nabla_{x_m} \mathcal{L}(\overline{f}_i(x^0))$, compute g_m^0 on the local data, and set $x_m^1 = x_m^0 - \gamma(\hat{g}_m^0 + g_m^0)$. Additionally, we set $\omega^0 = x^0$.

Algorithm 6 JPAGE*

1: **For** $k = 1, 2, \ldots, K$ **do:**
2: **With probability p:**
3: **With probability q:**
4: (1) $I_k = I$
5: **With probability $1 - q$:**
6: (2) $I_k \subset I$, $|I_k| = b$
7: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
8: Compute $\hat{f}_i(x_m^k)$ and, in case (2), $\hat{f}_i(\omega_m^{k-1})$ for $i \in I_k$ and send it to the server
9: **For** $i \in I_k$ **server do:**
10: $\overline{f}_i(x^k) = \frac{1}{M} \sum_{m=1}^M \hat{f}_i(x_m^k)$
11: (2) $\overline{f}_i(\omega^{k-1}) = \frac{1}{M} \sum_{m=1}^M \hat{f}_i(\omega_m^{k-1})$
12: Broadcast logits to nodes
13: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
14: $\begin{bmatrix} (1)\hat{g}_m^k = \lambda \frac{1}{N} \sum_{i=1}^N \nabla_{x_m} \mathcal{L}(\overline{f}_i(x^k)) \\ (2)\hat{g}_m^k = \hat{g}_m^{k-1} + \frac{\lambda}{b} \sum_{i \in I_k} \Big[\nabla_{x_m} \mathcal{L}(\overline{f}_i(x^k)) - \nabla_{\omega_m} \mathcal{L}(\overline{f}_i(\omega^{k-1}))\Big] \end{bmatrix}$
15: $\omega^k = x^k$
16: **With probability $1 - p$:**
17: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
18: $\hat{g}_m^k = \hat{g}_m^{k-1}$
19: $\omega^k = \omega^{k-1}$
20: **For** $m = 1, 2, \ldots, M$ **in parallel do:**
21: $g_m^k = \nabla_{x_m} \frac{1}{N_m} \sum_{i_m=1}^{N_m} \mathcal{L}_{i_m}\big(f_{i_m}(x_m^k)\big)$
22: $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

Theorem E.1 establishes convergence of JPAGE* and shows that the revised correction permits tighter control of the joint-oracle distortion.

Theorem E.1. *Let Assumptions 3.1, 3.4, and 3.5 hold. For $p, q \in (0,1]$, $b \in \{1,\dots,N\}$, and*

$$\gamma \le \min\left\{\frac{1}{2L},\ \frac{p}{2\sqrt{5}\lambda L_{joint}\sqrt{1+\frac{2(1-q)}{qb}}}\right\}, \tag{42}$$

Algorithm 6 satisfies

$$\mathbb{E}\|\nabla F(\overline{x}^K)\|^2 \le \frac{2(F(x^0)-F^*)}{K\gamma}, \qquad \overline{x}^K \sim U\{x^0,\dots,x^{K-1}\}. \tag{43}$$

Proof. The descent inequality (22) applies with oracle error $\hat{g}^k - \lambda\nabla F_{joint}(x^k)$. Assumption 3.4 gives

$$\|\hat{g}^k - \lambda\nabla F_{joint}(x^k)\|^2 \le 2\|\hat{g}^k - \lambda\nabla F_{joint}(\omega^k)\|^2 + 2\lambda^2 L_{joint}^2\|x^k-\omega^k\|^2. \tag{44}$$

Unlike the correction in (29), the mini-batch correction in Algorithm 6 compares x^{k+1} with ω^k. It is unbiased for $\lambda[\nabla F_{joint}(x^{k+1}) - \nabla F_{joint}(\omega^k)]$. Applying the same sampling variance argument as in (29), and using per-observation smoothness, yields

$$\begin{aligned}\mathbb{E}\|\hat{g}^{k+1} - \lambda\nabla F_{joint}(\omega^{k+1})\|^2 &\le (1-pq)\mathbb{E}\|\hat{g}^k - \lambda\nabla F_{joint}(\omega^k)\|^2 \\ &\quad + \frac{p(1-q)\lambda^2 L_{joint}^2}{b}\mathbb{E}\|x^{k+1}-\omega^k\|^2.\end{aligned} \tag{45}$$

On a joint update, $\omega^{k+1} = x^{k+1}$, otherwise $\omega^{k+1} = \omega^k$. Thus the reference-point drift obeys

$$\begin{aligned}\mathbb{E}\|x^{k+1}-\omega^{k+1}\|^2 &= (1-p)\mathbb{E}\|x^{k+1}-\omega^k\|^2 \\ &\le \left(1-\frac{p}{2}\right)\mathbb{E}\|x^k-\omega^k\|^2 + \frac{2}{p}\mathbb{E}\|x^{k+1}-x^k\|^2.\end{aligned} \tag{46}$$

The decomposition in (44) replaces the oracle error at x^k by the error at ω^k and isolates $\|x^k-\omega^k\|^2$, to which we apply Young's inequality in (46) with $c = \frac{p}{2(1-p)}$ for $p<1$ (the case $p=1$ is immediate), rather than $c_0 = \frac{pq}{2}$ used for the oracle error in (7). This yields a contraction of $1-\frac{p}{2}$ for the reference-point drift, instead of the $1-\frac{pq}{2}$ contraction in (30).

As in the proof of Theorem D.3, we combine the descent inequality with the oracle recursion (45) and the drift recursion (46). Weight the latter two by $A = \frac{\gamma}{pq}$ and $B = \frac{2\gamma\lambda^2 L_{joint}^2}{p}\left(1+\frac{2(1-q)}{qb}\right)$, respectively. By (44), the coefficients of the oracle error and the reference-point drift then telescope. The remaining coefficient of $\mathbb{E}\|x^{k+1}-x^k\|^2$ is at most

$$-\left(\frac{1}{2\gamma}-\frac{L}{2}\right) + \frac{2\gamma\lambda^2 L_{joint}^2(1-q)}{qb} + \frac{2B}{p} \le -\frac{1}{4\gamma} + \frac{5\gamma\lambda^2 L_{joint}^2}{p^2}\left(1+\frac{2(1-q)}{qb}\right) \le 0,$$

where the last inequality follows from (42). Summing over k, using $\hat{g}^0 = \lambda\nabla F_{joint}(x^0)$ and $\omega^0 = x^0$, and applying Assumption 3.5 proves (43). □

Theorem E.1 yields the following iteration, oracle, and communication bounds.

Corollary E.1. *Let Assumptions 3.1, 3.4, and 3.5 hold. Then Algorithm 6 with $b = \lceil\sqrt{N}\rceil$, $q = \frac{b}{N}$, and γ at the upper bound in (42) requires $\mathcal{O}\Big(\frac{1}{\varepsilon}\Big(L + \frac{\lambda L_{joint}}{p}\Big)\Big)$ iterations and $\mathcal{O}\Big(\frac{1}{\varepsilon}\sqrt{N}\,(pL+\lambda L_{joint})\Big)$ joint objective oracle calls and bits communicated from each client to the server to reach ε accuracy with respect to $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$.*

Proof. By (43), $K = \mathcal{O}\Big(\frac{1}{\gamma\varepsilon}\Big)$ iterations suffice. As in Corollary 4.1, we count joint-oracle calls and communication over these iterations. A full refresh evaluates N shared samples and sends N logit vectors with probability pq, a mini-batch correction evaluates its b samples at two states and sends $2b$ logit vectors with probability $p(1-q)$. Hence both costs per client are bounded by

$$\mathcal{O}\left(\frac{p\big(qN+2(1-q)b\big)}{\gamma\varepsilon}\right).$$

With $b = \lceil \sqrt{N} \rceil$ and $q = \frac{b}{N}$, the step size in (42) satisfies

$$\frac{1}{\gamma} = \max\left\{2L,\ \frac{2\sqrt{5}\lambda L_{joint}}{p}\sqrt{1 + \frac{2(N-b)}{b^2}}\right\} \leq 2L + \frac{2\sqrt{15}\lambda L_{joint}}{p}.$$

Since $qN = b$ and $b \leq 2\sqrt{N}$, substituting these bounds gives the stated complexities. □

Compared with Corollary 4.1, both joint objective oracle calls and client-to-server communication recover a $\sqrt{N}$ dependence.

F Restoring Mini-Batch Gains in CJPAGE

The local-step obstruction identified for JPAGE also appears in CJPAGE (Algorithm 4): its oracle distortion bound (39) contains a drift term proportional to $1/(pq)$, independently of b. Compression adds a second difficulty. In the mini-batch branch, only the states h_i with $i \in I^k$ change, whereas the correction in Line 16 weights their change by $1/b$. This leaves an additional compression-dependent error.

Let ω^k denote that reference point, as in Section E. In the mini-batch branch, the revised estimator is

$$\begin{aligned}\hat{g}_m^k = \hat{g}_m^{k-1} &+ \frac{\lambda}{N}\sum_{i\in I^k}\left[\nabla_{x_m}\mathcal{L}(h_i^k, x^k) - \nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k)\right] \\ &+ \frac{\lambda}{b}\sum_{i\in I^k}\left[\nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k) - \nabla_{\omega_m}\mathcal{L}(h_i^{k-1}, \omega^{k-1})\right].\end{aligned}$$

The first sum is the exact change in the full gradient induced by the updated compressed states (since $h_i^k = h_i^{k-1}$ for $i \notin I^k$, $\sum_{i\in I^k}\nabla_{x_m}[\mathcal{L}(h_i^k, x^k) - \mathcal{L}(h_i^{k-1}, x^k)] = \sum_{i=1}^N \nabla_{x_m}[\mathcal{L}(h_i^k, x^k) - \mathcal{L}(h_i^{k-1}, x^k)]$). The second is an unbiased estimate of the change caused by moving the model from ω^{k-1} to x^k. Their distinct weights, $1/N$ and $1/b$, preclude combining the corrections into a single term. Relative to CJPAGE, this requires one additional computation on the sampled mini-batch ($\nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k)$ for $i \in I^k$) but no additional client-to-server messages. As in CJPAGE, we initialize $h_{i,m}^0 = \hat{f}_i(x_m^0)$ and $\hat{g}_m^0 = \frac{\lambda}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(h_i^0, x^0)$, and set $\omega^0 = x^0$. Algorithm 7 shows the complete update, with changes relative to Algorithm 4 in green.

Algorithm 7 CJPAGE*

For $k = 1, 2, \dots, K$ **do:**
  With probability p:
    With probability q:
      (1) $I^k = I$
    With probability $1 - q$:
      (2) $I^k \subset I$, $|I^k| = b$
    For $m = 1, 2, \dots, M$ **in parallel do:**
      $\Delta_{i,m}^k = \mathcal{Q}_{(\cdot)}\left(\hat{f}_i(x_m^k) - h_{i,m}^{k-1}\right), i \in I^k$
      $h_{i,m}^k = h_{i,m}^{k-1} + \Delta_{i,m}^k,\ i \in I^k$
      $h_{i,m}^k = h_{i,m}^{k-1},\ i \in I \setminus I^k$
      Send $\Delta_{i,m}^k$ to the server
    For $i \in I^k$ **server do:**
      $h_i^k = h_i^{k-1} + \frac{1}{M}\sum_{m=1}^M \Delta_{i,m}^k$
      Broadcast h_i^k to nodes
    For $m = 1, 2, \dots, M$ **in parallel do:**
$$\left[\begin{array}{l} (1)\hat{g}_m^k = \lambda\frac{1}{N}\sum_{i=1}^N \nabla_{x_m}\mathcal{L}(h_i^k, x^k) \\ (2)\hat{g}_m^k = \hat{g}_m^{k-1} + \frac{\lambda}{N}\sum_{i\in I^k}\left[\nabla_{x_m}\mathcal{L}(h_i^k, x^k) - \nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k)\right] \\ \quad + \frac{\lambda}{b}\sum_{i\in I^k}\left[\nabla_{x_m}\mathcal{L}(h_i^{k-1}, x^k) - \nabla_{\omega_m}\mathcal{L}(h_i^{k-1}, \omega^{k-1})\right] \end{array}\right.$$
      $\omega^k = x^k$
  With probability $1 - p$:
    For $m = 1, 2, \dots, M$ **in parallel do:**
      $\hat{g}_m^k = \hat{g}_m^{k-1}$
      $h_{i,m}^k = h_{i,m}^{k-1}$
      $\omega^k = \omega^{k-1}$
  For $m = 1, 2, \dots, M$ **in parallel do:**
    $g_m^k = \nabla_{x_m}\frac{1}{N_m}\sum_{i_m=1}^{N_m}\mathcal{L}_{i_m}\left(f_{i_m}(x_m^k)\right)$
    $x_m^{k+1} = x_m^k - \gamma(\hat{g}_m^k + g_m^k)$

The following result shows that the revised correction controls both reference-point drift and compression error, with mini-batch variance governed by qb rather than an isolated q factor.

Theorem F.1. *Let Assumptions 3.1–3.5 hold. For $p, q \in (0, 1]$ and $b \in \{1, \dots, N\}$, suppose the step size satisfies*

$$\gamma \le \min\left\{\frac{1}{2L},\ \frac{p}{2\sqrt{5}\lambda}\left[\left(\frac{3}{2} + \frac{9(1-q)}{qb}\right)L_{joint}^2 + \frac{L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4}{M^2}\left(\frac{18(1-q)}{qb} + \frac{12\left(1 + \frac{9(1-q)}{qb}\right)}{\left(q\alpha_1 + (1-q)\frac{b}{N}\alpha_2\right)^2}\right)\right]^{-1/2}\right\}. \tag{47}$$

Then for Algorithm 7 holds

$$\mathbb{E}\|\nabla F(\overline{x}^K)\|^2 \le \frac{2(F(x^0) - F^*)}{K\gamma}, \qquad \overline{x}^K \sim U\{x^0, \dots, x^{K-1}\}. \tag{48}$$

Proof. To obtain an oracle distortion bound analogous to (39) we add and subtract the full compressed-state gradient at ω^{k+1} and the exact joint gradient at ω^{k+1}:

$$\begin{aligned} \hat{g}^{k+1} - \lambda\nabla F_{joint}(x^{k+1}) = &\left[\hat{g}^{k+1} - \frac{\lambda}{N}\sum_{i=1}^N \nabla\mathcal{L}(h_i^{k+1}, \omega^{k+1})\right] \\ &+ \frac{\lambda}{N}\sum_{i=1}^N\left[\nabla\mathcal{L}(h_i^{k+1}, \omega^{k+1}) - \nabla\mathcal{L}(\overline{f}_i(\omega^{k+1}), \omega^{k+1})\right] \\ &+ \lambda\left[\nabla F_{joint}(\omega^{k+1}) - \nabla F_{joint}(x^{k+1})\right]. \end{aligned} \tag{49}$$

Squaring this three-term sum with Cauchy–Schwarz, then bounding its second term by Lemma D.1 and its third by Assumption 3.4, gives

$$\begin{aligned}\|\hat{g}^{k+1} - \lambda\nabla F_{joint}(x^{k+1})\|^2 \le 3\left\|\hat{g}^{k+1} - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^{k+1},\omega^{k+1})\right\|^2 \\ + \frac{3\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{M^2N}\sum_{i=1}^{N}\sum_{m=1}^{M}\|h_{i,m}^{k+1} - \hat{f}_i(\omega_m^{k+1})\|^2 \\ + 3\lambda^2 L_{joint}^2\|x^{k+1} - \omega^{k+1}\|^2.\end{aligned}$$

The reference-point drift $\mathbb{E}\|x^{k+1} - \omega^{k+1}\|^2$ is controlled exactly as in (46). We now analyze the first term in (49). On the mini-batch branch, substituting the update of $\hat{g}^{k+1}$ from Algorithm 7, this term equals

$$\begin{aligned}&\hat{g}^k + \frac{\lambda}{N}\sum_{i\in I^{k+1}}\left[\nabla\mathcal{L}(h_i^{k+1},x^{k+1}) - \nabla\mathcal{L}(h_i^k,x^{k+1})\right] \\ &\quad + \frac{\lambda}{b}\sum_{i\in I^{k+1}}\left[\nabla\mathcal{L}(h_i^k,x^{k+1}) - \nabla\mathcal{L}(h_i^k,\omega^k)\right] - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^{k+1},x^{k+1}).\end{aligned}$$

Since $h_i^{k+1} = h_i^k$ for $i \notin I^{k+1}$, the second and fourth terms combine into $-\frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^k,x^{k+1})$. Adding and subtracting the full compressed-state gradient at ω^k gives

$$\begin{aligned}\hat{g}^{k+1} - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^{k+1},\omega^{k+1}) = \hat{g}^k - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^k,\omega^k) \\ + \left(\frac{\lambda}{b}\sum_{i\in I^{k+1}} - \frac{\lambda}{N}\sum_{i=1}^{N}\right)\left[\nabla\mathcal{L}(h_i^k,x^{k+1}) - \nabla\mathcal{L}(h_i^k,\omega^k)\right].\end{aligned} \tag{50}$$

On a full refresh (probability pq), the first term in (49) vanishes, a local step leaves it unchanged, on a mini-batch update, applying the `PAGE` variance argument of (29) and (45) to (50) yields

$$\begin{aligned}\mathbb{E}\left\|\hat{g}^{k+1} - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^{k+1},\omega^{k+1})\right\|^2 \le (1-pq)\mathbb{E}\left\|\hat{g}^k - \frac{\lambda}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(h_i^k,\omega^k)\right\|^2 \\ + \frac{9p(1-q)\lambda^2 L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^2}{M^2Nb}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\|h_{i,m}^k - \hat{f}_i(\omega_m^k)\|^2 \\ + \frac{3p(1-q)\lambda^2}{b}\left(L_{joint}^2 + \frac{2L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4}{M^2}\right)\mathbb{E}\|x^{k+1} - \omega^k\|^2.\end{aligned} \tag{51}$$

The compressed-logit states follow the contractive-update arguments of (18) and (20), using ω^k as the reference point. On a local step, both h and ω stay fixed. Young's inequality and Assumption 3.2 then give

$$\begin{aligned}&\frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\|h_{i,m}^{k+1} - \hat{f}_i(\omega_m^{k+1})\|^2 \\ &\le \left[1 - \frac{p}{2}\left(q\alpha_1 + (1-q)\frac{b}{N}\alpha_2\right)\right]\frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbb{E}\|h_{i,m}^k - \hat{f}_i(\omega_m^k)\|^2 \\ &\quad + \frac{2p\tilde{L}_{\hat{f}}^2}{M\left(q\alpha_1 + (1-q)\frac{b}{N}\alpha_2\right)}\mathbb{E}\|x^{k+1} - \omega^k\|^2.\end{aligned} \tag{52}$$

All required recursions are now in place. In the weighted sum we use

$$
\begin{aligned}
(22): &\quad 1,\\
(51): &\quad \frac{3\gamma}{2pq},\\
(52): &\quad \frac{3\gamma\lambda^2 L_{\mathcal{L}}^2 \tilde{L}_{\hat{f}}^2}{Mp\left(q\alpha_1 + (1-q)\frac{b}{N}\alpha_2\right)}\left(1 + \frac{9(1-q)}{qb}\right),\\
(46): &\quad \frac{2\gamma\lambda^2}{p}\left[\left(\frac{3}{2} + \frac{9(1-q)}{qb}\right)L_{joint}^2 + \frac{L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4}{M^2}\left(\frac{18(1-q)}{qb} + \frac{12\left(1+\frac{9(1-q)}{qb}\right)}{\left(q\alpha_1 + (1-q)\frac{b}{N}\alpha_2\right)^2}\right)\right].
\end{aligned}
$$

With the step size in (47), this sum cancels the oracle, compressed-logit, and reference-point errors and leaves a nonpositive coefficient of $\mathbb{E}\|x^{k+1} - x^k\|^2$. The initial errors vanish by $h_{i,m}^0 = \hat{f}_i(x_m^0)$, $\omega^0 = x^0$, and $\hat{g}^0 = \lambda\nabla F_{joint}(x^0)$. Summing over k and applying Assumption 3.5 proves (48). □

The theorem yields the following bounds under the same per-iteration accounting as Corollary 4.3.

Corollary F.1. *Let Assumptions 3.1–3.5 hold with $N \geq 3$. Set $b = \lceil\sqrt{N}\rceil$, $q = \frac{b}{N}$, and choose γ at the upper bound in* (47)*. Then Algorithm 7 reaches ε accuracy with respect to $\mathbb{E}\|\nabla F(\overline{x}^K)\|^2$ in*

$$
\mathcal{O}\left(\frac{1}{\varepsilon}\left[L + \frac{\lambda L_{joint}}{p} + \frac{\sqrt{N}\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mp(\alpha_1+\alpha_2)}\right]\right)
$$

iterations, using

$$
\mathcal{O}\left(\frac{1}{\varepsilon}\left[\sqrt{N}(pL + \lambda L_{joint}) + \frac{N\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{M(\alpha_1+\alpha_2)}\right]\right)
$$

joint objective oracle calls per client and

$$
\mathcal{O}\left(\frac{1}{\varepsilon}\left[\sqrt{N}(\alpha_1+\alpha_2)(pL + \lambda L_{joint}) + \frac{N\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{M}\right]\right)
$$

bits communicated from each client to the server.

Proof. By (48), $K = \mathcal{O}(\frac{1}{\gamma\varepsilon})$. A full refresh uses N joint-sample evaluations and sends N compressed states, and a mini-batch correction evaluates three joint-objective gradients per selected observation and sends one compressed state. Thus the oracle and uplink costs over K iterations are, respectively,

$$
\mathcal{O}\left(\frac{p(qN + 3(1-q)b)}{\gamma\varepsilon}\right), \qquad \mathcal{O}\left(\frac{p(\alpha_1 qN + \alpha_2(1-q)b)}{\gamma\varepsilon}\right).
$$

For the stated b and q, we have $qb \geq 1$. The bracketed expression in (47) is at most $\frac{21}{2}L_{joint}^2 + \frac{138N^2L_{\mathcal{L}}^2\tilde{L}_{\hat{f}}^4}{M^2b^2[\alpha_1+(1-\frac{b}{N})\alpha_2]^2}$. Hence the step-size condition gives $\gamma^{-1} = \mathcal{O}\left(L + \frac{\lambda L_{joint}}{p} + \frac{N\lambda L_{\mathcal{L}}\tilde{L}_{\hat{f}}^2}{Mpb[\alpha_1+(1-\frac{b}{N})\alpha_2]}\right)$. Substituting $qN = b \leq 2\sqrt{N}$ and $\alpha_1 qN + \alpha_2(1-q)b = b[\alpha_1 + (1-\frac{b}{N})\alpha_2]$ and applying $1 - b/N \geq 1/3$ for $N \geq 3$ gives the three bounds. □

Compared with Corollary 4.3, the L- and L_{joint}-dependent terms in both the joint-oracle and uplink bounds scale as $\sqrt{N}$. In contrast, the compression-induced terms retain a linear dependence on N. The reason is additional $\sqrt{N}$ term induced by effective compression rate. Indeed, for $b \sim q^{-1} \sim \sqrt{N}$ and fixed α_1, α_2, the effective compression rate $q\alpha_1 + (1-q)b\alpha_2/N$ scales as $N^{-1/2}$.

G Comparing Iteration and Communication Complexity

For a concise comparison, let $A = pL + \lambda L_{joint}$, $D = \lambda L_{\mathcal{L}} \tilde{L}_{\hat{f}}^2 / M$, and $\alpha_{\text{eff}} = q\alpha_1 + (1-q)b\alpha_2/N$. Table 9 compares iteration and uplink bounds for all six methods. The displayed terms follow Corollaries 3.1, 4.1, 4.2, 4.3, E.1, and F.1. For JPAGE* and CJPAGE*, we set $b = \lceil \sqrt{N} \rceil$ and $q = b/N$. The original JPAGE and CJPAGE have less tight bounds than their respective modifications and are listed below the dashed line.

Table 9: Iteration and per-client uplink bounds for ε accuracy.

Method	Iterations	Client-to-server bits
JEF21	$\mathcal{O}\left(\frac{1}{p\varepsilon}\left[A + \frac{D}{\alpha_1}\right]\right)$	$\mathcal{O}\left(\frac{N}{\varepsilon}\left[\alpha_1 A + D\right]\right)$
CJSAGA	$\mathcal{O}\left(\frac{1}{p\varepsilon}\left[A + \frac{ND}{\alpha_2 b}\right]\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\left[\alpha_2 bA + ND\right]\right)$
JPAGE*	$\mathcal{O}\left(\frac{A}{p\varepsilon}\right)$	$\mathcal{O}\left(\frac{\sqrt{N}}{\varepsilon}A\right)$
CJPAGE*	$\mathcal{O}\left(\frac{1}{p\varepsilon}\left[A + \frac{\sqrt{N}D}{\alpha_1+\alpha_2}\right]\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\left[\sqrt{N}(\alpha_1+\alpha_2)A + ND\right]\right)$
JPAGE	$\mathcal{O}\left(\frac{1}{p\varepsilon}\left[pL + \frac{\lambda L_{joint}}{q}\right]\right)$	$\mathcal{O}\left(\frac{qN+(1-q)b}{\varepsilon}\left[pL + \frac{\lambda L_{joint}}{q}\right]\right)$
CJPAGE	$\mathcal{O}\left(\frac{1}{p\varepsilon}\left[pL + \frac{\lambda L_{joint}}{q} + \frac{D}{q\alpha_{\text{eff}}}\right]\right)$	$\mathcal{O}\left(\frac{\alpha_1 qN+\alpha_2(1-q)b}{\varepsilon}\left[pL + \frac{\lambda L_{joint}}{q} + \frac{D}{q\alpha_{\text{eff}}}\right]\right)$

For a hyperparameter (p, q, b, α) selection one need to know the cost of local oracle call and communicated bit of information. Then, minimization of total cost will yield the optimal hyperparameter choice. This accounting must also include the cost of processing local observations, which varies with client dataset sizes and computational capacities. In case when communication dominates, decrease p until $pL \lesssim \lambda L_{joint}$ to balance the terms in A and, when feasible, choose $\alpha_1 \sim D/A$ for JEF21, $\alpha_2 b \sim ND/A$ for CJSAGA, and $\alpha_1 + \alpha_2 \sim \sqrt{N}D/A$ for CJPAGE*. Finally, theoretical bounds do not fully capture the practical benefits of local steps and contractive compression [Richtárik et al., 2021, Beznosikov et al., 2023, Luo et al., 2025]. Together with unknown problem constants, this makes hyperparameter selection an empirical question.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract claims four algorithms with convergence guarantees for communication-efficient federated ensembling. All four algorithms and their convergence guarantees are presented in the main text, with detailed proofs in Appendix D. Experiments support the empirical claims.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 7.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete proof?

   Answer: [Yes]

   Justification: Assumptions 3.1–3.5 are stated explicitly. Full proofs for all four algorithms are provided in Appendix D, with lemmas and corollaries properly numbered and cross-referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all information needed to reproduce the main experimental results?

   Answer: [Yes]

   Justification: Experimental setup, hyperparameters, and computational resources are described in Appendix C. No code is released.

5. **Open access to data and code**

   Question: Does the paper provide open access to data and code?

   Answer: [No]

   Justification: No code repository is provided. CIFAR-10 is publicly available, and implementation details are described in Appendix C.1 to allow reproduction.

6. **Experimental setting/details**

   Question: Does the paper specify all training and test details necessary to understand the results?

   Answer: [Yes]

   Justification: Data partitioning (Dirichlet α=0.5), client/shared dataset sizes, validation splits, and per-baseline hyperparameter configurations are all reported in Appendix C.

7. **Experiment statistical significance**

   Question: Does the paper report error bars or other information about statistical significance?

   Answer: [Yes]

   Justification: Figures 1, 2, 4, 5, and 6 show convergence trajectories. Tables 5, 6, 7, and 8 report variability across seeds.

8. **Experiments compute resources**

   Question: Does the paper provide sufficient information on compute resources?

   Answer: [Yes]

   Justification: Hardware and runtime details are reported in Appendix C.1.

9. **Code of ethics**

Question: Does the research conform with the NeurIPS Code of Ethics?

Answer: [Yes]

Justification: The work is foundational optimization research on public benchmark data (CIFAR-10) and poses no foreseeable ethical concerns.

10. **Broader impacts**

    Question: Does the paper discuss potential positive and negative societal impacts?

    Answer: [N/A]

    Justification: The paper presents theoretical algorithms for distributed optimization. There is no direct path to negative societal impact, and no broader-impact section is required for this type of foundational methods work.

11. **Safeguards**

    Question: Does the paper describe safeguards for responsible release of high-risk data or models?

    Answer: [N/A]

    Justification: No new datasets, pre-trained models, or high-risk assets are released.

12. **Licenses for existing assets**

    Question: Are existing assets properly credited with licenses mentioned?

    Answer: [No]

    Justification: All datasets and model architectures used in the experiments are cited, but their licenses are not specified.

13. **New assets**

    Question: Are new assets introduced in the paper well documented?

    Answer: [N/A]

    Justification: No new datasets, models, or code packages are released as part of this submission.

14. **Crowdsourcing and research with human subjects**

    Question: Does the paper include details about crowdsourcing or human subjects research?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals**

    Question: Does the paper describe IRB approvals for research with human subjects?

    Answer: [N/A]

    Justification: No human subjects are involved in this research.

16. **Declaration of LLM usage**

    Question: Does the paper describe usage of LLMs as an important component of the core methods?

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology. Any use was limited to writing assistance and does not affect scientific content or originality.