Loss Gap Parity for Fairness in Heterogeneous Federated Learning

Brahim Erraji[1], Michaël Perrot[1], and Aurélien Bellet[2]

[1]Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
[2]PreMeDICaL team, Inria, Idesp, Inserm, Université de Montpellier

Abstract

While clients may join federated learning to improve performance on data they rarely observe locally, they often remain self-interested, expecting the global model to perform well on their own data. This motivates an objective that ensures all clients achieve a similar *loss gap*—the difference in performance between the global model and the best model they could train using only their local data. To this end, we propose EAGLE, a novel federated learning algorithm that explicitly regularizes the global model to minimize disparities in loss gaps across clients. Our approach is particularly effective in heterogeneous settings, where the optimal local models of the clients may be misaligned. Unlike existing methods that encourage loss parity, potentially degrading performance for many clients, EAGLE targets fairness in relative improvements. We provide theoretical convergence guarantees for EAGLE under non-convex loss functions, and characterize how its iterates perform relative to the standard federated learning objective using a novel heterogeneity measure. Empirically, we demonstrate that EAGLE reduces the disparity in loss gaps among clients by prioritizing those furthest from their local optimal loss, while maintaining competitive utility in both convex and non-convex cases compared to strong baselines.

1 Introduction

Federated learning enables the training of machine learning models based on data stored across multiple clients without accessing or sharing it directly (McMahan et al., 2017). It has emerged as a promising training framework for applications with regulated data access such as healthcare (Long et al., 2020), banking (Rieke et al., 2020), and more (Li et al., 2020a). Federated learning is typically orchestrated by a centralized, often trusted server responsible for aggregating local client models and synchronizing updates. This is the setting we consider in our work.

Federated optimization algorithms typically aim to minimize a (weighted) average of the local client training losses by generating a sequence of models based on information received from the clients, such as the gradient of the loss function on their respective data. Clients participate in federated learning by contributing their resources in exchange for access to knowledge aggregated from other clients, enabling the training of a model that generalizes well to the overall data distribution. However, they often also expect the model to perform well on their own local data distribution. For example, consider a medical diagnosis system in which each hospital seeks a model that generalizes to rare or unseen patient cases, while still maintaining high performance on the types of patients commonly encountered in its region. In this context, a model that exhibits uneven performance across datasets from different hospitals can be viewed as unfair. In fact, Federated Averaging (FedAvg)—introduced by McMahan et al. (2017) and now the default algorithm in federated learning—often rewards clients that possess larger datasets, regardless of the relevance of their data to the overall utility. This happens because these clients are (i) typically weighted by their data size during aggregation; and (ii) allowed to run more local

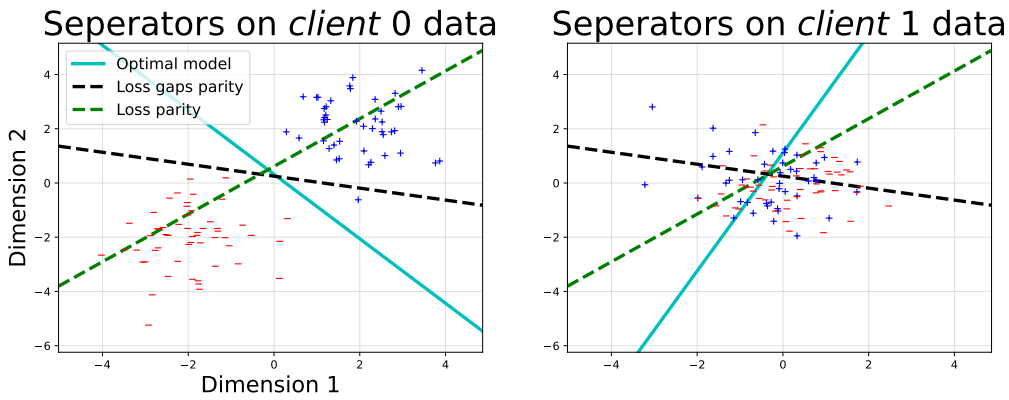


Figure 1: In this example, we use the *AFL* algorithm to enforce loss parity between the two clients, and our approach, *EAGLE*, to enforce loss gap parity. Enforcing loss parity favors *client* 1, which has noisier data and a more complex prediction task, resulting in poorer performance for *client* 0. In contrast, optimizing for loss gap parity handles this imbalance more effectively.

steps, since with a fixed batch size larger datasets require more updates per epoch (Wang et al., 2020), steering the global optimization toward their minimizers. These factors raise concerns about unfairness toward other clients, potentially discouraging their participation. As a result, promoting fairness has become a fundamental requirement in federated learning.

Similar to algorithmic fairness in centralized machine learning, determining what makes a federated model fair remains subject to debate and often depends on the specific application (Shi et al., 2023). A widely studied notion of fairness in federated learning is *loss parity* (Li et al., 2020b, 2021; Yue et al., 2023), which aims to equalize local losses between all participating clients. The underlying intuition is that similar loss values indicate the model has learned each client's data distribution equally well. However, this implicitly assumes that losses are directly comparable across clients—a condition that often fails in heterogeneous settings. In practice, the data distributions of some clients may be intrinsically more challenging to model or suffer from lower quality or resolution, causing their losses to be on inherently different scales. Enforcing equalized losses in such cases can distort optimization and reduce the utility of the global model, sometimes causing a *leveling down* effect—where fairness degrades outcomes for the best-off groups without benefiting the worst-off (Mittelstadt et al., 2023; Zietlow et al., 2022; Maheshwari et al., 2023). We illustrate this with a simple example involving two clients with different data distributions and a linear predictor in Figure 1. Similar problems can arise with *min-max* fairness (Mohri et al., 2019a), which optimizes for the worst-off client but may still compromise overall global performance.

To address these issues, we adopt a fairness objective for federated learning centered on encouraging *loss gap parity* across clients. The loss gap of a client is defined as the difference between the loss of the global model and the lowest loss achievable using only the local data of that client. We consider a global model fair when these gaps are similar across clients, ensuring that all clients benefit relatively equally from collaboration. To achieve this, we introduce EAGLE, a novel algorithm that augments FedAvg with a regularization term explicitly penalizing the variance of client loss gaps. By tuning the regularization weight, EAGLE allows one to trade off overall accuracy against fairness, consistently reducing disparities in relative improvements compared to optimal local model of each client. Theoretically, we provide convergence guarantees for EAGLE under non-convex local losses, showing that it reaches a saddle point of the regularized objective. We further connect these guarantees to the standard federated learning objective via a novel heterogeneity measure, highlighting how greater client heterogeneity can reduce global utility when enforcing fairness. Empirically, we demonstrate the effectiveness of EAGLE on synthetic data as well as the EMNIST and DirtyMNIST benchmarks, with both linear and

convolutional models. Our results show that EAGLE consistently reduces the variance in loss gaps across clients while maintaining comparable or improved overall performance relative to FedAvg and strong loss parity and min-max fairness baselines.

Related work. Several approaches in the literature define fairness in federated learning through client losses, either by reducing disparities in losses across clients or by optimizing for the worst-case loss (Li et al., 2020b, 2021; Mohri et al., 2019a). The motivation is that a fairer model should perform evenly across clients. However, in heterogeneous settings where clients have different optimal local losses and misaligned optimal models, defining fairness solely based on absolute losses can undesirably penalize clients with easier data, effectively forcing them to perform as poorly as clients with noisier or more complex data. To the best of our knowledge, FOCUS (Chu et al., 2023) is the first work to explicitly question this prevailing notion of fairness and to propose *loss gap parity* as a more suitable objective in heterogeneous settings. Their method builds personalized models for clusters of clients and evaluates fairness by measuring the loss gap of each client relative to its cluster model. However, their theoretical guarantees rely on very strong assumptions on the data distributions, limiting their applicability. In contrast, our approach is more general: it directly regularizes loss gap variance across all clients and provides explicit control over the fairness–utility trade-off via a tunable parameter λ. In a parallel work, Carey and Wu (2025) propose a unified framework for fairness in federated learning, introducing the notion of *desert fairness*, which aims to equalize performance in proportion to the optimal local loss of each client. While their objective is related to ours, their approach again offers limited flexibility in balancing fairness and utility, and lacks formal theoretical guarantees.

2 Fair Federated Learning: From Loss Parity To Loss Gap Parity

In this paper, we consider a cross-silo federated learning setup with K clients collaborating on a supervised learning task. Let $\mathcal{X}$ denote the feature space and $\mathcal{Y}$ the label space. Each client k has access to a local dataset $D_k = (x_k^i, y_k^i)_{i=1}^{n_k}$ of n_k examples in $\mathcal{X} \times \mathcal{Y}$. We consider models $\theta : \mathcal{X} \to \mathcal{Y}$ in a hypothesis space $\mathcal{H}$ equipped with a norm $\|\cdot\|$.

The general goal is to learn a global model that performs well across the datasets of all clients. Let $\ell(\theta, x, y)$ denote a loss function measuring the discrepancy between the prediction of a model $\theta(x)$ and the true label y. The standard federated learning objective is then

$$L(\theta) := \frac{1}{K}\sum_{k=1}^{K} L_k(\theta), \text{ with } L_k(\theta) := \frac{1}{n_k}\sum_{i=1}^{n_k} \ell(\theta, x_k^i, y_k^i),$$

which corresponds to minimizing the sum of local empirical losses $L_1, \ldots, L_K$.[1]

Optimizing the sum of local losses can lead to significant performance differences across clients when their data distributions are heterogeneous. Data heterogeneity occurs when clients' data differ in their feature distributions, label distributions, or the conditional relationship between features and labels. This variability poses a fundamental challenge in federated learning (Kairouz et al., 2021), calling into question whether collaboration truly benefits individual clients (Cho et al., 2022).

To address the uneven client performance caused by heterogeneity, a common strategy is to enforce some form of *loss parity* (Li et al., 2020b, 2021), aiming for a model that achieves roughly equal local losses across all clients. Here, we consider a definition of approximate loss parity based on the variance of the local losses.

Definition 1 (ϵ-Loss parity)**.** *A model $\theta \in \mathcal{H}$ satisfies ϵ-loss parity if the variance of the client losses $L_k(\theta)$ across the K clients is at most ϵ, that is:*

$$\mathbb{V}\left(\{L_k(\theta)\}_{k=1}^{K}\right) \leq \epsilon. \tag{1}$$

[1]While arbitrary weights can be applied to the clients, we consider uniform weights here for simplicity.

where $\mathbb{V}$ denotes the sample variance.

By directly comparing local losses across clients, loss parity implicitly assumes that losses are on the same scale. In heterogeneous settings, however, this assumption often fails, as some clients may have noisier or intrinsically harder-to-predict data. In such cases, achieving loss parity may require to drastically reduce the utility of the global model on best-off clients. In particular, for any model θ that satisfies exact loss parity, i.e., $\mathbb{V}\left(\{L_k(\theta)\}_{k=1}^K\right) = 0$, we have: $\forall k \in [K] : L_k(\theta) \geq \max_{k' \in [K]} L_{k'}^*$ where $L_k^* := \min_{\theta \in \mathcal{H}} L_k(\theta)$. Consequently, clients with lower optimal losses must give up performance to achieve loss parity.

To address this limitation, we adopt in this work a more suitable notion of fairness that focuses on equalizing *loss gaps* relative to the optimal local loss of each client.

Definition 2 (ϵ-Loss Gap Parity, extended from Chu et al. 2023)**.** *A model $\theta \in \mathcal{H}$ satisfies ϵ-loss gap parity if the variance of loss gaps $r_k(\theta) := L_k(\theta) - L_k^*$ across the K clients is at most ϵ, that is:*

$$\mathbb{V}\left(\{r_k(\theta)\}_{k=1}^K\right) \leq \epsilon. \tag{2}$$

Unlike loss parity, loss gap parity promotes fairness in *relative* performance. For instance, under exact loss gap parity (with $\epsilon = 0$),

each client achieves the same excess risk relative to its own optimal local loss. This avoids the leveling-down effect of loss parity: clients with noisier or harder-to-predict data naturally have higher optimal losses, so the global model is not forced to degrade the performance of easier clients to match them. On the other hand, when clients have identical or comparable optimal local losses, loss gap parity effectively reduces to loss parity, since the variance is unchanged by constant additive shifts. Moreover, in homogeneous data settings, the local losses of clients remain similar for any model, so unconstrained training with objective (1) already yields fairness under both definitions.

In the example of Figure 1, heterogeneity in data distributions creates different levels of task complexity across clients, leading their local optimal models to attain different minimum losses. Enforcing loss parity in this setting compels the global model to keep losses high even for clients with simpler distributions, which can be considered suboptimal in terms of overall performance. By contrast, loss gap parity normalizes losses relative to the local optimum of each client, making them comparable and promoting more balanced—and less trivial—solutions.

3 Eagle: A Federated Learning Algorithm For Loss Gap Parity

In this section, we introduce EAGLE, a new algorithm to enforce loss gap parity in federated learning. Throughout the presentation, we assume that the optimal local losses L_k^* are known, and defer the discussion on how to approximate them in practice to Section 3.2.

To enforce loss gap parity when learning federated models, we propose to add a new regularization term to the standard federated learning objective to penalize models with high variance in loss gaps. Hence, we consider the following optimization problem:

$$\arg\min_{\theta \in \mathcal{H}} F(\theta) := \frac{1}{K} \sum_{k=1}^K [L_k(\theta)] + \mathbb{V}\left(\{r_{k'}(\theta)\}_{k'=1}^K\right) \tag{3}$$

$$= \frac{1}{K} \sum_{k=1}^K \underbrace{\left[L_k(\theta) + \frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (r_k(\theta) - r_{k'}(\theta))^2 \right]}_{:= F_k(\theta)}, \tag{4}$$

where $r_k(\theta) := L_k(\theta) - L_k^*$ is the gap of client k and the regularization parameter $\lambda > 0$ controls the degree to which loss gap parity should be respected. We note that Problem (3) and

Algorithm 1 EAGLE algorithm, in red are the added steps compared to FedAvg.

Require:

K : number of clients, T : number of communication rounds, I : number of local epochs, η : learning rate, θ^0 : initial model parameters, λ : regularization parameter of EAGLE, $\{L_k^*\}_{k=1}^K$: optimal local losses for each client, $\{D_k\}_{k=1}^K$: local datasets for each client

procedure SERVEREXECUTE
  $\forall k \in [K], w_k^0 \leftarrow 1$
  for each round $t = 0, 1, ..., T-1$ **do**
    for each client $k \in [K]$ **in parallel do**
      $\theta_k^{t+1}, r_k^t \leftarrow \text{CLIENTUPDATE}(k, \theta^t, w_k^t)$
    end for
    Server calculates updated weights $\forall k \in [K], w_k^{t+1} \leftarrow 1 + \frac{4\lambda}{K-1} \sum_{1 \le k' \ne k \le K} (r_k^t - r_{k'}^t)$
    $\theta^{t+1} \leftarrow \frac{1}{K} \sum_{k \in [K]} \theta_k^{t+1}$
  end for
  return θ_T
end procedure
procedure CLIENTUPDATE(k, θ^t, w_k^t)
  Calculate the loss gap r_k^t on the model θ^t.
  $\theta_k^{t+1,0} \leftarrow \theta^t$
  for each local epoch $\tau = 0, 1, ..., I-1$ **do**
    Generate an unbiased gradient $\nabla L(\theta_k^{t+1,\tau}; \zeta_k)$
    $\theta_k^{t+1,\tau+1} \leftarrow \theta_k^{t+1,\tau} - \eta w_k^t \nabla L(\theta_k^{t+1,\tau}; \zeta_k)$
  end for
  return θ_k, r_k
end procedure

Problem (4) are strictly equivalent (Zhang et al., 2012), but choosing the latter makes it easier to solve the problem in a federated way. Indeed, the gradient of F can then be written as

$$\nabla F(\theta) := \frac{1}{K} \sum_{k=1}^{K} \underbrace{\Big(1 + \frac{4\lambda}{K-1} \sum_{1 \le k' \ne k \le K} (r_k(\theta) - r_{k'}(\theta))\Big)}_{:= w_k(\theta)} \nabla L_k(\theta)$$

To solve (4), we will adopt an optimization scheme similar to that of *FedAvg* (McMahan et al., 2017), where clients will perform multiple local steps without communicating with the server or the other clients, and then average the obtained local models. A key challenge in optimizing (4) with *FedAvg* is that computing $\nabla F_k(\theta)$—and, in particular, the weights $w_k(\theta)$—requires information from all clients. This necessitates inter-client communication after every local step, which contradicts the primary goal of reducing communication. To overcome this challenge, we, instead, choose to fix the weights after each round of communication and only update the local gradient loss after each local step. In other words, we approximate the local gradients by $\nabla \tilde{F}_k(\theta) = w_k(\theta') \nabla L_k(\theta)$ where θ' refers to the model received from the server at the second to last synchronization step. We summarize this approach, that we call EAGLE, in Algorithm 1.

Choosing the hyperparameter λ. Setting $\lambda = 0$ recovers the standard federated learning objective, while letting $\lambda \to \infty$ forces the solution of (4) to satisfy loss gap parity exactly for all clients. Empirically, we observe that beyond a certain threshold for λ, efforts to reduce the variance of loss gaps no longer benefit the worst-performing clients in terms of loss gap parity. Instead, overall performance suffers in order to achieve lower variance, resulting in a "leveling down" effect. The parameter λ thus allows practitioners to control the extent to which loss gap parity is enforced among clients, depending on the specific application.

3.1 Theoretical Analysis

In this section, we analyze the convergence of our approach from two different perspectives. On the one hand, we show that EAGLE converges to a stationary point of (4). It is worth noting that obtaining stronger convergence guarantees, for example in terms of closeness to the optimal model, might be challenging in our setting. Indeed, the structure of F contains a difference of functions L_k, $L_{k'}$ and thus, even if we assume convexity of the functions $\{L_k\}_{k=1}^K$, F remains non-convex. On the other hand, we show that EAGLE returns models that have relatively low gradient norms with respect to the standard FedAvg objective. It allows us to quantify the decrease in utility incurred by our approach compared to FedAvg when the client losses satisfy the Polyak-Łojasiewicz (PL) condition. We defer all proofs to Appendix A.

To derive our theoretical results, we need several regularity assumptions on the loss of each client, as summarized below.

Assumptions 1. *We assume that the loss functions $\{L_k\}_{k=1}^K$ have the following properties:*

- $\forall k \in [K] : L_k$ *is twice differentiable.*
- *The function L is bounded from below by L^*.*
- *(Unbiased stochastic gradients)* $\mathbb{E}_{\zeta_k}[\nabla L_k(\theta; \zeta_k)] = \nabla L_k(\theta)$.
- *(Bounded variance)* $\exists \sigma \geq 0 : \mathbb{E}_{\zeta_k}[\|\nabla L_k(\theta; \zeta_k) - \nabla L_k(\theta)\|^2] \leq \sigma^2$.
- *(Smoothness)* $\forall k \in [K] : L_k$ *is β-smooth:* $\forall \theta, \theta' \in \mathcal{H} : \|\nabla L_k(\theta) - \nabla L_k(\theta')\| \leq \beta \|\theta - \theta'\|$.
- *(Bounded stochastic gradients)* $\exists B > 0, \forall \theta \in \mathcal{H}, \forall k \in [K] : \|\nabla L_k(\theta; \zeta_k)\| \leq B$.

These assumptions are fairly common in the literature on the analysis of gradient descent-based algorithms (Wang et al., 2021). They are, for example, satisfied by the cross-entropy loss as long as the feature space $\mathcal{X}$ and hypothesis space $\mathcal{H}$ are bounded.

In our analysis, we propose a novel heterogeneity measure that depends on the loss gaps, namely $\Gamma := \sup_{\theta \in \mathcal{H}} \max_{k,k'} |r_k(\theta) - r_{k'}(\theta)|$. This quantity measures the level of heterogeneity of data across clients based on their optimal losses. Indeed, if the data is perfectly IID and each client has enough data points, then Γ would be small. Otherwise it grows as the functions $\{r_k\}_{k=1}^K$ differ from each other.

Based on the assumptions listed in Assumption 1 and our heterogeneity measure, we can now present the main convergence theorem of the EAGLE algorithm.

Theorem 1 (Convergence to a solution of (4))**.** *Let $\theta_k^{(t,\tau)}$ refer to the model of client k after t communication rounds and τ local steps and let $\bar{\theta}^{(t,\tau)} := \frac{1}{K} \sum_{k=1}^K \theta_k^{(t,\tau)}$. Let T be the total number of communication rounds and I be the number of local steps between each communication round. Under Assumption 1, for $\eta \leq \frac{1}{(1+4\lambda\Gamma)\beta + 8\lambda B^2}$, the sequence of models $\{\bar{\theta}^{(t,\tau)}\}_{t \geq 0, \tau \geq 0}$ generated by Algorithm 1 satisfies:*

$$\frac{1}{TI} \sum_{t=1}^{T} \sum_{\tau=1}^{I} \mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq 2 \frac{1}{\eta T I} (F(\bar{\theta}^{(1,1)}) - F^*) + \eta^2 I^2 \xi_1 + \eta \frac{\sigma^2}{K} \xi_2, \tag{5}$$

with $F^* := \arg\min_{\theta \in \mathcal{H}} F(\theta)$, $\xi_1 := 2B^2(1 + 4\lambda\Gamma)^2(\beta^2 + 32\lambda^2(\Gamma^2\beta^2 + B^4))$ and $\xi_2 := ((1 + 4\lambda\Gamma)\beta + 8\lambda B^2)(1 + 4\lambda\Gamma)^2$.

If we further choose $\eta = \frac{1}{\sqrt{TI}}$ and $1 \leq I \leq \sqrt{T}$, the rate can be shown to be sublinear in T, that is:

$$\frac{1}{TI} \sum_{t=1}^{T} \sum_{\tau=1}^{I} \mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq \underbrace{2 \frac{1}{\sqrt{IT}} (F(\bar{\theta}^1) - F^*) + \frac{I}{T} \xi_1 + \frac{\xi_2}{\sqrt{IT}} \frac{\sigma^2}{K}}_{\mathcal{O}(\frac{1}{\sqrt{T}})}. \tag{6}$$

Under standard assumptions on the loss function, we have shown that a carefully chosen step size allows us to recover the standard sublinear convergence rate of algorithms based on SGD in the general non-convex case (Yu et al., 2019). In this result, $I\xi_1$ represents the error related to running EAGLE for multiple local steps before averaging. Naturally, the larger the number of local epochs I and the heterogeneity measure Γ, the larger this error term. Similarly, ξ_2 is an upper bound on the the smoothness parameter of F. In fact, when $\lambda \to 0$, we have that $\xi_2 \to \beta$. This recovers the terms introduced in Yu et al. (2019) for FedAvg.

The next theorem investigates the convergence of EAGLE with respect to the original federated learning problem, that is $L(\theta) := \frac{1}{K}\sum_{k=1}^{K} L_k(\theta)$.

Theorem 2 (Convergence to a solution of federated learning)**.** *Let $\theta_k^{(t,\tau)}$ refer to the model of client k after t communication rounds and τ local steps and let $\bar{\theta}^{(t,\tau)} := \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau)}$. Let T be the total number of communication rounds and I be the number of local steps between each communication round. Under Assumption 1, for $\eta \leq \frac{1}{(1+4\lambda\Gamma)\beta+8\lambda B^2}$, the sequence of models $\{\bar{\theta}^{(t,\tau)}\}_{t\geq 0,\tau\geq 0}$ generated by Algorithm 1 satisfies:*

$$\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[\|\nabla L(\bar{\theta}^{(t,\tau)})\|^2] \leq 2\frac{1}{\eta TI}(L(\bar{\theta}^{(1,1)}) - L^*) + \underbrace{32\lambda^2\Gamma^2B^2}_{\substack{\text{neighborhood of the solution}\\ \text{relative to the heterogeneity}}} + \eta\frac{\sigma^2}{K}\xi_2 \quad (7)$$
$$+ 8\beta^2\eta^2I^2(1+4\lambda\Gamma)^2B^2,$$

with $\xi_2 := ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2$.

If we further assume that $L(\theta)$ is μ-Polyak-Łojasiewicz (PL), that is $\forall\theta \in \mathcal{H} : \frac{1}{2}\|\nabla L(\theta)\| \geq \mu(L(\theta) - L^*)$, then we have that:

$$\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[L(\bar{\theta}^{(t,\tau)}) - L^*] \leq 4\frac{\mu}{\eta TI}(L(\bar{\theta}^{(1,1)}) - L^*)$$
$$+ 2\eta\frac{\sigma^2}{K}\xi_2 + 16\beta^2\eta^2I^2(1+4\lambda\Gamma)^2B^2 + \underbrace{62\mu\lambda^2\Gamma^2B^2}_{\text{Loss in utility}}. \quad (8)$$

Theorem 2 bounds the potential loss of utility from the regularization added to ensure fairness since the only term that does not vanish with proper choice of η is $\mathcal{O}(\lambda^2\Gamma^2)$. Unsurprisingly, having less heterogeneity or choosing smaller values of λ results in a smaller decrease in utility.

3.2 Practical Considerations

In this section, we tackle two issues that may arise when using EAGLE in practice, namely the fact that optimal local losses are not available and that the added regularization term may create some instability in the optimization process.

Optimal local losses. So far, we assumed that the optimal losses L_k^* were available. However, this is an unrealistic assumption in practice since even getting a reliable approximation of this quantity would require each client to have access to sufficiently many examples to learn a good model on their own data distributions. In the experiments in Section 4, we tackle this issue by splitting the data of each client into a train and a validation set. We then learn the best possible model on the train set of each client and approximate L_k^* on their validation set. We provide additional details on how to obtain the best possible approximation of L_k^* in Appendix B.1.

Table 1: Performances of different baselines for very heterogenous split $dir(\alpha = 0.1)$. EAGLE achieves the best worst performing client in terms loss gaps while keeping a comparable accuracy to the other baselines.

Algorithm	$\alpha = 0.1$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy ↑	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	0.152 (± 0.207)	**-0.477 (± 0.155)**	0.687 (± 0.003)	0.037 (± 0.026)
FedAvg	0.106 (± 0.170)	-0.459 (± 0.211)	**0.692 (± 0.004)**	0.033 (± 0.021)
q-FFL $q = 1.0$	0.104 (± 0.178)	-0.464 (± 0.193)	**0.692 (± 0.002)**	0.032 (± 0.021)
q-FFL $q = 3.0$	0.109 (± 0.186)	-0.471 (± 0.178)	0.691 (± 0.004)	0.032 (± 0.022)
q-FFL $q = 5.0$	0.118 (± 0.189)	-0.471 (± 0.169)	0.689 (± 0.004)	0.033 (± 0.023)
EAGLE $\lambda = 0.1$	0.096 (± 0.155)	-0.450 (± 0.209)	0.691 (± 0.007)	0.030 (± 0.019)
EAGLE $\lambda = 1.0$	**0.073 (± 0.123)**	-0.381 (± 0.199)	0.687 (± 0.003)	0.020 (± 0.013)
EAGLE $\lambda = 2.0$	0.094 (± 0.166)	-0.355 (± 0.268)	0.681 (± 0.007)	**0.019 (± 0.010)**
EAGLE $\lambda = 3.0$	0.107 (± 0.171)	-0.335 (± 0.275)	0.677 (± 0.007)	0.019 (± 0.010)
EAGLE $\lambda = 5.0$	0.203 (± 0.181)	-0.245 (± 0.284)	0.650 (± 0.007)	0.020 (± 0.011)

Instability of EAGLE. Our approach can be seen as a reweighting approach where each client gets a new weight w_k^t at each communication round. These weights are adaptive and are based on the level of loss gap parity of the global model. This can be equivalently viewed as performing SGD locally with varying learning rates. Many model architectures are sensitive to this hyperparameter, and incorrect choices can lead to exploding gradients, even in simple models (Meng et al., 2024). To avoid this issue, we normalize the weights $\{w_k^t\}_{k=1}^K$ so that they have an L_2 norm of 1. This preserves the sign of the weights and their relative importance, while keeping their magnitude constant, preventing unintended variations in the effective learning rate.

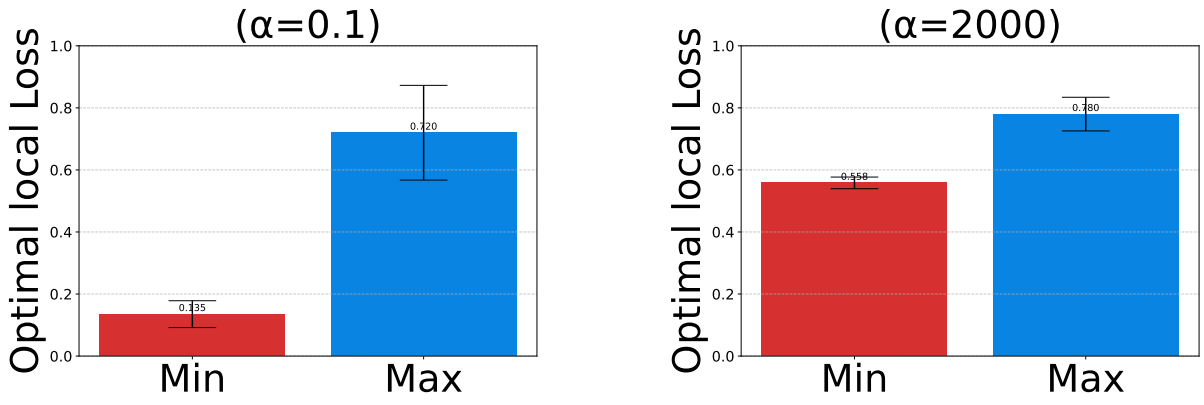
4 Experiments

In this section, we empirically evaluate EAGLE against *FedAvg* and two algorithms that aim for loss parity. This effectively allows us to highlight the issues of fairness unaware training (*FedAvg*) and that of the training that aims for a fairness definition (loss parity) that may not adequately handle data heterogeneity.

The section is structured as follows. In the first part we showcase experiments on logistic regression. We begin with a synthetic dataset where having clearly separable data in the input space is actually a disadvantage for clients under a loss-parity–enforcing algorithm. We then demonstrate how EAGLE overcomes this issue by assigning higher weights to clients with linearly separable data. Next, we show that such heterogeneity in optimal local loss also arises in a real dataset (EMNIST (Cohen et al., 2017)). In the second part, we present experiments with a CNN model to highlight the limitations of loss-parity–enforcing approaches and unconstrained training on more complex models. These experiments are conducted on both the EMNIST and DirtyMNIST datasets (Mukhoti et al., 2023).

Baselines Throughout this section, we compare EAGLE on accuracy and loss gaps fairness metrics with *FedAvg* for unconstrained training, *q-FedAvg (q-FFL)* (Li et al., 2020b), an approach that aims to reach models with balanced losses by raising the local losses to the power $q + 1$, where q is to be tuned, giving higher weights to the clients with higher losses, and with *Agnostic Federated Learning (AFL)* (Mohri et al., 2019b) which takes an *egalitarian* approach aiming to minimize the loss of the worst performing client.

Datasets



$\alpha = 0.1$ $\alpha = 2000$

Figure 2: $\min\{L_k^*\}_{k\in[10]}$ (red) and $\max\{L_k^*\}_{k\in[10]}$ (blue) for a heterogeneous data split ($\alpha = 0.1$) and a homogeneous one ($\alpha = 2000$) for a linear model on EMNIST data. The range of the optimal local losses is greater for non-iid data.

- **Synthetic data.** We consider three clients collaborating to fit a linear separator of 2-dimensional data into two classes $Y = \{-1, 1\}$, each client samples 100 data points from the following underlying distributions:
  - *client* 0, $X|Y = 1 \sim \mathcal{N}([2, 2], \mathbb{I}_2)$ and $X|Y = -1 \sim \mathcal{N}([-2, -2], \mathbb{I}_2)$
  - *client* 1, $X|Y = 1 \sim \mathcal{N}([1, 1], \mathbb{I}_2)$ and $X|Y = -1 \sim \mathcal{N}([-1, -1], \mathbb{I}_2)$
  - *client* 2, $X|Y = 1 \sim \mathcal{N}([0.5, 0.5], \mathbb{I}_2)$ and $X|Y = -1 \sim \mathcal{N}([-0.5, -0.5], \mathbb{I}_2)$

  Where $\mathbb{I}_2$ is the identity covariance matrix. We also rotate the data *client* 2 around $x_0 = \{0, 0\}$ with 45 degrees to avoid the case where all clients share an optimal separator. The resulting distributions are summarized in the Appendix (Figure 5).
- **EMNIST.** To simulate federated learning with variable degrees of data heterogeneity, we split the EMNIST data (Cohen et al., 2017) between 10 clients based on a splitting technique involving the Dirichlet distribution (Hsu et al., 2019). Details on the data splitting process are given in Appendix B.3. The heterogeneity of the data results in heterogeneity in optimal local metrics, as illustrated in Figure 2.
- **DirtyMNIST.** The examples in the DirtyMNIST dataset are labeled as clean or ambiguous depending on how confidently a human can tell their label. We use this attribute to distribute 8000 examples across 5 clients, each with different proportions of ambiguous images ranging from 0% in *client* 0 to 100% in *client* 5. The intuition is that the percentage of the ambiguous will be reflected in their optimal local loss.

In all experiments, the loss gaps $\{r_k(\theta)\}_{k=1}^K$ are calculated using a reserved validation set to avoid overfitting issues on the training set. With this choice, the loss gaps can sometimes be negative, meaning that federated learning helps clients generalize better. Indeed, the performance of the federated model on the validation set can be better than that obtained with local training. In cases where the loss gaps are negative, achieving loss gap parity implies that all clients benefit equally from federated learning, experiencing the same improvement in generalization on their validation sets.

4.1 Results

Here, we showcase the behavior of different algorithms on logistic regression tasks with a linear model and image classification tasks using a CNN model.

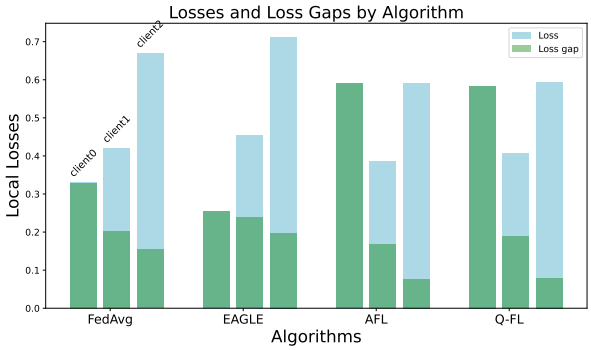


Figure 3: Loss losses and loss gaps for the three clients using the synthetic data. Here we run EAGLE with $\lambda = 2.0$ and q-FFL $q = 5.0$. While q-FFL and AFL try to learn a good classifier for *client*2 data and thus resulting in harming the performance of *client*0, EAGLE focuses on balancing the loss gaps of each client, which results in the clients benefiting equally from federated learning.

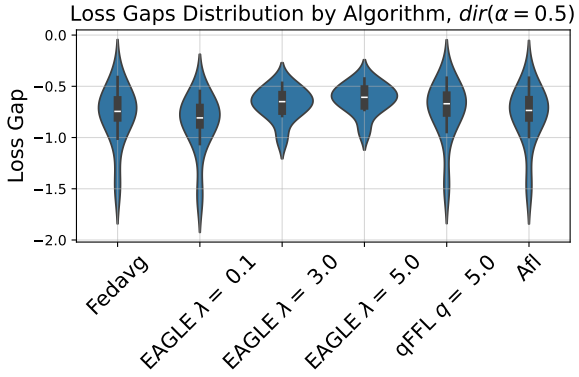


Figure 4: The distribution of loss gaps for the 10 clients for a heterogeneous split with CNN model for EMNIST dataset. We notice that in the case of high non-iid data, EAGLE reduces the gap of the worst performing client in terms of loss and achieves less variance then other baselines.

Synthetic data The choice of such distributions allows us to simulate variable levels of data separability between clients. As shown in Figure 3 , methods that aim to balance the losses of all clients give higher weights to clients 1 and 2, leading the optimization path towards their optimal model. This goes against the intended goal of balancing the losses, which is to learn all underlying distributions of the clients equally. By aiming to balance loss gaps instead of losses, EAGLE eliminates this behavior, and thus having a lower optimal training loss is no longer a disadvantage. The training losses evolution of *q-FFL*, *AFL* and *EAGLE* are described in Figure 3.

EMNIST data In Table 1, we describe the utility and fairness achieved by different baselines using a linear model on a heterogeneous data split $dir(0.1)$. EAGLE keeps comparable accuracies to other baselines, higher values of λ result in higher importance of the regularization term in the objective function, and thus result in models with smaller disparity of loss gaps at the cost of lower utility. In Figure 4, the distribution of loss gaps over the clients show that EAGLE aims to reduce the variance of loss gaps by helping the worst performing clients even for CNN models.

Table 2: Results of different baselines on DirtyMNIST data with a CNN model. Loss parity and loss gaps parity approaches achieves good accuracy because the clients with ambiguous examples provide gradients that are useful on the clean as well.

Algorithm	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-0.016 $\pm$ 0.013	-0.822 $\pm$ 0.017	0.882 $\pm$ 0.003	0.083 $\pm$ 0.000
FedAvg	-0.083 $\pm$ 0.004	-0.877 $\pm$ 0.046	0.889 $\pm$ 0.007	0.081 $\pm$ 0.010
q-FFL $q = 1.0$	-0.084 $\pm$ 0.004	-0.886 $\pm$ 0.022	0.889 $\pm$ 0.001	0.084 $\pm$ 0.003
q-FFL $q = 3.0$	-0.062 $\pm$ 0.013	-0.883 $\pm$ 0.020	0.883 $\pm$ 0.005	0.088 $\pm$ 0.006
q-FFL $q = 5.0$	-0.070 $\pm$ 0.003	-0.876 $\pm$ 0.018	0.879 $\pm$ 0.006	0.083 $\pm$ 0.005
EAGLE $\lambda = 0.1$	-0.095 $\pm$ 0.003	**-0.944 $\pm$ 0.018**	**0.897 $\pm$ 0.001**	0.093 $\pm$ 0.005
EAGLE $\lambda = 0.3$	**-0.099 $\pm$ 0.013**	-0.896 $\pm$ 0.038	**0.897 $\pm$ 0.002**	0.086 $\pm$ 0.003
EAGLE $\lambda = 0.5$	-0.094 $\pm$ 0.015	-0.869 $\pm$ 0.057	0.893 $\pm$ 0.004	0.082 $\pm$ 0.006
EAGLE $\lambda = 0.7$	-0.076 $\pm$ 0.023	-0.787 $\pm$ 0.090	0.878 $\pm$ 0.009	0.067 $\pm$ 0.013
EAGLE $\lambda = 1.0$	-0.002 $\pm$ 0.007	-0.506 $\pm$ 0.058	0.843 $\pm$ 0.003	0.034 $\pm$ 0.007
EAGLE $\lambda = 3.0$	0.322 $\pm$ 0.064	0.140 $\pm$ 0.086	0.818 $\pm$ 0.007	0.005 $\pm$ 0.003
EAGLE $\lambda = 5.0$	0.434 $\pm$ 0.104	0.249 $\pm$ 0.096	0.814 $\pm$ 0.004	**0.006 $\pm$ 0.002**

DirtyMNIST In the experiments on the DirtyMNIST dataset, we showcase the effect of having variable optimal local losses but still share the optimal local model. In such cases, the gradients provided by the clients point towards the same direction and therefore all methods that reweighs these gradients to achieve a certain fairness constraint will have similar behavior since the model learned by any client is also good for the others. We still notice that EAGLE achieves the best worst performing client.

5 Discussion And Conclusion

In this work, we tackled the problem of fairness between clients in federated learning. We first argued that the standard notion of loss parity may lead to severe leveling down effects and that the more recent notion of loss gap parity should be preferred. To enforce this fairness constraint, we proposed EAGLE, a new regularized federated objective able to explicitly control the trade-off between utility and loss gap parity.

Theoretically, we proved that our approach converges for non-convex losses. We also showed that it can be connected to the standard federated learning objective via a novel heterogeneity measure. Experimentally, we evaluated EAGLE on both synthetic and real data, showing that we can achieve the intended purpose of our approach by correctly choosing the hyperparameter we introduced to control the trade-off between utility and loss gap parity.

A key limitation of our approach lies in the assumption that $\{L_k^*\}_{k=1}^K$ can be computed with high precision and that all clients maintain continuous communication throughout the training process. These requirements make the method impractical for cross-edge applications, where both computational capacity and communication bandwidth are typically constrained. Furthermore, the performance of the method critically depends on the choice of the hyperparameter λ, which must be carefully tuned to avoid falling into a leveling-down regime.

Acknowledgments

This work is funded by the Groupe La Poste, sponsor of the Inria Foundation, in the framework of the FedMalin Inria Challenge, and ANR-22-PESN-0014 under the France 2030 program. Michaël Perrot is supported by the French National Research Agency (ANR) through the

grant ANR-23-CE23-0011-01 (Project FaCTor) and the France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project).

References

Carey, A. and Wu, X. (2025). Achieving distributive justice in federated learning via uncertainty quantification. *arXiv preprint arXiv:2504.15924*.

Cho, Y. J., Jhunjhunwala, D., Li, T., Smith, V., and Joshi, G. (2022). To federate or not to federate: Incentivizing client participation in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

Chu, W., Xie, C., Wang, B., Li, L., Yin, L., Nourian, A., Zhao, H., and Li, B. (2023). Focus: Fairness via agent-awareness for federated learning on heterogeneous data. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

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

Hsu, T.-M. H., Qi, H., and Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv*.

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

Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020a). A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854.

Li, T., Hu, S., Beirami, A., and Smith, V. (2021). Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2020b). Fair resource allocation in federated learning. In *International Conference on Learning Representations*.

Long, G., Tan, Y., Jiang, J., and Zhang, C. (2020). Federated learning for open banking. In *Federated learning: privacy and incentive*, pages 240–254. Springer.

Maheshwari, G., Bellet, A., Denis, P., and Keller, M. (2023). Fair without leveling down: A new intersectional fairness definition. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

Meng, S. Y., Orvieto, A., Cao, D. Y., and Sa, C. D. (2024). Gradient descent on logistic regression with non-separable data and large step sizes.

Mittelstadt, B., Wachter, S., and Russell, C. (2023). The unfairness of fair machine learning: Leveling down and strict egalitarianism by default. *Mich. Tech. L. Rev.*, 30:1.

Mohri, M., Sivek, G., and Suresh, A. T. (2019a). Agnostic federated learning. In *International conference on machine learning*, pages 4615–4625. PMLR.

Mohri, M., Sivek, G., and Suresh, A. T. (2019b). Agnostic federated learning. In *International conference on machine learning*, pages 4615–4625. PMLR.

Mukhoti, J., Kirsch, A., Van Amersfoort, J., Torr, P. H., and Gal, Y. (2023). Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. (2020). The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119.

Shi, Y., Yu, H., and Leung, C. (2023). Towards fairness-aware federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11922–11938.

Wang, J., Charles, Z. B., Xu, Z., Joshi, G., McMahan, B., Aguera-Arcas, B. H., Al-Shedivat, M., Andrew, G., Avestimehr, A. S., Daly, K., Data, D., Diggavi, S., Eichner, H., Gadhikar, A., Garrett, Z., Girgis, A. M., Hanzely, F., Hard, A., He, C., Horvath, S., Huo, Z., Ingerman, A., Jaggi, M., Javidi, T., Kairouz, P., Kale, S. C., Karimireddy, S. P., Konečný, J., Koyejo, S., Li, T., Liu, L., Mohri, M., Qi, H., Reddi, S., Richtarik, P., Singhal, K., Smith, V., Soltanolkotabi, M., Song, W., Suresh, A. T., Stich, S., Talwalkar, A., Wang, H., Woodworth, B., Wu, S., Yu, F., Yuan, H., Zaheer, M., Zhang, M., Zhang, T., Zheng, C. J., Zhu, C., and Zhu, W. (2021). A field guide to federated optimization. Technical report, arxiv.

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

Yu, H., Yang, S., and Zhu, S. (2019). Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5693–5700.

Yue, X., Nouiehed, M., and Al Kontar, R. (2023). Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2(1):10–23.

Zhang, Y., Wu, H., and Cheng, L. (2012). Some new deformation formulas about variance and covariance. In *2012 proceedings of international conference on modelling, identification and control*, pages 987–992. IEEE.

Zietlow, D., Lohaus, M., Balakrishnan, G., Kleindessner, M., Locatello, F., Schölkopf, B., and Russell, C. (2022). Leveling down in computer vision: Pareto inefficiencies in fair deep classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10410–10421.

Appendix

A Formal Statement of the Theoretical Results and Proofs

In this appendix, we provide the convergence proof for EAGLE. To facilitate understanding of the proof, all notations are defined in Table 3.

Recall that EAGLE optimizes the following objective:

$$\arg\min_{\theta\in\mathcal{H}} F(\theta) := \frac{1}{K}\sum_{k=1}^{K}\left[L_k(\theta) + \frac{\lambda}{K-1}\sum_{1\le k'\neq k\le K}(r_k(\theta) - r_{k'}(\theta))^2\right], \tag{A.9}$$

where $r_k(\theta) := L_k(\theta) - L_k^*$, and $\lambda > 0$ is a regularization parameter to be tuned in practice.

Our proof adapts the analysis of FedAvg in Yu et al. (2019), where we establish an upper bound on the smoothness parameter and a bound on the gradient norm for our new regularized objective, following a similar proof strategy.

We first observe that the gradient of F is equal to :

$$\nabla F(\theta) := \frac{1}{K}\sum_{k=1}^{K}\underbrace{\left(1 + \frac{4\lambda}{K-1}\sum_{1\le k'\neq k\le K}(r_k(\theta) - r_{k'}(\theta))\right)}_{w_k(\theta)}\nabla L_k(\theta)$$

We introduce a new "proxy" gradient

$$\nabla \tilde{F}_k(\theta,\theta') := \underbrace{\left(1 + \frac{4\lambda}{K-1}\sum_{1\le k'\neq k\le K}(r_k(\theta') - r_{k'}(\theta'))\right)}_{w_k(\theta')}\nabla L_k(\theta)$$

Here, the functions r_k and $r_{k'}$ are evaluated on a different model than the stochastic gradient ∇L_k, this will remove the need for communication between synchronization rounds.

We will use two time counters, t to denote the the number of synchronizations and τ to index the local step, and we will use k in subscript to index the client. For example, $\theta_k^{(t,\tau)}$ denotes the model of client k after t synchronizations and τ local steps. Additionally, we denote a virtual average model $\bar{\theta}^{(t,\tau)} := \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau)}$ to represent the average of local models within clients between two rounds communication. For all the results mentioned in this section, the expectation $\mathbb{E}$ with respect to some function of the model $\bar{\theta}^{(t,\tau)}$ is taken with respect to the minibatching noises $\zeta_1, \zeta_2, \ldots, \zeta_k$ at all times up to (t,τ).

With this notation, we have: $\theta_k^{(t+1,0)} := \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,I)} = \bar{\theta}^{(t,I)}$. In words, at the start of each local training phase, the local models are initialized with the average of the models of all clients.

We can derive a recursive update on the virtual model:

$$\bar{\theta}^{(t,\tau+1)} = \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau+1)} \tag{A.10}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau)} - \eta\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}; \zeta_k) \tag{A.11}$$

$$= \bar{\theta}^{(t,\tau)} - \eta\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}; \zeta_k). \tag{A.12}$$

We will first introduce lemmas that are useful in our proof of our main results. Then, we will restate the results of the main paper and provide their proofs.

As a preliminary step, we establish bounds on the weights $w_k(\theta), w_k(\theta')$, which will be useful in the subsequent analysis.

Notation	Description
λ	Regularization parameter to be controlled by the user
θ	An arbitrary model in the space
t	counter of synchronization rounds
τ	counter of local steps
$\theta_k^{(t,\tau)}$	Local model inside each client k, after t syncs and τ local steps.
$\bar{\theta}^{(t,\tau)}$	$\frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau)}$
K	Number of clients
$L_k(\theta)$	Loss of client k
$F_k(\theta)$	$L_k(\theta) + \frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\theta) - r_{k'}(\theta))^2$
$\nabla L_k(\theta)$	Gradient of L_k
$\nabla L_k(\theta, \zeta_k)$	(Minibatch) Stochastic Gradient of L_k
σ	Bound on the variance of stochastic gradients of L_k
β	Smoothness parameter of L_k
Γ	Heterogeneity measure, $\Gamma := \sup_{\theta\in\mathcal{H}}\max_{k,k'}\lvert r_k(\theta) - r_{k'}(\theta)\rvert$
B	Bound on $\lVert\nabla L_k(\theta;\zeta_k)\rVert^2$
$w_k(\theta)$	$(1 + 4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\theta) - r_{k'}(\theta)))$
$\nabla\tilde{F}_k(\theta)$	$w_k(\theta)\nabla L_k(\theta)$
$\nabla\tilde{F}_k(\theta,\theta')$	$w_k(\theta')\nabla L_k(\theta)$

Table 3: Table of notations and their descriptions

Lemma 1 (Properties of the weights of clients $w_k(\theta)$)**.** *For any two models $\theta, \theta' \in \mathcal{H}$, the weights $w_k(\theta), w_k(\theta')$ satisfy the following:*

$$|w_k(\theta)| \leq 1 + 4\lambda\Gamma, \tag{A.13}$$

$$|w_k(\theta) - w_k(\theta')| \leq 8\lambda B\|\theta - \theta'\|. \tag{A.14}$$

Proof. Indeed, for a $\theta \in \mathcal{H}$ we have for the first property:

$$|w_k(\theta)| = \left|1 + 4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\theta) - r_{k'}(\theta))\right|, \tag{A.15}$$

$$\overset{\text{Triangle inequality}}{\leq} 1 + 4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}|r_k(\theta) - r_{k'}(\theta)|, \tag{A.16}$$

$$\overset{|r_k(\theta)-r_{k'}(\theta)|\leq\Gamma}{\leq} 1 + 4\lambda\Gamma. \tag{A.17}$$

And for a $\theta, \theta' \in \mathcal{H}$ we have for the second property:

$$|w_k(\theta) - w_k(\theta')| \leq \left|4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\theta) - r_{k'}(\theta)) - 4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\theta') - r_{k'}(\theta'))\right|, \tag{A.18}$$

$$\overset{\text{Triangular inequality}}{\leq} 4\frac{\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}|r_k(\theta) - r_k(\theta') + r_{k'}(\theta') - r_{k'}(\theta)|, \tag{A.19}$$

$$\overset{L_k \text{ is } B\text{-Lipschitz}}{\leq} 8\lambda B\|\theta - \theta'\|. \qquad \square$$

The following lemma derives an upper on bound on the norm of $\|\nabla\tilde{F}_k(\theta,\theta';\zeta_k)\|$:

Lemma 2 (Bounded norm of the gradient of $\tilde{F}_k$)**.** *Under the assumption that the norm of the stochastic gradient of L_k is bounded by a constant B, for any two models $\theta, \theta' \in \mathcal{H}$, we have:*

$$\|\nabla \tilde{F}_k(\theta, \theta'; \zeta_k)\| \leq (1 + 4\lambda\Gamma)B. \tag{A.20}$$

Proof. We have $\forall \theta, \theta' \in \mathcal{H}$:

$$\|\nabla F_k(\theta, \theta'; \zeta_k)\| = \left\| w_k(\theta') \nabla L_k(\theta; \zeta_k) \right\| \tag{A.21}$$

$$= |w_k(\theta')| \cdot \|\nabla L_k(\theta; \zeta_k)\| \tag{A.22}$$

$$\overset{\text{Lemma 1}, \|\nabla L_k(\theta;\zeta_k)\| \leq B}{\leq} (1 + 4\lambda\Gamma)B. \qquad \square$$

The second lemma derives an upper bound on the smoothness parameter of the regularized local objective $F_k(\theta) := L_k(\theta) + \frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (r_k(\theta) - r_{k'}(\theta))^2$:

Lemma 3. *Under the assumptions 1 , $\forall k \in [K]$: F_k is smooth with a smoothness parameter equal to: $(1 + 4\lambda\Gamma)\beta + 8\lambda B^2$.*

$$\forall \theta \in \mathcal{H} : \|\nabla^2 F_k(\theta)\| \leq (1 + 4\lambda\Gamma)\beta + 8\lambda B^2. \tag{A.23}$$

Proof. We know that if F_k is twice differentiable, we have:

$$F_k \text{ is } \beta'\text{-smooth} \iff \forall \theta \in \mathcal{H} : \|\nabla^2 F_k(\theta)\| \leq \beta', \tag{A.24}$$

Let's derive an upper bound on β'. We have:

$$\nabla^2 F_k(\theta) = \frac{\partial}{\partial \theta} \left[w_k(\theta) \nabla L_k(\theta) \right]$$

$$= w_k(\theta) \nabla^2 L_k(\theta) + 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\nabla L_k(\theta) - \nabla L_{k'}(\theta)) \nabla L_k(\theta)^T$$

Thus:

$$\|\nabla^2 F_k(\theta)\| = \left\| w_k(\theta) \nabla^2 L_k(\theta) + 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\nabla L_k(\theta) - \nabla L_{k'}(\theta)) \nabla L_k(\theta)^T \right\|, \tag{A.25}$$

$$\|\nabla^2 F_k(\theta)\| \leq |w_k(\theta)| \underbrace{\|\nabla^2 L_k(\theta)\|}_{\leq \beta, \text{since } L_k \text{ is } \beta\text{-smooth}}$$

$$+ \left\| 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\nabla L_k(\theta) - \nabla L_{k'}(\theta)) \nabla L_k(\theta)^T \right\|, \tag{A.26}$$

$$\overset{\text{Lemma 1}}{\leq} (1 + 4\lambda\Gamma)\beta + \left\| 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\nabla L_k(\theta) - \nabla L_{k'}(\theta)) \nabla L_k(\theta)^T \right\|, \tag{A.27}$$

$$\leq (1 + 4\lambda\Gamma)\beta$$

$$+ 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\|(\nabla L_k(\theta) \nabla L_{k'}(\theta)^T\| + \|\nabla L_k(\theta) \nabla L_k(\theta)^T\|), \tag{A.28}$$

$$\overset{\|AB\| \leq \|A\| \|B\|}{\leq} (1 + 4\lambda\Gamma)\beta$$

$$+ 4\frac{\lambda}{K-1} \sum_{1 \leq k' \neq k \leq K} (\|(\nabla L_k(\theta)\| \|\nabla L_k(\theta)^T\| + \|\nabla L_{k'}(\theta)\| \|\nabla L_k(\theta)^T\|), \tag{A.29}$$

$$\overset{\|\nabla L_k(\theta;\zeta_k)\| \leq B}{\leq} \underbrace{\left((1 + 4\lambda\Gamma)\beta + 8\lambda B^2 \right)}_{\beta'}. \tag{A.30}$$

$\square$

The next lemma bounds the difference between the local models and the averaged model.

Lemma 4 (Bounded distance of local and average models)**.** *After t synchronization steps, and τ local steps, under assumptions 1, the algorithm 1 satisfies:*

$$\mathbb{E}[\|\bar{\theta}^{(t,\tau)} - \theta_k^{t,\tau}\|^2] \leq 4\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2. \tag{A.31}$$

Proof. We have:

$$\bar{\theta}^{(t,\tau)} = \frac{1}{K}\sum_{k=1}^{K} \theta_k^{(t,\tau)} \tag{A.32}$$

$$= \frac{1}{K}\sum_{k=1}^{K} \theta_k^{(t,\tau-1)} - \eta \nabla \tilde{F}_k(\theta_k^{(t,\tau-1)}, \bar{\theta}^{(t,0)}; \zeta_k) \tag{A.33}$$

$$= \bar{\theta}^{(t,\tau-1)} - \eta \frac{1}{K}\sum_{k=1}^{K} \nabla \tilde{F}_k(\theta_k^{(t,\tau-1)}, \bar{\theta}^{(t,0)}; \zeta_k) \tag{A.34}$$

$$= \bar{\theta}^{(t,0)} - \eta \frac{1}{K}\sum_{k=1}^{K}\sum_{\tau'=0}^{\tau-1} \nabla \tilde{F}_k(\theta_k^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_k) \tag{A.35}$$

And similarly:

$$\theta_k^{(t,\tau)} = \theta_k^{(t,\tau-1)} - \eta \nabla \tilde{F}_k(\theta_k^{(t,\tau-1)}, \bar{\theta}_k^{(t,0)}; \zeta_k) \tag{A.36}$$

$$= \bar{\theta}^{(t,0)} - \eta \sum_{\tau'=0}^{\tau-1} \nabla \tilde{F}_k(\theta_k^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_k) \tag{A.37}$$

Therefore:

$$\mathbb{E}[\|\bar{\theta}^{(t,\tau)} - \theta_k^{(t,\tau)}\|^2]$$

$$= \eta^2 \mathbb{E}\left[\left\| \sum_{\tau'=0}^{\tau-1} \frac{1}{K}\left(\sum_{k'=1}^{K} \nabla \tilde{F}_{k'}(\theta_{k'}^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_{k'}) \right) - \sum_{\tau'=0}^{\tau-1} \nabla \tilde{F}_k(\theta_k^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_k) \right\|^2\right] \tag{A.38}$$

$$\leq 2\eta^2 \tau \left(\sum_{\tau'=0}^{\tau-1} \mathbb{E}\left[\left\| \frac{1}{K}\left(\sum_{k'=1}^{K} \nabla \tilde{F}_{k'}(\theta_{k'}^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_{k'})\right)\right\|^2\right] + \mathbb{E}\left[\left\| \nabla \tilde{F}_k(\theta_k^{(t,\tau')}, \bar{\theta}^{(t,0)}; \zeta_k)\right\|^2\right] \right) \tag{A.39}$$

$$\overset{\text{Lemma } 2}{\leq} 4\eta^2 \tau^2 (1+4\lambda\Gamma)^2 B^2 \tag{A.40}$$

$$\overset{\tau \leq I}{\leq} 4\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2. \qquad \square$$

Theorem 1 (Convergence to a solution of (4))**.** *Let $\theta_k^{(t,\tau)}$ refer to the model of client k after t communication rounds and τ local steps and let $\bar{\theta}^{(t,\tau)} := \frac{1}{K}\sum_{k=1}^{K} \theta_k^{(t,\tau)}$. Let T be the total number of communication rounds and I be the number of local steps between each communication round. Under Assumption 1, for $\eta \leq \frac{1}{(1+4\lambda\Gamma)\beta + 8\lambda B^2}$, the sequence of models $\{\bar{\theta}^{(t,\tau)}\}_{t\geq 0, \tau \geq 0}$ generated by Algorithm 1 satisfies:*

$$\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I} \mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq 2\frac{1}{\eta T I}(F(\bar{\theta}^{(1,1)}) - F^*) + \eta^2 I^2 \xi_1 + \eta \frac{\sigma^2}{K}\xi_2,$$

with $F^* := \arg\min_{\theta \in \mathcal{H}} F(\theta), \xi_1 := 2B^2(1+4\lambda\Gamma)^2(\beta^2 + 32\lambda^2(\Gamma^2\beta^2 + B^4))$ and $\xi_2 := ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2$.

If we further choose $\eta = \frac{1}{\sqrt{TI}}$ *and* $1 \leq I \leq \sqrt{T}$*, the rate can be shown to be sublinear in* T*, that is:*

$$\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq \underbrace{2\frac{1}{\sqrt{IT}}(F(\bar{\theta}^1) - F^*) + \frac{I}{T}\xi_1 + \frac{\xi_2}{\sqrt{IT}}\frac{\sigma^2}{K}}_{\mathcal{O}(\frac{1}{\sqrt{T}})}.$$

Proof. First, let us start by upper bounding the following quantity (the difference of the applied gradient to the actual gradient of ∇F_k):

$$\|\nabla \tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}) - \nabla F_k(\bar{\theta}^{(t,\tau)})\|^2 = \left\|w_k(\bar{\theta}^{(t,0)})\nabla L_k(\theta_k^{(t,\tau)}) - w_k(\bar{\theta}^{(t,\tau)})\nabla L_k(\bar{\theta}^{(t,\tau)})\right\|^2 \tag{A.41}$$

$$= \|w_k(\bar{\theta}^{(t,0)})(\nabla L_k(\theta_k^{(t,\tau)}) - \nabla L_k(\bar{\theta}^{(t,\tau)})) + (w_k(\bar{\theta}^{(t,0)}) - w_k(\bar{\theta}^{(t,\tau)}))\nabla L_k(\bar{\theta}^{(t,\tau)})\|^2 \tag{A.42}$$

$$\leq \left[(1+4\lambda\Gamma)\|\nabla L_k(\theta_k^{(t,\tau)}) - \nabla L_k(\bar{\theta}^{(t,\tau)})\| + |w_k(\bar{\theta}^{(t,0)}) - w_k(\bar{\theta}^{(t,\tau)})|B\right]^2 \tag{A.43}$$

$$\overset{L_k \text{ is } B\text{-smooth, Lemma 1}}{\leq} \left[(1+4\lambda\Gamma)\beta\|\theta_k^{(t,\tau)} - \bar{\theta}^{(t,\tau)}\| + 8\lambda B^2\|\bar{\theta}^{(t,0)} - \bar{\theta}^{(t,\tau)}\|\right]^2 \tag{A.44}$$

$$\overset{(a+b)^2\leq 2a^2+2b^2}{\leq} 2\left[(1+4\lambda\Gamma)^2\beta^2\|\theta_k^{(t,\tau)} - \bar{\theta}^{(t,\tau)}\|^2 + 64\lambda^2B^4\|\bar{\theta}^{(t,0)} - \bar{\theta}^{(t,\tau)}\|^2\right] \tag{A.45}$$

$$\overset{\text{Lemma 4 and 2}}{\leq} 2\left[(1+4\lambda\Gamma)^2\beta^2 4\eta^2I^2(1+4\lambda\Gamma)^2B^2 + 64\lambda^2B^4 2\eta^2I^2(1+4\lambda)^2B^2\right] \tag{A.46}$$

$$= 8\eta^2I^2(1+4\lambda)^2B^2\left[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2B^4\right]. \tag{A.47}$$

Therefore we have:

$$\left\|\frac{1}{K}\sum_{k=1}^{K}\left[\nabla\tilde{F}_k(\bar{\theta}^{(t,\tau)}, \bar{\theta}^{(t,\tau)}) - \nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\right]\right\|^2$$

$$\leq \frac{1}{K}\sum_{k=1}^{K}\|\nabla\tilde{F}_k(\bar{\theta}^{(t,\tau)}, \bar{\theta}^{(t,\tau)}) - \nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2. \tag{A.48}$$

$$\leq 8\eta^2I^2(1+4\lambda)^2B^2\left[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2B^4\right]. \tag{A.49}$$

Since each F_k is smooth with smoothness equal to $(1+4\lambda\Gamma)\beta + 8\lambda B^2$, $F(\theta) := \frac{1}{K}\sum_{k=1}^{K}F_k(\theta)$ is also $(1+4\lambda\Gamma)\beta + 8\lambda B^2$-smooth. Therefore:

$$\mathbb{E}[F(\bar{\theta}^{(t,\tau+1)})]$$

$$\leq \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] + \mathbb{E}[\langle\nabla F(\bar{\theta}^{(t,\tau)}), \bar{\theta}^{(t,\tau+1)} - \bar{\theta}^{(t,\tau)}\rangle] + \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\mathbb{E}[\|\bar{\theta}^{(t,\tau+1)} - \bar{\theta}^{(t,\tau)}\|^2] \tag{A.50}$$

$$= \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \eta\mathbb{E}\left[\left\langle\nabla F(\bar{\theta}^{(t,\tau)}), \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}; \zeta_k)\right\rangle\right]$$

$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}; \zeta_k)\right\|^2\right] \tag{A.51}$$

$$\overset{\mathbb{E}_{\zeta_k}[\nabla\tilde{F}_k(\theta,\theta';\zeta_k)]=\nabla\tilde{F}_k(\theta,\theta')}{=} \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \eta\mathbb{E}\left\langle\nabla F(\bar{\theta}^{(t,\tau)}), \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\right\rangle$$

$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}; \zeta_k)\right\|^2\right] \tag{A.52}$$

We have:

$$\eta^2\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}^{(t,0)};\zeta_k)\right\|^2\right] \tag{A.53}$$

$$\overset{(a)}{=}\eta^2\mathbb{E}[\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}^{(t,0)};\zeta_k)-\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}^{(t,0)})\|^2]+\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}^{(t,0)})\|^2 \tag{A.54}$$

$$=\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}^{(t,0)})\|^2+$$

$$\eta^2\mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\left(1+\frac{4\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\bar{\theta}^{(t,0)})-r_{k'}(\bar{\theta}^{(t,0)}))\right)(\nabla L_k(\theta_k^{(t,\tau)};\zeta_k)-\nabla L_k(\theta_k^{(t,\tau)}))\right\|^2\right] \tag{A.55}$$

$$\overset{(b)}{=}\eta^2\frac{1}{K^2}\sum_{k=1}^{K}\left|1+\frac{4\lambda}{K-1}\sum_{1\leq k'\neq k\leq K}(r_k(\bar{\theta}_k^{(t,0)})-r_{k'}(\bar{\theta}^{(t,0)}))\right|\mathbb{E}[\|\nabla L_k(\theta_k^{(t,\tau)};\zeta_k)-\nabla L_k(\theta_k^{(t,\tau)})\|^2]$$

$$+\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}_k^{(t,0)})\|^2 \tag{A.56}$$

$$\overset{(c),(d)}{\leq}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K}+\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)},\bar{\theta}_k^{(t,0)})\|^2 \tag{A.57}$$

Here (a) is because for any random vector X, we have $\mathbb{E}[\|X\|^2]=\mathbb{E}[\|X-\mathbb{E}[X]\|^2]+\|\mathbb{E}[X]\|^2$, (b) is due to the fact that $\mathbb{E}[\|\sum_{k=1}^{K}x_k-\mathbb{E}[x_k]\|^2]=\sum_{k=1}^{K}\mathbb{E}[\|x_k-\mathbb{E}[x_k]\|^2]$ for all $\{x_k\}_{k=1}^{K}$ which are independent (c) is due to $\forall\theta\in\mathcal{H}:|1+\frac{\lambda}{K-1}\sum_{k=1}^{K}(r_k(\theta)-r_k(\theta'))|\leq(1+\lambda\Gamma)$ and (d) is because of the bounded variance of stochastic gradients of $L_k:\mathbb{E}[\|\nabla L_k(\theta;\zeta_k)-\nabla L_k(\theta)\|^2]\leq\sigma^2$.

Thus:

$$\mathbb{E}[F(\bar{\theta}^{(t,\tau+1)})] \tag{A.58}$$

$$= \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \eta\langle\nabla F(\bar{\theta}^{(t,\tau)}), \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\rangle$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}_k^{(t,0)})\|^2 + \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K} \tag{A.59}$$

$$\overset{(e)}{=} \mathbb{E}[F(\bar{\theta}^{(t,\tau)})]$$
$$- \frac{\eta}{2}\Bigg[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2 + \|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2 - \|\nabla F(\bar{\theta}^{(t,\tau)}) - \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2\Bigg]$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}_k^{(t,0)})\|^2$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K} \tag{A.60}$$

$$= \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \frac{\eta}{2}\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2 + \frac{\eta}{2}\|\nabla F(\bar{\theta}^{(t,\tau)}) - \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2$$
$$- \frac{\eta - \eta^2((1+4\lambda\Gamma)\beta + 8\lambda B^2)}{2}\|\frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K} \tag{A.61}$$

$$\leq \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \frac{\eta}{2}\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2 + \frac{\eta}{2}\|\nabla F(\bar{\theta}^{(t,\tau)}) - \frac{1}{K}\sum_{k=1}^{K}\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})\|^2$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K} \qquad \text{(This is for the choice of } \eta \leq \tfrac{1}{(1+4\lambda\Gamma)L + 8\lambda B^2}\text{)}$$
$$\overset{(A.49)}{\leq} \mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \frac{\eta}{2}\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2 + 4\eta^3 I^2(1+4\lambda)^2B^2\Big[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2B^4\Big]$$
$$+ \frac{(1+4\lambda\Gamma)\beta + 8\lambda B^2}{2}\eta^2(1+4\lambda\Gamma)^2\frac{\sigma^2}{K}. \tag{A.62}$$

(e) is using $\|a-b\|^2 = \|a\|^2 + \|b\|^2 - 2\langle a, b\rangle$.

Rearranging and dividing both sides by $\frac{\eta}{2}$ and summing over $t \in \{1, \ldots, T\}$ synchronization

rounds and $\tau \in \{1, \dots, I\}$ local steps, and finally dividing by TI:

$$\begin{aligned}
\mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq \frac{2}{\eta}[\mathbb{E}[F(\bar{\theta}^{(t,\tau)})] - \mathbb{E}[F(\bar{\theta}^{(t,\tau+1)})]] \\
+ 2\eta^2 I^2 (1+4\lambda)^2 B^2 \left[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2 B^4\right] \\
+ ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2 \eta \frac{\sigma^2}{K} \qquad \text{(A.63)} \\
\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[\|\nabla F(\bar{\theta}^{(t,\tau)})\|^2] \leq 2\frac{1}{\eta TI}\mathbb{E}[(F(\bar{\theta}^{(1,1)})] - \mathbb{E}[F(\bar{\theta}^{(T,I)})]) \\
+ 2\eta^2 I^2 (1+4\lambda)^2 B^2 \left[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2 B^4\right] \\
+ ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2 \eta \frac{\sigma^2}{K} \qquad \text{(A.64)} \\
\overset{F^*:=\arg\min_\theta F(\theta)}{\leq} 2\frac{1}{\eta TI}(F(\bar{\theta}^{(1,1)}) - F^*) \\
+ 2\eta^2 I^2 (1+4\lambda)^2 B^2 \left[(1+4\lambda\Gamma)^2\beta^2 + 32\lambda^2 B^4\right] \\
+ ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2 \eta \frac{\sigma^2}{K}. \qquad \square
\end{aligned}$$

Theorem 2 (Convergence to a solution of federated learning)**.** *Let $\theta_k^{(t,\tau)}$ refer to the model of client k after t communication rounds and τ local steps and let $\bar{\theta}^{(t,\tau)} := \frac{1}{K}\sum_{k=1}^{K}\theta_k^{(t,\tau)}$. Let T be the total number of communication rounds and I be the number of local steps between each communication round. Under Assumption 1, for $\eta \leq \frac{1}{(1+4\lambda\Gamma)\beta+8\lambda B^2}$, the sequence of models $\{\bar{\theta}^{(t,\tau)}\}_{t\geq 0,\tau\geq 0}$ generated by Algorithm 1 satisfies:*

$$\begin{aligned}
\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[\|\nabla L(\bar{\theta}^{(t,\tau)})\|^2] \leq 2\frac{1}{\eta TI}(L(\bar{\theta}^{(1,1)}) - L^*) + \underbrace{32\lambda^2\Gamma^2 B^2}_{\substack{\textit{neighborhood of the solution} \\ \textit{relative to the heterogeneity}}} + \eta\frac{\sigma^2}{K}\xi_2 \qquad \text{(A.65)} \\
+ 8\beta^2\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2,
\end{aligned}$$

with $\xi_2 := ((1+4\lambda\Gamma)\beta + 8\lambda B^2)(1+4\lambda\Gamma)^2$.

If we further assume that $L(\theta)$ is μ-Polyak-Łojasiewicz (PL), that is $\forall\theta \in \mathcal{H} : \frac{1}{2}\|\nabla L(\theta)\| \geq \mu(L(\theta) - L^*)$, then we have that:

$$\begin{aligned}
\frac{1}{TI}\sum_{t=1}^{T}\sum_{\tau=1}^{I}\mathbb{E}[L(\bar{\theta}^{(t,\tau)}) - L^*] \leq 4\frac{\mu}{\eta TI}(L(\bar{\theta}^{(1,1)}) - L^*) \\
+ 2\eta\frac{\sigma^2}{K}\xi_2 + 16\beta^2\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2 + \underbrace{62\mu\lambda^2\Gamma^2 B^2}_{\textit{Loss in utility}}. \qquad \text{(A.66)}
\end{aligned}$$

The main difference from the proof of Theorem 1 is the bound on the difference between the applied gradient $\nabla\tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)})$ and the gradient of the original federated learning

problem $\nabla L_k(\bar{\theta}^{(t,\tau)})$:

$$\|\nabla \tilde{F}_k(\theta_k^{(t,\tau)}, \bar{\theta}^{(t,0)}) - \nabla L_k(\bar{\theta}^{(t,\tau)})\|^2$$

$$= \left\| \left(1 + 4\frac{\lambda}{K-1} \sum_{1\leq k'\neq k\leq K} [r_k(\bar{\theta}^{(t,0)}) - r_{k'}(\bar{\theta}^{(t,0)})]\right) \nabla L_k(\theta_k^{(t,\tau)}) - \nabla L_k(\bar{\theta}^{(t,\tau)}) \right\|^2 \tag{A.67}$$

$$\leq 2 \underbrace{\|\nabla L_k(\theta_k^{(t,\tau)}) - \nabla L_k(\bar{\theta}^{(t,\tau)})\|^2}_{\text{Controllable using smoothness}}$$

$$+ 2\left\| \left(4\frac{\lambda}{K-1} \sum_{1\leq k'\neq k\leq K} [r_k(\bar{\theta}^{(t,0)}) - r_{k'}(\bar{\theta}^{(t,0)})]\right) \nabla L_k(\theta_k^{(t,\tau)}) \right\|^2 \tag{A.68}$$

$$\overset{\|\nabla L_k(\theta)\|\leq B}{\leq} 2\beta^2 \|\theta_k^{(t,\tau)} - \bar{\theta}^{(t,\tau)}\|^2 + \frac{32\lambda^2}{(K-1)^2} \left| \left(\sum_{1\leq k'\neq k\leq K} [r_k(\bar{\theta}^{(t,0)}) - r_{k'}(\bar{\theta}^{(t,0)})] \right) \right|^2 B^2 \tag{A.69}$$

$$\overset{\text{Lemma 4}}{\leq} 8\beta^2\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2 + \frac{32\lambda^2}{(K-1)^2} \left| \left(\sum_{1\leq k'\neq k\leq K} [r_k(\bar{\theta}^{(t,0)}) - r_{k'}(\bar{\theta}^{(t,0)})] \right) \right|^2 B^2 \tag{A.70}$$

$$\leq 8\beta^2\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2 + \frac{32\lambda^2}{K-1} \sum_{1\leq k'\neq k\leq K} \underbrace{\left| r_k(\bar{\theta}^{(t,0)}) - r_{k'}(\bar{\theta}^{(t,0)}) \right|^2}_{\leq \Gamma^2} B^2 \tag{A.71}$$

$$\leq 32\lambda^2\Gamma^2 B^2 + 8\beta^2\eta^2 I^2 (1+4\lambda\Gamma)^2 B^2. \tag{A.72}$$

The results in Theorem 2 follows after applying the smoothness of L_k between two consecutive virtual models $\bar{\theta}^{(t,\tau)}$ and $\bar{\theta}^{(t,\tau+1)}$, similarly as in the proof of Theorem 1.

B Experiments

Computational resources All experiments have been run by simulating federated learning on one machine (no communication between nodes). The machine has a NVIDIA A10 GPU chip with a linux as an operating system. The results can be reproduced with the seeds [0, 42, 100, 200] for model initialization and the splitting of the data across clients. All experiments have been run for 10000 epochs with full batch gradients to ensure convergence of all methods.

Model and hyper parameters We learn a linear model and a convolutional layer neural network that classifies an image $x \in \mathbb{R}^{784}$ as a single label $y \in [62]$. We use SGD optimizer for local steps with variable learning rates tuned for each algorithm based on preliminary experiments. For q-FFL, we use a grid search on q in $\{0.001, 0.01, 0.1, 1, 2, 5, 10\}$ as proposed in the original work (Li et al., 2020b). We run EAGLE with multiple values of λ in $\{0.1, 0.3, 0.50.7, 1.0, 2.0, 3.0, 5.0\}$ to illustrate the utility-fairness trade-off.

Metrics In all experiments, we evaluate EAGLE against the baselines in terms of fairness—measured by the variance of loss gaps—and utility, quantified as the balanced accuracy across clients. Additionally, we report the performance of the best and worst clients (based on loss gaps) to assess whether fairness is achieved by improving the utility of the worst performing clients or we are in a leveling down regime.

B.1 Approximation of L_k^*

In the implementation of EAGLE, we start with an initialization phase in which clients run local training with the global model architecture and hyperparameters that are shared by the server. In practice, the client already has an approximation of L_k^* from previous local training before

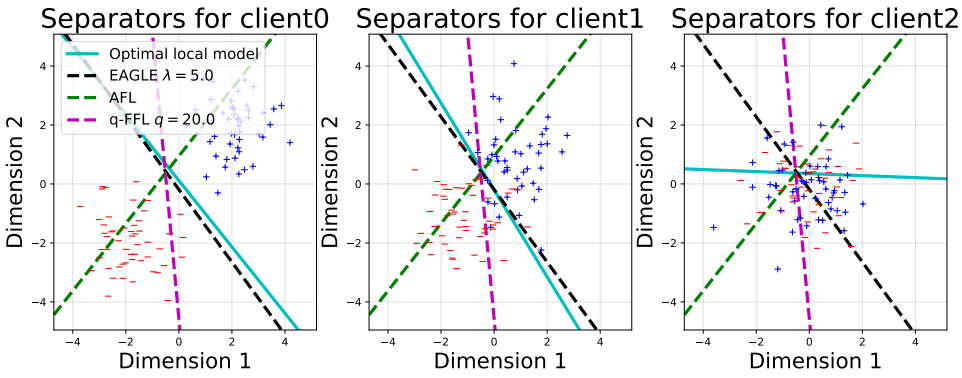


Figure 5: Synthetic data distributions of the three clients. Rotating data resulted in *client* 1 and *client* 2 sharing the same optimal separator which is different than the one for *client* 0. This difference resuls in conflicting gradient directions between local optimizers for different clients.

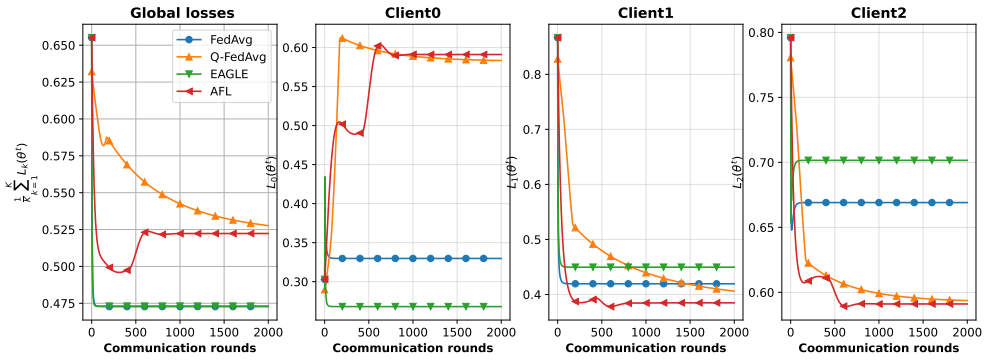


Figure 6: The optimization behavior of different algorithms on the proposed synthetic data, *AFL* and *q-FFL* deliberately increase the training loss of *client* 0 to balance it with those of *client* 2. And since *client* 1 and *client* 2 share the same optimal model and differ in data separability, the methods fail to balance their losses.

joining federated learning. Each client runs multiple epochs of SGD until one of the stopping criteria is met, the maximum number of allowed epochs is reached, or the training loss history curve flattens. This heuristic convergence test is not guaranteed to reach the global optimum, especially in nonconvex cases. For the experiments on linear models, we use a logistic regression model with nonseparable data, resulting in a strongly convex local objective and thus a single global optimum; therefore, our convergence check works in this case. We leave more elaborate strategies to approximate L_k^* for future work.

B.2 Synthetic data

In Figure 6, we highlight the optimization behavior of each algorithm, showing that having low optimal local loss leads to a disadvantage when enforcing *loss parity*.

B.3 EMNIST

Dirichlet splitting consists of generating a probability vector with the size of the number of clients using the Dirichlet distribution with a given parameter α. The parameter α controls the level of heterogeneity or uniformity in the generated probability vector, with $\alpha \to 0$ resulting in very heterogeneous probability vectors and $\alpha \to \infty$ in very uniform ones. To generate a data split across 10 clients, we generate probability vectors for each target value of $\{0, 1, \ldots, 61\}$ with

Table 4: Maximum, minimum and variance of loss gaps and accuracy different baselines with a linear model reported on the test split for a heterogeneous split $dir(\alpha = 0.1)$ with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 0.1$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	0.152 (± 0.207)	**-0.477 (± 0.155)**	0.687 (± 0.003)	0.037 (± 0.026)
FedAvg	0.106 (± 0.170)	-0.459 (± 0.211)	**0.692 (± 0.004)**	0.033 (± 0.021)
q-FFL $q = 1.0$	0.104 (± 0.178)	-0.464 (± 0.193)	0.692 (± 0.005)	0.032 (± 0.021)
q-FFL $q = 3.0$	0.109 (± 0.186)	-0.471 (± 0.178)	0.691 (± 0.004)	0.032 (± 0.022)
q-FFL $q = 5.0$	0.118 (± 0.189)	-0.471 (± 0.169)	0.689 (± 0.004)	0.033 (± 0.023)
EAGLE $\lambda = 0.1$	0.096 (± 0.155)	-0.450 (± 0.209)	0.691 (± 0.007)	0.030 (± 0.019)
EAGLE $\lambda = 1.0$	**0.073 (± 0.123)**	-0.381 (± 0.199)	0.687 (± 0.003)	0.020 (± 0.013)
EAGLE $\lambda = 2.0$	0.094 (± 0.166)	-0.355 (± 0.268)	0.681 (± 0.007)	**0.019 (± 0.010)**
EAGLE $\lambda = 3.0$	0.107 (± 0.171)	-0.335 (± 0.275)	0.677 (± 0.007)	0.019 (± 0.010)
EAGLE $\lambda = 5.0$	0.203 (± 0.181)	-0.245 (± 0.284)	0.650 (± 0.007)	0.020 (± 0.011)

Table 5: Maximum, minimum and variance of loss gaps and accuracy different baselines with a linear model reported on the test split for a less heterogeneous split $dir(\alpha = 0.5)$ with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 0.5$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-0.949 (± 0.106)	-1.710 (± 0.054)	0.686 (± 0.004)	0.053 (± 0.012)
FedAvg	-0.961 (± 0.094)	-1.715 (± 0.044)	**0.691 (± 0.004)**	0.053 (± 0.014)
q-FFL $q = 1$	-0.963 (± 0.096)	-1.717 (± 0.048)	0.690 (± 0.004)	0.053 (± 0.013)
q-FFL $q = 3$	-0.965 (± 0.097)	-1.720 (± 0.051)	0.690 (± 0.004)	0.053 (± 0.013)
q-FFL $q = 5.0$	-0.965 (± 0.098)	**-1.723 (± 0.051)**	0.690 (± 0.003)	0.053 (± 0.012)
EAGLE $\lambda = 0.1$	-0.989 (± 0.086)	-1.719 (± 0.045)	**0.691** (± 0.002)	0.050 (± 0.013)
EAGLE $\lambda = 1.0$	-1.045 (± 0.066)	-1.617 (± 0.046)	0.679 (± 0.005)	0.032 (± 0.011)
EAGLE $\lambda = 2.0$	**-1.046 (± 0.062)**	-1.491 (± 0.037)	0.662 (± 0.009)	0.022 (± 0.008)
EAGLE $\lambda = 3.0$	-0.990 (± 0.085)	-1.410 (± 0.028)	0.650 (± 0.013)	0.018 (± 0.006)
EAGLE $\lambda = 5.0$	-0.926 (± 0.124)	-1.347 (± 0.038)	0.640 (± 0.016)	**0.016 (± 0.006)**

the size of the number of clients for a given α parameter. Following the the generated probability vectors we partition the data of each target value across clients. No data points are repeated in multiple clients and the probability vectors are generated independently.

B.4 Additional experimental results

B.4.1 Experiment on EMNIST data with dirichlet splitting

Table 6: Maximum, minimum and variance of loss gaps and accuracy different baselines with a linear model reported on the test split for a moderately heterogeneous split $dir(\alpha = 1.0)$ with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 1.0$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-1.495 (± 0.141)	-2.011 (± 0.067)	0.688 (± 0.002)	0.028 (± 0.016)
FedAvg	-1.503 (± 0.132)	-2.015 (± 0.074)	0.693 (± 0.002)	0.028 (± 0.018)
q-FFL $q = 1$	-1.505 (± 0.134)	-2.015 (± 0.072)	0.693 (± 0.002)	0.028 (± 0.017)
q-FFL $q = 3$	-1.507 (± 0.135)	-2.016 (± 0.071)	0.692 (± 0.002)	0.028 (± 0.017)
q-FFL $q = 5.0$	-1.508 (± 0.136)	-2.017 (± 0.070)	0.692 (± 0.002)	0.028 (± 0.017)
EAGLE $\lambda = 0.1$	**-1.513 (± 0.131)**	**-2.019 (± 0.069)**	**0.693 (± 0.001)**	0.028 (± 0.017)
EAGLE $\lambda = 1.0$	-1.497 (± 0.150)	-1.972 (± 0.034)	0.685 (± 0.003)	0.023 (± 0.014)
EAGLE $\lambda = 2.0$	-1.457 (± 0.163)	-1.898 (± 0.035)	0.670 (± 0.008)	0.019 (± 0.011)
EAGLE $\lambda = 3.0$	-1.407 (± 0.160)	-1.849 (± 0.047)	0.657 (± 0.012)	0.017 (± 0.009)
EAGLE $\lambda = 5.0$	-1.333 (± 0.155)	-1.780 (± 0.069)	0.639 (± 0.015)	**0.015 (± 0.008)**

Table 7: Maximum, minimum and variance of loss gaps and accuracy different baselines with a linear model reported on the test split for homogeneous split $dir(\alpha = 2000.0)$ (IID data) with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 2000.0$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-2.075 (± 0.126)	-2.571 (± 0.087)	0.692 (± 0.001)	0.017 (± 0.010)
FedAvg	-2.076 (± 0.126)	-2.570 (± 0.087)	**0.693 (± 0.002)**	0.017 (± 0.010)
qFL $q = 1.0$	-2.072 (± 0.126)	-2.565 (± 0.087)	0.692 (± 0.002)	**0.017 (± 0.010)**
qFL $q = 3.0$	-2.072 (± 0.126)	-2.566 (± 0.086)	0.692 (± 0.002)	0.017 (± 0.010)
qFL $q = 5.0$	-2.072 (± 0.126)	-2.566 (± 0.086)	0.692 (± 0.001)	0.017 (± 0.010)
EAGLE $\lambda = 0.1$	**-2.080 (± 0.127)**	**-2.581 (± 0.088)**	0.692 (± 0.001)	0.018 (± 0.010)
EAGLE $\lambda = 1.0$	-2.079 (± 0.127)	-2.580 (± 0.088)	0.692 (± 0.000)	0.018 (± 0.010)
EAGLE $\lambda = 2.0$	-2.059 (± 0.132)	-2.554 (± 0.092)	0.684 (± 0.002)	0.018 (± 0.010)
EAGLE $\lambda = 3.0$	-2.029 (± 0.137)	-2.521 (± 0.092)	0.673 (± 0.005)	0.017 (± 0.010)
EAGLE $\lambda = 5.0$	-2.068 (± 0.130)	-2.572 (± 0.085)	0.690 (± 0.001)	0.018 (± 0.011)

Table 8: Maximum, minimum and variance of loss gaps and accuracy different baselines with a CNN model reported on the test split for a heterogeneous split $dir(\alpha = 0.1)$ with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 0.1$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	0.144 (± 0.096)	-0.497 (± 0.159)	0.843 (± 0.001)	0.036 (± 0.018)
FedAvg	0.294 (± 0.196)	**-0.512 (± 0.131)**	0.844 (± 0.005)	0.047 (± 0.020)
q-FFL $q = 1.0$	0.097 (± 0.067)	-0.473 (± 0.120)	0.835 (± 0.003)	0.028 (± 0.013)
q-FFL $q = 3.0$	0.244 (± 0.089)	-0.330 (± 0.147)	0.800 (± 0.009)	0.030 (± 0.015)
q-FFL $q = 5.0$	0.386 (± 0.089)	-0.201 (± 0.159)	0.765 (± 0.011)	0.032 (± 0.017)
EAGLE $\lambda = 0.1$	0.091 (± 0.136)	-0.506 (± 0.101)	**0.848 (± 0.003)**	0.029 (± 0.017)
EAGLE $\lambda = 1.0$	**0.063 (± 0.058)**	-0.348 (± 0.103)	0.844 (± 0.009)	**0.013 (± 0.003)**
EAGLE $\lambda = 2.0$	0.178 (± 0.118)	-0.315 (± 0.084)	0.838 (± 0.008)	0.017 (± 0.010)
EAGLE $\lambda = 3.0$	0.153 (± 0.101)	-0.294 (± 0.077)	0.834 (± 0.007)	0.014 (± 0.007)
EAGLE $\lambda = 5.0$	0.192 (± 0.137)	-0.268 (± 0.083)	0.828 (± 0.007)	0.017 (± 0.009)

Table 9: Maximum, minimum and variance of loss gaps and accuracy different baselines with a CNN model reported on the test split for a less heterogeneous split $dir(\alpha = 0.5)$ with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 0.5$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-0.416 (± 0.132)	-1.480 (± 0.674)	0.841 (± 0.004)	0.110 (± 0.137)
FedAvg	-0.409 (± 0.130)	-1.477 (± 0.675)	0.840 (± 0.003)	0.112 (± 0.140)
q-FFL $q = 1.0$	-0.534 (± 0.111)	-1.558 (± 0.670)	0.843 (± 0.002)	0.109 (± 0.141)
q-FFL $q = 3.0$	-0.476 (± 0.115)	-1.524 (± 0.681)	0.833 (± 0.002)	0.114 (± 0.145)
q-FFL $q = 5.0$	-0.416 (± 0.108)	-1.470 (± 0.679)	0.820 (± 0.005)	0.115 (± 0.144)
EAGLE $\lambda = 0.1$	**-0.546 (± 0.103)**	**-1.567 (± 0.671)**	**0.850 (± 0.002)**	0.108 (± 0.140)
EAGLE $\lambda = 1.0$	-0.502 (± 0.080)	-1.202 (± 0.289)	0.832 (± 0.011)	0.044 (± 0.037)
EAGLE $\lambda = 2.0$	-0.483 (± 0.069)	-1.078 (± 0.257)	0.823 (± 0.004)	0.032 (± 0.029)
EAGLE $\lambda = 3.0$	-0.466 (± 0.068)	-1.011 (± 0.247)	0.816 (± 0.005)	0.028 (± 0.026)
EAGLE $\lambda = 5.0$	-0.424 (± 0.071)	-0.938 (± 0.223)	0.805 (± 0.011)	**0.024 (± 0.022)**

Table 10: Maximum, minimum and variance of loss gaps and accuracy different baselines with CNN model reported on the test split for a moderately heterogeneous split $dir(\alpha = 1.0)$ with EMNIST dataset, results are aggregated over four independent repetitions

Algorithm	$\alpha = 1$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-0.726 (± 0.078)	-1.367 (± 0.082)	0.845 (± 0.003)	0.038 (± 0.012)
FedAvg	-0.742 (± 0.052)	-1.400 (± 0.065)	0.843 (± 0.003)	0.039 (± 0.014)
q-FFL $q = 1.0$	-0.760 (± 0.077)	-1.431 (± 0.096)	0.846 (± 0.004)	0.041 (± 0.016)
q-FFL $q = 3.0$	-0.764 (± 0.079)	-1.437 (± 0.100)	0.842 (± 0.003)	0.042 (± 0.018)
q-FFL $q = 5.0$	-0.745 (± 0.081)	-1.432 (± 0.102)	0.839 (± 0.004)	0.043 (± 0.018)
EAGLE $\lambda = 0.1$	**-0.794 (± 0.073)**	**-1.461 (± 0.097)**	**0.851 (± 0.001)**	0.041 (± 0.017)
EAGLE $\lambda = 1.0$	-0.744 (± 0.081)	-1.294 (± 0.063)	0.833 (± 0.012)	0.029 (± 0.010)
EAGLE $\lambda = 2.0$	-0.621 (± 0.115)	-1.062 (± 0.101)	0.809 (± 0.014)	0.019 (± 0.007)
EAGLE $\lambda = 3.0$	-0.567 (± 0.136)	-0.939 (± 0.126)	0.800 (± 0.016)	0.016 (± 0.005)
EAGLE $\lambda = 5.0$	-0.543 (± 0.151)	-0.864 (± 0.117)	0.789 (± 0.017)	**0.012 (± 0.003)**

Table 11: Maximum, minimum and variance of loss gaps and accuracy different baselines with a linear model reported on the test split for homogeneous split $dir(\alpha = 2000.0)$ (IID data) with EMNIST dataset, results are aggregated over four independent repetitions.

Algorithm	$\alpha = 2000$			
	$\max_{k\in[K]} r_k(\theta) \downarrow$	$\min_{k\in[K]} r_k(\theta) \downarrow$	accuracy $\uparrow$	$\mathbb{V}_{k\in[K]} r_k(\theta) \downarrow$
AFL	-1.140 (± 0.063)	-1.591 (± 0.060)	0.845 (± 0.002)	0.015 (± 0.002)
FedAvg	-1.140 (± 0.063)	-1.591 (± 0.060)	0.845 (± 0.002)	0.015 (± 0.002)
qFL $q = 1.0$	-1.170 (± 0.053)	-1.615 (± 0.053)	0.849 (± 0.002)	0.015 (± 0.001)
qFL $q = 3.0$	-1.180 (± 0.062)	-1.605 (± 0.069)	0.845 (± 0.004)	0.014 (± 0.002)
qFL $q = 5.0$	-1.193 (± 0.040)	-1.618 (± 0.054)	0.846 (± 0.003)	0.014 (± 0.002)
EAGLE $\lambda = 0.1$	**-1.217 (± 0.051)**	**-1.650 (± 0.060)**	**0.850 (± 0.000)**	0.014 (± 0.002)
EAGLE $\lambda = 1.0$	-1.209 (± 0.050)	-1.637 (± 0.060)	0.848 (± 0.001)	0.014 (± 0.002)
EAGLE $\lambda = 2.0$	-1.162 (± 0.051)	-1.582 (± 0.062)	0.838 (± 0.006)	0.013 (± 0.002)
EAGLE $\lambda = 3.0$	-0.962 (± 0.164)	-1.364 (± 0.173)	0.801 (± 0.026)	**0.013 (± 0.002)**
EAGLE $\lambda = 5.0$	-0.570 (± 0.074)	-0.976 (± 0.078)	0.690 (± 0.022)	0.013 (± 0.003)