SP-CACW: Convergence-Aware Client Weighting for Selfish Personalized Learning

Yaron Kiselman
Technion - Israel Institute of Technology
Haifa, Israel

Kfir Y. Levy
Technion - Israel Institute of Technology
Haifa, Israel

Abstract

Collaborative learning is sustainable only when it benefits each participant. Standard federated learning optimizes a global average objective, which can under perform for clients whose data distributions differ substantially from the population. We study *selfish personalization*: how a designated target client can use peer gradients to minimize its own risk while avoiding negative transfer. We propose *SP-CACW*, a convergence-aware client-weighting framework that selects aggregation weights by minimizing an upper bound on the target client's convergence error. The resulting rule explicitly trades off peer bias against stochastic variance and can assign zero weight to harmful peers. We provide convergence guarantees under smoothness and bounded-variance assumptions and evaluate the method on MNIST, CIFAR-100, and LEAF Shakespeare, where it is competitive with or improves over strong personalized and clustering baselines.

1 Introduction

In today's data-driven world, information is inherently distributed across a large number of clients, each collecting data locally and independently. This setting poses substantial challenges for traditional centralized learning approaches, particularly regarding privacy preservation, communication overhead, and data ownership constraints. Federated Learning (FL) has emerged as a promising paradigm that enables the training of a shared global model by coordinating learning across multiple decentralized data sources, while eliminating the need for direct access to clients' raw data Hu et al. [2024], Majeed et al. [2022], Xu et al. [2021].

While global modeling is effective, it struggles with non-IID data Hanzely and Richtárik [2020], Yu et al. [2020], Zhao et al. [2018], necessitating Personalized Federated Learning (PFL) Fallah et al. [2020]. However, standard PFL approaches that blend global and local representations often limit personalization for clients with distinct distributions Dinh et al. [2020], Collins et al. [2021b], Lai et al. [2024]. Consequently, recent works favor clustering strategies where clients collaborate only with similar peers to minimize negative transfer Ghosh et al. [2022], Werner et al. [2023], placing FL paradigms on a spectrum from fully global to client-specific optimization. This spectrum spans from traditional FL (a single global model) through Personalized FL (PFL) to Selfish Federated Learning (SFL), where a client leverages the network solely to maximize its own performance. While FL and PFL are extensively studied, SFL remains largely overlooked. SFL is a fundamental problem that necessitates dedicated study: understanding how to optimally leverage a federated environment for a single client uncovers structural insights that may extend to broader multi-client challenges.

The practical necessity of SFL is driven by two real-world paradigms. In **Inter-Silo Networks** (e.g., hospital consortium), SFL facilitates a *reciprocal knowledge exchange* where clients share gradients to maximize their distinct local objectives without submitting to a compromised global consensus. Additionally, for **Public Data Exploitation**, SFL acts as a dynamic valuation mechanism that safely filters massive, heterogeneous public datasets, extracting useful signals while preventing negative transfer and harmful bias.

While many existing PFL methods aim to learn a personalized model for each client by leveraging data from all clients, their theoretical analyses typically focus on convergence guarantees of the averaged clients Huang et al. [2021]. While these analyses are well-suited for the PFL setting, they do not directly translate to the SFL. Furthermore, many existing works rely on strong assumptions regarding data distributions or optimization functions Werner et al. [2023], Mansour et al. [2020], Ghosh et al. [2022], participation patterns, or model structure. Consequently, they provide limited insight into the optimization behavior of an individual target client. Conversely, our work explicitly focuses on the convergence properties of a single client's objective.

Our main contributions are summarized as follows:

• We introduce **Convergence-Aware Client Weighting for Selfish Federated Learning (SP-CACW)**, a principled framework that determines optimal client aggregation weights by minimizing an upper bound on the target client's convergence rate, yielding efficient and stable training dynamics.

• We derive guarantees which substantiate the benefit of **SP-CACW** for the target client. In the specific case of a cluster structured problems, we show that our method achieves a rate of $O\left(\sigma/\sqrt{n_0 T}\right)$, effectively leveraging the cluster size n_0 to reduce variance (Section 3.2).

• We demonstrate the effectiveness of SP-CACW on MNIST, CIFAR, and LEAF Shakespeare datasets, showing consistent improvements over existing personalized and clustering-based baselines.

Related Work. Personalized Federated Learning (PFL) has been extensively studied as a solution to the performance degradation caused by non-IID data in global model aggregation. Seminal approaches, such as Ditto Li et al. [2021] and pFedMe Dinh et al. [2020], introduce client-side regularization to stabilize local updates while maintaining proximity to a shared global anchor. Similarly, meta-learning frameworks like Per-FedAvg Fallah et al. [2020] aim to learn an adaptable initial model that can be quickly fine-tuned to local distributions. Other works facilitate selective information sharing by leveraging multi-task learning Zhang and Yang [2022], knowledge distillation Mansourian et al. [2025], or discrepancy-aware collaborations; e.g. approaches like FedDisco Ye et al. [2023] utilize categorical data statistics to optimally weight peer contributions. While foundational, these methods often rely on implicit assumptions regarding client alignment or introduce complex auxiliary objectives that can hinder scalability and increase computational overhead.

To address data heterogeneity more explicitly, two dominant paradigms have emerged: **(i) Clustering based methods:** like IFCA and FeSEM Ghosh et al. [2022], Werner et al. [2023], partition clients into groups based on gradient or model affinity, training a specialized model for each cluster. While effective when clear cluster structures exist, these approaches are sensitive to ambiguous decision boundaries and intermittent client participation, often leading to unstable cluster assignments. Alternatively, **(ii) similarity-based approaches:** like pFedSim and FedAMP Tan et al. [2023], Huang et al. [2021] circumvent rigid grouping by learning pairwise relationships to weight peer contributions. However, these methods typically rely on heuristic similarity metrics (e.g., Euclidean distance or cosine similarity) which do not necessarily correlate with useful descent direction for a specific client.

In contrast to these paradigms, our work adopts a Selfish Federated Learning (SFL) perspective. We depart from the reliance on heuristic similarity measures or latent cluster structures. Instead, we introduce a *convergence-aware* weighting mechanism. By explicitly deriving aggregation weights that minimize the upper bound on the target client's convergence rate, we ensure that peer contributions are valued based on their *optimization utility*—effectively, how much they accelerate the target's training—rather than their static data similarity. This provides a principled alternative to heuristic personalization strategies.

2 Problem Setup

Selfish Personalization (SP). Consider a federated learning setting with M clients indexed $i \in \{0, 1, \ldots, M-1\}$. Each client i draws samples from a *local data distribution* $\mathcal{D}^{(i)}$, the distributions $\{\mathcal{D}^{(i)}\}_i$ may be distinct and are unknown. Our aim is to find a model $w \in \mathbb{R}^d$ specifically for a designated **target client**, denoted as client 0. Formally, the expected loss (population risk) of the target client is defined as follows:

$$\min_{w \in \mathbb{R}^d} f^{(0)}(w), \quad \text{where} \quad f^{(0)}(w) := \mathbb{E}_{\xi \sim \mathcal{D}^{(0)}}[\ell(w; \xi)].$$

Here, $\ell(w;\xi)$ represents the loss function for a model parameter w and a data sample ξ. We focus on non-convex optimization, where global loss minimization is generally intractable, so our goal is finding an approximate stationary point, i.e., a point w for which the expected gradient norm $\mathbb{E}\|\nabla f^{(0)}(w)\|^2$ tends to zero.

To focus on the core challenge we assume that the target client may communicate with all other clients. At every round t, client 0 may send its current weights w_t to all peers, and in return receive a gradient estimate $g_t^{(i)} := \nabla\ell(w_t;\xi_t^{(i)})$, where $\xi_t^{(i)} \sim \mathcal{D}^{(i)}$. Client 0 then uses the set of received gradients $\{g_t^{(i)}\}$ to update the local model w and query new estimates. We assume that the samples $\{\xi_t^{(i)}\}_t$ are independent across time and across clients.

Communication Efficiency Note. To simplify the exposition our formulation focuses on the Cross-Silo FL setting with full communication. Nevertheless, Our SP-CACW is broader and operates efficiently without full participation. Concretely, as we detail in Appendix E, our framework generalizes to accommodate *partial participation* via unbiased peer sampling. Moreover, as depicted in 3.1 our approach naturally induces *learned sparsity*, enabling it to prune edges to unhelpful peers.

Assumption 2.1 (Smoothness)**.** *The objective function of client 0 is L-smooth. That is, there exists* $L > 0$ *such that,* $\|\nabla f^{(0)}(w) - \nabla f^{(0)}(u)\| \le L\|w-u\|, \ \ \forall w,u \in \mathbb{R}^d$

Assumption 2.2 (Bounded Variance)**.** *For each client i, there exists* $\sigma^{(i)} > 0$ *such that for any* $w \in \mathbb{R}^d$*:* $\mathbb{E}_{\xi\sim\mathcal{D}^{(i)}}\left[\|\nabla\ell(w;\xi) - \nabla f^{(i)}(w)\|^2\right] \le (\sigma^{(i)})^2$.

3 Our Basic Approach

Design Philosophy: Directly Targeting Convergence Rather than Similarity. Classic personalized FL approaches typically focus on estimating *data similarity*—clustering clients or weighting them based on parameter proximity Ghosh et al. [2022], Huang et al. [2021]. While intuitive, this approach relies on the heuristic assumption that "similar data implies useful gradients." This is not always the case. For example, a peer with a slightly shifted distribution but extremely low gradient noise might be more valuable for learning than a highly similar peer with noisy gradients. This demonstrates that similarity alone is an insufficient proxy for peer quality.

To mitigate this, *we design client aggregation rules that directly minimize a theoretical upper bound on the target client's convergence rate.* By optimizing this bound directly, we allow the algorithm to optimally trade-off bias (dissimilarity) with variance (noise) of different clients, which facilitates practical and sample efficient training.

Formulation: Weighted Aggregation Our approach towards utilizing the information from other (possibly biased) clients is to employ a *weighted* gradient update rule. Concretely, at each round t, client 0 sends the current model w_t to all clients and receives $\{g_t^{(i)}\}_{i\in[M]}$: gradient estimate based on their local data. Instead of uniformly aggregating these gradients like in standard FL, the target client performs a **weighted-SGD** aggregation using $\alpha_t \in \Delta_M$,

$$w_{t+1} = w_t - \eta g_t, \quad g_t = \sum_{i=0}^{M-1} \alpha_t^{(i)} g_t^{(i)} \tag{1}$$

where $g_t^{(i)} = \nabla\ell(w_t;\xi_t^{(i)})$; $\xi_t^{(i)} \sim \mathcal{D}^{(i)}$, and η is the stepsize. Clearly, the choice of $\alpha_t^{(i)}$ determines the effects of $g_t^{(i)}$ on the aggregated gradient g_t which in turn effects convergence. As we show in the next section, the convergence rates of the weighted SGD above are related to a tradeoff between the bias and variance of $g_t^{(i)}$:

- **Gradient Bias ($b_t^{(i)}$)** of client i at time t is defined as: $b_t^{(i)} = \nabla f^{(i)}(w_t) - \nabla f^{(0)}(w_t)$.
- **Gradient Variance ($\sigma^{(i)^2}$):** is the stochastic noise inherent in client i's local update (see Assumption 2.2).

Overall, the bias and variance of aggregated gradient g_t depend on α_t and $\{b_t^{(i)}, \sigma^{(i)}\}_{i\in[M]}$, and effect the overall convergence of the algorithm. Curiously, the effects of bias and variance are uneven, and roughly speaking—a high bias is much more harmful than a high variance. Thus, our aim is to pick the weights α_t to optimize the tradeoff between bias with variance, thus directly facilitating better convergence.

Convergence Induced Cost. For every round t let us define a cost function over weights $\alpha \in \Delta_M$,

$$\mathcal{L}_t(\alpha) := \| \sum_{i\in[M]} \alpha^{(i)} b_t^{(i)} \|^2 + \eta L \| \sum_{i\in[M]} \alpha^{(i)} g_t^{(i)} \|^2 \,. \tag{2}$$

Note that the bias terms directly effect $\mathcal{L}_t(\cdot)$ but the variance terms appears only implicitly via the second term. As we show later(A.2), one can decompose $\| \sum_{i\in[M]} \alpha^{(i)} g_t^{(i)} \|^2 \le 2\| \sum_{i\in[M]} \alpha^{(i)} b_t^{(i)} \|^2 + \sum_{i\in[M]} (\alpha^{(i)})^2 (\sigma^{(i)})^2 + 2\|\nabla f^{(0)}(w_t)\|^2$, and therefore, $\mathcal{L}_t(\alpha) \le (1+2\eta L)\| \sum_{i\in[M]} \alpha^{(i)} b_t^{(i)} \|^2 + 2\eta L \sum_{i\in[M]} (\alpha^{(i)})^2 (\sigma^{(i)})^2$. Since η is typically small i.e. $\eta \propto 1/\sqrt{T}$ (where T is the total number of SGD updates) then $1+\eta L \approx 1$. This shows that the bias terms in $\mathcal{L}_t(\cdot)$ come with a factor of ≈ 1 where variance terms come with a factor of $\eta L \ll 1$; implying that bias has a much stronger effect on $\mathcal{L}_t(\cdot)$.

The next theorem shows that the cumulative cost directly relates to the convergence of **weighted-SGD**.

Theorem 3.1. *Suppose assumptions 2.2, 2.1 hold. Then applying the weighted-SGD Algorithm (Eq. (1)) for T rounds, with a fixed weights $\alpha_t = \alpha \in \Delta_M$, and $\eta > 0$, ensures,*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \le \frac{2}{T}\left(\frac{\delta_1}{\eta} + \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_t(\alpha)] \right) \text{ , where}$$

$\delta_1 := f^{(0)}(w_1) - f^{(0)}(w^*)$, and w^* = is a global minimizer of $f^{(0)}$, and $\mathcal{L}_t(\cdot)$ is defined in Eq. (2).

The above suggests that by picking weights α that minimize the cumulative costs directly leads to minimizing the upper bound on the convergence rate that serves the utility of target client.(full proof in A.1)

Challenges. There are two basic issues with this principled approach: **(i)** Computing the cost $\mathcal{L}_t(\cdot)$, requires the knowledge of $\{b_t^{(i)}\}_{i\in[M]}$ and $\{g_t^{(i)}\}_{i\in[M]}$. While we can simply compute the gradients, we do not have a direct access to the biases and therefore need to estimate them online. **(ii)** We do not know the costs $\mathcal{L}_t(\cdot)$'s in advance so we need to learn the α's online during the training process. We address both of these challenges in the Section 4, leading to an efficient and practical algorithm.

In the next subsection we shall explore the properties of the optimal choice of α as well as substantiate the potential benefit of this approach in an illustrative example.

3.1 Optimal Weighting and Potential Benefits

To gain analytical insights regarding the optimal choice of $\min_\alpha \sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_t(\alpha)]$ we shall simplify the expression for $\mathbb{E}[\mathcal{L}_t(\cdot)]$, and analyze an upper bound for this expression. Concretely, define,

$$\bar{\mathcal{L}}_t(\alpha) := (1+2\eta L) \sum_{i\in[M]} \alpha^{(i)} \|b_t^{(i)}\|^2 + \eta L \sum_{i\in[M]} (\alpha^{(i)})^2 (\sigma^{(i)})^2 + 2\eta L \|\nabla f^{(0)}(w_t)\|^2$$

Then it can be shown that for any $\alpha \in \Delta_M$ the following holds: $\mathbb{E}[\mathcal{L}_t(\alpha)] \le \mathbb{E}\bar{\mathcal{L}}_t(\alpha)$. This follows as a result of $\| \sum_{i\in[M]} \alpha^{(i)} b_t^{(i)} \|^2 \le \sum_{i\in[M]} \alpha^{(i)} \|b_t^{(i)}\|^2$ which holds due to the fact that $\alpha \in \Delta_M$, as well as to a decomposition of $g_t := \sum_{i\in[M]} \alpha_t^{(i)} g_t^{(i)}$ into bias and variance, and taking expectation. To further simplify the $\bar{\mathcal{L}}_t$ lets us assume that the bias norms are fixed, i.e. $\|b_t^{(i)}\| = \|b^{(i)}\|$, for all i, t. This immediately implies that the α-dependent part is time-independent up to an additive term independent of α, and therefore $\frac{1}{T}\sum_{t=1}^{T} \bar{\mathcal{L}}_t(\alpha) = \bar{\mathcal{L}}(\alpha) := (1+2\eta L) \sum_{i\in[M]} \alpha^{(i)} \|b^{(i)}\|^2 + \eta L \sum_{i\in[M]} (\alpha^{(i)})^2 (\sigma^{(i)})^2 + 2\eta L \|\nabla f^{(0)}(w_t)\|^2$.(full proof in A.2).

The next lemma characterizes the optimal solution of $\min_{\alpha\in_m} \bar{\mathcal{L}}(\alpha)$ (see the proof in Appendix A.2).

Lemma 3.1 (Optimal Weights)**.** *Consider the problem of minimizing the proxy function $\bar{\mathcal{L}}(\alpha)$ over the simplex. Then there exists $\lambda > 0$ such that an optimal solution is of the following form,*

$$\alpha^{(i)*} = \left[\frac{\lambda - C_1 \|b^{(i)}\|^2}{2C_2 (\sigma^{(i)})^2} \right]_+ ,$$

and we define $[z]_+ =: \max\{z, 0\}$, and $C_1 = 2L\eta + 1$, $C_2 = L\eta$.

The above lemma provides two fundamental insights:

• **Variance Scaling:** Optimal weights are inversely proportional to the variance $(\sigma^{(i)})^2$. This aligns with the natural intuition: noisier clients should receive lower trust.

• **Bias Thresholding:** The term $\lambda - C_1\|b^{(i)}\|^2$ introduces a strict selection mechanism. Unlike Softmax or similarity-based weighting—which assign non-zero mass to all clients—our solution *sparsifies* the federation. Clients with bias exceeding the threshold λ/C_1 are assigned exactly zero weight, effectively removing their contribution to prevent negative transfer.

3.2 Illustrative Case Study: Pure Clusters

To illustrate a concrete benefit of our approach, consider the following scenario: Assume that the clients are divided into two distinct clusters:
• **A Target Cluster** I_0 where all clients draws samples from the same distribution as the target distribution; i.e. $\mathcal{D}^{(i)} = \mathcal{D}^{(0)},\ \forall i \in I_0$; implying that $b^{(i)} = 0,\ \forall i \in I_0$.
• **An off-target Cluster** I_{off}, where all clients draws samples from the same distribution $\mathcal{D}_{\text{off}} \neq \mathcal{D}^{(0)}$, i.e. $\mathcal{D}^{(i)} = \mathcal{D}_{\text{off}},\ \forall i \in I_{\text{off}}$, implying that $b^{(i)} = b,\ \forall i \in I_{\text{off}}$. We also assume that the target and off target distributions are distinct enough such that $\|b\| = O(1)$.

Finally, we assume that the gradient variance of all machines is equal, implying $\sigma^{(i)} = \sigma;\ \forall i \in [M]$.

We show in Lemma C.4 that in this setting our bound is $\text{err}_T^{\textbf{Ours}} := O\left(\frac{\sigma}{\sqrt{n_0 T}}\right)$. where $\text{err}_T := \frac{1}{T}\sum_{t=1}^{T}\|\nabla f^{(0)}(w_t)\|^2$. This bound **substantially improves over these natural baselines:** **(i)** Autarky (training alone), **(ii)** Standard FL and **(iii)** Federated Clustering (FC) Werner et al. [2023],

$$\text{err}_T^{\textbf{Autarky}} := O\left(\frac{\sigma}{\sqrt{T}}\right)\ ,\ \text{err}_T^{\textbf{FL}} := O\left(\|b\|^2 + \frac{\sigma}{\sqrt{MT}}\right)\ ,\ \text{err}_T^{\textbf{FC}} := O\left(\sqrt{\frac{\sigma^2}{n_0 T} + \frac{\sigma^3}{\Delta T}}\right)\ .$$

Where bounds are optimized over η. Notably, our rate improves over Autarky by a factor of $\sqrt{n_0}$ and removes the cluster separability (Δ) dependence found in the FC approach. This is crucial because when Δ is on the order of σ, the additional FC term becomes order σ^2/T inside the square root, which can dominate the cluster-variance term when n_0 is large. (Complete derivations for all comparative baselines are provided in Appendix C).

4 Our Refined Approach

The previous section implies that in order to facilitate personalization, we should pick weights $\alpha \in \Delta_M$ so as to minimize the cumulative cost $\mathbb{E}\sum_{t=1}^{T}\mathcal{L}_t(\alpha)$ (Eq. (2)). Recall we face two challenges with this principled approach: **(i)** Computing the cost $\mathcal{L}_t(\cdot)$, requires the knowledge of $\{b_t^{(i)}\}_{i\in[M]}$ and $\{g_t^{(i)}\}_{i\in[M]}$. While we can simply compute the gradients, we do not have a direct access to the biases and therefore need to estimate them online. **(ii)** We do not know the costs $\mathcal{L}_t(\cdot)$'s in advance so we need to learn the α's online during the training process. In this section we address these challenges, thereby leading to a practical and efficient algorithm.

4.1 Estimating the Biases

The bias of client 0 is clearly $b^{(0)} := 0$. As for biases of other clients compared to the Target client, we propose to estimate them by employing a running average during the training process. This is depicted in Equation (3).

$$\tilde{b}_t^{(i)} := g_t^{(i)} - g_t^{(0)}, \quad \widehat{b}_t^{(i)} = (1-\beta_t)\widehat{b}_{t-1}^{(i)} + \beta_t \tilde{b}_t^{(i)} \tag{3}$$

Equation (3) describes a recursive exponential moving average (EMA). The new estimate $\widehat{b}_t^{(i)}$ is calculated as a convex combination of the previous estimate $\widehat{b}_{t-1}^{(i)}$ and the current observation $b_t^{(i)}$. The parameter $\beta_t \in (0,1)$ acts as a smoothing factor, controlling the trade-off between historical stability and responsiveness to new data. Naturally, we can now define an approximate cost function $\widehat{\mathcal{L}_t(\alpha)}$ which is based on these bias estimates, $\widehat{\mathcal{L}_t(\alpha)} := \|\sum_{i\in[M]}\alpha^{(i)}\widehat{b}_t^{(i)}\|^2 + \eta L\|\sum_{i\in[M]}\alpha^{(i)}g_t^{(i)}\|^2$.

4.2 Implications of Employing $\widehat{\mathcal{L}_t(\alpha)}$

In this part we show that the cumulative proxy $\sum_{t=1}^{T} \widehat{\mathcal{L}_t(\cdot)}$ upper bounds the exact cumulative cost $\sum_{t=1}^{T} \mathcal{L}_t(\cdot)$ up to a small error $\mathcal{E}_T = O(\log T)$ and a constant factor. Thus, aiming to minimize the cumulative proxy costs, ensures good performance to the true cost. To simplify the analysis we assume that the biases are fixed in time, i.e. $b_t^{(i)} = b^{(i)}$, $\forall i, t$.

Lemma 4.1. ***(Cumulative Bias-Variance Decomposition)*** *Under Assumption 2.2 and under stationary bias assumption $b_t^{(i)} = b^{(i)}$ for all i, t, let the bias estimates $\{\widehat{b}_t\}_{t=1}^{T}$ be updated using Eq. (3). The cumulative true bias is bounded by the cumulative estimated bias plus a logarithmic penalty $\mathcal{E}_T$: $\mathbb{E}\left[\sum_{t=1}^{T} \|\alpha_t b_t\|^2\right] \leq 2\mathbb{E}\left[\sum_{t=1}^{T} \|\alpha_t \widehat{b}_t\|^2\right] + \mathcal{E}_T$. Specifically, under the **Sample Mean Schedule** ($\beta_t = 1/t$), the penalty term satisfies: $\mathcal{E}_T \leq 2\sigma_{bound}^2(1 + \log T)$. (Full proof in B.2)*

Corollary 4.1.1. *Lemma 4.1 directly implies the following: $\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}(\alpha_t)\right] \leq 2\mathbb{E}\left[\sum_{t=1}^{T} \widehat{\mathcal{L}}(\alpha_t)\right] + \mathcal{O}(\log T)$. Thus, this establishes that minimizing the cumulative proxy costs explicitly minimizes the true cumulative cost, up to a multiplicative constant and an additive logarithmic error term. Recalling Theorem 3.1, this implies that the convergence guarantees that are induced by finding α that minimizes the cumulative proxy costs, are equivalent to the error achieved by minimizing the true cumulative cost, up to a multiplicative factor of 2 and additive factor of $O(\log T/T)$.*

Remark 4.1. ***(Practical Adaptation via Constant Smoothing)*** *While Lemma 4.1 utilizes $\beta_t = 1/t$ to guarantee extremely accurate bias estimators, practical deep learning settings often involve non-stationary bias (drifting as parameters update). In such cases, a decaying schedule reacts too slowly to changes. Therefore, in our experiments, we employ a **Constant Smoothing Schedule** ($\beta_t = \beta$). This choice maintains a good balance: it effectively handles "moving bias" by exponentially weighting recent observations, which provides the necessary adaptability for stable training in dynamic environments. On the other hand it ensures an error reduction of $O(\beta)$ for the bias estimates.*

4.3 Overcoming Uncertainty via Regret Minimization

Our objective is to find weights α_t that minimize the auxiliary loss $\mathcal{L}_t(\alpha)$. This presents a twofold challenge: first, $b_t^{(i)}$ and $g_t^{(i)}$ are unknown when α_t is chosen; second, the true bias $b_t^{(i)}$ remains unobservable even after communication and must be approximated. We resolve this by framing the problem as *Online Convex Optimization* (OCO). In this model, the learner commits to a decision (α_t) before the loss function is revealed (or estimated). This allows us to circumvent the lack of *prior* information by minimizing the *regret*—a metric quantifying the performance gap between our adaptive strategy and the optimal static weighting. Formally, the regret R_T over T rounds is defined as: $R_T = \sum_{t=1}^{T} \mathcal{L}_t(\alpha_t) - \min_{\alpha \in \Delta_M} \sum_{t=1}^{T} \mathcal{L}_t(\alpha)$. By minimizing R_T, we guarantee that the algorithm performs competitively against the best fixed collaboration strategy chosen in hindsight.

Properties of $\mathcal{L}_t$ We observe that our loss function $\mathcal{L}_t(\alpha)$ is quadratic and convex in α. Furthermore, we show it is γ-exp-concave (D.1), with $\gamma = \frac{1}{(8+2L\eta)g_{max}^2}$, and its gradient is bound by $G_{\mathcal{L}} \leq (8+2L\eta)g_{max}^2$, here g_{max} is an upper bound on the norm of stochastic gradients (proof in App. D.1).

Exp-concavity is a powerful property allowing to achieve logarithmic regret bounds Hazan et al. [2007]. Concretely, exp-concavity allows to employ a regret minimization procedure called FOLLOW THE APPROXIMATE LEADER (FTAL). The latter resembles *Follow The Leader* (FTL), which selects weights minimizing the cumulative loss of past rounds, i.e. $\alpha_{t+1} = \arg\min_{\alpha} \sum_{\tau=1}^{t} \mathcal{L}_t(\alpha)$. Instead of minimizing raw past losses, FTAL minimizes a sequence of *quadratic approximations* that incorporate second-order curvature information. The complete implementation of the FTAL algorithm is detailed in Appendix D.2. Due to exp-concavity we can invoke Theorem 6 from Hazan et al. [2007]:

Theorem 4.1. *Hazan et al. [2007] Assume that for all t, the loss function $\mathcal{L}_t : \mathcal{P} \in \mathbb{R}^M \to \mathbb{R}$ satisfies the following properties for all $\alpha \in \mathcal{P}$: **(i)** The gradient is bounded: $\|\nabla \mathcal{L}_t(\alpha)\| \leq G_{\mathcal{L}}$. **(ii)** The function is γ-exp-concave, meaning that $\exp(-\gamma \mathcal{L}_t((\alpha)))$ is concave in α. Then,* FOLLOW THE APPROXIMATE LEADER *(FTAL) Algorithm with step size parameter $\beta = \frac{1}{2}\min\left\{\frac{1}{4G_{\mathcal{L}}D}, \gamma\right\}$*

satisfies the following regret bound: $\text{Regret}_T(\text{FTAL}) \leq 64\left(\frac{1}{\gamma} + G_{\mathcal{L}} D\right) M(\log(T) + 1)$ *where* n *is the dimension of the feasible set and* D *is the diameter of* $\mathcal{P}$.

While the standard FTAL bound assumes access to the true loss functions, our setting relies on *estimated* bias. The following theorem (see proof in App. D.3) formally bridges this gap, establishing convergence even when minimizing the estimated auxiliary loss.

Theorem 4.2 (Convergence with Estimated Bias)**.** *Let Assumptions 2.1, 2.2 and let there be unchanging bias* $b_t = b$ *for all* t *hold and the gradient of* f *be bounded by* g_{max}*. Additionally, let the bias estimates* $\{\widehat{b}^{(i)}\}_{i=0}^{M}$ *be obtained using Algorithm 3 with decay schedule* $\beta_t = \frac{1}{t}$*. If the algorithm minimizes the estimated auxiliary loss* $\widehat{\mathcal{L}}_t$ *with curvature constant* γ*, the convergence bound as,*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}_t(\alpha_t)\right] \leq 4 \sum_{t=1}^{T} \mathcal{L}_t(\alpha^*) + 2R_T + 6 \sum_{t=1}^{T} \mathbb{E}[\|\xi_t\|^2], \tag{4}$$

where the regret term is $R_T = 64\left(\frac{1}{\gamma} + \sqrt{2}\,(4 + 2L\eta)\, g_{max}^2\right) M(1 + \log T)$.

Discussion. Theorem 4.2 confirms the theoretical validity of minimizing the estimated auxiliary loss by cleanly decoupling the oracle performance from the tracking mechanics. The first term, $4\sum \mathcal{L}_t(\alpha^*)$, demonstrates that our algorithm remains highly competitive with the optimal fixed aggregation up to a constant factor. The final term explicitly isolates the aggregate penalty caused by relying on the estimated bias (ξ_t). Crucially, by applying Lemma B.1 to rigorously bound this cumulative tracking error, we prove that $\sum \mathbb{E}[\|\xi_t\|^2]$ grows only logarithmically over time. Because both the FTAL regret R_T and the bounded estimation noise scale as $\mathcal{O}(\log T)$, their combined contribution to the average excess gradient decays rapidly at a rate of $\mathcal{O}(\frac{\log T}{T})$. This guarantees that the bias estimation mechanism imposes a vanishingly small overhead on the network's overall convergence.

Relaxation of the Stationary Bias Assumption It is important to note that while our primary theoretical analysis assumes a stationary true bias for clarity of exposition, this assumption can be easily relaxed to the more standard assumption of bounded bias. Instead of assuming the bias remains constant, we can assume that the bias for each peer i is upper-bounded by a maximum magnitude (representing the maximum cluster separation or distribution drift parameter). Let us denote this upper bound as $B^{(i)}$, such that $\|b_t^{(i)}\| \leq B^{(i)}$ for all t. By applying the triangle inequality, our per-step cost function $\mathcal{L}_t(\alpha)$ can be rigorously upper-bounded: $\mathcal{L}_t(\alpha) := \left\|\sum_{i\in[M]} \alpha^{(i)} b_t^{(i)}\right\|^2 + \eta L \left\|\sum_{i\in[M]} \alpha^{(i)} g_t^{(i)}\right\|^2 \leq \left(\sum_{i\in[M]} \alpha^{(i)} B^{(i)}\right)^2 + \eta L \left\|\sum_{i\in[M]} \alpha^{(i)} g_t^{(i)}\right\|^2$ By substituting the dynamic bias with this worst-case upper bound, we effectively replace the strict stationary bias constraint with the classic bounded-bias assumption. Since our empirical EMA estimation will not exceed this bound, this demonstrates how the constraint can be safely relaxed in practice, even though we maintain it in our primary analysis for theoretical clarity.

4.4 Final Algorithm

Algorithm 1 outlines the execution flow of the proposed SP-CACW framework. In each communication round t, the target client receives gradients from the federation and updates its estimates of peer bias $\hat{b}^{(i)}$ via a running average. These estimates are used to construct a weighted aggregate gradient g_t, which drives the standard SGD update for the model parameters w_t. The core innovation occurs at the end of each iteration: the collaboration weights α_t are re-optimized using a Follow-The-Approximate-Leader (FTAL) strategy on the cumulative proxy loss. As established in Theorem 4.2, this strategy guarantees that the algorithm achieves a solution close to the optimal static weighting (up to logarithmic regret and a factor of 4), effectively filtering out harmful peers while progressively amplifying those that demonstrably aid convergence. (See Appendix F.2 for an analysis of computational complexity.)

Algorithm 1 SP-CACW

Require: Initial model w_0, learning rate η, tracking parameter β, number of clients M, number of rounds T

Initialize weights $\alpha_0^{(i)} = 0$ for all $i \neq 0$, and $\alpha_0^{(0)} = 1$.
Initialize bias estimates $\widehat{b}_0^{(i)} = 0$ for all i.
for $t = 1$ to T **do**
  Broadcast model w_{t-1} to all participating clients.
  Receive stochastic gradients $g_t^{(i)}$ from all participating clients.
  Update local bias estimates using the exponential moving average:

$$\widehat{b}_t^{(i)} \leftarrow (1-\beta)\widehat{b}_{t-1}^{(i)} + \beta(g_t^{(i)} - g_t^{(0)})$$

  Compute the aggregated gradient:

$$g_t = \sum_{i=0}^{M-1} \alpha_{t-1}^{(i)} g_t^{(i)}$$

  Update the global model: $w_t \leftarrow w_{t-1} - \eta g_t$.
  Calculate new collaboration weights via Follow The Approximate Leader (FTAL):

$$\alpha_t \leftarrow \text{FTAL}(\widehat{\mathcal{L}}_1, \ldots, \widehat{\mathcal{L}}_t)$$

end for
Return optimized model w_T.

5 Experiments

In this section, we evaluate the empirical efficacy of our proposed framework across three diverse datasets (MNIST, CIFAR-100, and the Shakespeare dataset from the LEAF benchmark), structured into 6 heterogeneity settings. Detailed specifications regarding our final algorithmic implementation, including complete network architectures and Hyperparameters, are provided in Appendices F and G.

In our experiments, we evaluate the standard SP-CACW algorithm alongside two variants (detailed in Appendix F). To stabilize collaboration weights, the regularized variant applies the following penalty:

$$\mathcal{L}_{\text{reg}}(\alpha) = \lambda_{\text{reg}} \sum_{i=0}^{M-1} (\alpha^{(i)})^3 + \mathcal{L}_{\text{old}}(\alpha).$$

The second variant performs PCA on the gradients.

Baselines. We compare our proposed method (both the standard and regularized variants) against several established baselines in Personalized Federated Learning: **FedAMP** Huang et al. [2021], **FC** (Federated Clustering) Werner et al. [2023], **Ditto** Li et al. [2021],**FedRep** Collins et al. [2021a], and **FedDisco** Ye et al. [2023]. Furthermore, we benchmark against a **Local-Only** model trained exclusively on the target client's (Client 0) local data. Finally, we report the performance of an **Oracle Bound** (an idealized training scenario, such as training exclusively on the aggregated data of a client's true latent cluster not always optimal), which serves as a rigorous empirical upper bound for achievable accuracy.

Experimental Protocol. To ensure statistical robustness, we repeat each experiment across five distinct random seeds. The results presented in the accompanying figures and tables report the mean accuracy across these independent trials, with error bars denoting one standard deviation to account for variance. Further details regarding the variance analysis can be found in Section H.2 and compute resources of the experiment in G.4 while the algorithm used for data partitioning can be found in H.

5.1 Results and Analysis

To provide a comprehensive overview of our framework's adaptability across varying types of data heterogeneity, we summarize our performance results in Tables 1 and 2. Notably, the top three

performing approaches for each setting are highlighted in blue and bold. As shown, our SP-CACW variants consistently secure a top-three position across all evaluated scenarios:

- **MNIST Label-Switching (Concept Shift):** We simulate 7 clients across 3 latent clusters with permuted class labels. **SP-CACW** demonstrates rapid cluster identification, converging to up to 94.98% accuracy. This outperforms baselines by $\sim$2.8%—a significant margin on a saturated benchmark—and effectively matches the Oracle Bound (95.14%).
- **MNIST Rotation (Feature Shift):** We simulate a continuous manifold of 89 clients with sequential 4° angular rotations. **SP-CACW** achieves up to 89.52% accuracy, demonstrating that our method remains highly competitive even when heterogeneity is continuous rather than distinctly clustered.
- **CIFAR-100 (Cluster & Noise Robustness):** We partition 20 clients into 4 distinct clusters (assigned 25 disjoint classes each) and evaluate under varying intra-cluster noise levels ($\sigma \in \{0.01, 10\}$). Across all noise profiles, SP-CACW and its regularized variant establish a strong convergence trajectory (reaching up to 50.17% accuracy). SP-CACW is consistently competitive and outperforms FedAMP/FC/Ditto, while FedDisco is strongest (reaching 53.89%) in the high-noise CIFAR setting.
- **CIFAR-100 Label-Switching (Contradictory Labels):** To explicitly test robustness against contradictory data, we simulate a concept shift on a 10-class subset of CIFAR-100 partitioned across 21 clients in 3 clusters. We systematically shift the labels for each cluster by a fixed constant, creating severely contradictory local environments while evaluating on an untouched test set. Under these conditions, baselines relying on rigid label assumptions, such as FedDisco, suffer catastrophic collapse (plummeting to 2.00% accuracy). In stark contrast, SP-CACW's dynamic bias-thresholding isolates and down-weights these unreliable peers, achieving a leading 45.28% accuracy and demonstrating definitive practical superiority in adversarial regimes.
- **LEAF Shakespeare (Sequence Heterogeneity):** We evaluate next-character prediction across $n = 15$ and $n = 30$ clients. Rather than relying on synthetic latent clusters, this natural partition constructs each client's local dataset from the combined dialogue of $m = 7$ and $m = 3$ unique Shakespearean characters (speaking roles). Because every role possesses a distinct vocabulary, cadence, and sequence length, this creates a severely non-IID, cluster-free environment. SP-CACW successfully adapts to these complex, highly individualized linguistic patterns (achieving up to 54.65% accuracy) without assuming any predefined group structure.
- **LEAF Shakespeare Large-Scale & Partial Participation:** We expand the Shakespeare environment to $n = 200$ clients to evaluate scalability and communication efficiency. Under full participation, SP-CACW variants achieve a strong overall accuracy (up to 54.06%). Crucially, when restricted to a 10% partial participation regime, the regularized SP-CACW exhibits remarkable resilience, successfully maintaining a highly competitive 51.92% accuracy. This validates that our framework scales efficiently to larger federations without requiring full communication bandwidth, and continues to outperform baselines like FedRep (50.18%) under restricted participation.

Full graphs detailing the optimization trajectories for all experiments can be found in Appendix I.

5.2 Analysis of Counter-Intuitive Empirical Trends

Our evaluation revealed notable empirical anomalies driven by data distribution, architecture, and peer reliability. First, we observed a stark dichotomy in **FedDisco's** performance: it excels when peer data does not "lie" (e.g., MNIST rotation) by anchoring to a shared model, but suffers catastrophic collapse in adversarial settings with contradictory data (e.g., CIFAR-100 label switching). its cased by contradictory labels due to the algometer static global assumptions (no bad date). In contrast, SP-CACW dynamically isolates and down-weights unreliable peers, trading slight peak accuracy in idealized conditions for definitive adversarial robustness.

Second, an architectural constraint explains **FedRep's relative underperformance in the MNIST trials**. FedRep requires partitioning the network into a shared representation base and localized

Table 1: Max Average Performance Per Dataset (Part 1: MNIST and CIFAR100). The top 3 results per column are highlighted. Error bars denote 1 standard deviation across random seeds.

	MNIST		CIFAR-100		
	relabel	rotate	sigma=0.01	sigma=10	sigma=10 relabel
SP-CACW	94.87 ± 0.64	88.85 ± 2.34	43.48 ± 0.86	39.28 ± 4.34	45.28 ± 4.77
SP-CACW reg no pca	94.98 ± 0.41	89.52 ± 0.95	44.79 ± 2.24	50.17 ± 1.17	40.34 ± 8.72
SP-CACW pca no reg	94.54 ± 0.35	88.49 ± 2.87	45.29 ± 1.20	39.06 ± 4.55	45.26 ± 4.96
FedDisco	35.14 ± 4.76	55.08 ± 4.97	52.52 ± 1.21	53.89 ± 0.62	2.00 ± 4.00
Ditto	90.41 ± 1.52	81.83 ± 0.96	18.04 ± 2.42	20.88 ± 1.25	13.22 ± 3.05
FedAmp	90.37 ± 0.97	80.45 ± 2.08	8.80 ± 0.62	14.37 ± 0.70	18.30 ± 2.27
Fed Cluster	92.15 ± 0.86	84.33 ± 1.51	16.76 ± 5.33	24.44 ± 0.99	11.38 ± 1.87
FedRep	88.32 ± 2.11	78.05 ± 1.54	22.16 ± 1.13	27.67 ± 0.98	27.04 ± 2.90
Local-Only	92.12 ± 0.34	84.74 ± 0.82	17.30 ± 0.57	25.85 ± 0.73	16.86 ± 3.52
ORACLE	95.14 ± 0.36	-	38.12 ± 1.33	33.15 ± 1.52	40.32 ± 3.88

Table 2: Max Average Performance Per Dataset (Part 2: Shakespeare). The top 3 results per column are highlighted. Error bars denote 1 standard deviation across random seeds.

	Shakespeare		
	(n=15 m=7)	(n=30 m=3)	(n=200 m=5)
SP-CACW	54.20 ± 0.14	47.77 ± 0.32	54.06 ± 0.14
SP-CACW reg no pca	54.65 ± 0.20	47.36 ± 0.14	53.63 ± 0.37
SP-CACW pca no reg	54.21 ± 0.16	47.69 ± 0.54	54.01 ± 0.26
FedDisco	53.71 ± 0.18	44.96 ± 0.32	49.36 ± 0.29
Ditto	51.28 ± 0.15	46.12 ± 0.23	48.55 ± 0.43
FedAmp	47.52 ± 0.29	45.73 ± 0.45	47.41 ± 0.12
Fed Cluster	53.72 ± 0.24	46.76 ± 0.33	52.35 ± 0.17
FedRep	52.01 ± 0.26	48.99 ± 0.40	50.42 ± 0.21
Local-Only	48.17 ± 0.19	42.25 ± 0.36	45.33 ± 0.45
FedRep 10% partial participation	-	-	50.18 ± 0.23
SP-CACW 10% partial participation	-	-	51.92 ± 0.00

classifier. Because our MNIST experiments use a compact, shallow MLP, splitting it leaves almost no meaningful representation to share, leading to sub-optimal dynamics compared to full-model sharing.

Finally, notable scale and participation dynamics emerged. SP-CACW demonstrated remarkable stability on LEAF Shakespeare when scaling from 15 to 200 clients, maintaining $\sim$54% accuracy. Larger cohorts yield finer local distributions, enabling SP-CACW to precisely isolate correlated peers. Furthermore, regularized SP-CACW and FedRep exhibited extreme resilience under 10% partial participation. Notably, FedRep performs even better under 10% partial participation than under full participation. Because the target client remains continuously active in our SFL setup, restricting peer participation allows the target to disproportionately dictate the updates to FedRep's shared global representation. By constantly swaying this global component of the model in its own favor, the target client prevents its localized features from being diluted by the broader network's heterogeneity.

6 Conclusion

In this work, we introduced SP-CACW, a Selfish Federated Learning framework that leverages online bound-minimization to optimize the bias-variance trade-off in heterogeneous networks. By replacing heuristic metrics with principled collaborator selection, and integrating adaptive cubic regularization with bias estimation, SP-CACW effectively prevents negative transfer. Empirical evaluations demonstrate that it is consistently competitive and often outperforms strong personalized and clustering baselines across MNIST, LEAF Shakespeare, and CIFAR-100.

Acknowledgments

This work was partially supported by Israel PBC-VATAT, by the Technion Artificial Intelligence Hub (Tech.AI), and by the Israel Science Foundation (grant No. 3109/24). The source code for all experiments is publicly available at: https://github.com/nheh3833/cacw-code.

References

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021a.

Lucas Collins, Hamed Hassani, Aryan Mokhtari, and Srinivas Shakkottai. Exploiting shared representations for Personalized Federated Learning (FedRep). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pages 2089–2099. PMLR, 2021b.

Cuong T. Dinh, Ngoc H. Tran, and Trong D. Nguyen. Personalized Federated Learning with Moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21394–21405. Curran Associates, Inc., 2020.

Alireza Fallah, Aryan Mokhtari, and Asu Ozdaglar. Personalized Federated Learning with theoretical guarantees: A Model-Agnostic Meta-Learning approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning (ifca). *IEEE Transactions on Information Theory*, 68(12):8076–8091, 2022. doi: 10.1109/TIT.2022.3192506.

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

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Kun Hu, Sheng Gong, Qiang Zhang, Chao Seng, Meng Xia, and Shu Jiang. An overview of implementing security and privacy in federated learning. *Artificial Intelligence Review*, 57:204, 2024. doi: 10.1007/s10462-024-10846-8.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7865–7873. AAAI Press, 2021. doi: 10.1609/aaai.v35i14.16960.

Ying-Hsun Lai, Shin-Yeh Chen, Wen-Chi Chou, Hua-Yang Hsu, and Han-Chieh Chao. Personalized federated learning with adaptive feature extraction and category prediction in non-iid datasets. *Future Internet*, 16(3):95, 2024. doi: 10.3390/fi16030095.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

I. A. Majeed, S. Kaushik, A. Bardhan, V. S. K. Tadi, H.-K. Min, K. Kumaraguru, and R. D. Muni. Comparative assessment of Federated and Centralized Machine Learning, 2022.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 15001–15014. Curran Associates, Inc., 2020.

Amir M. Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. A comprehensive survey on knowledge distillation, 2025.

Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994. doi: 10.1162/NECO.1994.6.1.147.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Jiahao Tan, Yipeng Zhou, Gang Liu, Jessie Hui Wang, and Shui Yu. pfedsim: Similarity-aware model aggregation towards personalized federated learning, 2023.

Mariel A. Werner, Lie He, Michael I. Jordan, Martin Jaggi, and Sai Praneeth Karimireddy. Provably personalized and robust federated learning. *Transactions on Machine Learning Research*, 2023.

Junyuan Xu, Benjamin S. Glicksberg, Chuxu Su, Patrick Walker, Jiang Bian, and Fei Wang. Federated Learning for Healthcare Informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021. doi: 10.1007/s41666-020-00082-4.

Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. FedDisco: Federated learning with discrepancy-aware collaboration. In *Proceedings of the 40th International Conference on Machine Learning*, pages 39879–39902. PMLR, 2023.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging Federated Learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020. doi: 10.48550/arXiv.2002.04758.

Yu Zhang and Qiang Yang. A survey on Multi-Task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022.

Yue Zhao, Meng Li, Lixin Lai, Nishanth Suda, Damon Civin, and Vaibhav Chandra. Federated Learning with Non-IID data. *arXiv preprint arXiv:1806.00582*, 2018.

A Proofs and Details for Section 3

A.1 Proof of Theorem 3.1

Proof. We start by invoking the L-smoothness of $f^{(0)}$ (Assumption 2.1). For the update step $w_{t+1} = w_t - \eta g_t$, the smoothness inequality implies:

$$\begin{aligned} f^{(0)}(w_{t+1}) &\leq f^{(0)}(w_t) + \langle \nabla f^{(0)}(w_t), w_{t+1} - w_t \rangle + \frac{L}{2}\|w_{t+1} - w_t\|^2 \\ &= f^{(0)}(w_t) - \eta \langle \nabla f^{(0)}(w_t), g_t \rangle + \frac{L\eta^2}{2}\|g_t\|^2. \end{aligned}$$

Let $\mathcal{F}_t$ denote the filtration representing the history up to round t. We assume the aggregation weights α_t are $\mathcal{F}_t$-measurable (i.e., computed before the stochastic gradients at round t are sampled). We define the expected aggregated gradient and the true population bias as:

$$\begin{aligned} \mathbb{E}_t[g_t] &= \sum_{i=0}^{M-1} \alpha_t^{(i)} \nabla f^{(i)}(w_t) \\ \bar{b}_t(\alpha_t) &:= \sum_{i=0}^{M-1} \alpha_t^{(i)} \left(\nabla f^{(i)}(w_t) - \nabla f^{(0)}(w_t) \right) \end{aligned}$$

This allows us to cleanly express the expected gradient as $\mathbb{E}_t[g_t] = \nabla f^{(0)}(w_t) + \bar{b}_t(\alpha_t)$.

Taking the conditional expectation $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_t]$ of the smoothness inequality eliminates the stochastic noise from the inner product:

$$\begin{aligned} \mathbb{E}_t[f^{(0)}(w_{t+1})] &\leq f^{(0)}(w_t) - \eta \langle \nabla f^{(0)}(w_t), \mathbb{E}_t[g_t] \rangle + \frac{L\eta^2}{2}\mathbb{E}_t\|g_t\|^2 \\ &= f^{(0)}(w_t) - \eta \langle \nabla f^{(0)}(w_t), \nabla f^{(0)}(w_t) + \bar{b}_t(\alpha_t) \rangle + \frac{L\eta^2}{2}\mathbb{E}_t\|g_t\|^2 \\ &= f^{(0)}(w_t) - \eta\|\nabla f^{(0)}(w_t)\|^2 - \eta \langle \nabla f^{(0)}(w_t), \bar{b}_t(\alpha_t) \rangle + \frac{L\eta^2}{2}\mathbb{E}_t\|g_t\|^2. \end{aligned}$$

Using Young's inequality, specifically $-\langle x, y \rangle \leq \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2$, we bound the noise-free cross-term:

$$-\eta \langle \nabla f^{(0)}(w_t), \bar{b}_t(\alpha_t) \rangle \leq \frac{\eta}{2}\|\nabla f^{(0)}(w_t)\|^2 + \frac{\eta}{2}\|\bar{b}_t(\alpha_t)\|^2.$$

Plugging this bound back yields:

$$\begin{aligned} \mathbb{E}_t[f^{(0)}(w_{t+1})] &\leq f^{(0)}(w_t) - \eta\|\nabla f^{(0)}(w_t)\|^2 + \frac{\eta}{2}\|\nabla f^{(0)}(w_t)\|^2 + \frac{\eta}{2}\|\bar{b}_t(\alpha_t)\|^2 + \frac{L\eta^2}{2}\mathbb{E}_t\|g_t\|^2 \\ &= f^{(0)}(w_t) - \frac{\eta}{2}\|\nabla f^{(0)}(w_t)\|^2 + \frac{\eta}{2}\underbrace{\left(\|\bar{b}_t(\alpha_t)\|^2 + L\eta\mathbb{E}_t\|g_t\|^2\right)}_{\mathcal{L}_t(\alpha_t)}. \end{aligned}$$

Rearranging terms to isolate the squared target gradient norm:

$$\frac{\eta}{2}\|\nabla f^{(0)}(w_t)\|^2 \leq f^{(0)}(w_t) - \mathbb{E}_t[f^{(0)}(w_{t+1})] + \frac{\eta}{2}\mathcal{L}_t(\alpha_t).$$

Taking the total expectation and summing over $t = 1, \ldots, T$:

$$\frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \leq \sum_{t=1}^{T}\left(\mathbb{E}[f^{(0)}(w_t)] - \mathbb{E}[f^{(0)}(w_{t+1})]\right) + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}[\mathcal{L}_t(\alpha_t)].$$

The first sum on the RHS is a telescoping sum that collapses to $\mathbb{E}[f^{(0)}(w_1)] - \mathbb{E}[f^{(0)}(w_{T+1})]$. Since w^* is a global minimizer, $f^{(0)}(w_{T+1}) \geq f^{(0)}(w^*)$, and thus:

$$\sum_{t=1}^{T}\left(\mathbb{E}[f^{(0)}(w_t)] - \mathbb{E}[f^{(0)}(w_{t+1})]\right) \leq f^{(0)}(w_1) - f^{(0)}(w^*) = \delta_1.$$

Substituting this initial distance gap bound:

$$\frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \leq \delta_1 + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}[\mathcal{L}_t(\alpha_t)].$$

Finally, multiplying by $\frac{2}{\eta T}$ to obtain the average gradient norm establishes the result:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \leq \frac{2\delta_1}{\eta T} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\mathcal{L}_t(\alpha_t)].$$

This completes the proof. □

A.1.1 Proof of lemma A.1

Lemma A.1. *Let $x, y \geq 0$. If $x \leq y + 4\eta L x$ with $\eta L \leq \frac{1}{8}$, then $x \leq 2y$.*

Proof. Rearranging the inequality yields $x(1-\eta L) \leq y$. Since $\eta L \leq \frac{1}{2}$, we have $1-\eta L \geq \frac{1}{2}$. Dividing by this positive factor gives:

$$x \leq \frac{y}{1-4\eta L} \leq \frac{y}{1/2} = 2y.$$

□

A.1.2 Proof of lemma A.2

Lemma A.2 (Second Moment Bound)**.** *Let $g_t^{(i)}$ be the stochastic gradient for client $i \in [M]$ such that $g_t^{(i)} = \nabla f^{(0)}(w_t) + b_t^{(i)} + \xi_t^{(i)}$, where the noise $\xi_t^{(i)}$ has zero mean and variance bounded by $(\sigma^{(i)})^2$.*

For any aggregation weights $\alpha \in \Delta_M$, the conditional expected squared norm of the aggregated gradient is bounded by:

$$\mathbb{E}_t\left[\left\|\sum_{i\in[M]}\alpha^{(i)}g_t^{(i)}\right\|^2\right] \leq 2\|\nabla f^{(0)}(w_t)\|^2 + 2\left\|\sum_{i\in[M]}\alpha^{(i)}b_t^{(i)}\right\|^2 + \sum_{i\in[M]}(\alpha^{(i)})^2(\sigma^{(i)})^2$$

Proof. Let $G_t(\alpha) = \sum_{i\in[M]}\alpha^{(i)}g_t^{(i)}$. Using the standard bias-variance decomposition, we have $\mathbb{E}_t[\|G_t(\alpha)\|^2] = \|\mathbb{E}_t[G_t(\alpha)]\|^2 + \mathrm{Var}(G_t(\alpha))$.

1. Variance Term:
Assuming independence of the stochastic noise across clients, the variance of the weighted sum is the sum of the individual variances. Applying the bounded noise assumption:

$$\mathrm{Var}(G_t(\alpha)) = \sum_{i\in[M]}(\alpha^{(i)})^2\mathbb{E}_t\|\xi_t^{(i)}\|^2 \leq \sum_{i\in[M]}(\alpha^{(i)})^2(\sigma^{(i)})^2$$

2. Expectation Term:
Because the noise is zero-mean ($\mathbb{E}_t[\xi_t^{(i)}] = 0$) and the weights reside on the probability simplex ($\sum_i \alpha^{(i)} = 1$), the target gradient factors out cleanly:

$$\mathbb{E}_t[G_t(\alpha)] = \sum_{i\in[M]}\alpha^{(i)}\left(\nabla f^{(0)}(w_t) + b_t^{(i)}\right) = \nabla f^{(0)}(w_t) + \sum_{i\in[M]}\alpha^{(i)}b_t^{(i)}$$

Applying the elementary inequality $\|x+y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$, we bound the squared norm of this expectation:

$$\|\mathbb{E}_t[G_t(\alpha)]\|^2 \leq 2\|\nabla f^{(0)}(w_t)\|^2 + 2\left\|\sum_{i\in[M]}\alpha^{(i)}b_t^{(i)}\right\|^2$$

Summing the bounded variance and expectation terms completes the proof. □

A.2 Proof of Lemma 3.1

Proof. We minimize the objective function subject to simplex constraints:

$$\min_{\alpha} \quad C_1 \sum_{i=0}^{M-1} \alpha^{(i)} \|b^{(i)}\|^2 + C_2 \sum_{i=0}^{M-1} (\alpha^{(i)})^2 (\sigma^{(i)})^2,$$

subject to $\alpha^{(i)} \geq 0$ for all i, and $\sum_{i=0}^{M-1} \alpha^{(i)} = 1$.

The Lagrangian for this problem is:

$$\mathcal{L}(\alpha, \lambda, \mu) = C_1 \sum_{i=0}^{M-1} \alpha^{(i)} \|b^{(i)}\|^2 + C_2 \sum_{i=0}^{M-1} (\alpha^{(i)})^2 (\sigma^{(i)})^2 - \lambda \left(\sum_{i=0}^{M-1} \alpha^{(i)} - 1 \right) - \sum_{i=0}^{M-1} \mu_i \alpha^{(i)},$$

where λ is the multiplier for the equality constraint and $\mu_i \geq 0$ correspond to the inequality constraints. The KKT stationarity condition with respect to $\alpha^{(i)}$ is:

$$\frac{\partial \mathcal{L}}{\partial \alpha^{(i)}} = C_1 \|b^{(i)}\|^2 + 2C_2 (\sigma^{(i)})^2 \alpha^{(i)} - \lambda - \mu_i = 0.$$

Using the complementary slackness condition $\mu_i \alpha^{(i)} = 0$, we analyze two cases:

- **Case 1 ($\alpha^{(i)} > 0$):** Here $\mu_i = 0$. The stationarity condition yields:

  $$2C_2 (\sigma^{(i)})^2 \alpha^{(i)} = \lambda - C_1 \|b^{(i)}\|^2 \quad \Longrightarrow \quad \alpha^{(i)} = \frac{\lambda - C_1 \|b^{(i)}\|^2}{2C_2 (\sigma^{(i)})^2}.$$

  For this to be consistent with $\alpha^{(i)} > 0$, we must have $\lambda > C_1 \|b^{(i)}\|^2$.

- **Case 2 ($\alpha^{(i)} = 0$):** Here $\mu_i \geq 0$. The stationarity condition implies $\lambda - C_1 \|b^{(i)}\|^2 = -\mu_i \leq 0$, so $\lambda \leq C_1 \|b^{(i)}\|^2$.

Combining these cases, we can express the optimal weights compactly using the operator $[z]_+ = \max\{0, z\}$:

$$\alpha^{(i)*} = \left[\frac{\lambda - C_1 \|b^{(i)}\|^2}{2C_2 (\sigma^{(i)})^2} \right]_+.$$

Finally, the value of λ is determined by substituting these weights into the equality constraint:

$$\sum_{i=0}^{M-1} \left[\frac{\lambda - C_1 \|b^{(i)}\|^2}{2C_2 (\sigma^{(i)})^2} \right]_+ = 1.$$

Since the Left-Hand Side is a continuous, strictly increasing piecewise linear function of λ (for λ large enough to activate at least one weight), there exists a unique λ satisfying this equation. □

B Proof of lemma 4.1 under Stationary Bias

B.1 Proof of lemma B.1 (Stationary Case)

Before proving the estimator's accuracy, we explicitly define the bounded variance of the bias observation. The observation $\tilde{b}_t^{(i)} = g_t^{(i)} - g_t^{(0)}$ contains independent zero-mean stochastic gradient noise from both the local client i and the target client 0. Consequently, the variance is bounded by:

$$\mathbb{E}_t\|\tilde{b}_t^{(i)} - b_t^{(i)}\|^2 \leq (\sigma^{(i)})^2 + (\sigma^{(0)})^2 \leq \sigma_{\text{bound}}^2$$

where we define the global worst-case noise threshold as $\sigma_{\text{bound}}^2 := \max_{i\neq 0}\left((\sigma^{(i)})^2 + (\sigma^{(0)})^2\right)$.

Lemma B.1 (Bias Estimation Error under Stationary Bias)**.** *Let Assumption 2.2 hold, and assume the true population bias is strictly stationary across time steps ($b_k^{(i)} = b_t^{(i)} = b^{(i)}$). Suppose the bias estimates for client i are updated via the recurrence:*

$$\widehat{b}_t^{(i)} = (1-\beta_t)\widehat{b}_{t-1}^{(i)} + \beta_t\tilde{b}_t^{(i)},$$

where $\beta_t = 1/t$. The expected squared estimation error is tightly bounded by:

$$\mathbb{E}\left[\|\widehat{b}_t^{(i)} - b^{(i)}\|^2\right] \leq \frac{\sigma_{\text{bound}}^2}{t}.$$

Proof. For the schedule $\beta_t = 1/t$, the update rule $\widehat{b}_t^{(i)} = \frac{t-1}{t}\widehat{b}_{t-1}^{(i)} + \frac{1}{t}\tilde{b}_t^{(i)}$ is mathematically equivalent to computing the simple cumulative average of all observed gradients up to time t:

$$\widehat{b}_t^{(i)} = \frac{1}{t}\sum_{k=1}^{t}\tilde{b}_k^{(i)}.$$

Because the population bias is stationary, the expected value of the observation is exactly the true bias ($\mathbb{E}[\tilde{b}_k^{(i)}] = b^{(i)}$). Thus, the estimation error is purely driven by zero-mean stochastic noise:

$$\widehat{b}_t^{(i)} - b^{(i)} = \frac{1}{t}\sum_{k=1}^{t}(\tilde{b}_k^{(i)} - b^{(i)}).$$

Taking the expected squared norm, and utilizing the fact that the stochastic noise at each step k is independent, the cross-terms vanish. The expectation of the squared sum becomes the sum of the variances:

$$\mathbb{E}\left[\left\|\frac{1}{t}\sum_{k=1}^{t}(\tilde{b}_k^{(i)} - b^{(i)})\right\|^2\right] = \frac{1}{t^2}\sum_{k=1}^{t}\mathbb{E}[\|\tilde{b}_k^{(i)} - b^{(i)}\|^2].$$

Applying the baseline variance bound yields the exact convergence rate of the estimator:

$$\frac{1}{t^2}\sum_{k=1}^{t}\mathbb{E}[\|\tilde{b}_k^{(i)} - b^{(i)}\|^2] \leq \frac{1}{t^2}(t\sigma_{\text{bound}}^2) = \frac{\sigma_{\text{bound}}^2}{t}.$$

This concludes the proof. □

B.2 Proof of the Cumulative Tracking Penalty

Let $B \in \mathbb{R}^{d\times M}$ denote the true stationary population bias matrix, and let $\widehat{B}_t \in \mathbb{R}^{d\times M}$ be its empirical estimate at time step t. Let $\alpha_t \in \Delta_M$ represent the active weight vector residing on the M-dimensional probability simplex. Assuming the column-wise estimation error, defined as $\xi_t^{(i)} = b^{(i)} - \widehat{b}_t^{(i)}$, satisfies $\mathbb{E}\|\xi_t^{(i)}\|^2 \leq \frac{\sigma_{\text{bound}}^2}{t}$ for all $i \in \{0,\ldots,M-1\}$, the expected cumulative true bias loss over T steps is bounded by

Proof. To ensure rigorous dimensionality, we define the true stationary population bias matrix and its empirical estimate at step t as:

$$B := \begin{bmatrix} b^{(0)} & b^{(1)} & \cdots & b^{(M-1)}\end{bmatrix} \in \mathbb{R}^{d\times M}, \quad \widehat{B}_t := \begin{bmatrix}\widehat{b}_t^{(0)} & \widehat{b}_t^{(1)} & \cdots & \widehat{b}_t^{(M-1)}\end{bmatrix} \in \mathbb{R}^{d\times M}$$

We decompose the true bias matrix as $B = \widehat{B}_t + \Xi_t$, where $\Xi_t = \begin{bmatrix} \xi_t^{(0)} & \cdots & \xi_t^{(M-1)} \end{bmatrix}$ represents the column-wise estimation error. Our goal is to bound the cumulative true bias loss for the active weights $\alpha_t \in \Delta_M$:

$$L = \sum_{t=1}^{T} \|B\alpha_t\|^2 = \sum_{t=1}^{T} \|(\widehat{B}_t + \Xi_t)\alpha_t\|^2.$$

Using the elementary inequality $\|x+y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$, we bound the aggregated term at each step t:

$$\|B\alpha_t\|^2 \leq 2\|\widehat{B}_t\alpha_t\|^2 + 2\|\Xi_t\alpha_t\|^2. \tag{5}$$

Taking the expectation and summing over $t = 1$ to T:

$$\mathbb{E}\left[\sum_{t=1}^{T} \|B\alpha_t\|^2\right] \leq 2\mathbb{E}\left[\sum_{t=1}^{T} \|\widehat{B}_t\alpha_t\|^2\right] + 2\sum_{t=1}^{T} \mathbb{E}\left[\|\Xi_t\alpha_t\|^2\right]. \tag{6}$$

We now focus on bounding the tracking penalty $\mathcal{E}_T = 2\sum_{t=1}^{T} \mathbb{E}[\|\Xi_t\alpha_t\|^2]$. Because the weight vector α_t resides strictly on the probability simplex ($\sum_i \alpha_t^{(i)} = 1, \alpha_t^{(i)} \geq 0$) and the squared L_2-norm is convex, we apply Jensen's inequality:

$$\|\Xi_t\alpha_t\|^2 = \left\|\sum_{i=0}^{M-1} \alpha_t^{(i)}\xi_t^{(i)}\right\|^2 \leq \sum_{i=0}^{M-1} \alpha_t^{(i)}\|\xi_t^{(i)}\|^2.$$

Taking the expectation, we can upper-bound this sum by the maximum expected client error:

$$\mathbb{E}\|\Xi_t\alpha_t\|^2 \leq \sum_{i=0}^{M-1} \alpha_t^{(i)}\mathbb{E}\|\xi_t^{(i)}\|^2 \leq \max_{i\in[M]} \mathbb{E}\|\xi_t^{(i)}\|^2.$$

Substituting the exact stationary variance bound from Lemma B.1 ($\mathbb{E}\|\xi_t^{(i)}\|^2 \leq \frac{\sigma^2_{\text{bound}}}{t}$):

$$\mathcal{E}_T \leq 2\sum_{t=1}^{T} \frac{\sigma^2_{\text{bound}}}{t} = 2\sigma^2_{\text{bound}} \sum_{t=1}^{T} \frac{1}{t}. \tag{7}$$

Using the harmonic series bound $\sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log T$, we obtain the final tight penalty:

$$\mathcal{E}_T \leq 2\sigma^2_{\text{bound}}(1 + \log T). \tag{8}$$

Because the underlying population distribution is stationary, the total tracking overhead collapses entirely to the vanishing stochastic variance, yielding an optimal $\mathcal{O}(\frac{\log T}{T})$ estimation bound. □

C Derivation of Baseline Rates Regarding Cluster Example in Section 3.2

In this section, we derive the convergence rates for the baselines using the general bound provided in Theorem 3.1. Recall that the theorem bounds the convergence rate by:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \leq \frac{2}{T}\left(\frac{\delta_1}{\eta} + \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_t(\alpha)]\right) \tag{9}$$

where the cost function term effectively decomposes into variance and bias components: $\mathbb{E}[\mathcal{L}_t(\alpha)] \approx \eta\left(\sum_{i=0}^{M-1}(\alpha^{(i)})^2\sigma^2 + \text{Bias}(\alpha)\right)$.

Lemma C.1 (Autarky Convergence)**.** *In the Autarky setting (training only on the target client 0), the convergence rate is:*

$$err_T^{Autarky} = O\left(\frac{\sigma}{\sqrt{T}}\right).$$

Proof. In Autarky, the weights are set to isolate the target distribution. Specifically, $\alpha^{(0)} = 1$ and 0 otherwise. Substituting these weights into the cost term:

- **Variance:** $\sum(\alpha^{(i)})^2\sigma^2 = \sigma^2$.
- **Bias:** Since only target clients are used, the bias is zero: $\|\sum \alpha^{(i)}\nabla f^{(i)} - \nabla f^{(0)}\|^2 = 0$.

Thus, $\mathbb{E}[\mathcal{L}_t(\alpha)] = O(\eta\sigma^2)$. Substituting this into Theorem 3.1:

$$\text{err}_T \leq \frac{2\delta_1}{\eta T} + O(\eta\sigma^2).$$

Optimizing the step size $\eta \asymp \frac{1}{\sigma\sqrt{T}}$ yields the rate $O(\frac{\sigma}{\sqrt{T}})$. □

Lemma C.2 (Standard FL Convergence)**.** *In the Standard Federated Learning setting (uniform averaging), the convergence rate is:*

$$err_T^{FL} = O\left(\|b\|^2 + \frac{\sigma}{\sqrt{MT}}\right).$$

Proof. For Standard FL, the server assigns uniform weights $\alpha^{(i)} = \frac{1}{M}$ for all $i \in [M]$. Substituting these weights into the cost term:

- **Variance:** The effective variance reduces due to averaging over M clients:

$$\sum_{i=0}^{M-1}(\alpha^{(i)})^2\sigma^2 = \sum_{i=0}^{M-1}\frac{1}{M^2}\sigma^2 = \frac{M}{M^2}\sigma^2 = \frac{\sigma^2}{M}.$$

- **Bias:** The weights include off-target clients, introducing a non-zero bias. Based on the "Pure Clusters" scenario, the bias term is proportional to the distributions shift b:

$$\text{Bias}(\alpha) = \left\|\sum_{i=0}^{M-1}\frac{1}{M}\nabla f^{(i)}(w) - \nabla f^{(0)}(w)\right\|^2 = O(\|b\|^2).$$

Thus, the cost term is $\mathbb{E}[\mathcal{L}_t(\alpha)] = O(\eta[\frac{\sigma^2}{M} + b])$. Substituting this into Theorem 3.1:

$$\text{err}_T \leq \frac{2\delta_1}{\eta T} + O\left(\eta\frac{\sigma^2}{M}\right) + O(\eta b).$$

Optimizing $\eta \asymp \frac{1}{\sqrt{T}}$ balances the first two terms to $O(\frac{\sigma}{\sqrt{MT}})$, but the bias term $O(b)$ remains constant (or dominates depending on η), leading to the final rate $O(\|b\|^2 + \frac{\sigma}{\sqrt{MT}})$. □

C.1 Comparison with Federated-Clustering (FC)

In this section, we analyze the convergence rate of the Federated-Clustering (FC) algorithm Werner et al. [2023] in the **Example 1: Pure Clusters** setting described in Section 3.2.

Lemma C.3 (Federated-Clustering Convergence in Pure Clusters)**.** *Consider the "Pure Clusters" setting where the target cluster I_0 has size n_0. Assume the inter-cluster separation is Δ and there are no malicious clients ($\beta = 0$). The convergence rate of the Federated-Clustering (FC) algorithm is bounded by:*

$$err_T^{FC} := O\left(\sqrt{\frac{\sigma^2}{n_0 T} + \frac{\sigma^3}{\Delta T}}\right).$$

Proof. We rely on Theorem 3 from Werner et al. [2023], which provides the convergence guarantee for FC for general non-convex loss functions. The theorem states that:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f_i(x_{i,t-1})\|^2 \leq \sqrt{\frac{\max(1, A^2)(\sigma^2/n_i + \sigma^3/\Delta + \beta_i\sigma\Delta)}{T}}.$$

We adapt this bound to our specific scenario as follows:

1. **Target Cluster Size:** We focus on a client in the target cluster I_0, so we set $n_i = n_0$.
2. **No Malicious Clients:** We set $\beta_i = 0$, which eliminates the robustness term $\beta_i\sigma\Delta$.
3. **Simplification:** We treat constants A as $O(1)$.

Substituting these values into the general theorem yields:

$$\text{err}_T^{\text{FC}} \lesssim \sqrt{\frac{\sigma^2/n_0 + \sigma^3/\Delta}{T}} = \sqrt{\frac{\sigma^2}{n_0 T} + \frac{\sigma^3}{\Delta T}}.$$

□

To establish an idealized baseline, this lemma applies the general bound from Theorem 4.2, though the formal proof of that theorem is deferred to Appendix D.3. Readers may prefer to complete the appendix to understand the full dynamic framework before returning to analyze this specific stationary example.

C.2 Proof Regarding Cluster Example in Section 3.2

Lemma C.4 (Convergence in the Presence of an Unbiased Cluster)**.** *Consider the setting of Example 3.2, where there exists a cluster of clients (Cluster 0) of size n_0 such that for all $i \in$ Cluster 0, the bias is zero ($b_t^{(i)} = 0$). If the step size is optimally tuned via clamping, the algorithm achieves the following convergence rate:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 = \mathcal{O}\left(\frac{1}{T} + \frac{\log T}{T} + \frac{\sigma}{\sqrt{n_0 T}}\right).$$

where σ^2 is the variance of the stochastic gradients.

Proof. In the setting described in Example 3.2, using the bound described in the Corollary of Theorem 4.2 combined with lemma 4.1, our algorithm satisfies:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \leq \frac{2}{T}\left(\frac{\delta_1}{\eta} + 2\sum_{t=1}^{T}\mathcal{L}_t(\alpha^*) + 2R_T + 6\sigma_{bound}^2(1 + \log T)\right). \tag{10}$$

Since α^* minimizes the cumulative loss, the inequality holds if we replace α^* with any other fixed weight selection. Consider the specific candidate weights $\alpha^{(c)}$ defined by placing uniform weights exclusively on all clients in Cluster 0 and zero weights elsewhere. Let n_0 be the size of Cluster 0.

By definition, for clients in Cluster 0, the bias is strictly zero ($b_t^{(i)} = 0$). Because this uniform candidate selection creates a perfectly unbiased gradient estimator, its second moment decomposes exactly into the squared norm of the mean and its variance, completely bypassing Young's inequality. The proxy cost simplifies cleanly to:

$$\mathcal{L}_t(\alpha^{(c)}) = \left\| \sum_{i\in[M]} \alpha^{(c)^{(i)}} b_t^{(i)} \right\|^2 + \eta L \mathbb{E} \left\| \sum_{i\in[M]} \alpha^{(c)^{(i)}} g_t^{(i)} \right\|^2 = \eta L \left(\|\nabla f^{(0)}(w_t)\|^2 + \frac{\sigma^2}{n_0} \right).$$

Substituting this exact baseline cost back into Equation (10):

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \le \frac{2}{T}\left(\frac{\delta_1}{\eta} + 2\sum_{t=1}^{T} \eta L \left(\|\nabla f^{(0)}(w_t)\|^2 + \frac{\sigma^2}{n_0} \right) + 2R_T + 6\sigma_{bound}^2(1+\log T) \right).$$

Rearranging the inequality to isolate the target gradient sum on the right-hand side, we obtain:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 \le \frac{1}{1-4\eta L}\left[\frac{2}{T}\left(\frac{\delta_1}{\eta} + 2T\eta L \frac{\sigma^2}{n_0} + 2R_T + 6\sigma_{bound}^2(1+\log T) \right) \right].$$

To safely absorb the gradient penalty on the right-hand side, we must enforce $4\eta L \le \frac{1}{2}$, which establishes a strict physical speed limit of $\eta \le \frac{1}{8L}$.

To minimize the upper bound, we balance the dominant deterministic gap $\frac{\delta_1}{\eta}$ and the stochastic variance $2\eta\frac{L\sigma^2}{n_0}$. To ensure mathematical stability for perfectly noiseless clusters ($\sigma \to 0$) while optimizing empirical performance, we define the learning rate using the step-size clamping technique:

$$\eta = \min\left(\frac{1}{8L}, \sqrt{\frac{\delta_1 n_0}{2L\sigma^2 T}} \right).$$

This guarantees the scaling factor satisfies $\frac{1}{1-4\eta L} \le 2$. To evaluate the dominant terms under the clamped learning rate, we utilize the algebraic properties of the minimum function. For any $\eta = \min(\eta_1, \eta_2)$, it strictly holds that $\frac{1}{\eta} \le \frac{1}{\eta_1} + \frac{1}{\eta_2}$ and $\eta \le \eta_2$.

Let the two candidate rates be the deterministic speed limit $\eta_1 = \frac{1}{8L}$ and the stochastic-dominant rate $\eta_2 = \sqrt{\frac{\delta_1 n_0}{2L\sigma^2 T}}$. Applying the inverse inequality to the initial distance gap yields:

$$\frac{\delta_1}{T\eta} \le \frac{\delta_1}{T\eta_1} + \frac{\delta_1}{T\eta_2} = \frac{8L\delta_1}{T} + \frac{\delta_1}{T}\sqrt{\frac{2L\sigma^2 T}{\delta_1 n_0}} = \frac{8L\delta_1}{T} + \sqrt{\frac{2\delta_1 L\sigma^2}{n_0 T}} = \mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{\sigma}{\sqrt{n_0 T}}\right)$$

Applying the direct upper bound ($\eta \le \eta_2$) to the stochastic variance term safely caps its maximum penalty:

$$\frac{2\eta L\sigma^2}{n_0} \le \frac{2\eta_2 L\sigma^2}{n_0} = 2\left(\sqrt{\frac{\delta_1 n_0}{2L\sigma^2 T}}\right)\frac{L\sigma^2}{n_0} = \sqrt{\frac{2\delta_1 L\sigma^2}{n_0 T}} = \mathcal{O}\left(\frac{\sigma}{\sqrt{n_0 T}}\right)$$

Summing these terms bounds the primary optimization penalties:

$$\frac{\delta_1}{T\eta} + \frac{2\eta L\sigma^2}{n_0} \le \mathcal{O}\left(\frac{1}{T}\right) + 2\cdot\mathcal{O}\left(\frac{\sigma}{\sqrt{n_0 T}}\right)$$

Finally, we account for the residual estimation noise (R_T and the algorithmic tracking terms). By the FTAL guarantee, this regret decays strictly as $\mathcal{O}\left(\frac{\log T}{T}\right)$.

To explicitly delineate the three distinct sources of convergence error—the deterministic optimization geometry, the active-set tracking regret, and the clustered statistical variance—we present the uncollapsed overall convergence rate. The bound is tightly given by:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f^{(0)}(w_t)\|^2 = \mathcal{O}\left(\frac{1}{T} + \frac{\log T}{T} + \frac{\sigma}{\sqrt{n_0 T}} \right).$$

This uncollapsed formulation transparently proves our three core theoretical claims: the base model architecture converges at an accelerated $\mathcal{O}(1/T)$ rate, the dynamic tracking overhead is vanishingly small ($\mathcal{O}(\log T/T)$), and the algorithm organically isolates the target cluster to achieve the optimal $\mathcal{O}(\sigma/\sqrt{n_0 T})$ Oracle variance reduction. □

D Details and Proofs for Subsection 4.3

D.1 Proof of Exp-Concavity Property

Lemma D.1. *The function $\mathcal{L}_t(\alpha)$ is γ-exp-concave with parameter:*

$$\gamma = \frac{1}{(8 + 2L\eta)\, g_{max}^2} \tag{11}$$

and its gradient is bounded by $G_{\mathcal{L}} \leq (8 + 2L\eta)\, g_{max}^2$.

Proof. To simplify the analysis, we first express the cost function $\mathcal{L}_t(\alpha)$ in a compact quadratic form. Let B_t be the matrix of population biases, and G_t be the matrix of stochastic gradients. Following our formulation utilizing the conditional expected second moment, the loss evaluates to:

$$\mathcal{L}_t(\alpha) = \|\widehat{B}_t\alpha\|^2 + \eta L \mathbb{E}_t\|G_t\alpha\|^2 = \alpha^\top \underbrace{\left(\widehat{B}_t^\top \widehat{B}_t + \eta L \mathbb{E}_t[G_t^\top G_t]\right)}_{\mathbf{H}_t} \alpha. \tag{12}$$

Because $\mathbf{H}_t$ is constructed from Gram matrices and expected outer products, it is positive semi-definite (PSD). Consequently, the function $\mathcal{L}_t(\alpha)$ is convex (though not necessarily strictly convex unless $\mathbf{H}_t \succ 0$).

Definition D.1 (γ-Exp-Concavity)**.** *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set. A function $g : \mathcal{K} \to \mathbb{R}$ is said to be γ-exp-concave for some parameter $\gamma > 0$ if the mapping*

$$x \mapsto \exp(-\gamma g(x)) \tag{13}$$

is convex over $\mathcal{K}$. Conceptually, exp-concavity bridges the gap between standard convexity and strong convexity by guaranteeing sufficient local curvature relative to the gradient magnitude. Furthermore, if g is twice continuously differentiable, this property is mathematically equivalent to the following Hessian condition for all $x \in \mathcal{K}$:

$$\nabla^2 g(x) \succeq \gamma \nabla g(x) \nabla g(x)^\top, \tag{14}$$

where $\succeq$ denotes the Loewner partial order for positive semi-definite matrices (i.e., $A \succeq B$ means $A - B$ is a positive semi-definite matrix).

1. Formulating the Exp-Concavity Condition
To prove our function $\mathcal{L}_t(\alpha)$ is γ-exp-concave, we compute its first and second derivatives. By standard matrix calculus for quadratic forms, the gradient is:

$$\nabla \mathcal{L}_t(\alpha) = 2\mathbf{H}_t\alpha \tag{15}$$

and the Hessian is the constant matrix:

$$\nabla^2 \mathcal{L}_t(\alpha) = 2\mathbf{H}_t. \tag{16}$$

Substituting these into the definition, we require:

$$2\mathbf{H}_t \succeq \gamma(2\mathbf{H}_t\alpha)(2\mathbf{H}_t\alpha)^\top = 4\gamma \mathbf{H}_t\alpha\alpha^\top\mathbf{H}_t. \tag{17}$$

2. Verification via Induced Inner Products
By the definition of the Loewner order, $A \succeq B$ if and only if $v^\top Av \geq v^\top Bv$ for any arbitrary vector $v \in \mathbb{R}^M$. Projecting both sides onto v yields:

$$2(v^\top \mathbf{H}_t v) \geq 4\gamma(v^\top \mathbf{H}_t\alpha)(\alpha^\top \mathbf{H}_t v). \tag{18}$$

Because $\mathbf{H}_t$ is symmetric and PSD, it defines a valid induced inner product space $\langle x, y\rangle_{\mathbf{H}} = x^\top \mathbf{H}_t y$ with squared norm $\|x\|_{\mathbf{H}}^2 = x^\top \mathbf{H}_t x$. Rewriting our inequality:

$$2\|v\|_{\mathbf{H}}^2 \geq 4\gamma \langle v, \alpha\rangle_{\mathbf{H}}^2. \tag{19}$$

Applying the Cauchy-Schwarz inequality ($\langle v, \alpha\rangle_{\mathbf{H}}^2 \leq \|v\|_{\mathbf{H}}^2 \|\alpha\|_{\mathbf{H}}^2$), it is mathematically sufficient to ensure:

$$2\|v\|_{\mathbf{H}}^2 \geq 4\gamma \|v\|_{\mathbf{H}}^2 \|\alpha\|_{\mathbf{H}}^2. \tag{20}$$

Dividing by the non-negative scalar $\|v\|_{\mathbf{H}}^2$ eliminates the dependence on v:

$$1 \geq 2\gamma \|\alpha\|_{\mathbf{H}}^2 = 2\gamma \mathcal{L}_t(\alpha). \tag{21}$$

Rearranging for γ, the function satisfies exp-concavity for any parameter bounded by the maximum possible value of the loss: $\gamma \leq \frac{1}{2 \max_\alpha \mathcal{L}_t(\alpha)}$.

3. Bounding the Maximum Loss and Gradient
Because $\mathcal{L}_t(\alpha)$ is convex over the probability simplex Δ_M, its maximum occurs at a vertex. To rigorously bound this maximum, we introduce the explicit assumption that the local gradients and true gradients are almost surely bounded such that $\|g_t^{(i)}\| \leq g_{max}$ and $\|\nabla f^{(i)}(w_t)\| \leq g_{max}$ for all i.

Evaluating the loss at a vertex isolates individual client bounds. Utilizing the implied bias bound $\|b_t^{(i)}\|^2 \leq 4g_{max}^2$, we bound the maximal vertex loss:

$$\max_{\alpha \in \Delta_M} \mathcal{L}_t(\alpha) \leq \max_i \left(\|b_t^{(i)}\|^2 + L\eta \mathbb{E}_t \|g_t^{(i)}\|^2 \right) \leq 4g_{max}^2 + L\eta g_{max}^2 = (4 + L\eta) g_{max}^2. \tag{22}$$

Substituting this maximal loss into our condition gives the final parameter:

$$\gamma = \frac{1}{2(4 + L\eta) g_{max}^2} = \frac{1}{(8 + 2L\eta) g_{max}^2}. \tag{23}$$

Finally, we bound the gradient norm $G_{\mathcal{L}}$. Given $\|\nabla \mathcal{L}_t(\alpha)\|_2 = 2\|\mathbf{H}_t \alpha\|_2 \leq 2\lambda_{max}(\mathbf{H}_t)\|\alpha\|_2$. Since the maximum quadratic form over the simplex bounds the largest eigenvalue ($\lambda_{max}(\mathbf{H}_t) \leq (4 + L\eta) g_{max}^2$) and $\|\alpha\|_2 \leq \|\alpha\|_1 = 1$, the gradient norm is strictly bounded by:

$$G_{\mathcal{L}} \leq 2(4 + L\eta) g_{max}^2 = (8 + 2L\eta) g_{max}^2. \tag{24}$$

This unifies the constants and concludes the proof. □

D.2 The FTAL Algorithm

FTAL As discussed in Section 4.3, our implementation utilizes a variation of the Follow the Leader (FTL) framework. While standard "Follow the Leader" (FTL) strategies can perform poorly in adversarial settings due to instability, FTAL introduces controlled noise to ensure robustness and minimize regret.

We present the approximate algorithm below D.2 for readers who may be unfamiliar with its mechanics.

Algorithm 2 FOLLOW THE APPROXIMATE LEADER (version 1)

convex set $\mathcal{P} \subset \mathbb{R}^n$, and the parameter β.

- On period 1, play an arbitrary $\mathbf{x}_1 \in \mathcal{P}$.
- On period t, play the leader $\mathbf{x}_t$ defined as

$$\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{P}} \sum_{\tau=1}^{t-1} \tilde{f}_\tau(\mathbf{x})$$

Where for $\tau = 1, \ldots, t-1$, define $\nabla_\tau = \nabla f_\tau(\mathbf{x}_\tau)$ and

$$\tilde{f}_\tau(\mathbf{x}) = f_\tau(\mathbf{x}_\tau) + \nabla_\tau^\top (\mathbf{x} - \mathbf{x}_\tau) + \frac{\beta}{2} (\mathbf{x} - \mathbf{x}_\tau)^\top \nabla_\tau \nabla_\tau^\top (\mathbf{x} - \mathbf{x}_\tau)$$

D.3 Proof of Theorem 4.2

In this section, we provide the complete proof for the convergence of the algorithm under estimated bias. We begin by establishing a bound on the estimation error of the bias tracking mechanism, and then utilize this bound to prove the main convergence theorem.

Theorem 4.2. *Let Assumptions 2.1 and 2.2 hold. If the algorithm minimizes the estimated auxiliary loss $\widehat{\mathcal{L}}_t$ with curvature constant γ, the convergence bound is given by:*

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathcal{L}_t(\alpha_t)\right] \leq 4\sum_{t=1}^{T}\mathcal{L}_t(\alpha^*) + 2R_T + 6\sum_{t=1}^{T}\mathbb{E}[\|\xi_t\|^2] \tag{25}$$

Proof. Define $\mathcal{V}_t(\alpha) = L\eta\mathbb{E}[\|\alpha G_t\|^2]$ as the variance term (common to both as it depends on gradient norms).

And let $\xi_t = b_t - \widehat{b}_t$ denote the estimation error. For any vector $\alpha \in \Delta_M$, we relate the true and estimated bias terms using the standard inequality $\|x+y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$.

Step 1: Upper Bound on Realized Loss. For the chosen action α_t, we substitute the true bias with its estimate plus the error ($b_t = \widehat{b}_t + \xi_t$). Applying the aforementioned norm inequality and noting that the variance term is non-negative ($\mathcal{V}_t(\alpha_t) \leq 2\mathcal{V}_t(\alpha_t)$), we obtain:

$$\begin{aligned}\mathcal{L}_t(\alpha_t) &= \|\alpha_t(\widehat{b}_t + \xi_t)\|^2 + L\eta\alpha_t\mathbb{E}[\|\alpha G_t\|^2] \\ &\leq 2\|\alpha_t\widehat{b}_t\|^2 + 2\|\alpha_t\xi_t\|^2 + L\eta\alpha_t\mathbb{E}[\|\alpha G_t\|^2] \\ &= 2\widehat{\mathcal{L}}_t(\alpha_t) + 2\|\alpha_t\xi_t\|^2. \end{aligned} \tag{26}$$

Step 2: Lower Bound on Optimal Loss. Similarly, for the optimal comparator α^*, we substitute the estimated bias with the true bias minus the error ($\widehat{b}_t = b_t - \xi_t$) and apply the same inequality bounds:

$$\begin{aligned}\widehat{\mathcal{L}}_t(\alpha^*) &= \|\alpha^*(b_t - \xi_t)\|^2 + \mathcal{V}_t(\alpha^*) \\ &\leq 2\|\alpha^* b_t\|^2 + 2\|\alpha^*\xi_t\|^2 + \mathcal{V}_t(\alpha^*) \\ &\leq 2\mathcal{L}_t(\alpha^*) + 2\|\alpha^*\xi_t\|^2. \end{aligned} \tag{27}$$

Step 3: Applying Regret Bounds. The algorithm minimizes the sequence $\widehat{\mathcal{L}}_t$. Since $\widehat{\mathcal{L}}_t$ satisfies the exp-concavity property (as established in Section 4.3), the FTAL algorithm guarantees a regret bound R_T on this estimated sequence:

$$\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\alpha_t) \leq \sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\alpha^*) + R_T, \tag{28}$$

where $R_T = 64\left(\frac{1}{\gamma} + GD\right)M(1+\log T)$.

We now chain these inequalities together. Substituting the upper bound from Step 1, applying the FTAL regret bound, and then substituting the lower bound from Step 2, we get:

$$\begin{aligned}\sum_{t=1}^{T}\mathcal{L}_t(\alpha_t) &\leq 2\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\alpha_t) + 2\sum_{t=1}^{T}\|\alpha_t\xi_t\|^2 \\ &\leq 2\left(\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\alpha^*) + R_T\right) + 2\sum_{t=1}^{T}\|\alpha_t\xi_t\|^2 \\ &\leq 2\left(\sum_{t=1}^{T}\left(2\mathcal{L}_t(\alpha^*) + 2\|\alpha^*\xi_t\|^2\right) + R_T\right) + 2\sum_{t=1}^{T}\|\alpha_t\xi_t\|^2 \\ &= 4\sum_{t=1}^{T}\mathcal{L}_t(\alpha^*) + 2R_T + \sum_{t=1}^{T}\left(4\|\alpha^*\xi_t\|^2 + 2\|\alpha_t\xi_t\|^2\right). \end{aligned} \tag{29}$$

Step 4: Expectation and Noise Bound. We now take the expectation with respect to the stochasticity of the bias estimation. Since $\alpha_t, \alpha^* \in \Delta_M$, we have $\|\alpha\xi_t\|^2 \leq \|\alpha\|_1^2 \|\xi_t\|_\infty^2$, but more simply via Cauchy-Schwarz on the vector product $\|\alpha\xi_t\| \leq \|\alpha\|\|\xi_t\|$. Since α is in the simplex, $\|\alpha\| \leq 1$. Thus, $\|\alpha\xi_t\|^2 \leq \|\xi_t\|^2$. obtain:

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(4\|\alpha^*\xi_t\|^2 + 2\|\alpha_t\xi_t\|^2\right)\right] \leq 6\sum_{t=1}^{T}\mathbb{E}[\|\xi_t\|^2]$$

Substituting this noise term back into the main inequality yields the final bound:

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathcal{L}_t(\alpha_t)\right] \leq 4\sum_{t=1}^{T}\mathcal{L}_t(\alpha^*) + 2R_T + 6\sum_{t=1}^{T}\mathbb{E}[\|\xi_t\|^2] \tag{30}$$

To fully parameterize R_T, we use the fact that we optimize over the probability simplex Δ_M, which trivially bounds the domain diameter $D \leq \sqrt{2}$. Combining this with Lemma D.1 (which provides the bound on the gradient norm $G_{\mathcal{L}}$ of the loss function) completes the proof. □

E Extension to Partial Participation

While the primary analysis of SP-CACW is presented under the assumption of full participation, the framework naturally extends to partial participation settings via a standard mathematical reduction. Assume each client i participates at communication round t with a uniform probability $p \in (0, 1]$. Let $z_{i,t} \sim \text{Bernoulli}(p)$ be independent random indicator variables denoting whether client i is selected in round t.

To account for the missing updates from unselected clients, we employ the unbiased gradient estimator:

$$\tilde{g}_t^{(i)} = \frac{g_t^{(i)}}{p} z_{i,t} \tag{31}$$

The server aggregates these estimators using the FTAL collaboration weights α_t. The following proposition formally establishes the statistical properties of the aggregated estimator, ensuring the fundamental bias-variance tradeoff optimized by FTAL strictly holds.

Proposition 1 (Aggregated Partial Participation Estimator)**.** *Let the aggregated estimator be defined as* $\tilde{G}_t(\alpha_t) := \sum_{i=0}^{M-1} \alpha_t^{(i)} \tilde{g}_t^{(i)}$*. Assuming the client sampling indicators* $z_{i,t}$ *are independent, the estimator satisfies the following properties conditioned on the current gradients:*

1. ***Unbiasedness:*** $\mathbb{E}_Z[\tilde{G}_t(\alpha_t)] = \sum_{i=0}^{M-1} \alpha_t^{(i)} g_t^{(i)}$

2. ***Variance Bound:*** $Var_Z(\tilde{G}_t(\alpha_t)) = \sum_{i=0}^{M-1} (\alpha_t^{(i)})^2 \left(\frac{1}{p_i} - 1\right) \|g_t^{(i)}\|^2$ *where* $\mathbb{E}_Z$ *and* Var_Z *denote the expectation and variance over the sampling indicators* $z_{i,t}$*.*

Proof. **1. Unbiasedness:**
By linearity of expectation and the definition of the Bernoulli variable ($\mathbb{E}_Z[z_{i,t}] = p_i$):

$$\mathbb{E}_Z[\tilde{G}_t(\alpha_t)] = \sum_{i=0}^{M-1} \alpha_t^{(i)} \mathbb{E}_Z\left[\frac{g_t^{(i)}}{p_i} z_{i,t}\right] = \sum_{i=0}^{M-1} \alpha_t^{(i)} g_t^{(i)}$$

2. Variance Bound:
By definition, the variance of the sum of independent random variables is the sum of their variances. Since the client sampling indicators are independent across the network, we have:

$$\text{Var}_Z(\tilde{G}_t(\alpha_t)) = \sum_{i=0}^{M-1} (\alpha_t^{(i)})^2 \text{Var}_Z\left(\frac{g_t^{(i)}}{p_i} z_{i,t}\right)$$

For a Bernoulli random variable $z_{i,t}$ with parameter p_i, the variance is $\text{Var}_Z(z_{i,t}) = p_i(1-p_i)$. Pulling out the deterministic constants yields:

$$\begin{aligned}\text{Var}_Z(\tilde{G}_t(\alpha_t)) &= \sum_{i=0}^{M-1} (\alpha_t^{(i)})^2 \frac{\|g_t^{(i)}\|^2}{p_i^2} p_i(1-p_i) \\ &= \sum_{i=0}^{M-1} (\alpha_t^{(i)})^2 \left(\frac{1}{p_i} - 1\right) \|g_t^{(i)}\|^2\end{aligned}$$

□

This rigorous formulation demonstrates that SP-CACW seamlessly extends to partial participation. The algorithm continues to track the true target gradient perfectly, with the only penalty being a variance inflation factor $\left(\frac{1}{p_i} - 1\right)$ strictly localized to the sampled clients. Notably, this noise inflation is aggressively suppressed because it scales quadratically with the FTAL weights $(\alpha_t^{(i)})^2$, proving that our proxy-loss framework fundamentally mitigates sampling variance alongside structural heterogeneity.

F Practical Algorithmic Details and Refinements

While our theoretical analysis assumes access to population statistics, practical deployment requires refinements to handle stochastic noise, non-stationary dynamics, and numerical stability. We introduce a key modifications to the core algorithm to bridge this gap.

1. Cubic Regularization. To prevent the algorithm from prematurely collapsing into a trivial solution (e.g., placing probability 1 on the target client or a single peer), we introduce a regularization term. We select a cubic penalty $\sum(\alpha^{(i)})^3$ rather than the standard quadratic (L_2) penalty. Cubic regularization provides a flatter gradient near zero, allowing small but useful contributions to persist, while penalizing large dominant weights more aggressively than L_2. This promotes a more robust and distributed collaboration.

With these modifications, the practical objective function minimized at round T is given by the empirical sum:

$$\mathcal{L}_{\text{prac}}(\alpha) = \sum_{t=T_{\text{start}}}^{T} \underbrace{\left\| \sum_{i=0}^{M-1} \alpha^{(i)} \hat{b}_t^{(i)} \right\|^2}_{\text{Estimated Bias}} + \sum_{t=T_{\text{start}}}^{T} L\eta_t \underbrace{\left\| \sum_{i=0}^{M-1} \alpha^{(i)} g_t^{(i)} \right\|^2}_{\text{Empirical Variance Proxy}} + \lambda_{\text{reg}} \sum_{i=0}^{M-1} (\alpha^{(i)})^3. \tag{32}$$

F.1 Implementation Details

For the experimental evaluation, we implemented the model optimization through the following technical procedures:

1. **Lipschitz Constant Estimation:** The smoothness constant L is estimated dynamically using local curvature information. To maintain computational efficiency, we approximate the spectral norm of the Hessian via Power Iteration, computing Hessian-vector products using the method of Pearlmutter [1994]. This avoids materializing the full Hessian matrix while providing a lightweight approximation of L at each iteration.
2. **Dynamic Regularization:** The regularization coefficient λ_{reg} is adaptive, scaling with the signal strength of the target client. We set $\lambda_{\text{reg}} = \gamma \cdot \|var(g_t^{(0)})\|^2$ (where $\gamma = 1$), ensuring the penalty naturally decays as the model approaches convergence and the gradient magnitude diminishes.
3. **Numerical Solver (PGD):** To efficiently find the optimal weights α at each step, we employ **Projected Gradient Descent (PGD)**. This allows us to optimize the objective while strictly enforcing the simplex constraint $\sum \alpha^{(i)} = 1, \alpha^{(i)} \geq 0$ via Euclidean projection.

4. **Lazy Weight Updates (Epoch-Based):** To improve stability and reduce computational cost, we update the aggregation weights α_t only once every K iterations. In our experiments, we set K to correspond to one full pass over the local data (one epoch). This strategy allows the bias estimates to accumulate sufficient statistics from the entire data distribution before shifting the collaboration topology, thereby preventing the weights from overfitting to specific transient batches.

5. **Dimensionality Reduction via PCA (PCA Variant):** In deep neural networks, the gradient parameter space P is exceptionally large, making the direct optimization of aggregation weights computationally heavy. For the PCA-enabled variant ("SP-CACW pca no reg"), we apply a low-rank Principal Component Analysis (PCA) to the high-dimensional bias and gradient tensors before computing the collaboration weights. Specifically, we project the P-dimensional bias and gradient matrices down to an M-dimensional subspace (where M is the number of clients) using a randomized, uncentered low-rank PCA approximation. The FTAL weight solver then calculates the optimal α weights using these dense, low-dimensional representations ($\mathbb{R}^M$ instead of $\mathbb{R}^P$), which significantly reduces computational overhead and filters out high-frequency gradient noise while preserving the primary directions of variance.

F.2 Comprehensive Complexity Analysis

The comprehensive and communication cost can be seen in this table:

Component	Complexity	Note
Up link	$O(d)$	Identical to Standard FL
Down link	$O(M \cdot d)$	Reception of peer updates
Aggregation & Bias	$O(M \cdot d)$	Linear vector operations and EMA tracking
Weight Optimization	$O(I \cdot M \cdot d)$	Efficient optimization of collaboration weights via PGD

To formally establish this, we analyze the per-round overhead of Algorithm 1. Let d denote the model dimension, M the number of participating clients, and I the number of local projection iterations required for the optimization solver. The total algorithmic cost per round (C_{total}) is explicitly defined as the sum of the network communication cost (C_{comm}) and the local computational complexity (C_{comp}).

1. Communication Cost (C_{comm}):
The only network exchange occurs when the server broadcasts the model and receives the local stochastic gradients $g_t^{(i)}$ from the active peers (Line 3).

$$\begin{aligned} C_{comm} = & \underbrace{\mathcal{O}(d)}_{\text{Uplink Broadcast}} + \underbrace{\mathcal{O}(M \cdot d)}_{\text{Down link Reception (Line 3)}} \\ & = \mathcal{O}(M \cdot d) \end{aligned} \tag{33}$$

2. Computational Cost (C_{comp}):
Once the local gradients are received, all subsequent personalization and tracking mechanics are executed entirely in-memory via linear vector operations. The computation breaks down into the bias EMA tracking (Line 4), the weighted gradient aggregation (Line 5), the model update (Line 6), and the FTAL weight optimization via projected gradient descent (Line 7).

$$\begin{aligned} C_{comp} = & \underbrace{\mathcal{O}(M \cdot d)}_{\text{Bias \& Aggregation (Lines 4-5)}} + \underbrace{\mathcal{O}(d)}_{\text{Model Update (Line 6)}} + \underbrace{\mathcal{O}(I \cdot M \cdot d)}_{\text{FTAL Weight Opt. (Line 7)}} \\ & = \mathcal{O}(I \cdot M \cdot d) \end{aligned} \tag{34}$$

Total Algorithmic Overhead:
Summing the explicitly derived communication and computational requirements yields the total

worst-case complexity per communication round:

$$\begin{aligned} C_{total} &= C_{comm} + C_{comp} \\ &= \mathcal{O}(M \cdot d) + \mathcal{O}(I \cdot M \cdot d) \\ &= \mathcal{O}(I \cdot M \cdot d) \end{aligned} \tag{35}$$

Epoch-Level Amortization: To ensure that the $\mathcal{O}(I \cdot M \cdot d)$ overhead remains practically scalable, we deploy an epoch-level amortization strategy. While the FTAL weight solver (Line 7) represents the dominant computational burden, these collaboration weights only need to be updated once per global communication epoch. By amortizing this local calculation across all intermediate local backpropagation steps, the effective computational overhead of SP-CACW approaches the theoretical baseline of standard Federated Learning, without requiring a single additional byte of network bandwidth.

G Network Architectures and Hyperparameter

G.1 MNIST Experimental Setup

We employ a compact Multi-Layer Perceptron (MLP). The architecture accepts flattened inputs (784 dimensions), containing a single hidden layer with 64 units (ReLU activation) and a final output layer for the 10 classes.

Hyperparameter.

- Learning Rate: $\eta = 0.01$, Momentum: $\mu = 0.9$
- Weight Decay: $\lambda = 5 \times 10^{-4}$,max norm of gradient=1
- batch size: 32

G.2 CIFAR-100 Experimental Setup

For CIFAR-100, we utilize a modified VGG16 architecture Simonyan and Zisserman [2015] adapted specifically for lower-resolution 32×32 inputs. Key modifications to the standard VGG design include the aggressive integration of **Batch Normalization** and 2D spatial dropout ($p = 0.3$) immediately following every Convolution layer to stabilize training and prevent overfitting. Additionally, the traditional massive fully connected layers are replaced with a streamlined classifier featuring a single 512-dimensional hidden layer, regularized with 1D Batch Normalization and dropout ($p = 0.5$).

Hyperparameter. Optimization is performed using SGD:

- Learning Rate: $\eta = 0.1$, Momentum: $\mu = 0.9$
- Weight Decay: $\lambda = 5 \times 10^{-4}$,max norm of gradient=5
- batch size: 32

G.3 Shakespeare (LEAF) Experimental Setup

For the Shakespeare dataset, we utilize a stacked Character-Level Long Short-Term Memory (CharLSTM) network for next-character prediction. The architecture maps discrete input character tokens into a dense continuous space using an embedding layer. This is followed by a 2-layer LSTM feature extractor configured with a hidden state dimension of 256 to capture long-term sequential dependencies. Finally, the unrolled sequence outputs are passed through a fully connected linear layer to predict the probability distribution over the entire vocabulary space. The recurrent hidden and cell states are zero-initialized at the start of each batch.

Hyperparameter. Optimization is performed using SGD:

Table 3: **VGG16-CIFAR100 Network Architecture.** The feature extractor uses sequential 3×3 convolutions. All Convolution layers are followed by 2D Batch Norm, an in-place ReLU, and 2D spatial dropout ($p = 0.3$). The classifier uses 1D Batch Norm and dropout ($p = 0.5$).

Layer Type	In	Out	Kernel	Stride	Padding
Conv2d	3	64	3×3	1	1
Conv2d	64	64	3×3	1	1
MaxPool2d	64	64	2×2	2	0
Conv2d	64	128	3×3	1	1
Conv2d	128	128	3×3	1	1
MaxPool2d	128	128	2×2	2	0
Conv2d	128	256	3×3	1	1
Conv2d	256	256	3×3	1	1
Conv2d	256	256	3×3	1	1
MaxPool2d	256	256	2×2	2	0
Conv2d	256	512	3×3	1	1
Conv2d	512	512	3×3	1	1
Conv2d	512	512	3×3	1	1
MaxPool2d	512	512	2×2	2	0
Conv2d	512	512	3×3	1	1
Conv2d	512	512	3×3	1	1
Conv2d	512	512	3×3	1	1
MaxPool2d	512	512	2×2	2	0
Flatten	512	512	-	-	-
Linear (Classifier)	512	512	-	-	-
Linear (Output)	512	100	-	-	-

Table 4: **CharLSTM Network Architecture.** The network processes variable-length sequences. The vocabulary size is denoted by $|V|$. The LSTM layers output the full sequence (`batch_first=True`), which is then linearly mapped to the vocabulary size.

Layer Type	In Dimension	Out Dimension	Details
Embedding	$\|V\|$	256	-
LSTM	256	256	2 Layers
Linear (Classifier)	256	$\|V\|$	-

- Sequence Length: 50 batch size: 32
- Learning Rate: $\eta = 0.01$, Momentum: $\mu = 0.9$
- Weight Decay: $\lambda = 1 \times 10^{-4}$,max norm of gradient=1

G.4 Compute resources

Due to varying computational demands across our datasets, the experiments were conducted using two distinct hardware setups. The MNIST experiments were executed locally on a machine equipped with an Intel Core i7-1165G7 CPU (2.80GHz) and 16GB of RAM, requiring approximately 6 hours of compute time per experimental run. Conversely, due to the increased complexity of the image and text data, the CIFAR-100 and LEAF Shakespeare experiments were executed on a dedicated workstation equipped with an NVIDIA GeForce RTX 4090 GPU (24GB VRAM). Each of these larger experimental runs required approximately 14 hours of compute time to complete.

H Data Partitioning Algorithms

In this section, we detail the data partitioning procedures employed in Section 5 to simulate realistic non-IID distributions. We utilize a noise parameter ϵ to control the degree of stochastic hetero-

geneity, ensuring that even clients with similar distributions possess varying data volumes and class imbalances.

H.1 CIFAR-100 Data Partitioning (Latent Clusters)

This algorithm is used for the CIFAR-100 experiments (Section 6.3) to simulate a *Label Skew* scenario with latent community structure. Clients are grouped into latent clusters, where each cluster is responsible for a distinct subset of the 100 classes.

Mechanism: The initialization assigns each of the N clients to one of C clusters, and the 100 classes are divided disjointedly among these clusters to form the "primary signal." Crucially, to simulate realistic imperfections, we add Gaussian noise $\mathcal{N}(0, \sqrt{\epsilon})$ to the probability mass of **every** class for every client, regardless of cluster membership.

This noise addition serves two purposes:

1. **Intra-Cluster Heterogeneity:** It varies the distributions of the primary classes, ensuring clients in the same cluster are not identical.
2. **Inter-Cluster Leakage:** It introduces small, non-zero probabilities for classes *outside* the client's assigned cluster. This breaks the strict disjoint assumption, creating a "background noise" level that tests the algorithm's robustness to irrelevant data.

Algorithm 3 CIFAR-100 Partitioning (Latent Clusters)

Require: N (Clients), C (Clusters), M (Clients per cluster), ϵ (Heterogeneity noise)
Ensure: Allocation matrix $\mathbf{X} \in \mathbb{R}^{N \times 100}$

Configuration:
  $P \leftarrow 100/C$ {Number of unique classes per cluster}
  $B \leftarrow 1/M$ {Base probability mass}
Initialize: $\mathbf{X} \leftarrow \mathbf{0}_{N \times 100}$
Generate Noise: $\mathbf{N} \sim \mathcal{N}(0, \sqrt{\epsilon})$
for each client $i \in \{0, \ldots, N-1\}$ **do**
  $k \leftarrow \lfloor i/M \rfloor$ {Determine latent cluster index for client i}
  for each class $j \in \{0, \ldots, 99\}$ **do**
    Primary Signal:
    if $k \cdot P \leq j < (k+1) \cdot P$ **then**
      $X_{i,j} \leftarrow B$ {Assign base mass if class is in cluster}
    end if
    Global Noise Injection:
    $X_{i,j} \leftarrow X_{i,j} + N_{i,j}$ {Adds noise to ALL classes (in and out of cluster)}
    Rectify: $X_{i,j} \leftarrow \max(0, X_{i,j})$
  end for
end for
Normalize: Perform column-wise normalization on $\mathbf{X}$ to ensure valid probability distribution.
return $\mathbf{X}$

H.2 Detailed Performance Metrics and Variance Analysis

In all experiments, we report the maximum average performance across N independent runs initialized with different random seeds. To capture the variance across these runs and ensure statistical reliability, we report 1-sigma error bars representing the sample standard deviation.

Specifically, for a given method, let x_i be the performance metric of the i-th run at the epoch that achieved the highest mean validation performance. The mean μ and sample standard deviation σ are calculated as:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{36}$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{37}$$

I Full Plots of the Empirical Study

I.1 CIFAR-100 Experiments

To evaluate tracking performance and convergence behavior, we conducted experiments on the CIFAR-100 image classification task under varying degrees of data heterogeneity, controlled by σ. Figures in this section track the test accuracy over 100 epochs for $\sigma = 10$ and $\sigma = 0.01$, respectively. Across both heterogeneity settings, FedDisco achieves top-tier final test accuracy ($\sim 52 - 54\%$), with SP-CACW variants establishing a strong secondary convergence trajectory ($\sim 45 - 50\%$). Notably, SP-CACW maintains a highly smooth and robust convergence, completely avoiding the early-epoch volatility and underperformance exhibited by the ORACLE baseline. In contrast, methods such as Ditto and the localized "Only My Data" baseline plateau significantly earlier, failing to surpass 20% accuracy under severe heterogeneity ($\sigma = 0.01$).

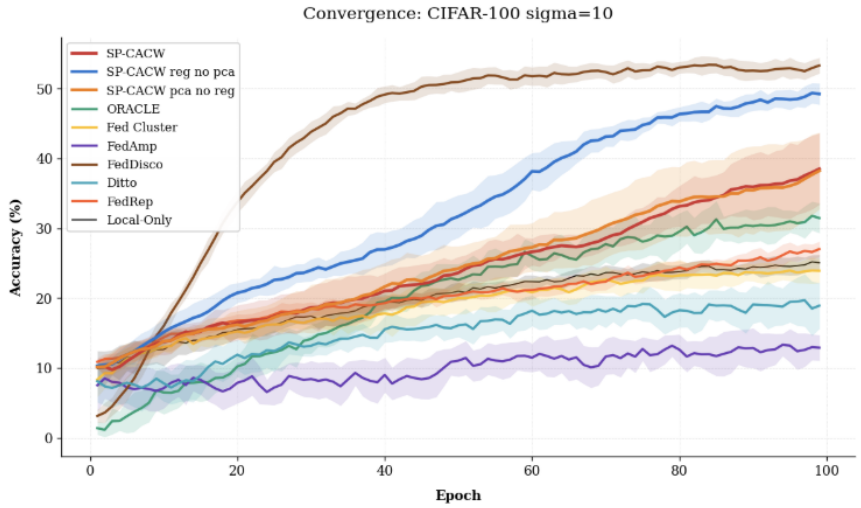

Figure 1: Test accuracy over 100 epochs for CIFAR-100 with intra-cluster noise $\sigma = 10$.

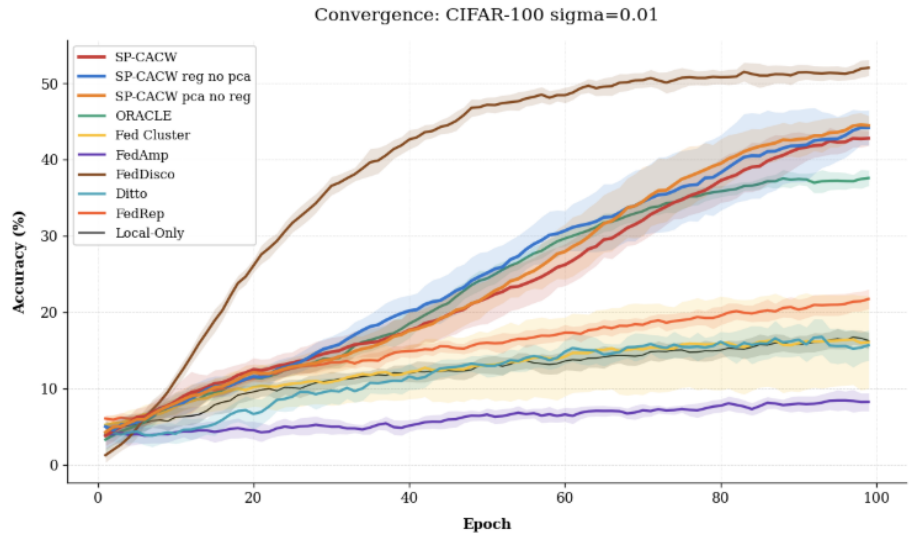

Figure 2: Test accuracy over 100 epochs for CIFAR-100 with intra-cluster noise $\sigma = 0.01$.

I.2 CIFAR-100 Label-Switching Convergence Dynamics

Our figure illustrates the optimization trajectories for the CIFAR-100 label-switching experiment over 100 epochs. This adversarial environment, characterized by severely contradictory data across clusters, clearly delineates the robustness of the evaluated methods. Most notably, FedDisco flatlines near 2% accuracy from the outset, suffering a catastrophic collapse due to its reliance on rigid, static label assumptions that are entirely violated in this setting. In stark contrast, both SP-CACW variants (regularized and unregularized) demonstrate rapid adaptation, successfully identifying and utilizing helpful peers while filtering out contradictory ones. The variants peak exceptionally high at roughly 45% accuracy around epoch 60. While both SP-CACW variants exhibit notable variance and a slight performance degradation in the final epochs—likely an artifact of the extreme noise inherent to the label-switched regime—they maintain a definitive and substantial advantage over standard baselines such as FedRep, Ditto, and Fed Cluster, which fail to achieve comparable peak performance.

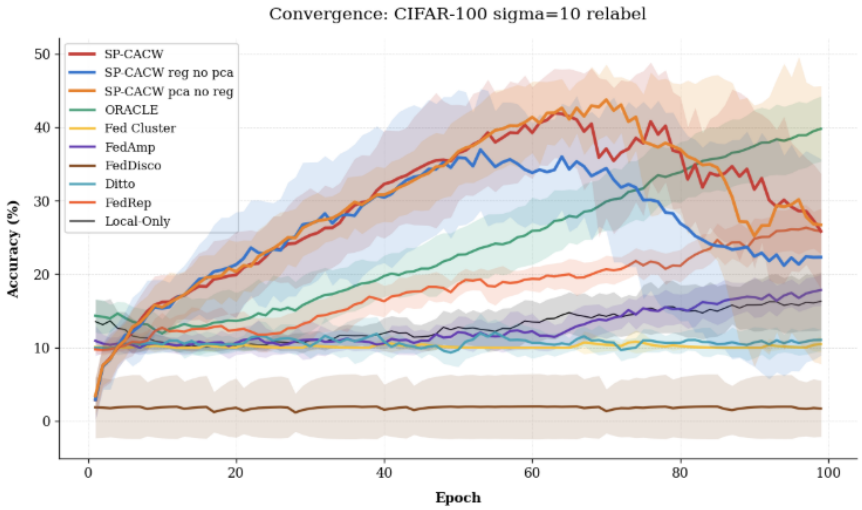

Figure 3: Optimization trajectories for the CIFAR-100 label-switching experiment over 100 epochs.

I.3 LEAF Shakespeare Experiments

This subsection details the empirical results evaluated on the LEAF Shakespeare language modeling task. To accurately reflect the strict heterogeneity of realistic federated networks, the data distribution here is partitioned naturally by individual character speaking roles rather than relying on synthetic client clusters. Figures in this section track the metric value over 100 epochs across two distinct participant sampling configurations ($n = 15, m = 7$ and $m = 30, n = 3$). SP-CACW (both with and without regularization) achieves highly competitive final metric values (up to $\sim 54\%$ and $\sim 48\%$, respectively), matching or exceeding FedDisco. Crucially, baseline methods such as FedAmp and Ditto suffer from significant performance gaps, while the "Only My Data" approach visibly degrades after epoch 40 due to severe local overfitting on the character-specific dialog.

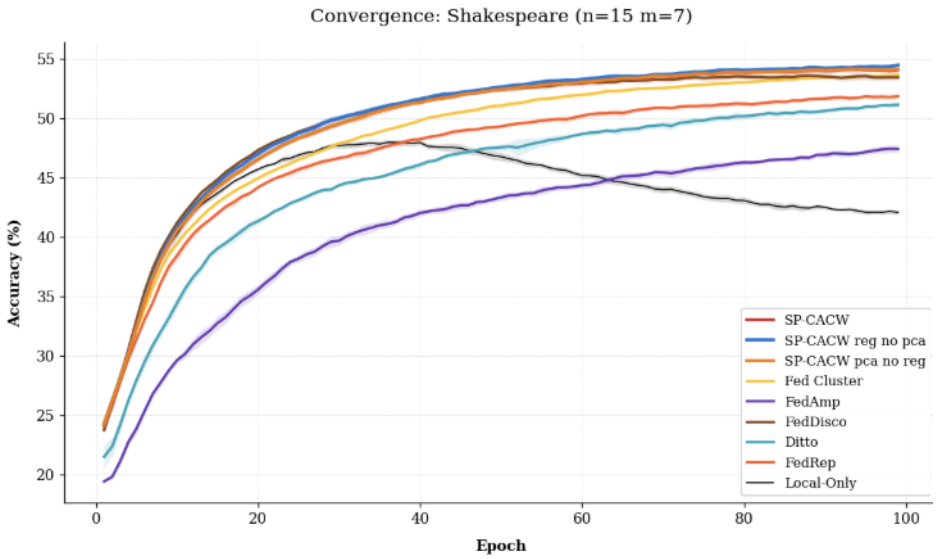

Figure 4: Test accuracy over 100 epochs on the LEAF Shakespeare dataset with $n = 15$ clients and $m = 7$ roles.

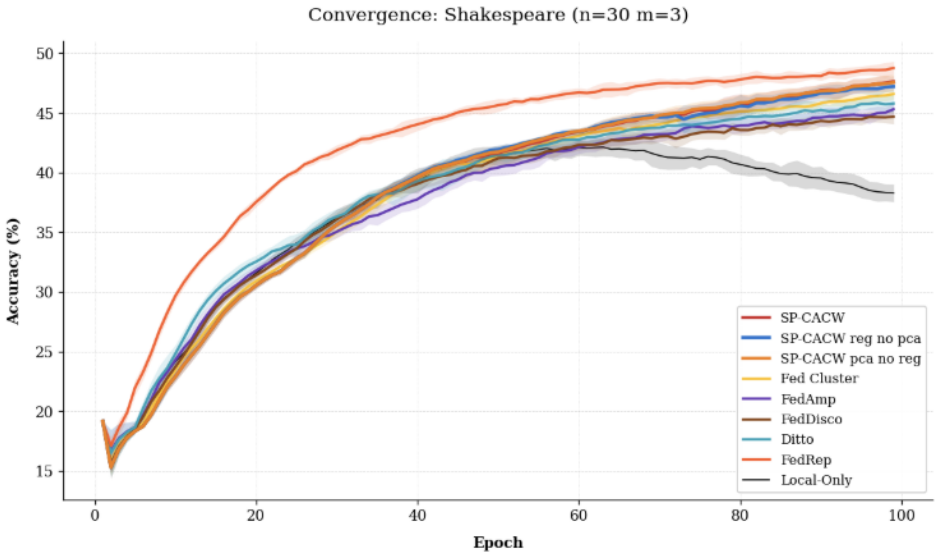

Figure 5: Test accuracy over 100 epochs on the LEAF Shakespeare dataset with $n = 30$ clients and $m = 3$ roles.

I.4 MNIST Experiments

The final set of plots showcases the algorithmic performance on the MNIST handwritten digit dataset, encompassing structural challenges such as feature rotation and label distribution skews. Figures in this section illustrate the test accuracy over 50 epochs for the rotation and relabeling scenarios, respectively. A critical observation in these experiments is the catastrophic failure of FedDisco; while highly competitive on standard feature skew, it completely collapses under extreme structural heterogeneity, flatlining below 55% on rotation and near 30% on relabeling. In stark contrast, SP-CACW demonstrates exceptional robustness, rapidly converging to the optimal $\sim 89 - 95\%$ accuracy band alongside FedCluster and ORACLE. Furthermore, on the rotation task, the regularized variant ("SP-CACW reg no pca") pulls slightly ahead, suggesting that the regularizer in environments with highly orthogonal local gradients allows the active set to effectively discard irrelevant clients without sacrificing overall stability.

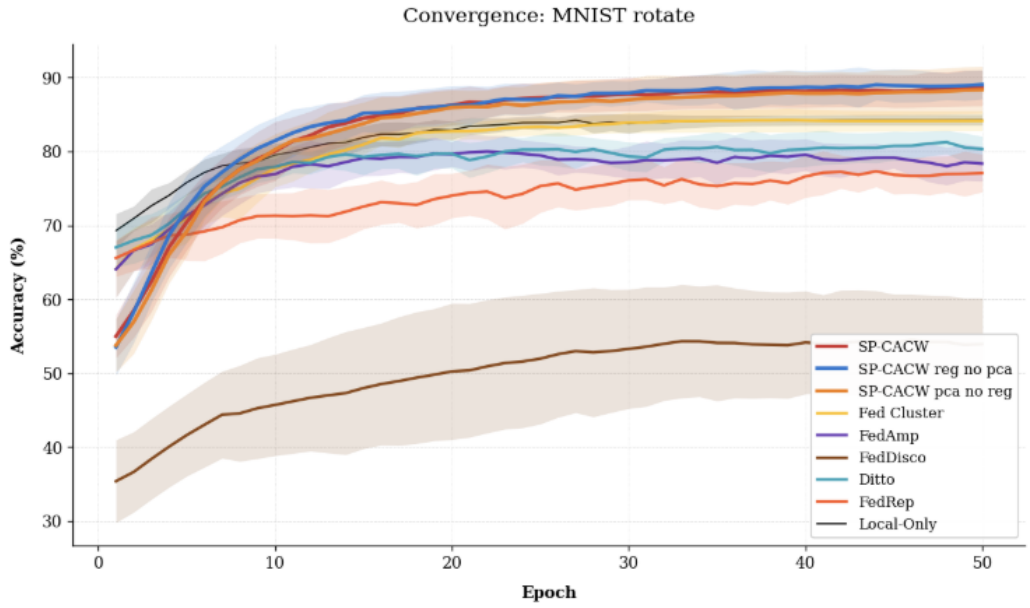

Figure 6: Test accuracy over 50 epochs for the MNIST feature rotation scenario.

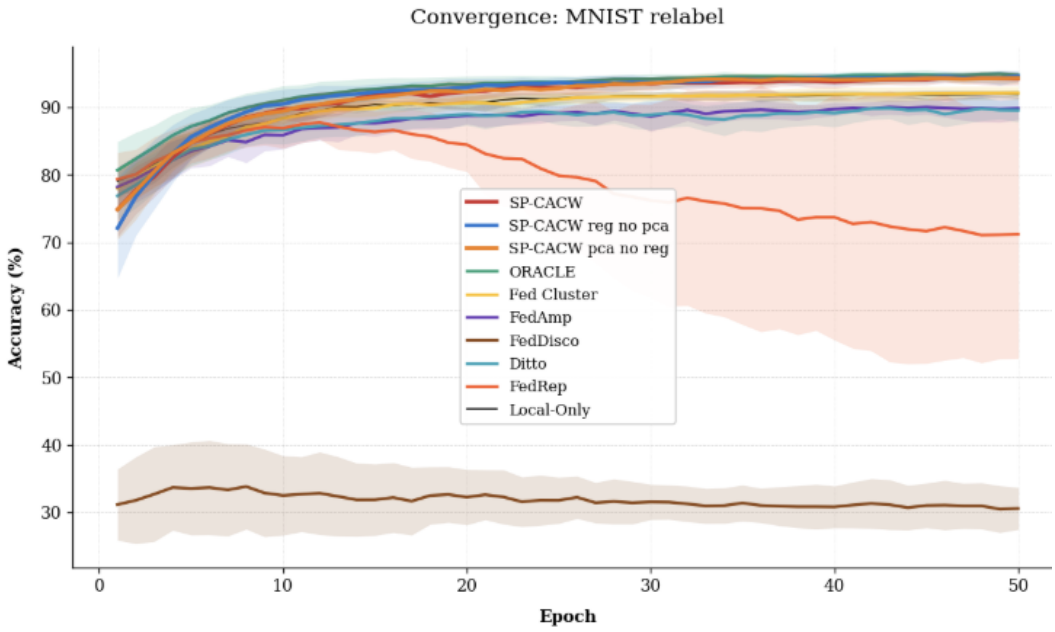

Figure 7: Test accuracy over 50 epochs for the MNIST label relabeling scenario.

I.5 Partial Participation on LEAF Shakespeare

This experiment evaluates the scalability and communication efficiency of our framework on the LEAF Shakespeare dataset, expanded to a large-scale setting with $n = 200$ clients. The experiment illustrates the test accuracy over 100 epochs, specifically comparing full collaboration against a 10% partial participation regime. A critical observation is that SP-CACW achieves the highest overall accuracy under full participation ($\sim 54\%$). Furthermore, when restricted to only 10% partial participation ("SP-CACW reg 10% partial participation"), the framework exhibits remarkable resilience, tracking closely behind the full-participation baseline (at $\sim 52\%$) with minimal performance degradation. Interestingly, FedRep demonstrates a similar trend, maintaining stable performance ($\sim 50\%$) under partial participation. As analyzed previously, this occurs because random peer subsampling acts as a regularizer, allowing the continuously active target client to maintain a dominant influence over the shared global encoder without being overwhelmed by the collective heterogeneity of all 200 peers. Meanwhile, baselines such as FedDisco struggle significantly in this massive, highly heterogeneous environment, with its performance visibly degrading in later epochs.

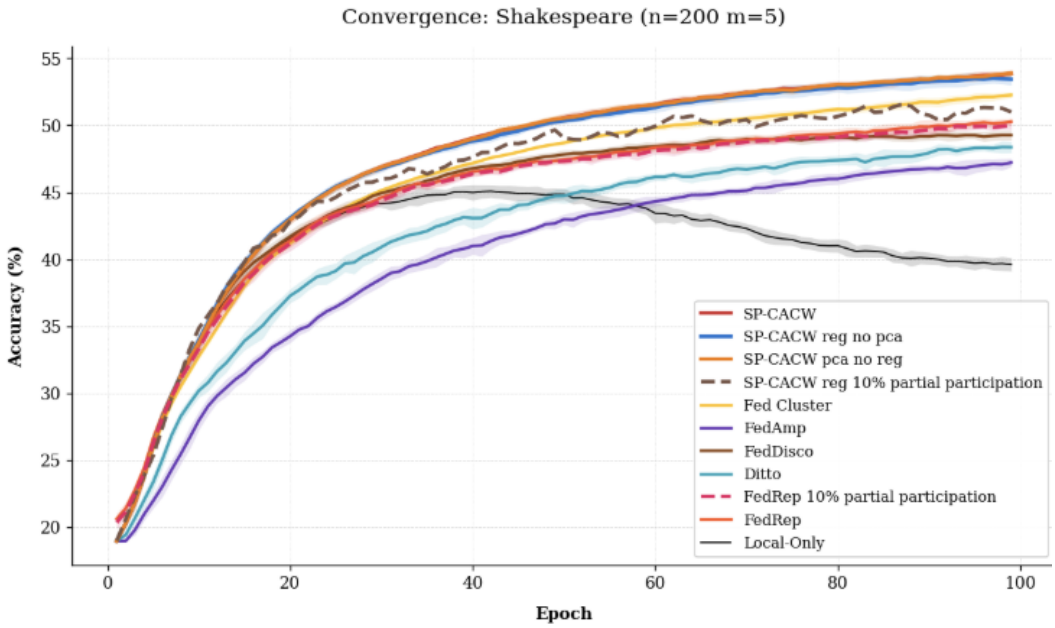

Figure 8: Test accuracy over 100 epochs on the LEAF Shakespeare dataset with $n = 200$ clients, comparing full and 10% partial participation.