On the Power of Adaptive Weighted Aggregation in Heterogeneous Federated Learning and Beyond

Dun Zeng
University of Electronic Science
and Technology of China
Chengdu, China

Zenglin Xu*
Fudan University
Shanghai Academy of AI for Science
Shanghai, China
zenglinxu@fudan.edu.cn

Shiyu Liu
Southwestern University
of Finance and Economics
Chengdu, China

Yu Pan
Harbin Institute of Technology
Shenzhen, China

Qifan Wang
Meta AI
Menlo Park, USA

Xiaoying Tang
Chinese University of Hong Kong
Shenzhen, China

Abstract

Federated averaging (FedAvg) is the most fundamental algorithm in Federated learning (FL). Previous theoretical results assert that FedAvg convergence and generalization degenerate under heterogeneous clients. However, recent empirical results show that FedAvg can perform well in many real-world heterogeneous tasks. These results reveal an inconsistency between FL theory and practice that is not fully explained. In this paper, we show that common heterogeneity measures contribute to this inconsistency based on rigorous convergence analysis. Furthermore, we introduce a new measure *client consensus dynamics* and prove that *FedAvg can effectively handle client heterogeneity when an appropriate aggregation strategy is used.* Building on this theoretical insight, we present a simple and effective FedAvg variant termed FedAWARE. Extensive experiments on three datasets and two modern neural network architectures demonstrate that FedAWARE ensures faster convergence and better generalization in heterogeneous client settings. Moreover, our results show that FedAWARE can significantly enhance the generalization performance of advanced FL algorithms when used as a plug-in module. The source code is available at https://github.com/dunzeng/FedAWARE.

1 INTRODUCTION

Federated Learning (FL) is an emerging distributed training paradigm (Kairouz et al., 2021; Zhang et al., 2023a, 2024), where a central server orchestrates multiple clients jointly to optimize a machine learning model. The pioneering algorithm, federated averaging (FedAvg) (McMahan et al., 2017), provides the general local-update framework. It only requires infrequent communication between a server and clients. Thus, FedAvg is especially suitable for FL settings where communication costs are a major bottleneck. FedAvg's simplicity and empirical effectiveness made it the basis of almost all subsequent FL algorithms.

However, the convergence and generalization performance of FedAvg is hindered by client heterogeneity. Specifically, as local updates on clients become more diverse (i.e., as data heterogeneity increases), FedAvg may require more communication rounds to converge. Additionally, this process often results in unstable generalization performance, as seen in the fluctuations of test accuracy (the "spikes" problem). To understand the effects of heterogeneity, plenty of theoretical results (Zhao et al., 2018; Karimireddy et al., 2020; Jhunjhunwala et al., 2022) provided a clear explanation via rigorous convergence analysis. These results often use common heterogeneity measures, such as bounded gradient dissimilarity (Li et al., 2020b), and implicitly assume that federated learning is continuously impacted by worst-case heterogeneity. However, this theoretical understanding does not always align with empirical findings (Reddi et al., 2020; Charles et al., 2021; Wu et al., 2023; Wang et al., 2024), which show that FedAvg can perform comparably to advanced FL methods (Karimireddy et al., 2020) in real-world heterogeneous tasks (Caldas et al., 2018). This dis-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

crepancy may arise because FedAvg only encounters severe heterogeneity in a small fraction of the training process (Wang et al., 2024). These results highlight an inconsistency between FL theory and practice. Understanding this gap is crucial for designing and evaluating future federated algorithms. This motivates the question *are we overlooking key properties that could explain why FedAvg and its variants perform well in practice under general heterogeneous conditions?*

We intend to answer the question by revisiting the training dynamics of FedAvg via convergence analysis. Firstly, to accurately describe the heterogeneity impacts, we introduce a concept of *client consensus dynamics*, defined as the cumulative expected norm of aggregated local updates over the communication rounds. Through rigorous theoretical analysis, we show that *the impact of data heterogeneity on FL training can be mitigated if the attained client consensus dynamics is small* (see Theorem 4.2). This suggests that coordinating heterogeneity impacts through carefully designed server-side aggregation could lead to faster convergence of FL training, as shown in Corollary 4.1. Furthermore, we induce a heterogeneity measure, *local update diversity*, to quantify the divergence between local and global updates in practice. We state that *FL is amenable to better generalization if the training procedure attains high local update diversity dynamics.*

Based on these analyses, we propose a simple and effective variant of FedAvg termed Federated Adaptive Weighted AggREgator (Fed**AWARE**). Through intensive experiments on training modern neural networks using image and text datasets, we demonstrate that FedAWARE achieves faster convergence and better generalization than most baselines under heterogeneous clients. Furthermore, the AWARE module works with existing federated algorithms, significantly enhancing their generalization performance in data heterogeneity by improving the local update diversity dynamics. This paper contributes key insights into the convergence of FedAvg variants. The results can guide the design and evaluation of future federated algorithms in heterogeneous optimization.

2 PRELIMINARIES & RELATED WORKS

We consider a standard cross-device FL task (Kairouz et al., 2021; Wu et al., 2024), which minimizes a finite sum of local empirical objectives:

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) := \sum\nolimits_{i=1}^{N} \boldsymbol{\lambda}_i f_i(\boldsymbol{x}) := \sum\nolimits_{i=1}^{N} \boldsymbol{\lambda}_i \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[F_i(\boldsymbol{x},\xi_i)],$$

where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ is parameters of machine learning model, $f(\boldsymbol{x})$ is the global objective weighted by $\boldsymbol{\lambda}_i(s.t., \sum_{i=1}^{N} \boldsymbol{\lambda}_i = 1, \boldsymbol{\lambda}_i \geq 0, \forall i)$, ξ_i is stochastic batch data, and $\mathcal{D}_i$ denotes dataset on the i-th client ($i \in \{1, 2, \dots, N\}$). The FedAvg minimizes the global objective involving alternative client optimization and server optimization procedures (Reddi et al., 2020; McMahan et al., 2017):

$$\begin{aligned} &\textbf{Client: } \boldsymbol{g}_i^t = \boldsymbol{x}_i^{t,K} - \boldsymbol{x}_i^{t,0} = \eta_l \sum\nolimits_{k=0}^{K-1} \nabla F_i(\boldsymbol{x}_i^{t,k}); \\ &\textbf{Server: } \boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \sum\nolimits_{i=1}^{N} \boldsymbol{\lambda}_i \boldsymbol{g}_i^t = \boldsymbol{x}^t - \eta_g \boldsymbol{G}^t \end{aligned} \tag{1}$$

where $\nabla F_i(\boldsymbol{x})$ denotes stochastic gradients over a mini-batch of samples, $\boldsymbol{G}^t$ is the *pseudo-gradient* for global gradient descent, $\boldsymbol{x}_i^{t,k}$ denotes client i's model after the k local update steps at the t-th communication round, and η_l is the client learning rate. The federated optimization framework covers a broader range of subsequent FL algorithms (McMahan et al., 2017; Reddi et al., 2020; Wang et al., 2022, 2021).

Our goal is to train a robust and well-generalized global machine-learning model under the potential negative impacts of heterogeneous clients. Therefore, this work is related to previous studies on heterogeneous federated optimization and adaptive aggregation strategies.

Heterogeneous federated optimization The basic federated optimization algorithm, FedAvg (McMahan et al., 2017), significantly reduces communication costs. Subsequent works built upon FedAvg to address challenges related to convergence guarantees and heterogeneity issues. For example, some approaches introduced a regularization term in the client objectives (Li et al., 2020b), while others incorporated server momentum (Hsu et al., 2019). Several studies have analyzed the convergence rate of FedAvg and demonstrated its degradation with system heterogeneity (Li et al., 2020b; Wang et al., 2019) and statistical heterogeneity (Zhao et al., 2018; Khaled et al., 2019). SCAFFOLD (Karimireddy et al., 2020) utilizes control variates to mitigate client drift and achieve convergence rates independent of the level of heterogeneity. FedNova (Wang et al., 2020) addresses objective inconsistency issues arising from system heterogeneity through local update regularization. Besides, adaptive methods (Zaheer et al., 2018; Reddi et al., 2019; Xie et al., 2019) have proven effective in non-convex optimizations. In the context of federated optimization, FedYogi (Reddi et al., 2020; Zaheer et al., 2018) and FedAMS (Acar et al., 2020) are representative adaptive federated optimization algorithms that incorporate Adam-like momentum and adaptive terms to address heterogeneity issues. For more detailed comparisons, we refer to the survey (Kairouz et al., 2021). These related works demonstrate the ongoing efforts to address heterogeneity issues in FL.

Adaptive weighting in FL The aggregation weights typically represent the importance of each local function in the FL global objective. Local function reweighting scheme has been adopted to improve the fairness (Li et al., 2019a; Mohri et al., 2019), robustness (Li et al., 2020a), and generalization (Li et al., 2023) via adjusting λ for FedAvg. However, existing adaptive weighting strategies are still based on heuristics. They typically assign a score to each client based on the local dataset properties (McMahan et al., 2017; Zhao and Shen, 2024; Ye et al., 2023; Li et al., 2023), local empirical loss (Li et al., 2019a, 2020a), or local updates information (Chen et al., 2024). Then, the aggregation of each client depends on the normalized score among a certain set of clients. Despite these methods being empirically efficient, they provide no explicit theoretical objective for the weight design. In contrast, FedAWARE's non-heuristic weight strategy relies on a clear objective with a closed-form solution. Moreover, we present a unified convergence analysis, and the results cover a broader range of FedAvg variants.

Algorithm 1 FedAWARE

```
Require: $\boldsymbol{x}^0, \boldsymbol{m}^0, \alpha$
 1: for round $t \in [T]$ do
 2:     Server sample clients $S^t$ and broadcast $\boldsymbol{x}^t$
 3:     for client $i \in S^t$ in parallel do
 4:         $\boldsymbol{x}_i^{t,0} = \boldsymbol{x}^t$
 5:         for local update step $k \in [K]$ do
 6:             $\boldsymbol{x}_i^{t,k} = \boldsymbol{x}_i^{t,k-1} - \eta_l \nabla F_i(\boldsymbol{x}_i^{t,k-1})$
 7:         end for
 8:         Client uploads local updates $\boldsymbol{g}_i^t = \boldsymbol{x}^{t,0} - \boldsymbol{x}^{t,K}$
 9:     end for
10:     Server updates momentum $\boldsymbol{m}_i^t$ by Eq. (3)
11:     Server computes $\boldsymbol{\lambda}^t$ by Eq. (2)
12:     Server computes $\boldsymbol{d}^t = \sum_{i=1}^N \boldsymbol{\lambda}_i^t \boldsymbol{m}_i^t$
13:     Server updates $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \boldsymbol{d}^t$
14: end for
```

3 PROPOSED ALGORITHM: FedAWARE

We present the details of FedAWARE in Algorithm 1. It suggests the FL server conducts gradient descent with a simple *adaptive averaging* of *moving-averaged* local updates $\boldsymbol{d}^t = \sum_{i=1}^N \boldsymbol{\lambda}_i^t \boldsymbol{m}_i^t$ by solving

$$\boldsymbol{\lambda}^t = \min_{\boldsymbol{\lambda}} \Big\{ \| \sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{m}_i^t \|^2 \big| \sum_{i=1}^N \boldsymbol{\lambda}_i = 1, \boldsymbol{\lambda}_i \geq 0 \ \forall i \Big\}, \quad (2)$$

where $\|\cdot\|$ denotes ℓ_2 norm and $\boldsymbol{m}_i^t$ is moving-averaged local updates $\boldsymbol{g}_i^t$ for the i-th client at communication round t. Specifically, this algorithm involves two vital components. FedAWARE can also work with other advanced federated algorithms.

Moving-averaged local updates supports for partial client participation We use historical moving-averaged local updates to approximate the local updates of the nonparticipants at a round. In detail, we use a coefficient α to control the approximation with update rules as:

$$\boldsymbol{m}_i^t = \begin{cases} (1-\alpha)\boldsymbol{m}_i^{t-1} + \alpha \boldsymbol{g}_i^t, & \text{if } i \in S^t \\ \boldsymbol{m}_i^{t-1}, & \text{if } i \notin S^t, \end{cases} \quad (3)$$

where S^t is a set of selected clients at round t. Moreover, the moving-averaged local updates can further stabilize the degree of local heterogeneity as discussed in momentum-based FL methods (Hsu et al., 2019).

Adaptive aggregation via global norm minimization In Eq. (2), we consider inputted $\{\boldsymbol{m}_i^t\}_{i\in[N]}$ as base vectors in d-dimensional linear space. This constrained minimization problem involves finding a minimum-norm point in the convex hull. Therefore, the attained estimates $\boldsymbol{d}$ is a vector from initial model parameters to this minimum-norm point. Besides, we particularly consider the non-convex optimization problem in FL, where the dimension of gradients can be millions due to the neural network scale (i.e., $d \gg N$). We use the Frank-Wolfe algorithm (Jaggi, 2013) to solve it. Moreover, we argue that the norm of the global estimate is always non-zero during the whole FL training process, i.e., $\|\boldsymbol{d}^t\| > 0$ for all $t \in [T]$. In other words, these cases mean that all the vectors should be *linearly independent* (Greub, 2012). Hence, we state that Algorithm 1 does not fail by reaching some corner cases with $\|\boldsymbol{d}^t\| = 0$ in training neural network practices.

Plugging AWARE in advanced federated algorithms AWARE refers to the server-side aggregation module in Lines 10-12, Algorithm 1, which outputs a pseudo-gradient $\boldsymbol{d}^t$. It can enhance the generalization performance of existing federated algorithms by using the following extension:

Proposition 3.1 (AWARE extension) *Given a federated optimization method updating the global model by* $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \tilde{\boldsymbol{d}}^t$*, where* $\tilde{\boldsymbol{d}}^t$ *is estimated by the method. We project the* $\tilde{\boldsymbol{d}}^t$ *to the direction of AWARE* $\boldsymbol{d}^t$ *by computing* $\boldsymbol{d}_{proj}^t = \frac{\langle \tilde{\boldsymbol{d}}^t, \boldsymbol{d}^t \rangle}{\langle \boldsymbol{d}^t, \boldsymbol{d}^t \rangle} \boldsymbol{d}^t$*, and then conduct* $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \boldsymbol{d}_{proj}^t$.

This procedure only modifies the server-side gradient descent of applied algorithms to the direction that enhances generalization. We elaborate on its insights in Section 4.3 and evaluate it in Section 5.

Terms	Definition	References
grad. dissimilarity	$\mathbb{E}\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \sigma_g^2$	(Woodworth et al., 2020)
grad. diversity	$\mathbb{E}\|\nabla f_i(\boldsymbol{x})\| \leq \gamma^2 \|\nabla f(\boldsymbol{x})\|$	(Koloskova et al., 2020)
general grad. diversity	$\mathbb{E}\|\nabla f_i(\boldsymbol{x})\| \leq \gamma^2 \|\nabla f(\boldsymbol{x})\| + \sigma_g^2$	(Li et al., 2020b)
grad. norm	$\mathbb{E}\|\nabla f_i(\boldsymbol{x})\| \leq \sigma_g^2$	(Li et al., 2019b)

Table 1: Summary of data heterogeneity measures, adapted from Table 6 (Kairouz et al., 2021).

4 CONVERGENCE GUARANTEES

In this section, we first provide deep insights into FedAvg's convergence using empirical properties. Then, by comparing with FedAvg, we theoretically demonstrate that using FedAWARE improves FL convergence. Then, we connect the results with existing theory to illustrate the generalization benefits.

4.1 Common Theoretical Analysis of FedAvg

We begin our analysis with the analytic assumption. Conventional convergence analysis typically rely on common non-convex optimization assumptions (Reddi et al., 2020; Wang et al., 2022; Acar et al., 2020) on local objectives $f_i(\boldsymbol{x}), i \in [N]$:

Assumption 4.1 (Bounded dissimilarity) *We assume the averaged global variance is bounded, i.e.,* $\sum_{i=1}^N \boldsymbol{\lambda}_i \mathbb{E}\left\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right\|^2 \leq \sigma_g^2$ *for all* $x \in \mathcal{X}$.

Assumption 4.2 (Smoothness) *Each objective* $f_i(\boldsymbol{x})$ *for all* $i \in [N]$ *is* L*-smooth, inducing that for all* $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, *it holds* $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.

Assumption 4.3 (Unbiasedness) *For each* $i \in [N]$ *and* $\boldsymbol{x} \in \mathbb{R}^d$, *we assume the access to an unbiased stochastic gradient* $\nabla F_i(\boldsymbol{x}, \xi_i)$ *of client's true gradient* $\nabla f_i(\boldsymbol{x})$, *i.e.,*$\mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[\nabla F_i(\boldsymbol{x}, \xi_i)\right] = \nabla f_i(\boldsymbol{x})$. *The function* f_i *have* σ_l*-bounded (local) variance i.e.,*$\mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[\|\nabla F_i(\boldsymbol{x}, \xi_i) - \nabla f_i(\boldsymbol{x})\|^2\right] \leq \sigma_l^2$.

The smoothness and unbiasedness are common assumptions in stochastic optimization analysis. The first assumption is specially made to analyze the convergence of FedAvg under heterogeneous clients. In Table 1, we summarize the analogous heterogeneity measures used in the literature. We note that the existing convergence analysis of FedAvg has a similar dependency on σ_g^2, though σ_g's definition is different.

Pessimistic convergence of FedAvg These assumptions can be pessimistic in practice (Wang et al., 2024). Under the above assumptions, previous works derived an upper bound for the optimization error with non-convex objective functions, for example Jhunjhunwala et al. (2022):

Theorem 4.1 *Under Assumptions 4.1 to 4.3, if FedAvg learning rates satisfy* $\eta_l \leq 1/8LK, \eta \leq 1/24LK$, *then the global gradient norm* $\min_t \mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2$ *can be upper bounded by*

$$\mathcal{O}\left(\frac{F^*}{\eta K T}\right) + \mathcal{O}\left(\eta\sigma_l^2 + \eta_l^2 K \sigma_l^2\right) + \mathcal{O}\left(\eta_l^2 K^2 \sigma_g^2\right), \quad (4)$$

where $\eta = \eta_l \eta_g$, $f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*) \leq F^*$ *and* $\boldsymbol{x}^* = \arg\min f(\boldsymbol{x})$.

Theorem 4.1 shows that when the learning rates are fixed and the communication round T is limited, data heterogeneity always introduces an additional term $\mathcal{O}(\sigma_g^2)$ to the optimization error bound. Under heterogeneous clients, it indicates that FedAvg cannot outperform the simple mini-batch SGD (Woodworth et al., 2020) or vanilla GD (Khaled et al., 2020). Meanwhile, these upper bounds match a lower bound of FedAvg (Glasgow et al., 2022), suggesting they are tight in the worst case. Therefore, we do not improve these bounds as they are already tight. Instead, we argue that the common heterogeneity measures are too pessimistic to cover the FedAvg practice, as the training dynamics of FedAvg do not always match the worst-case scenarios. We contend that existing convergence analyses overlook key properties in FedAvg's practical implementation, which are crucial for guiding the design and evaluation of federated algorithms.

4.2 Convergence of FedAWARE under Client Consensus Dynamics

We extend the advanced notion of heterogeneity, called *client consensus hypothesis* (Wang et al., 2024). This hypothesis states that *the data heterogeneity does not have any negative impacts on the convergence of FedAvg if the norm of averaged local updates at* ***the global optimum*** *is close to zero.* This statement is proven under the strong convexity assumption. In contrast, we are interested in the non-convex optimization problem for providing accurate statements on the training of neural networks. To this end, we propose the *client consensus dynamics* as shown below:

Property 4.1 (Client Consensus Dynamics) *The norm of* $\boldsymbol{\lambda}$*-averaged local updates (pseudo-gradient on server-size optimization in Eq.* (1)*)*

$$\rho^t(\boldsymbol{\lambda}) \triangleq \|\mathbb{E}[\boldsymbol{G}^t]\|^2 = \left\|\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{g}_i^t\right]\right\|^2$$

decays along with federated optimization procedures, i.e., $\frac{1}{T}\sum_{t=1}^T \rho^t(\boldsymbol{\lambda}) \leq \mathcal{O}(T^{-c})$ *and* $c > 0$ *in heterogeneous federated optimization.*

We argue that the *client consensus dynamics* decays with FL convergence, as the global model typically converges to the points within the convex hull formed by heterogeneous local solutions. And, the local updates $\boldsymbol{g}_i^t$ reduces as it is closing to the local minimum along with the FL procedure. We provide empirical observations on this property in Figure 1. Moreover, we note that *client consensus dynamics* is a weaker assumption than the *client consensus hypothesis* as we do not assume the information at the global minimum. It allows us to analyze the FedAvg training dynamics without using pessimistic assumptions. Now, we derive a new convergence analysis of FedAvg:

Theorem 4.2 *Under assumptions 4.2, 4.3 and property 4.1, there exists static FedAvg learning rates η_l, η_g such that the global gradient norm $\min_t \mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2$ be upper bounded by*

$$\mathcal{O}(\frac{F^*}{\eta KT}) + \mathcal{O}(\eta\sigma_l^2 + \eta_l^2 K\sigma_l^2) + \mathcal{O}(\eta\Psi),$$

where

$$\Psi = \frac{1}{T}\sum_{t=0}^{T-1}\frac{\rho^t(\boldsymbol{\lambda})}{\eta_l^2 K}.$$

Revisiting FedAvg convergence via client consensus dynamics We can obtain the same results as Theorem 4.1 if we explicitly derive the upper bound of term Ψ using Assumption 4.1. For example, the local updates are upper bounded by $\boldsymbol{g}_i^t \leq K\eta_l^2(\sigma_l^2 + 6K\sigma_g^2)$ by Reddi et al. (2020). In other words, the dynamics term Ψ absorbs the impacts of data heterogeneity and local gradient variances. The above theorem suggests that *FedAvg empirically converges with a better rate than the rate described by Theorem 4.1.* This gap comes from the differences between pessimistic assumptions and empirical client consensus dynamics. Moreover, using an optimized learning rate (e.g., Lemma A.2) in Appendix A.1, the above upper bound can achieve a dominated rate of $\mathcal{O}(\Psi/\sqrt{T})$. It indicates that *the data heterogeneity has less impact on the training procedure (i.e., $\{\boldsymbol{x}^t\}_{t=0}^{T-1}$) if the attained client consensus dynamics Ψ is small.* Therefore, a better aggregation strategy that reduces client consensus dynamics is a promising way to enhance FL under heterogeneity scenarios.

However, existing convergence analyses of FedAvg typically use uniform weights or pessimistic heterogeneity measures. They omitted the power of global aggregation in FedAvg. And how the aggregation weights $\boldsymbol{\lambda}$ work in the convergence of FedAvg remains ambiguous. The commonly used static weight $\boldsymbol{\lambda}$ is typically related to the global objective. The previous discussion motivates us to investigate the convergence of FedAvg when the global objective weight $\boldsymbol{\lambda}$ and the applied aggregation weight $\tilde{\boldsymbol{\lambda}}$ are not identical. To describe the power of aggregation weights, we present the following Corollary 4.1:

Corollary 4.1 *Suppose FL defines its global objective with a static weight $\boldsymbol{\lambda}$ while aggregating local updates with (possibly adaptive) weights $\tilde{\boldsymbol{\lambda}}$ such that $\tilde{\boldsymbol{G}}^t = \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{g}_i^t$. If FedAvg uses $\tilde{\boldsymbol{G}}^t$ for global gradient descent instead of $\boldsymbol{G}^t$ in Eq. (1). Then, the global gradient norm $\min_t \mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2$ can be upper bounded by*

$$\mathcal{O}(\frac{F^*}{\eta KT}) + \mathcal{O}(\eta\sigma_l^2 + \eta_l^2 K\sigma_l^2) + \mathcal{O}(\eta\tilde{\Psi}),$$

where

$$\tilde{\Psi} = \frac{1}{T}\sum_{t=0}^{T-1}\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\cdot\frac{\rho^t(\tilde{\boldsymbol{\lambda}})}{\eta_l^2 K},\ \chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}} = \sum_{i=1}^N\frac{(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^2}{\boldsymbol{\lambda}_i}.$$

Compared with Theorem 4.2, the above corollary indicates that *adptive aggregation strategy accelerates the FL convergence if it induces lower client consensus dynamics $\tilde{\Psi}$ on a training trajectory of $\{\boldsymbol{x}^t\}_{t=0}^{T-1}$.* It shows that a key step in enhancing FedAvg is to design the pseudo-gradient for global updates and achieve a promising client consensus, i.e., minimize the norm of the applied pseudo-gradient.

Convergence of FedAWARE The convergence benefits of FedAWARE follow the afore-discussed insights. In the methodology, we take the moving-averaged local updates $\boldsymbol{m}_i^t$ as the approximation of local updates $\boldsymbol{g}_i^t$. The solution of Eq.(2) is to minimize client consensus for the current communication round greedily. When $\alpha = 1$ and with full client participation, the above Corollary matches the convergence of FedAWARE, ensuring it converges faster than vanilla FedAvg in heterogeneous environments. Further empirical study demonstrates surprisingly superior empirical convergence of FedAWARE compared to more advanced federated algorithms. In the Appendix A.4, we present the convergence analysis of FedAWARE with partial client participation, which is analogous to the corollary. The results indicate that a proper selection of α can further mitigate the impacts of local update variances. Further ablation studies provide evidence of its efficacy.

4.3 Why does FedAWARE Generalize Better?

FedAWARE achieves better generalization performance by potentially enlarging the gradient diversity over the FL training process. Gradient diversity is first introduced in data-centralized distributed learning (Yin et al., 2018), which quantifies the degree to which individual gradients diverge from each other. It is proved that *distributed mini-batch SGD is amenable to better speedups and generalization if the problems attain high gradient diversity.* Moreover, the common heterogeneity measures are its extensions in federated optimization analysis (Haddadpour and Mahdavi, 2019;

Li et al., 2020b). However, vanilla gradient diversity typically serves as a theoretical analysis tool, lacking empirical guidance for federated algorithm design. To resolve this drawback, we replace the local first-order gradient $\nabla f_i(\boldsymbol{x}^t)$ with local updates $\boldsymbol{g}_i^t$. Formally, we present an empirical measurement termed as *Local Update Diversity* (LUD):

Definition 4.1 (Local Update Diversity) *For any round $t \in [T]$ of the FL training process, we define the local update diversity by*

$$\delta_D^t \triangleq \sqrt{\frac{\sum_{i=1}^N \boldsymbol{\lambda}_i \mathbb{E}\|\boldsymbol{g}_i^t\|^2}{\mathbb{E}\|\boldsymbol{G}^t\|^2}} = \sqrt{\frac{\sum_{i=1}^N \boldsymbol{\lambda}_i \mathbb{E}\|\boldsymbol{g}_i^t\|^2}{\mathbb{V}(\boldsymbol{G}^t) + \rho^t(\boldsymbol{\lambda})}}. \qquad (5)$$

LUD quantifies the degree of local updates $\boldsymbol{g}_i^t$ diverse from global pseudo-gradient $\boldsymbol{G}^t$. It also reflects the ratio of convergence rates between local and global update norms. In non-convex FL, we propose to track the LUD dynamics, that is, the evolution of LUD values in an FL training process. Here, we discuss deep insights into the LUD dynamics.

FL is a process of enlarging LUD Intuitively, when the model is far from the global solution, most of the gradients point in a similar direction, which means that gradient diversity is initially small. When the model approaches the global solution, the LUD starts to grow because of the heterogeneity at the local optimums. From another perspective, this intuition also matches the client consensus dynamics, as shown in Eq. (5). Straightforwardly, the client's consensus $\rho^t(\boldsymbol{\lambda})$ typically decreases faster than local updates during training due to data heterogeneity. Therefore, the empirical LUD values are increasing during training.

To illustrate this statement, we observe the training dynamics of FedAvg on different degrees of Non-IID partitioned CIFAR-10 experiments in Figure 1. Note that different curves indicate the different levels of data heterogeneity that impact FedAvg convergence. We observe that the LUD values increase over the communication round in all heterogeneity settings. Moreover, Figure 1(b) demonstrates that FedAvg attains higher LUD when the heterogeneity level is low. It obtains faster convergence curves, as shown in Figure 1(d). Besides, Figure 1(c) provides empirical evidence for the Assumption 4.1, showing that the norm of global pseudo-gradient decays during FL training.

FedAWARE further enlarges LUD dynamics Gradient diversityYin et al. (2018) has shown that distributed mini-batch SGD gains its generalization ability with larger gradient diversity. As FedAvg can be interpreted as a perturbed distributed SGD, this conclusion should apply to FedAvg and its variants. For example, Figure 1 demonstrates that FedAvg archives higher local update diversity (Figure 1 (b)) and tends to obtain better test accuracy (Figure 1 (a)). In the experimental evaluation, we empirically illustrate that FedAWARE further enlarges the LUD dynamics during training, thus obtaining better generalization performance. Future extension of algorithmic stability Hardt et al. (2016); Lei and Ying (2020) on FedAWARE may provide deeper theoretical insights into its generalization ability.

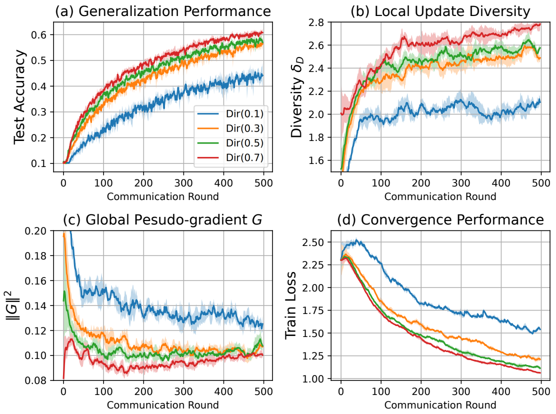


Figure 1: Training dynamics of FedAvg on Non-IID partitioned CIFAR-10 task. We use Dirichlet distribution to allocate clients' data as described in Section 5. Dir(0.1) indicates the most heterogeneous FL setting.

Discussion: connection with client coherence Client coherence (Chatterjee, 2020; Li et al., 2023), defined as the sum of cosine similarities between the local updates of distinct clients, is also used to quantify heterogeneity across clients. The global pseudo-gradient norm (LUD denominator) is related to client coherence, incorporating additional information about the local update norms. A recent study (Li et al., 2023) states: *Gradient diversity argues that higher similarities between workers' gradients degrade performance in distributed mini-batch SGD, while gradient coherence claims that higher similarities between sample gradients enhance generalization.* However, these concepts are not contradictory and pertain to different training stages. In FL, client coherence is relevant only before a critical point where overall coherence is positive (Li et al., 2023), typically during the early training rounds. By contrast, LUD dynamics characterize the entire FL training process, with larger overall LUD dynamics consistently associated with improved generalization, as demonstrated in our experiments. We further elaborate on these points in Appendix B.2.

Discussion: LUD quantifies heterogeneity better Common data heterogeneity measures/assumptions in Table 1 typically build on the local and global first-order gradients. In FL, these common measures cannot

Table 2: Evaluation of CIFAR-10, CIFAR-100 and AGNews tasks. In the table, Raw denotes the original version of all algorithms. And ×AWARE denotes using for FedAWARE extension in Proposition 3.1 on corresponding methods. And, e-LUDD denotes e-LUD dynamics (i.e., $\sum_{t=0}^{T-1} \tilde{\delta}_D^t/T$) during training. The numbers in red indicate the improvements after applying ×AWARE. We run T communication rounds for stable test accuracy. We report the mean test accuracy of the last 10% communication rounds, indicating generalization stability.

Settings	CIFAR-10, $T = 500$				CIFAR-100, $T = 600$				AGNews, $T = 300$			
	Raw		×AWARE		Raw		×AWARE		Raw		×AWARE	
	Acc.	e-LUDD	Acc.	e-LUDD	Acc.	e-LUDD	Acc.	e-LUDD	Acc.	e-LUDD	Acc.	e-LUDD
FedAvg	42.78	2.58	49.25$_{+6.47}$	3.08$_{+0.50}$	28.69	2.94	32.72$_{+4.03}$	3.61$_{+0.67}$	79.27	1.25	80.21$_{+0.94}$	1.98$_{+0.73}$
FedAvgM	50.07	2.17	**59.58**$_{+9.51}$	2.92$_{+0.75}$	31.04	2.72	40.66$_{+9.62}$	3.37$_{+0.65}$	81.85	1.38	83.15$_{+1.30}$	1.85$_{+0.47}$
FedCM	47.84	2.99	49.13$_{+1.29}$	3.09$_{+0.10}$	30.62	2.94	33.09$_{+2.04}$	3.63$_{+0.69}$	80.79	1.26	80.24$_{-0.55}$	1.76$_{+0.50}$
FedDyn	54.87	-	-	-	**39.04**	-	-	-	84.47	-	-	-
FedProx	46.33	2.81	49.40$_{+3.07}$	2.87$_{+0.06}$	27.20	2.95	32.68$_{+5.48}$	3.62$_{+0.67}$	76.81	1.15	80.13$_{+3.33}$	1.83$_{+0.68}$
FedYogi	54.62	2.47	54.92$_{+0.30}$	2.66$_{+0.19}$	37.12	2.61	44.02$_{+6.90}$	3.27$_{+0.66}$	83.15	1.43	85.89$_{+2.74}$	2.01$_{+0.58}$
FedAMS	57.09	2.46	58.45$_{+1.36}$	2.67$_{+0.21}$	38.19	2.42	**45.20**$_{+7.01}$	3.13$_{+0.71}$	83.33	1.34	**86.08**$_{+2.75}$	1.92$_{+0.58}$
FedAWARE	**59.78**	2.90	-	-	38.77	2.75	-	-	**87.70**	1.73	-	-

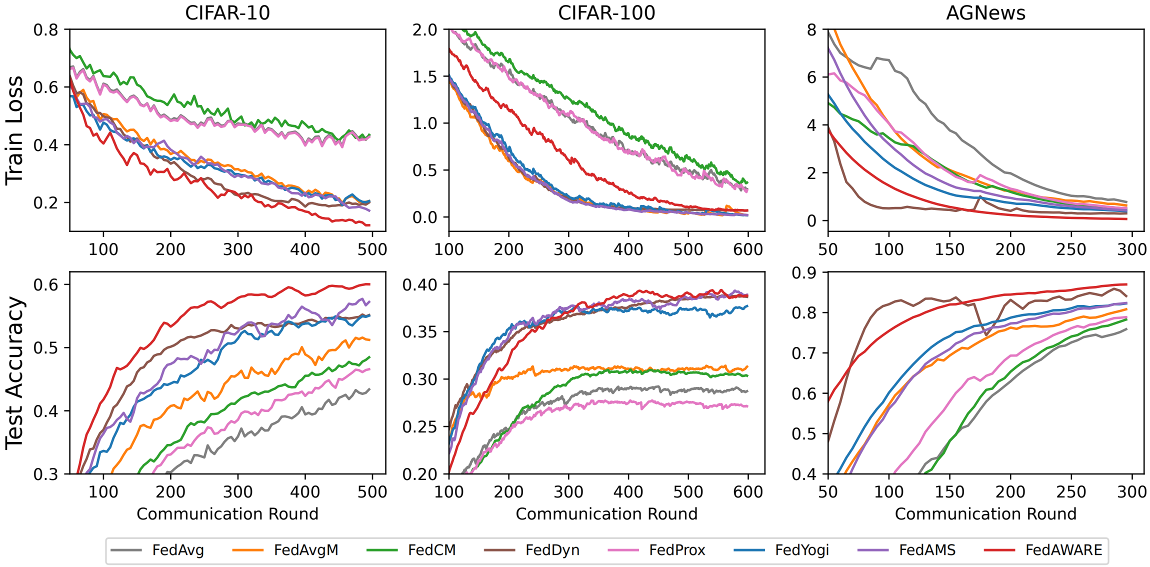


Figure 2: Training dynamics of raw algorithms. The training loss indicates the convergence speed on training datasets, while the test accuracy indicates the generalization stability against heterogeneous clients.

be computed in practice due to privacy risk (Zhu et al., 2019) or communication efficiency (McMahan et al., 2017). Client coherence only measures the heterogeneity in local update directions. Hence, they may overlook the effects of unbalanced local client updates. For example, local updates typically consist of cumulative stochastic mini-batch gradients computed over multiple epochs on the local dataset. When the number of local data samples varies significantly across clients, LUD also reflects the imbalance in local updates. Therefore, the LUD metric provides more comprehensive information than gradient diversity in FL systems. We propose tracking the evolution of δ_D^t to quantify the impact of heterogeneity and assess the quality of FL convergence.

5 EXPERIMENTAL EVALUATION

Baselines We compare advanced FL algorithms and basic baselines related to the methodologies of FedAWARE. Our baselines include standard baseline FedAvg (McMahan et al., 2017), FedAvgM (Hsu et al., 2019), and FedCM (Wang et al., 2022) for the static aggregation and momentum-based algorithms. We also compare with local regularization-based algorithms FedProx (Li et al., 2020b) and FedDyn (Acar et al., 2020). We compare the adaptive federated optimization algorithms with FedYogi (Reddi et al., 2020) and FedAMS (Wang et al., 2022).

Settings We evaluate all algorithms on three setups: (1) train 5-layer CNN on Non-IID partitioned CIFAR-10 dataset. (2) train Resnet-18 (group norm) (He et al.,

2016) on Non-IID partitioned CIFAR-100 dataset. (3) fine-tune pretrained GPT2 model Pythia-70M (Biderman et al., 2023) on Non-IID partitioned AGNews dataset (Zhang et al., 2015). CIFAR-10 and CIFAR-100 are image classification tasks with 50,000 training data samples and 10,000 test data samples. AGNews is a collection of more than 1 million news articles. It is a four-label text classification task with 120,000 train samples and 7,600 test samples. For all datasets, we conduct Non-IID data partitioning following the latent Dirichlet allocation over labels (Hsu et al., 2019) with parameters 0.1 into $N = 100$ clients, indicating extreme data heterogeneity. The visualization of datasets is provided in the Appendix. For the training setup, we set the learning rate $\eta_l = 0.01$ (CIFAR) or $\eta_l = 0.0001$ (AGNews) for local training parameters, batch size 64, and local epoch 3 for all clients. For each communication round, we randomly select 10 clients. Then, we grid search global learning rate η_g and the algorithm-specific hyperparameters for all algorithms as elaborated in the Appendix. All results are the mean values of three independent runs over random seeds.

Metrics We evaluate the test accuracy of algorithms on RAW test datasets and observe the training loss dynamics to compare the convergence speed. Moreover, we also observe the LUD dynamics of FL. However, we cannot accurately compute the LUD in practice due to the partial participation of clients. Instead, we define a metric called an empirical LUD (e-LUD):

$$\tilde{\delta}_D^t \triangleq \sqrt{\frac{\frac{1}{|S^t|}\sum_{i\in S^t}\|\boldsymbol{g}_i^t\|^2}{\|\frac{1}{|S^t|}\sum_{i\in S^t}\boldsymbol{g}_i^t\|^2}},$$

where S^t is the selected client set at the t-th round and $\boldsymbol{g}_i^t$ is the uploaded local updates from selected clients. For a fair comparison, the e-LUD of all algorithms is computed the same. Their values indicate the status of global solutions and their relation to local minimums.

Vanilla FedAWARE converges faster and generalizes better In Figure 2, we examine the convergence performance of algorithms. The results highlight FedAWARE's superior performance. On CIFAR-10 and AGNews tasks, we surprisingly observed that FedAWARE converges faster than most baselines and even outperforms adaptive federated optimization methods (FedYogi and FedAMS). On the CIFAR-100 task, despite FedAWARE converging slightly slower than the SOTA methods at early communication rounds, it still reaches a good solution with identical train loss. Meanwhile, the test accuracy of FedAWARE also implements marginal improvement to FedAvgM, FedYogi, and FedAMS methods, as shown in Table 2. Given the empirical results, we argue that FedAWARE, a novel variant of FedAvg, is more efficient for handling heterogeneous local updates. Besides noting that the key performance improvement of FedAWARE is obtained from the adaptive weighted aggregation strategies, the empirical results also provide solid support for our Corollary 4.1.

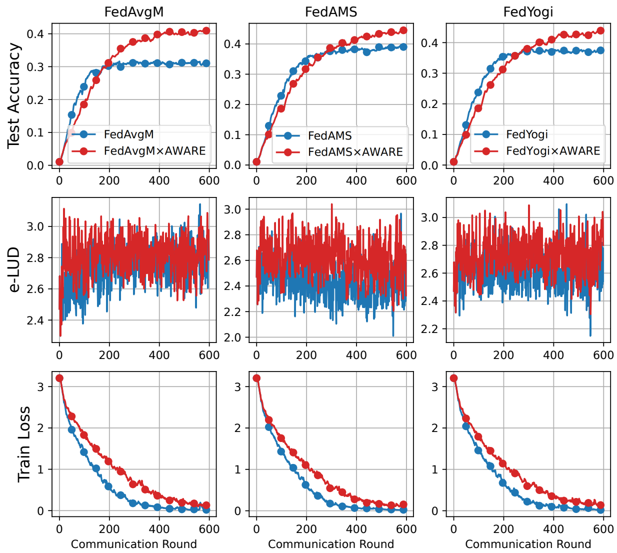


Figure 3: Training dynamics of FedAWARE extension on CIFAR-100 setting. The results of CIFAR-10 and AGNews are presented in the Appendix.

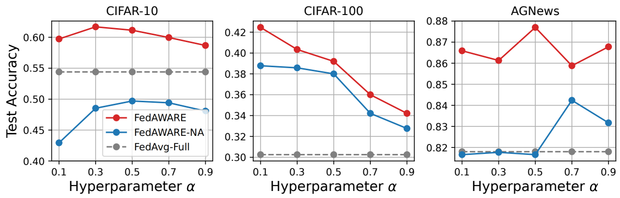


Figure 4: Ablation study. "FedAWARE-NA" means we only use moving-averaged local updates without adaptive aggregation. "FedAvg-Full" is the result of vanilla FedAvg with full client participation.

Enlarging e-LUD dynamics of algorithms obtain better generalization performance In Table 2, we apply the FedAWARE extension (Proposition 3.1) on all baseline algorithms, excluding FedDyn, which does not follow the global gradient descent rule. Compared with the raw results of algorithms, the FedAWARE module enlarges the e-LUDD of ALL algorithms. Meanwhile, the test accuracy of most algorithms is significantly improved. Surprisingly, the FedAMS×AWARE on CIFAR100 implements superior accuracy than all raw algorithms. The results provide empirical evidence on the relation between LUD and generalization, that is, *federated learning is amenable to better generalization if the training procedure attains high LUD.* We believe this finding can enlighten future techniques for enhancing FL generalization.

FedAWARE extension may slow convergence slightly in exchange for generalization boost In Table 2, we observe that FedAWARE greatly enhances FedAvgM, FedYogi, and FedAMS. To provide deeper insights. we further examine the training dynamics in CIFAR-100 task after applying the FedAWARE extension, as shown in Figure 3. The convergence speed of applied algorithms has decreased slightly, and we observed that the test accuracy curves reached a higher final accuracy. Moreover, additional results on the CIFAR-10 and AGNews tasks in the Appendix show that the FedAWARE extension mitigates the test accuracy spikes problem, indicating stabilized generalization performance. The results demonstrate that enlarged e-LUD dynamics of algorithms also stabilize the generalization of a training procedure.

Ablation Study In Figure 4, we present the ablation study to evaluate the components of FedAWARE. We run FedAWARE with different moving-average parameters $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ to evaluate its sensitivity. The results show that the proper selection of α can enhance the generalization performance, while the best selection varies on different tasks. We also conducted the same experiments without adaptive aggregation; FedAWARE's results are better than all FedAWARE-NA results. Meanwhile, it is worth noting that FedAWARE's e-LUDD is also higher than FedAWARE-NA's. Compared with FedAvg-Full, moving-averaged local updates only provide marginal improvements on CIFAR-100 and AGNews tasks. This indicates the critical ability of adaptive aggregation.

Limitation and feasible solutions Running Algorithm 1 requires the server to store the local momentum. We argue that the FL server has sufficient storage and computing resources. Besides, we argue that this memory consumption is worth using FedAWARE to enhance other algorithms. This concern can be alleviated by the following options to save the storage: *1. Computing* (2) *with the last few layers of a neural network.* This is because numerous studies (Xu et al., 2020; Kirichenko et al., 2022; Burns et al., 2023) have shown that the last layers contain crucial network information. Hence, we can trade off the memory storage and quality of the aggregation. *2. Adopting client clustering techniques* (Sattler et al., 2020; Zeng et al., 2023a; Long et al., 2023). We can cluster clients respecting their similarity and save the momentum of cluster-averaged gradients. The server only costs the storage proportional to the number of client clusters.

We have empirically demonstrated that enlarging the LUD dynamics of baseline algorithms by incorporating AWARE can improve generalization performance. However, the causal relation between LUD and model performance remains inconclusive due to a gap between the practical use of e-LUD and the theoretical LUD measure. For example, methods with the highest accuracy do not necessarily have the highest e-LUDD values and may even have relatively low values (e.g., FedAMS in "CIFAR-100 xAWARE") in Table 2. Hence, investigating correlations between absolute heterogeneity measures and model performance may provide valuable insights for future research. Additionally, we propose that the theoretical relationship between LUD and generalization performance could be established through algorithmic stability analyses (Lei and Ying, 2020) or our recent framework for analyzing excess risk dynamics (Zeng et al., 2024), which we leave for future exploration.

6 CONCLUSION

In this paper, we revisit the power of FedAvg aggregation in theory and practice. Our results further enrich the understanding of FedAvg and its variants by highlighting client consensus dynamics in convergence analysis and local update diversity in generalization. Based on our findings, we derive a simple and effective FedAvg variant, FedAWARE, which minimizes the norm of global aggregation results for convergence and generalization performance. Importantly, given the strong relationship between LUD dynamics and FL generalization illustrated by intensive empirical results, we believe LUD can be a critical measurement for designing and evaluating future federated algorithms.

Acknowledgements

We sincerely thank the anonymous reviewers for their time and constructive comments. The computations in this research were performed using the CFFF platform of Fudan University.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. (2020). Federated learning based on dynamic regularization. In *International Conference on Learning Representations.*

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., Chen, Y., Ecoffet, A., Joglekar, M., Leike, J., et al. (2023). Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

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

Charles, Z., Garrett, Z., Huo, Z., Shmulyian, S., and Smith, V. (2021). On large-cohort training for federated learning. *Advances in neural information processing systems*, 34:20461–20475.

Chatterjee, S. (2020). Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *ICLR*. OpenReview.net.

Chen, Y., He, N., and Sun, L. (2024). Fedawa: Aggregation weight adjustment in federated domain generalization. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 451–457. IEEE.

Glasgow, M. R., Yuan, H., and Ma, T. (2022). Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 9050–9090. PMLR.

Greub, W. H. (2012). *Linear algebra*, volume 23. Springer Science & Business Media.

Gu, X., Lyu, K., Huang, L., and Arora, S. (2023). Why (and when) does local SGD generalize better than sgd? In *ICLR*. OpenReview.net.

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

Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

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:1909.06335*.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR.

Jhunjhunwala, D., Sharma, P., Nagarkatti, A., and Joshi, G. (2022). Fedvarp: Tackling the variance due to partial client participation in federated learning. In *Uncertainty in Artificial Intelligence*, pages 906–916. PMLR.

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

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

Khaled, A., Mishchenko, K., and Richtárik, P. (2019). First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*.

Khaled, A., Mishchenko, K., and Richtárik, P. (2020). Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR.

Kirichenko, P., Izmailov, P., and Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. (2020). A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR.

Lei, Y. and Ying, Y. (2020). Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pages 5809–5819. PMLR.

Li, T., Beirami, A., Sanjabi, M., and Smith, V. (2020a). Tilted empirical risk minimization. In *International Conference on Learning Representations*.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020b). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2019a). Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2019b). On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Li, Z., Lin, T., Shang, X., and Wu, C. (2023). Revisiting weighted aggregation in federated learning with neural networks. In *International Conference on Machine Learning*, pages 19767–19788. PMLR.

Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., and Jiang, J. (2023). Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500.

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.

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

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.

Sattler, F., Müller, K.-R., and Samek, W. (2020). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. (2021). A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*.

Wang, J., Das, R., Joshi, G., Kale, S., Xu, Z., and Zhang, T. (2024). On the unreasonable effectiveness of federated averaging with heterogeneous data. *Transactions on Machine Learning Research*.

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

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. (2019). Adaptive federated learning in resource constrained edge computing systems. *IEEE journal on selected areas in communications*, 37(6):1205–1221.

Wang, Y., Lin, L., and Chen, J. (2022). Communication-efficient adaptive federated learning. In *International Conference on Machine Learning*, pages 22802–22838. PMLR.

Woodworth, B. E., Patel, K. K., and Srebro, N. (2020). Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292.

Wu, F., Guo, S., Qu, Z., He, S., Liu, Z., and Gao, J. (2023). Anchor sampling for federated learning with partial client participation. In *International Conference on Machine Learning*, pages 37379–37416. PMLR.

Wu, Z., Xu, Z., Zeng, D., Wang, Q., and Liu, J. (2024). Advocating for the silent: Enhancing federated generalization for nonparticipating clients. *IEEE Transactions on Neural Networks and Learning Systems*.

Xie, C., Koyejo, O., Gupta, I., and Lin, H. (2019). Local adaalter: Communication-efficient stochastic gradient descent with adaptive learning rates. *arXiv preprint arXiv:1911.09030*.

Xu, P., Wen, Z., Zhao, H., and Gu, Q. (2020). Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*.

Ye, R., Xu, M., Wang, J., Xu, C., Chen, S., and Wang, Y. (2023). Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, pages 39879–39902. PMLR.

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. (2018). Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1998–2007. PMLR.

Zaheer, M., Reddi, S., Sachan, D., Kale, S., and Kumar, S. (2018). Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31.

Zeng, D., Hu, X., Liu, S., Yu, Y., Wang, Q., and Xu, Z. (2023a). Stochastic clustered federated learning. *arXiv preprint arXiv:2303.00897*.

Zeng, D., Liang, S., Hu, X., Wang, H., and Xu, Z. (2023b). Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7.

Zeng, D., Wu, Z., Liu, S., Pan, Y., Tang, X., and Xu, Z. (2024). Understanding generalization of federated learning: the trade-off between model stability and optimization. *arXiv preprint arXiv:2411.16303*.

Zhang, X., Zhao, J. J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *NIPS*.

Zhang, Y., Zeng, D., Luo, J., Fu, X., Chen, G., Xu, Z., and King, I. (2024). A survey of trustworthy federated learning: Issues, solutions, and challenges. *ACM Transactions on Intelligent Systems and Technology*, 15(6):1–47.

Zhang, Y., Zeng, D., Luo, J., Xu, Z., and King, I. (2023a). A survey of trustworthy federated learning with perspectives on security, robustness, and privacy. *arXiv preprint arXiv:2302.10637*.

Zhang, Z., Zhang, Y., Chen, G., Qu, L., Zhou, X., Wang, H., and Xu, Z. (2023b). From continuous pre-training to alignment: A comprehensive toolkit for large language models in federated learning.

Available at SSRN: https://ssrn.com/abstract=5087720.

Zhao, X. and Shen, D. (2024). Fedsw: Federated learning with adaptive sample weights. *Information Sciences*, 654:119873.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

APPENDIX

A Convergence Guarantees

A.1 Useful Lemmas

Lemma A.1 (Bounded local updates (Reddi et al., 2020)) *Let Assumption 4.3 4.1 hold. For all client $i \in [N]$ with arbitrary local iteration steps $k \in [K]$ and $\eta_l \leq \frac{1}{8KL}$, the local updates can be bounded as follows,*

$$\mathbb{E}\left\|\boldsymbol{x}_i^{t,k} - \boldsymbol{x}^t\right\|^2 \leq 5K\eta_l^2(\sigma_l^2 + 6K\sigma_g^2 + 6K\left\|\nabla f\left(\boldsymbol{x}^t\right)\right\|^2).$$

Lemma A.2 (Tuning the stepsize (Koloskova et al., 2020)) *For any parameters $r_0 \geq 0, b \geq 0, e \geq 0, d \geq 0$ there exists constant stepsize $\eta \leq \frac{1}{d}$ such that*

$$\Psi_T := \frac{r_0}{\eta T} + b\eta + e\eta^2 \leq 2\left(\frac{br_0}{T}\right)^{\frac{1}{2}} + 2e^{1/3}\left(\frac{r_0}{T}\right)^{\frac{2}{3}} + \frac{dr_0}{T}.$$

A.2 Proof of Theorem 4.2

Here, we prove the convergence rate of FedAvg:

$$\textbf{Client: } \boldsymbol{g}_i^t = \boldsymbol{x}_i^{t,K} - \boldsymbol{x}_i^{t,0} = \eta_l \sum_{k=0}^{K-1} \nabla F_i(\boldsymbol{x}_i^{t,k});$$

$$\textbf{Server: } \boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \sum_{i=1}^{N} \boldsymbol{\lambda}_i \boldsymbol{g}_i^t = \boldsymbol{x}^t - \eta_g \boldsymbol{G}^t.$$

Specifically, we analyze the update rule on the server-side gradient descent

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_g \boldsymbol{G}^t.$$

For ease of writing, we rewrite the above equations as follows

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \tilde{\eta}\hat{\boldsymbol{G}}^t,$$

where $\tilde{\eta} = K\eta_l\eta_g$ and $\hat{\boldsymbol{G}}^t = \boldsymbol{G}^t/\eta_l K = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\nabla F_i(\boldsymbol{x}_i^{t,k})$.

Using the smoothness, we have

$$
\begin{aligned}
f\left(\boldsymbol{x}^{t+1}\right) = f\left(\boldsymbol{x}^{t} - \tilde{\eta}\hat{\boldsymbol{G}}^{t}\right) &\leq f(\boldsymbol{x}^{t}) - \tilde{\eta}\left\langle \nabla f(\boldsymbol{x}^{t}), \hat{\boldsymbol{G}}^{t} \right\rangle + \frac{L}{2}\tilde{\eta}^2 \left\| \hat{\boldsymbol{G}}^{t} \right\|^2 \\
&\leq f(\boldsymbol{x}^{t}) - \tilde{\eta}\left\langle \nabla f(\boldsymbol{x}^{t}), \hat{\boldsymbol{G}}^{t} - \nabla f(\boldsymbol{x}^{t}) + \nabla f(\boldsymbol{x}^{t}) \right\rangle + \frac{L}{2}\tilde{\eta}^2 \left\| \hat{\boldsymbol{G}}^{t} \right\|^2 \\
&\leq f(\boldsymbol{x}^{t}) - \tilde{\eta}\|\nabla f(\boldsymbol{x}^{t})\|^2 + \tilde{\eta}\left\langle \nabla f(\boldsymbol{x}^{t}), \nabla f(\boldsymbol{x}^{t}) - \hat{\boldsymbol{G}}^{t} \right\rangle + \frac{L}{2}\tilde{\eta}^2 \left\| \hat{\boldsymbol{G}}^{t} \right\|^2 \\
&\leq f(\boldsymbol{x}^{t}) - \frac{\tilde{\eta}}{2}\|\nabla f(\boldsymbol{x}^{t})\|^2 + \frac{\tilde{\eta}}{2}\|\nabla f(\boldsymbol{x}^{t}) - \hat{\boldsymbol{G}}^{t}\|^2 + \frac{L}{2}\tilde{\eta}^2 \left\| \hat{\boldsymbol{G}}^{t} \right\|^2 .
\end{aligned}
$$

Taking full expectation over randomness at time step t on both sides, we have

$$
\mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right] - f(\boldsymbol{x}^{t}) \leq -\frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t})\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t}) - \hat{\boldsymbol{G}}^{t}\|^2 + \frac{L}{2}\tilde{\eta}^2\mathbb{E}[\|\hat{\boldsymbol{G}}^{t}\|^2] \tag{6}
$$

Then, we investigate

$$
\begin{aligned}
\mathbb{E}\|\nabla f(\boldsymbol{x}^{t}) - \hat{\boldsymbol{G}}^{t}\|^2 &= \mathbb{E}\|\sum_{i=1}^{N}\boldsymbol{\lambda}_i(\nabla f_i\left(\boldsymbol{x}^{t}\right) - \frac{1}{\eta_l K}\boldsymbol{g}_i^t)\|^2 \leq \sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\|\nabla f_i\left(\boldsymbol{x}^{t}\right) - \frac{1}{\eta_l K}\boldsymbol{g}_i^t\|^2 \\
&= \sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\left(\nabla f_i\left(\boldsymbol{x}^{t}\right) - \frac{1}{K}\sum_{k=0}^{K-1}\nabla F_i(\boldsymbol{x}_i^{t,k})\right)\right\|^2 \\
&= \sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\left(\frac{1}{K}\sum_{k=0}^{K-1}(\nabla f_i\left(\boldsymbol{x}^{t}\right) - \nabla F_i(\boldsymbol{x}_i^{t,k}))\right)\right\|^2 \\
&\leq \frac{1}{K^2}\sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\left(\sum_{k=0}^{K-1}(\nabla f_i\left(\boldsymbol{x}^{t}\right) \pm \nabla f_i(\boldsymbol{x}^{t,k}) - \nabla F_i(\boldsymbol{x}_i^{t,k}))\right)\right\|^2 \\
&\leq 2\frac{1}{K^2}\sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\sum_{k=0}^{K-1}(\nabla f_i\left(\boldsymbol{x}^{t}\right) - \nabla f_i(\boldsymbol{x}^{t,k}))\right\|^2 + 2\frac{1}{K^2}\sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\sum_{k=0}^{K-1}(\nabla f_i(\boldsymbol{x}^{t,k}) - \nabla F_i(\boldsymbol{x}_i^{t,k}))\right\|^2 \\
&\leq 2\frac{1}{K^2}\sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\sum_{k=0}^{K-1}(\nabla f_i\left(\boldsymbol{x}^{t}\right) - \nabla f_i(\boldsymbol{x}^{t,k}))\right\|^2 + 2\sigma_l^2/K \\
&\leq 2\frac{1}{K^2}L^2\sum_{i=1}^{N}\boldsymbol{\lambda}_i K\sum_{k=0}^{K-1}\mathbb{E}\left\|\boldsymbol{x}^{t} - \boldsymbol{x}_i^{t,k-1}\right\|^2 + 2c\eta_l^2 L^2 K\sigma_l^2 \qquad \triangleright \text{Letting } \frac{1}{\sqrt{c}KL} \leq \eta_l \\
&\leq 2L^2\sum_{i=1}^{N}\boldsymbol{\lambda}_i\mathbb{E}\left\|\boldsymbol{g}_i^t\right\|^2 + 2c\eta_l^2 L^2 K\sigma_l^2
\end{aligned}
$$

This work mainly focuses on the benefits of using adaptive aggregation. To simplify the analysis, we make a mild assumption on the relation between the local update and the global pseudo-gradient norm:

Assumption A.1 *We assume there is a constant γ makes the local update $\boldsymbol{g}_i^t$ and applied global pseudo-gradient $\boldsymbol{G}^t$ satisfies that*

$$
\mathbb{E}\|\boldsymbol{g}_i^t\|^2 \leq \gamma\mathbb{E}\|\boldsymbol{G}^t\|^2,
$$

$\forall t \in [T], i \in [N], \boldsymbol{\lambda} \in \Delta_N$, *where Δ_N is the N-dimensional simplex. Please note the γ value will not exceed a minor constant in practice.*

Using the above assumption, we obtain

$$
\mathbb{E}\|\nabla f(\boldsymbol{x}^{t}) - \hat{\boldsymbol{G}}^{t}\|^2 \leq 2L^2\gamma\mathbb{E}\|\boldsymbol{G}^t\|^2 + 2c\eta_l^2 L^2 K\sigma_l^2. \tag{7}
$$

Combining (6) and (7), we have

$$\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 \leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{\tilde{\eta}} + 2c\eta_l^2 L^2 K \sigma_l^2 + (2L^2\gamma + \frac{L\tilde{\eta}}{\eta_l^2 K^2})\mathbb{E}[\|\boldsymbol{G}^t\|^2]. \tag{8}$$

Then, we investigate $\mathbb{E}\|\boldsymbol{G}^t\|^2$ using Assumption 4.1. Straightforwardly, we know that

$$\mathbb{E}\|\boldsymbol{G}^t\|^2 = \mathbb{V}[\boldsymbol{G}^t] + \|\mathbb{E}[\boldsymbol{G}^t]\|^2 = \mathbb{V}[\boldsymbol{G}^t] + \rho^t(\boldsymbol{\lambda}).$$

Here, the variance of $\boldsymbol{G}^t$ is taken with respect to random noise in stochastic local updates. The upper bound only depends on the dynamics of local SGD, which has been well understood in literature (Khaled et al., 2020). Concretely, author of Khaled et al. (2020) show that

$$\begin{aligned}\mathbb{V}[\boldsymbol{G}^t] = \mathbb{E}\|\boldsymbol{G}^t - \mathbb{E}[\boldsymbol{G}^t]\|^2 &= \mathbb{E}\|\sum_{i=1}^N \boldsymbol{\lambda}_i(\boldsymbol{g}_i^t - \mathbb{E}[\boldsymbol{g}_i^t])\|^2 \\ &\leq \sum_{i=1}^N \boldsymbol{\lambda}_i \mathbb{E}\|\boldsymbol{g}_i^t - \mathbb{E}[\boldsymbol{g}_i^t]\|^2 \leq K\eta_l^2\sigma_l^2.\end{aligned}$$

Hence, we have the descent form as

$$\begin{aligned}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{\tilde{\eta}} + 2c\eta_l^2 L^2 K\sigma_l^2 + (2L^2\gamma + \frac{L\tilde{\eta}}{\eta_l^2 K^2})(K\eta_l^2\sigma_l^2 + \rho^t(\boldsymbol{\lambda})) \\ &\leq \frac{2(f(\boldsymbol{x}^0) - \mathbb{E}\left[f(\boldsymbol{x}^T)\right])}{TK\eta} + 2(c+\gamma)\eta_l^2 L^2 K\sigma_l^2 + \eta L\sigma_l^2 + \eta L(\frac{2L\gamma}{\eta} + \frac{1}{\eta_l^2 K})\rho^t(\boldsymbol{\lambda}) \\ &\leq \frac{2(f(\boldsymbol{x}^0) - \mathbb{E}\left[f(\boldsymbol{x}^T)\right])}{TK\eta} + 2(c+\gamma)\eta_l^2 L^2 K\sigma_l^2 + \eta L\sigma_l^2 + \eta L(1 + \frac{\gamma}{4\eta_g})\frac{\rho^t(\boldsymbol{\lambda})}{\eta_l^2 K}\end{aligned} \tag{9}$$

where we define $\eta = \eta_l\eta_g$ and use $\eta_l \leq \frac{1}{8LK}$ from Lemma A.1. Taking averaging of both sides from time $t = 0$ to $T-1$, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 \leq \frac{2(f(\boldsymbol{x}^0) - \mathbb{E}\left[f(\boldsymbol{x}^T)\right])}{TK\eta} + 2(c+\gamma)\eta_l^2 L^2 K\sigma_l^2 + \eta L\sigma_l^2 + \eta L(1 + \frac{\gamma}{4\eta_g})\frac{1}{T}\sum_{t=0}^{T-1}\frac{\rho^t(\boldsymbol{\lambda})}{\eta_l^2 K}, \tag{10}$$

which concludes the proof.

A.3 Proof of Corollary 4.1

Now, we investigate the convergence of FedAvg when the global objective weight $\boldsymbol{\lambda}$ and the applied aggregation weight $\tilde{\boldsymbol{\lambda}}$ are not identical. The update rule on the server side becomes

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \tilde{\eta}\tilde{\boldsymbol{G}}^t, \quad \tilde{\boldsymbol{G}}^t = \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i^t \boldsymbol{g}_i^t,$$

where $\tilde{\boldsymbol{\lambda}}^t$ is given by any aggregation strategy on FL server. Noting that original weights $\boldsymbol{\lambda}$ is used to define global objective, inducing $\nabla f(\boldsymbol{x}) = \sum_{i=}^N \boldsymbol{\lambda}_i \nabla f_i(\boldsymbol{x})$.

For ease of writing, we investigate the normalized update rule:

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \tilde{\eta}\bar{\boldsymbol{G}}^t, \quad \bar{\boldsymbol{G}}^t = \frac{1}{\eta_l K}\tilde{\boldsymbol{G}}^t.$$

Using the smoothness, we have:

$$f\left(\boldsymbol{x}^{t+1}\right) \leq f(\boldsymbol{x}^t) - \frac{\tilde{\eta}}{2}\|\nabla f(\boldsymbol{x}^t)\|^2 + \frac{\tilde{\eta}}{2}\|\nabla f(\boldsymbol{x}^t) - \bar{\boldsymbol{G}}^t\|^2 + \frac{L}{2}\tilde{\eta}^2\left\|\bar{\boldsymbol{G}}^t\right\|^2$$

Taking full expectation over randomness at time step t on both sides, we have:

$$\begin{aligned}\mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right] - f(\boldsymbol{x}^t) &\leq -\frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \bar{\boldsymbol{G}}^t\|^2 + \frac{L}{2}\tilde{\eta}^2\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2] \\ &\leq -\frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^t) \pm \hat{\boldsymbol{G}}^t - \bar{\boldsymbol{G}}^t\|^2 + \frac{L}{2}\tilde{\eta}^2\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2] \\ &\leq -\frac{\tilde{\eta}}{2}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + \tilde{\eta}\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2 + \tilde{\eta}\mathbb{E}\|\hat{\boldsymbol{G}}^t - \bar{\boldsymbol{G}}^t\|^2 + \frac{L}{2}\tilde{\eta}^2\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2]\end{aligned} \tag{11}$$

To compare with vanilla FedAvg (static weight), we investigate the gap between expected pseudo-gradient $\boldsymbol{G}^t$ and applied pseudo-gradient $\tilde{\boldsymbol{G}}^t$. By definition, we know

$$\boldsymbol{G}^t - \tilde{\boldsymbol{G}}^t = \sum_{i=1}^N (\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)\boldsymbol{g}_i^t = \sum_{i=1}^N \frac{(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)}{\sqrt{\boldsymbol{\lambda}_i}}\sqrt{\boldsymbol{\lambda}_i}\boldsymbol{g}_i^t.$$

Then, applying Cauchy-Schwarz inequality, it induces that

$$\mathbb{E}\|\hat{\boldsymbol{G}}^t - \bar{\boldsymbol{G}}^t\|^2 = \frac{1}{\eta_l^2 K^2}\mathbb{E}\|\boldsymbol{G}^t - \tilde{\boldsymbol{G}}^t\|^2 \leq \frac{1}{\eta_l^2 K^2}\left[\sum_{i=1}^N \frac{(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^2}{\boldsymbol{\lambda}_i}\right] \cdot \left[\sum_{i=1}^N \boldsymbol{\lambda}_i \left\|\boldsymbol{g}_i^t\right\|^2\right] = \frac{\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2}\sum_{i=1}^N \boldsymbol{\lambda}_i \left\|\boldsymbol{g}_i^t\right\|^2, \tag{12}$$

where we define $\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}} = \sum_{i=1}^N \frac{(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^2}{\boldsymbol{\lambda}_i}$. Then, reorganizing the terms in (11), we have

$$\begin{aligned}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + 2\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2 + 2\mathbb{E}\|\hat{\boldsymbol{G}}^t - \bar{\boldsymbol{G}}^t\|^2 + L\tilde{\eta}\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2] \\ &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + (4L^2 + \frac{2\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2})\sum_{i=1}^N \boldsymbol{\lambda}_i\mathbb{E}\left\|\boldsymbol{g}_i^t\right\|^2 + 4\eta_l^2 K\sigma_l^2 + L\tilde{\eta}\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2],\end{aligned}$$

where the last inequality uses (7) and (12).

Then, we use Assumption A.1 to have

$$\begin{aligned}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + (4L^2 + \frac{2\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2})\gamma\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + 4\eta_l^2 K\sigma_l^2 + L\tilde{\eta}\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2] \\ &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + (4L^2 + \frac{2\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2})\gamma\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + 4\eta_l^2 K\sigma_l^2 + L\tilde{\eta}\mathbb{E}[\|\bar{\boldsymbol{G}}^t\|^2] \\ &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + 4\eta_l^2 K\sigma_l^2 + ((4L^2 + \frac{2\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2})\gamma + \frac{L\tilde{\eta}}{\eta_l^2 K^2})\mathbb{E}[\|\tilde{\boldsymbol{G}}^t\|^2] \\ &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + 4\eta_l^2 K\sigma_l^2 + ((4L^2 + \frac{2\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2})\gamma + \frac{L\tilde{\eta}}{\eta_l^2 K^2}) \cdot (K\eta_l^2\sigma_l^2 + \rho^t(\tilde{\boldsymbol{\lambda}})).\end{aligned}$$

Now, we conducting similar operations in (9) to obtain

$$\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 \leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{K\eta} + 4((L^2 + \frac{\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{2\eta_l^2 K^2})\gamma + 1)\eta_l^2 K\sigma_l^2 + \eta L\sigma_l^2 + \eta L(4(L + \frac{\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{2\eta KL})\gamma + 1)\frac{\rho^t(\tilde{\boldsymbol{\lambda}})}{\eta_l^2 K}.$$

Taking averaging of both sides from time $t = 0$ to $T - 1$ and defining $\tilde{\chi} = \frac{1}{T}\sum_{t=0}^{T-1}\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}$ for notation simplicity, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 \leq \frac{2(f(\boldsymbol{x}^0) - \mathbb{E}\left[f(\boldsymbol{x}^T)\right])}{TK\eta} + \eta L\sigma_l^2 + 4(c + \gamma + \frac{\tilde{\chi}\gamma}{2\eta_l^2 L^2 K^2}) \cdot \eta_l^2 L^2 K\sigma_l^2 \\ + \eta L(1 + \frac{L\gamma}{2\eta_g} + \frac{2\tilde{\chi}\gamma}{\eta KL})\frac{1}{T}\sum_{t=0}^{T-1}\frac{\rho^t(\tilde{\boldsymbol{\lambda}})}{\eta_l^2 K}.\end{aligned} \tag{13}$$

A.4 Convergence Analysis of FedAWARE with Partial Participation

We recall the update rule of Algorithm 1 is:

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \boldsymbol{d}^t,$$

where

$$\boldsymbol{d}^t = \sum_{i=1}^N \tilde{\lambda}_i^t \boldsymbol{m}_i^t, \text{s.t. } \tilde{\lambda}^t = \arg\min_{\lambda} \left\| \sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{m}_i^t \right\|^2.$$

We rewrite the update rule with normalized pseudo-gradient:

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \tilde{\eta} \hat{\boldsymbol{d}}^t,$$

where $\tilde{\eta} = K\eta_l \eta_g$ and $\hat{\boldsymbol{d}}^t = \boldsymbol{d}^t / \eta_l K$. Using the smoothness, we have:

$$f\left(\boldsymbol{x}^{t+1}\right) \leq f(\boldsymbol{x}^t) - \frac{\tilde{\eta}}{2} \|\nabla f(\boldsymbol{x}^t)\|^2 + \frac{\tilde{\eta}}{2} \|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{d}}^t\|^2 + \frac{L}{2}\tilde{\eta}^2 \left\|\hat{\boldsymbol{d}}^t\right\|^2.$$

Taking full expectation over randomness at time step t on both sides, we have:

$$\mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right] - f(\boldsymbol{x}^t) \leq -\frac{\tilde{\eta}}{2} \mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + \frac{\tilde{\eta}}{2} \underbrace{\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{d}}^t\|^2}_{T_1} + \frac{L}{2}\tilde{\eta}^2 \underbrace{\mathbb{E}[\|\hat{\boldsymbol{d}}^t\|^2]}_{T_2}. \tag{14}$$

Bounding T_1 Following the definition, we have

$$\begin{aligned}
\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{d}}^t\|^2 &\leq \mathbb{E}\|\nabla f(\boldsymbol{x}^t) \pm \hat{\boldsymbol{G}}^t - \hat{\boldsymbol{d}}^t\|^2 \\
&\leq 2\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2 + 2\mathbb{E}\|\hat{\boldsymbol{G}}^t - \hat{\boldsymbol{d}}^t\|^2 \\
&\leq 2\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2 + 2\frac{1}{\eta_l^2 K^2}\mathbb{E}\|\sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{g}_i^t - \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{m}_i^t\|^2 \\
&\leq 2\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2 + 2\frac{1}{\eta_l^2 K^2}\mathbb{E}\|\sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{g}_i^t \pm \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{g}_i^t - \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{m}_i^t\|^2 \\
&\leq 2\underbrace{\mathbb{E}\|\nabla f(\boldsymbol{x}^t) - \hat{\boldsymbol{G}}^t\|^2}_{(7)} + 4\underbrace{\frac{1}{\eta_l^2 K^2}\mathbb{E}\|\sum_{i=1}^N \boldsymbol{\lambda}_i \boldsymbol{g}_i^t - \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{g}_i^t\|^2}_{(12)} + 4\frac{1}{\eta_l^2 K^2}\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{g}_i^t - \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{m}_i^t\|^2,
\end{aligned}$$

where the first and second terms have been bounded.

We know

$$\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{g}_i^t - \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \boldsymbol{m}_i^t\|^2 = \mathbb{E}\|\sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i}{\sqrt{\boldsymbol{\lambda}_i}} \cdot \sqrt{\boldsymbol{\lambda}_i}(\boldsymbol{g}_i^t - \boldsymbol{m}_i^t)\|^2 \leq \sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i^2}{\boldsymbol{\lambda}_i} \cdot \sum_{i=1}^N \boldsymbol{\lambda}_i \mathbb{E}\|\boldsymbol{g}_i^t - \boldsymbol{m}_i^t\|^2 \tag{15}$$

by Cauchy-Schwarz inequality. Now, we turn to bound the approximation error of moving-averaged local updates. Letting $p_i = \text{Prob}(i \in S^t)$ be the probability of i-th clients be selected at the round, we have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{m}_i^t - \boldsymbol{g}_i^t\|^2 &= \mathbb{E}\|(1-p_i)\boldsymbol{m}_i^{t-1} + p_i((1-\alpha)\boldsymbol{m}_i^{t-1} + \alpha \boldsymbol{g}_i^t) - \boldsymbol{g}_i^t\|^2 \\
&= \mathbb{E}\|(1-\alpha p_i)\boldsymbol{m}_i^{t-1} \pm (1-\alpha p_i)\boldsymbol{g}_i^{t-1} - (1-\alpha p_i)\boldsymbol{g}_i^t\|^2 \\
&\leq (1-\alpha p_i)^2 \mathbb{E}\|\boldsymbol{m}_i^{t-1} - \boldsymbol{g}_i^{t-1}\|^2 + (1-\alpha p_i)^2 \mathbb{E}\|\boldsymbol{g}_i^{t-1} - \boldsymbol{g}_i^t\|^2 \\
&\leq (1-\alpha p_i)^2 \mathbb{E}\|\boldsymbol{m}_i^{t-1} - \boldsymbol{g}_i^{t-1}\|^2 + (1-\alpha p_i)^2 \mathbb{E}\|\boldsymbol{g}_i^{t-1}\|^2 + (1-\alpha p_i)^2 \mathbb{E}\|\boldsymbol{g}_i^t\|^2.
\end{aligned}$$

Then, letting $\beta_i = (1-\alpha p_i)^2 \ll 1$ for simple notion and unrolling the recursion from time 0 to t with the initialization such that $\mathbb{E}\|\boldsymbol{m}_i^0 - \boldsymbol{g}_i^0\|^2 = 0$, we have

$$\mathbb{E}\|\boldsymbol{m}_i^t - \boldsymbol{g}_i^t\|^2 \le \sum_{\tau=0}^{t-1} \beta_i^{t-\tau}(\mathbb{E}\|\boldsymbol{g}_i^\tau\|^2 + \mathbb{E}\|\boldsymbol{g}_i^{\tau+1}\|^2).$$

Noting that $\beta_i \ll 1, \forall i \in [N]$, we can simplify the expression by observing that the terms involving $\beta^{t-\tau}$ decay rapidly. This allows us to approximate the sum by focusing primarily on the latest terms t, effectively reducing the accumulation effect. We assume a constant ρ that simplifies the inequality as

$$\mathbb{E}\|\boldsymbol{m}_i^t - \boldsymbol{g}_i^t\|^2 \le 2\beta_i\rho(\mathbb{E}\|\boldsymbol{g}_i^{t-1}\|^2 + \mathbb{E}\|\boldsymbol{g}_i^t\|^2) \le 2\beta_i\rho\max(\mathbb{E}\|\boldsymbol{g}_i^{t-1}\|^2, \mathbb{E}\|\boldsymbol{g}_i^t\|^2),$$

Without loss of generality, we take

$$\mathbb{E}\|\boldsymbol{m}_i^t - \boldsymbol{g}_i^t\|^2 \le 2\beta\rho\mathbb{E}\|\boldsymbol{g}_i^t\|^2.$$

Combining the above terms, we obtain

$$\begin{aligned}
T_1 &\le 4L^2\gamma\mathbb{E}\|\boldsymbol{G}^t\|^2 + 4c\eta_l^2L^2K\sigma_l^2 + 4\frac{\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2K^2}\sum_{i=1}^N \boldsymbol{\lambda}_i \left\|\boldsymbol{g}_i^t\right\|^2 + 8\frac{\beta\rho}{\eta_l^2K^2}\sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i^2}{\boldsymbol{\lambda}_i} \cdot \sum_{i=1}^N \boldsymbol{\lambda}_i\mathbb{E}\|\boldsymbol{g}_i^t\|^2 \\
&= 4\left(L^2 + \frac{\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2K^2} + \frac{2\beta\rho}{\eta_l^2K^2}\sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i^2}{\boldsymbol{\lambda}_i}\right)\gamma\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + 4c\eta_l^2L^2K\sigma_l^2 \\
&\le 4\left(L^2 + \frac{(1+2\beta\rho)\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2K^2}\right)\gamma\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + 4c\eta_l^2L^2K\sigma_l^2
\end{aligned}$$

where the last inequality assumes $\sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i^2}{\boldsymbol{\lambda}_i} \le \chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}} = \sum_{i=1}^N \frac{(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^2}{\boldsymbol{\lambda}_i}$ without loss of generality.

Bounding T_2 Following the definition, we have

$$\begin{aligned}
\mathbb{E}\|\hat{\boldsymbol{d}}^t\|^2 &= \frac{1}{\eta_l^2K^2}\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i\boldsymbol{m}_i^t \pm \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i\boldsymbol{g}_i^t\|^2 \\
&\le \frac{2}{\eta_l^2K^2}\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i(\boldsymbol{m}_i^t - \boldsymbol{g}_i^t)\|^2 + \frac{2}{\eta_l^2K^2}\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i\boldsymbol{g}_i^t\|^2 \\
&= \frac{2}{\eta_l^2K^2}\underbrace{\mathbb{E}\|\sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i(\boldsymbol{m}_i^t - \boldsymbol{g}_i^t)\|^2}_{(15)} + \frac{2}{\eta_l^2K^2}\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2
\end{aligned}$$

Substituting corresponding terms, we have

$$\begin{aligned}
T_2 &\le 8\beta\sum_{i=1}^N \frac{\tilde{\boldsymbol{\lambda}}_i^2}{\boldsymbol{\lambda}_i} \cdot \sum_{i=1}^N \boldsymbol{\lambda}_i\mathbb{E}\|\boldsymbol{g}_i^t\|^2 + \frac{2}{\eta_l^2K^2}\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 \\
&\le \frac{8}{\eta_l^2K^2}\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + \frac{2}{\eta_l^2K^2}\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 \\
&= 2\frac{1 + 4\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma}{\eta_l^2K^2}\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2
\end{aligned}$$

Putting together Substituting T_1 and T_2 into (14), we have

$$
\begin{aligned}
\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 &\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{\eta K} + 4c\eta_l^2 L^2 K \sigma_l^2 \\
&\quad + 4\left(L^2 + \frac{(1+2\beta\rho)\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2}\right)\gamma \mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 + L\tilde{\eta}\cdot 2\frac{1+4\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma}{\eta_l^2 K^2}\mathbb{E}\|\tilde{\boldsymbol{G}}^t\|^2 \\
&\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{\eta K} + 4c\eta_l^2 L^2 K \sigma_l^2 \\
&\quad + 4\left(L^2 + \frac{(1+2\beta\rho)\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 K^2}\right)\gamma \cdot (K\eta_l^2\sigma_l^2 + \rho^t(\tilde{\boldsymbol{\lambda}})) + L\tilde{\eta}\cdot 2\frac{1+4\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma}{\eta_l^2 K^2}\cdot (K\eta_l^2\sigma_l^2 + \rho^t(\tilde{\boldsymbol{\lambda}})) \\
&\leq \frac{2(f(\boldsymbol{x}^t) - \mathbb{E}\left[f(\boldsymbol{x}^{t+1})\right])}{\eta K} + 4(c + \left(1 + \frac{(1+2\beta\rho)\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}}{\eta_l^2 L^2 K^2}\right)\gamma)\cdot \eta_l^2 L^2 K\sigma_l^2 + 2L(1+4\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma)\cdot\eta\sigma_l^2 \\
&\quad + 2\eta L\cdot\left(1 + \frac{L\gamma}{8\eta_g} + \frac{2(1+2\beta\rho)\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma}{\eta K L} + 4\beta\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}\gamma\right)\cdot\frac{\rho^t(\tilde{\boldsymbol{\lambda}})}{\eta_l^2 K}.
\end{aligned}
$$

Taking averaging of both sides from time $t = 0$ to $T-1$, and defining $\tilde{\chi} = \frac{1}{T}\sum_{t=0}^{T-1}\chi^2_{\boldsymbol{\lambda}\|\tilde{\boldsymbol{\lambda}}}$ for notation simplicity, we obtain

$$
\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 \leq \frac{2(f(\boldsymbol{x}^0) - \mathbb{E}\left[f(\boldsymbol{x}^T)\right])}{\eta T K} + 4\left(c + \gamma + \frac{(1+2\beta\rho)\tilde{\chi}\gamma}{\eta_l^2 L^2 K^2}\right)\cdot\eta_l^2 L^2 K\sigma_l^2 + 2L(1+4\beta\tilde{\chi}\gamma)\cdot\eta\sigma_l^2 \\
+ 2\eta L\cdot\left(1 + \frac{L\gamma}{8\eta_g} + \frac{2(1+2\beta\rho)\tilde{\chi}\gamma}{\eta K L} + 4\beta\tilde{\chi}\gamma\right)\frac{1}{T}\sum_{t=0}^{T-1}\frac{\rho^t(\tilde{\boldsymbol{\lambda}})}{\eta_l^2 K}.
\end{aligned}
\tag{16}
$$

Compared with (13), we see that FedAWARE with partial participation only induces additional minor coefficients multiplied by β. As $\beta \ll 1$ in practice, we argue that FedAWARE is robust to partial client participation in theory. Moreover, empirical evidence in the main paper also proves this point.

B Experiment Details

B.1 Experiment Details

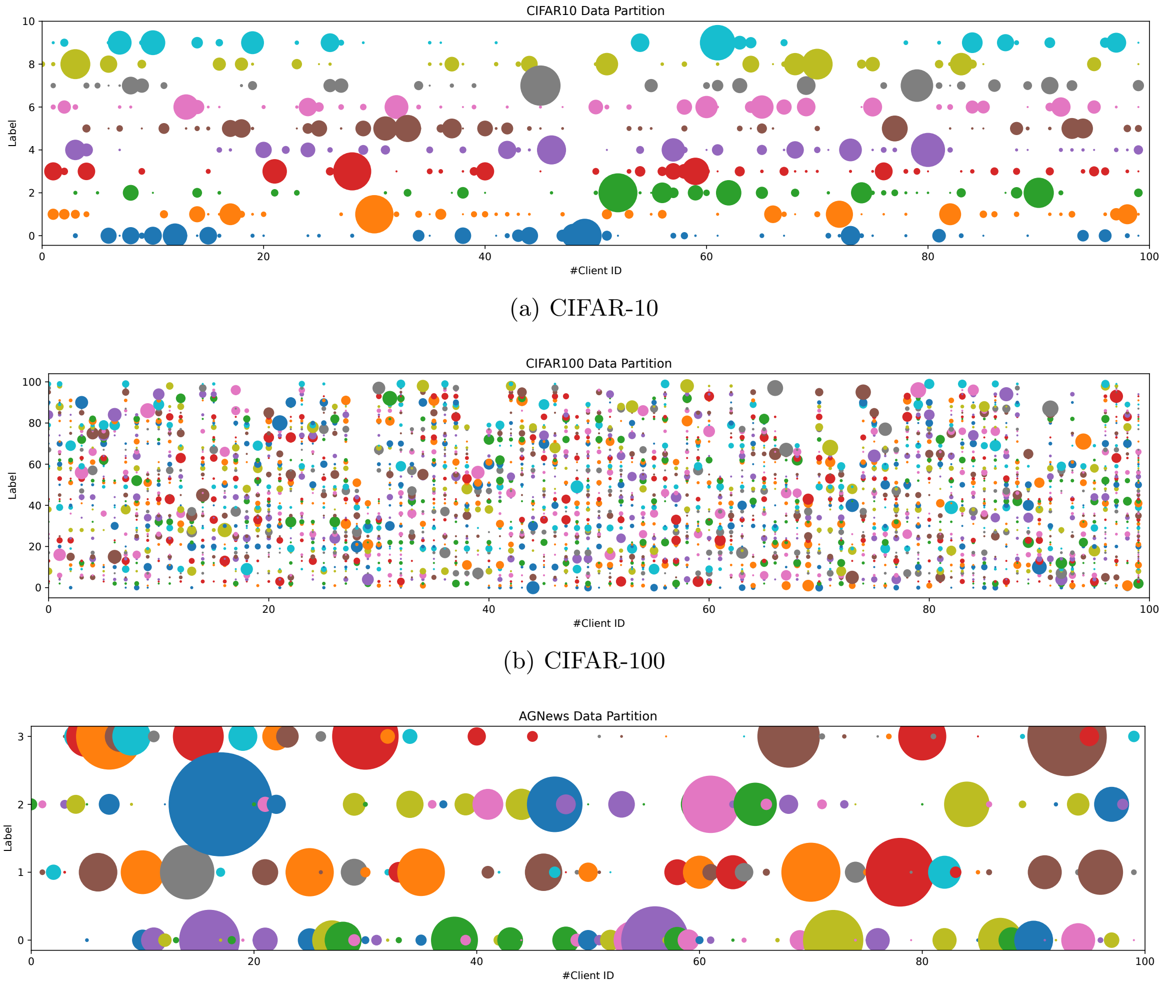
Platform The experiment implementations are supported by FedLab (Zeng et al., 2023b) and Codebase (Zhang et al., 2023b). Our experiments run on a Linux server with 4*2080Ti GPU.

Data partition We present the data distribution of datasets in Figure 5.

Algorithm-specified hyperparameters We set the rate of client participation to be 10%, and use $\eta_g = 1$ for FedAvg, FedAvgM, FedProx, FedCM, and FedAWARE. For the momentum parameter, FedAvgM is chosen from $\{0.7, 0.9, 0.97, 0.997\}$ following the original paper, and FedCM is chosen from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For weights of the penalty term in FedProx, we tune it from the grid $\{0.01, 0.1, 1, 10\}$. For FedYogi, we set momentum parameter $\beta_1 = 0.9$, second-momentum parameter $\beta_2 = 0.99$, and adaptivity $\tau = 10^{-4}$ following the original paper. Besides, We select η_g for FedYogi and FedAMS by grid-searching tuning from $\{10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{1}\}$. The parameter of FedDyn is chosen among $\{0.1, 0.01, 0.001\}$ from the original paper. For FedAMS, we set $\beta_1 = 0.9, \beta_2 = 0.99$ follows the original paper. Then, we grid search for the best global learning rate $\eta_g = \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and the best stabilization term $\epsilon = \{10^{-8}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. We set $\alpha = 0.5$ for FedAWARE.

B.2 Additional Experimental Results

Experiments results on CIFAR-10 and AGNews We present the FedAWARE extension results of CIFAR-10 and AGNews in Figure 8. We observe that raw federated algorithms test accuracy curves with heavy spike problems, indicating the heterogeneous data impacts on FL generalization. On both tasks, FedAWARE enlarges



(a) CIFAR-10

(b) CIFAR-100

(c) AGNews

Figure 5: Visualization of data distribution.

the e-LUD values of all algorithms and thus stabilizes the test accuracy curves. Moreover, connecting the results of CIFAR-100 in the main paper, we observe that the FedAWARE extension may reduce convergence slightly on CIFAR-10 and CIFAR-100 tasks while the convergence of AGNews tasks is improved.

Relation between client coherence and LUD To illustrate the nuances, we roughly reproduced the experiments from Figure 5 in FedLAW (Li et al., 2023), as shown in Figure 6. The experiments involve 20 clients, where the first 10 have label-balanced datasets, and the remaining 10 have label-unbalanced datasets. We observe consistent trends across three different measures, suggesting their interrelation. Additionally, we visualize cosine similarity matrices at training stages $T = 3, 60, 150$, which reveal that local updates become less similar in later training rounds.

Discussion: the meaning of cosine similarity in FL. Cosine similarity plays a different role in FL training. We outline the varying meanings of consistent gradients (cosine values) during the training process:

- **Initialization and early stage:** High cosine values indicate similar data distributions (even IID) in the FL system (Zeng et al., 2023a; Sattler et al., 2020). It is well-established that distributed learning with IID data outperforms non-IID data in terms of generalization. In this phase, *gradient coherence claims that higher similarities between the gradients of samples will boost generalization.*
- **Training stage:** As training progresses, cosine similarity reflects conflicts between local objectives. Increased dissimilarity in gradients signals divergence from local minima. We observe that cosine values generally

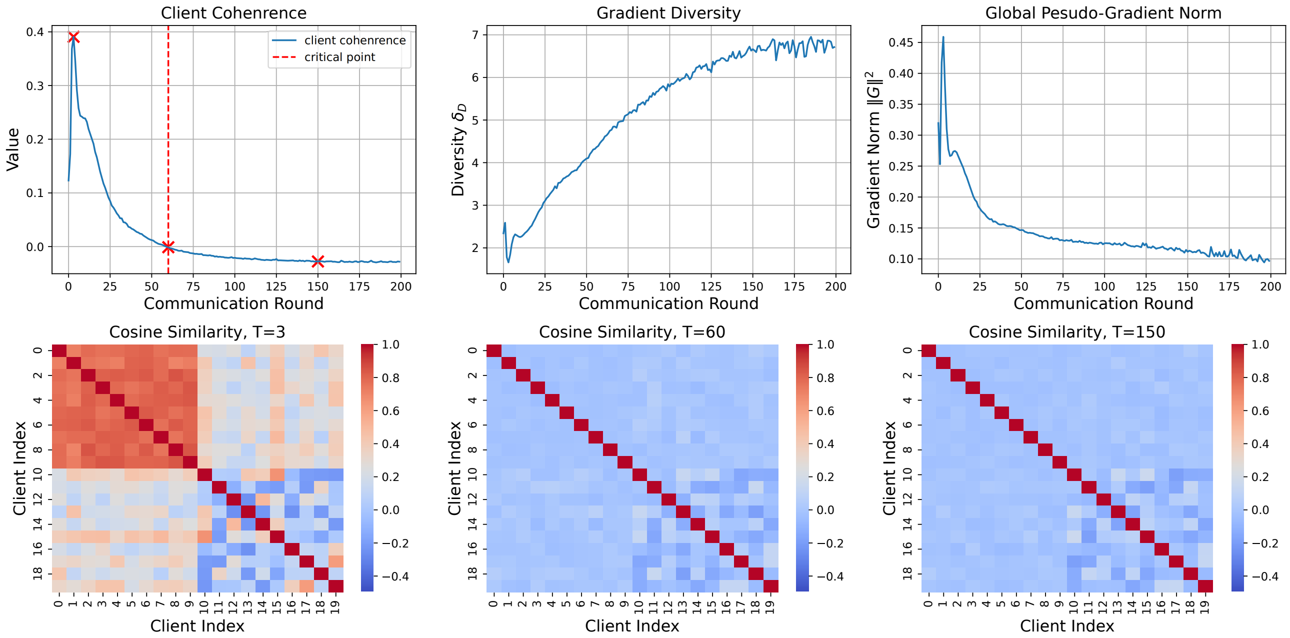


Figure 6: Visualization of client coherence and cosine similarity.

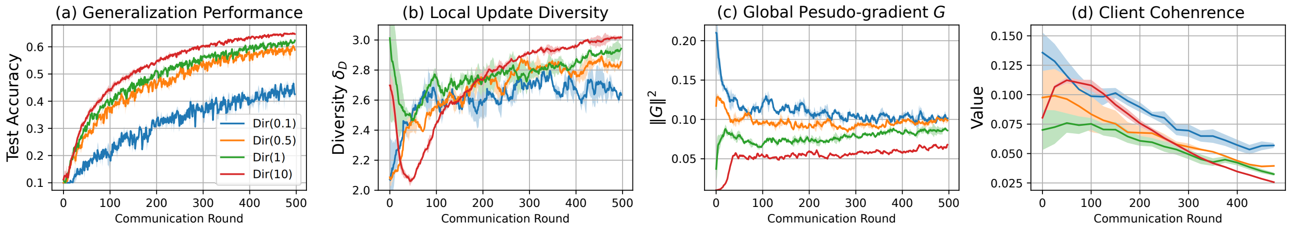
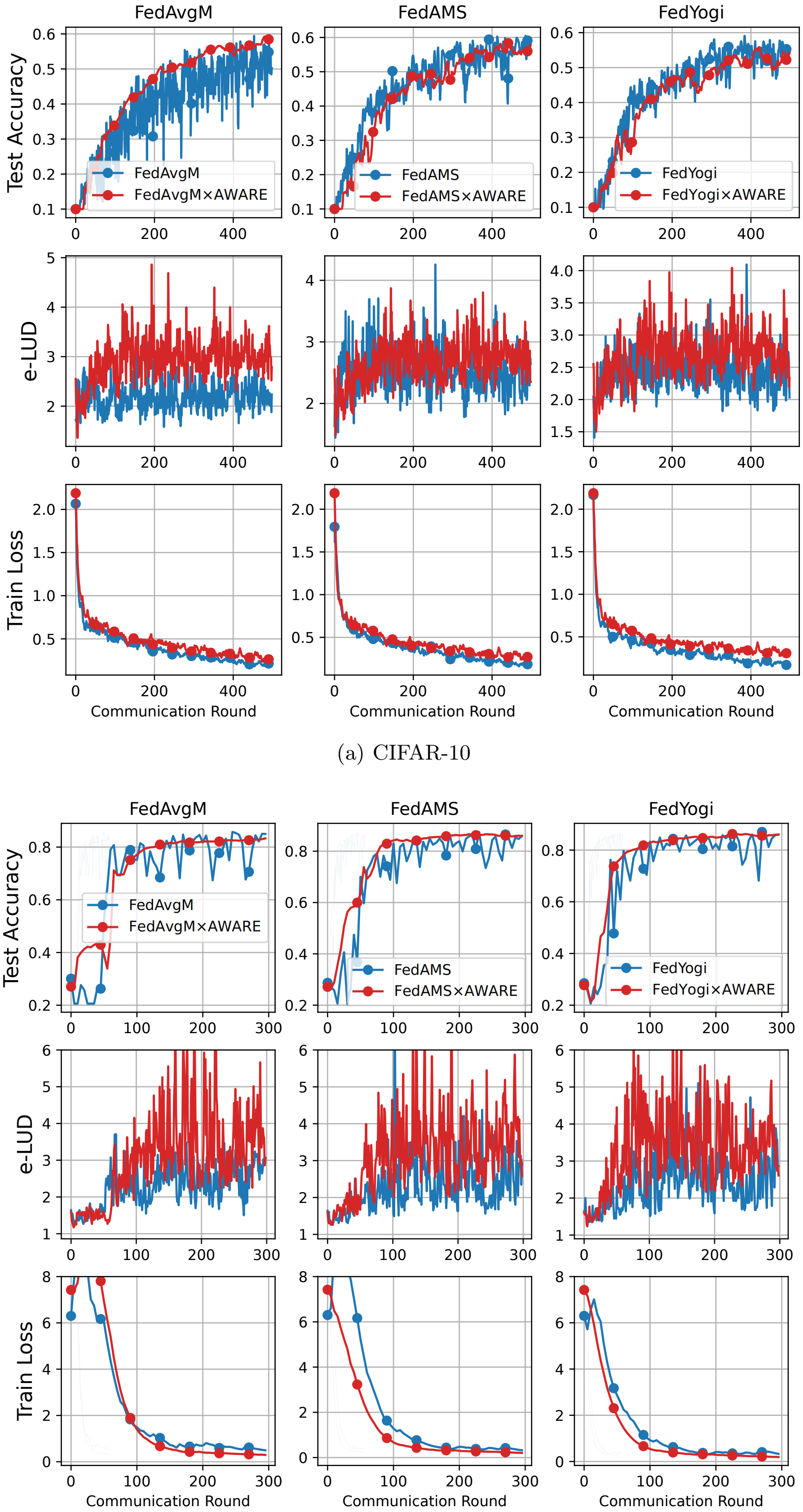


Figure 7: Training dynamics of IID and Non-IID settings.

decrease over time, signaling growing gradient dissimilarity (Sattler et al., 2020; Li et al., 2023). Interestingly, we find that larger dissimilarity in later training rounds correlates with better generalization. This trend is further supported in Figure 7, where we compare client coherence and LUD across different heterogeneous settings.

In summary, we clarify that the findings on client coherence (Li et al., 2023; Chatterjee, 2020) and gradient diversity (Yin et al., 2018) in federated learning are not contradictory.

Observation experiments on IID settings Figure 7 compares Dirichlet partitions Dir(1) and Dir(10) as IID settings with Dir(0.1) and Dir(0.5) as Non-IID settings. We observe distinct trends between the two. In IID settings, the global gradient norm remains lower than in Non-IID settings. However, the LUD curve is higher in IID settings, as the global norm stabilizes while local gradient norms increase due to stochasticity. In essence, FL in IID settings behaves similarly to *Local SGD* (Gu et al., 2023). Notably, the decaying property 4.1 of global norms may not hold in IID settings. Despite that, our convergence analysis of Theorem 4.2 also holds, as the cumulative client consensus remains lower in IID settings. Furthermore, the relationship between LUD and generalization continues to align with the theoretical findings on gradient diversity (Yin et al., 2018).



(a) CIFAR-10

(b) AGNews

Figure 8: Training dynamics of FedAWARE extension.