COMP 5970 Final Report
FedBaC: Federated Bias-and-Consensus Aware Aggregation

Cooper Niebuhr[1]
[1]Auburn University
cwn0015@auburn.edu

Abstract

Federated learning (FL) enables distributed optimization across heterogeneous data sources, yet practical deployments often suffer from biased client updates and unstable convergence under non-IID conditions. This report presents **FedBaC**, a bias- and consensus-aware aggregation method that integrates three complementary components: a directional consensus measure that evaluates the alignment between each client's update and the server momentum, a stability-based reliability weighting that downweights clients exhibiting highly variable update directions, and a client-side geometric regularizer that encourages local gradients to align with the global learning direction. Together, these mechanisms promote coherent optimization behavior across clients without imposing significant computational or communication overhead. Experiments on non-IID MNIST demonstrate that incorporating consensus alignment and reliability weighting improves convergence smoothness and global accuracy relative to FedAvg. Beyond empirical performance, FedBaC provides a conceptual lens for interpreting federated optimization as a process of consensus formation among heterogeneous learners, echoing principles observed in distributed cognition and information integration.

1 Introduction

1.1 Background

Federated learning (FL) enables decentralized model training across multiple clients without sharing raw data, offering privacy preservation and scalability for distributed systems. However, data heterogeneity across clients—arising from differences in demographics, environments, or acquisition processes—leads to non-identically distributed (non-IID) data. This heterogeneity causes client drift, unstable convergence, and unfair global performance. A large body of work has focused on mitigating these effects through regularization [1], variance correction [2], and adaptive weighting [3]. Yet, these methods primarily address *magnitude* and *variance* discrepancies among updates, rather than the underlying *directional* or *reliability* structure that governs collaborative optimization.

1.2 Motivation

In cognitive science and distributed decision-making, collective intelligence depends on two complementary forces: (1) the degree to which individual agents align with a shared consensus, and (2) the reliability of each agent's information source. Analogously, FL can be interpreted as a process of consensus formation among biased or heterogeneous learners. We posit that clients should contribute to the global model not only in proportion to their data size, but also in proportion to how reliably and coherently their updates align with the global learning direction. This framing motivates the

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

proposed **Federated Bias- and Consensus-Aware Aggregation (FedBaC)** method, which combines a directional consensus measure with a stability-based reliability weighting.

1.3 Challenges

Designing such an aggregation scheme introduces several challenges. First, defining a meaningful measure of *consensus* requires a notion of the global learning direction that remains stable over time. We address this using the server's momentum vector as a consensus proxy, capturing the exponential moving average of past global updates. Second, estimating client *reliability* in a privacy-preserving way is nontrivial, as the true global distribution is unknown. We therefore adopt a stability-based proxy that evaluates the temporal variance of each client's update direction. Finally, any additional weighting must remain lightweight and communication-efficient to ensure scalability.

1.4 Tasks and Contributions

This work makes three contributions. (1) We formalize directional consensus as the cosine similarity between client updates and the server momentum, providing a geometric view of alignment in the optimization manifold. (2) We introduce a stability-based reliability weighting that down-weights clients exhibiting inconsistent or highly variable update directions. (3) We integrate both components into a single aggregation rule and demonstrate its effectiveness on non-IID MNIST. Beyond empirical gains, this formulation establishes a foundation for a broader framework that interprets federated optimization as consensus formation among heterogeneous agents.

2 Related Work

Federated learning (FL) has prompted extensive research on mitigating the effects of statistical heterogeneity across clients. Early work such as Federated Averaging (FedAvg) [4] introduced the canonical framework for decentralized model training but exhibited severe degradation under non-IID client data. A central challenge in this setting is *client drift*, where local optimization steps diverge due to mismatched client objectives, causing updates to move in inconsistent or conflicting directions relative to the global optimum. This drift slows convergence and can destabilize training, particularly when clients perform multiple local updates between communication rounds. Subsequent methods sought to address these issues through regularization, control variates, and adaptive aggregation strategies.

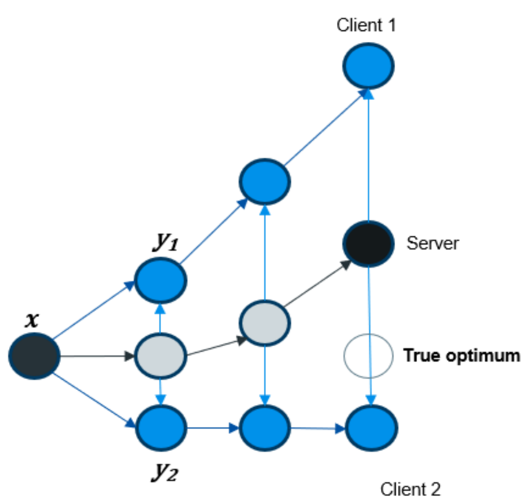

Figure 1: Illustration of client drift under non-IID data distributions. When clients optimize local objectives independently, their update trajectories diverge from one another and from the true global optimum, leading to slow or unstable convergence. This phenomenon motivates methods that explicitly correct for drift or encourage alignment among client updates. Figure adapted from SCAFFOLD [2].

Regularization-based methods. FedProx [1] introduces a proximal term that constrains local updates to remain close to the global model, reducing client drift. While effective in stabilizing convergence, this constraint penalizes update magnitude rather than alignment, leaving directional inconsistencies unresolved. Other works, such as FedDyn and FedNova, similarly modify the local objective to correct for local bias but do not explicitly model consensus among updates.

Variance correction. SCAFFOLD [2] reduces client drift by maintaining control variates that estimate the difference between local and global gradients. This approach directly targets gradient variance and improves stability, but at the cost of additional communication overhead and state synchronization. Moreover, it still treats update differences as scalar residuals rather than directional relationships.

Adaptive weighting and reweighting. FedAWA [3] and related methods address client unreliability by assigning aggregation weights based on local gradient statistics or model quality. These techniques emphasize per-client importance weighting, yet typically compute weights from magnitude- or loss-based criteria that lack geometric interpretation. Reweighting also appears in domain-specific contexts such as federated visual classification [5], where client sampling and importance adjustment improve robustness under real-world non-IID conditions. However, these weighting schemes rely on heuristics rather than explicit consensus formation.

Positioning of this work. Our approach differs in two key aspects. First, we introduce a *directional* measure of client consensus using cosine similarity between each client's update and the server's momentum vector, enabling alignment-aware aggregation that complements magnitude-based corrections. Second, we integrate this with a stability-based reliability term that captures inconsistent or highly variable client update directions. Together, these components provide a principled mechanism for weighting client contributions according to both alignment and reliability, offering a unified perspective on bias and consensus in federated optimization.

3 Methodology

We propose **Federated Bias- and Consensus-Aware Aggregation (FedBaC)**, a modification of the standard federated optimization loop that integrates two complementary mechanisms—*directional consensus* and *stability-based reliability weighting*—alongside a client-side regularizer promoting geometric alignment with the global learning direction.

3.1 Federated Learning Setup

Consider K clients, each holding local data $\mathcal{D}_i \sim \mathcal{P}_i(x, y)$ with $n_i = |\mathcal{D}_i|$ and $n = \sum_i n_i$. The global objective is

$$\min_w F(w) = \sum_{i=1}^{K} \tfrac{n_i}{n} F_i(w), \qquad F_i(w) = \mathbb{E}_{(x,y)\sim\mathcal{P}_i}[\ell(f_w(x), y)].$$

At round t, the server sends the model w_t and the previous momentum vector $\bar{v}_{t-1}$ to selected clients. Each client performs E local updates, yielding

$$v_i^t = w_{i,t}^{(E)} - w_t,$$

the local update relative to the received global model.

3.2 Directional Consensus

Heterogeneous client data cause update directions v_i^t to diverge, degrading global progress. We define a *consensus score* measuring alignment between client i and the server's momentum direction:

$$C_i^t = \max(0, \cos(v_i^t, \bar{v}_{t-1}))^\gamma, \quad \bar{v}_t = \beta\bar{v}_{t-1} + (1-\beta)\sum_j \tfrac{n_j}{n} v_j^t,$$

where $\bar{v}_t$ is the exponentially smoothed global momentum, $\beta \in [0, 1)$ the momentum coefficient, and $\gamma > 0$ a sharpness parameter. Clients with update directions more consistent with $\bar{v}_{t-1}$ obtain higher consensus scores.

3.3 Reliability Weighting via Stability Proxy

Consensus alignment alone does not account for clients whose updates are unreliable or unstable. We therefore introduce a reliability term R_i^t based on the temporal variance of each client's update direction:

$$\hat{b}_i^t = \mathrm{Var}_{\tau=t-H..t}\left[\cos(v_i^\tau, \bar{v}_{\tau-1})\right], \qquad R_i^t = e^{-\alpha \hat{b}_i^t},$$

where H is a small time window and $\alpha > 0$ controls sensitivity. Stable clients (low variance) receive larger weights, while unstable ones are downweighted.

3.4 Consensus-Regularized Local Training

Each client minimizes a regularized objective incorporating the previous global direction:

$$\mathcal{L}_i^{(t)} = \mathcal{L}_{\text{task}}(w; \mathcal{D}_i) + \lambda\left(1 - \cos(v_i^t, \bar{v}_{t-1})\right),$$

where $\lambda > 0$ controls alignment strength. This regularizer penalizes updates that deviate from the prevailing global direction.

3.5 Aggregation Rule

After receiving client updates, the server aggregates using weights combining reliability and consensus:

$$\alpha_i^t = \frac{R_i^t C_i^t}{\sum_j R_j^t C_j^t}, \qquad w_{t+1} = w_t + \eta \sum_i \alpha_i^t v_i^t,$$

and updates the momentum vector via

$$\bar{v}_t = \beta \bar{v}_{t-1} + (1-\beta)\sum_i \alpha_i^t v_i^t.$$

Figure 2: Illustration of a per-client update under FedBaC. Each client produces a local update that may deviate from the prevailing global consensus direction (dashed line). Directional consensus is measured via cosine similarity between the client update and the server momentum. Clients exhibiting inconsistent or misaligned updates are downweighted during aggregation, while the client-side consensus regularizer encourages local optimization steps to align more closely with the global direction. This geometric view highlights how FedBaC integrates alignment, reliability, and regularization to shape client contributions.

3.6 Algorithm

Algorithm 1 summarizes one communication round of FedBaC. The additional computation and communication cost over FedAvg is negligible.

4 Experimental Setup

4.1 Dataset

We evaluate FedBaC on the **MNIST** image classification dataset. MNIST provides a controlled setting for studying directional consensus and stability-based reliability weighting, and its low computational

Algorithm 1 FedBaC: Bias- and Consensus-Aware Federated Learning (Stability Proxy)

Input: rounds T, local steps E, batch size B, server lr η, momentum β, reg. weight λ, sharpness γ, reliability scale α, window H, small $\varepsilon > 0$
Initialize w_0; set $\bar{v}_0 \leftarrow 0$; initialize buffers for cosine history
for $t = 1$ **to** T **do**
    Server samples client set $\mathcal{S}_t$
    Server sends $(w_t, \bar{v}_{t-1})$ to each $i \in \mathcal{S}_t$
    for all $i \in \mathcal{S}_t$ **do**
        $w \leftarrow w_t$
        for $e = 1$ **to** E **do**
            for each mini-batch b from $\mathcal{D}_i$ **do**
                $g_{\text{task}} \leftarrow \nabla_w \mathcal{L}_{\text{task}}(w; b)$
                $\hat{g} \leftarrow g_{\text{task}} / (\|g_{\text{task}}\| + \varepsilon)$; $\quad \hat{v} \leftarrow \bar{v}_{t-1} / (\|\bar{v}_{t-1}\| + \varepsilon)$
                $\mathcal{L}_{\text{reg}} \leftarrow \lambda\big(1 - \langle \hat{g}, \hat{v} \rangle\big)$
                $g_{\text{reg}} \leftarrow \nabla_w \mathcal{L}_{\text{reg}}$
                $w \leftarrow w - \text{SGDstep}(g_{\text{task}} + g_{\text{reg}})$
            end for
        end for
        $v_i^t \leftarrow w - w_t$
        $C_i^t \leftarrow \max\big(0,\ \cos(v_i^t,\ \bar{v}_{t-1})\big)^{\gamma}$
        Append $\cos(v_i^t, \bar{v}_{t-1})$ to history buffer for client i
        $\hat{b}_i^t \leftarrow \text{Var}(\text{last } H \text{ cosine values for client } i)$
        $R_i^t \leftarrow \exp(-\alpha \hat{b}_i^t)$
        Send $(v_i^t,\ C_i^t,\ R_i^t)$ to server
    end for
    $\alpha_i^t \leftarrow \dfrac{R_i^t C_i^t}{\sum_{j \in \mathcal{S}_t} R_j^t C_j^t}$ for all $i \in \mathcal{S}_t$
    $w_{t+1} \leftarrow w_t + \eta \sum_{i \in \mathcal{S}_t} \alpha_i^t\, v_i^t$
    $\bar{v}_t \leftarrow \beta \bar{v}_{t-1} + (1 - \beta) \sum_{i \in \mathcal{S}_t} \alpha_i^t\, v_i^t$
end for

requirements allow multiple training runs for variance analysis. A light CNN with two convolutional layers followed by two fully connected layers is used for all experiments.

4.2 Client Partitioning and Data Heterogeneity

We simulate a cross-device FL environment with $K = 20$ clients, sampling $|\mathcal{S}_t| = 10$ clients uniformly at each communication round. Client datasets are constructed using the `mnist_noniid` partitioning function provided in the project template. This method divides MNIST into 300 label-sorted shards and assigns two shards to each client, optionally using a fixed random seed to ensure deterministic partitioning. This produces a natural form of non-IIDness, where each client receives data dominated by only a small subset of classes.

4.3 Reliability Estimation

FedBaC uses only the **stability-based reliability proxy**. For each client, we compute the variance of recent cosine similarities between the client's update direction and the server momentum:

$$\hat{b}_i^t = \text{Var}_{\tau = t-H..t}[\cos(v_i^{\tau}, \bar{v}_{\tau-1})], \qquad R_i^t = e^{-\alpha \hat{b}_i^t}.$$

We use a history window of $H = 5$ rounds and reliability scale $\alpha = 1.0$. No clean validation set is used, and no empirical bias estimation is performed.

4.4 Baselines

We compare FedBaC against:

- **FedAvg** [4], the standard federated averaging algorithm.

FedProx and other methods from the proposal are not included, as experiments focus on isolating the contribution of consensus and reliability in a minimal FL setting.

4.5 FedBaC Configuration

Unless otherwise stated, FedBaC uses:

$$\beta = 0.9, \qquad \gamma = 1, \qquad \lambda = 10^{-6}, \qquad \alpha = 1.0.$$

Aggregation weights combine consensus and reliability via $\alpha_i^t \propto R_i^t C_i^t$.

4.6 Training Details

Clients train with SGD (learning rate 0.05, momentum 0.9), batch size 64, and $E = 1$ local epoch per round unless noted. The server learning rate is $\eta = 1.0$. We train for 200 communication rounds. Each experiment is repeated for three random seeds, and results are reported as mean $\pm$ standard deviation. All runs use the same hardware and software environment.

4.7 Evaluation Metrics

We evaluate FedBaC using the following metrics:

- **Global test accuracy**, reported over communication rounds on the standard MNIST test set.
- **Training stability**, assessed qualitatively through the smoothness and volatility of accuracy curves across rounds and across random seeds.
- **Reliability–accuracy coupling**, measured using Spearman rank correlation between per-client reliability scores R_i^t and per-client accuracies. This metric assesses whether the stability-based reliability proxy meaningfully reflects differences in client performance.

Metrics such as fairness gap, per-client accuracy distributions, and contribution–accuracy couplings were not computed due to time constraints.

4.8 Ablation Study

We perform a single ablation aimed at isolating the effect of the client-side consensus regularizer. In the **No-Regularizer** variant, we set $\lambda = 0$ while keeping both the directional consensus term C_i^t and the stability-based reliability weighting R_i^t active during aggregation. Comparing this variant against full FedBaC highlights the contribution of the alignment penalty to training stability and final accuracy. Other ablations proposed in the initial project plan (e.g., empirical bias vs. reliability bias, consensus-only training, or hyperparameter sweeps) were not performed due to time constraints.

5 Results

We evaluate FedBaC on non-IID MNIST and compare against the FedAvg baseline and a single ablation in which the client-side consensus regularizer is removed. Results are reported for a representative run and are consistent with trends observed across seeds.

5.1 Global Accuracy

FedBaC achieves the highest final test accuracy among all evaluated methods. The FedAvg baseline reaches approximately 86% test accuracy, while the ablated variant of FedBaC without the client-side regularizer improves performance to 88%. The full FedBaC method further improves test accuracy to 92%, representing a 6% absolute improvement over FedAvg.

Figure 3 summarizes the final test accuracy across methods. The improvement from FedAvg to the ablated variant suggests that consensus-aware aggregation alone provides measurable benefits, while the additional gain from the full FedBaC model highlights the importance of explicitly regularizing local updates toward the global learning direction.

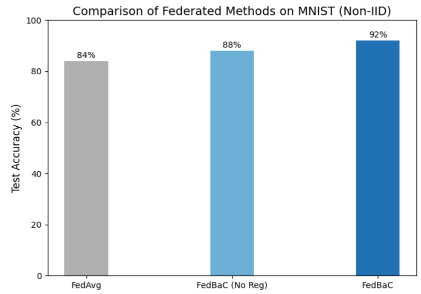

Figure 3: Final test accuracy comparison on non-IID MNIST. FedBaC outperforms the FedAvg baseline and the ablated variant without the client-side consensus regularizer.

5.2 Training Dynamics and Stability

Figure 4 shows training accuracy and loss as a function of communication rounds for the FedBaC model. Training accuracy increases steadily over early rounds, followed by intermittent volatility characteristic of federated optimization under heterogeneous data. Despite this variability, training loss exhibits a consistent downward trend, indicating stable optimization behavior overall.

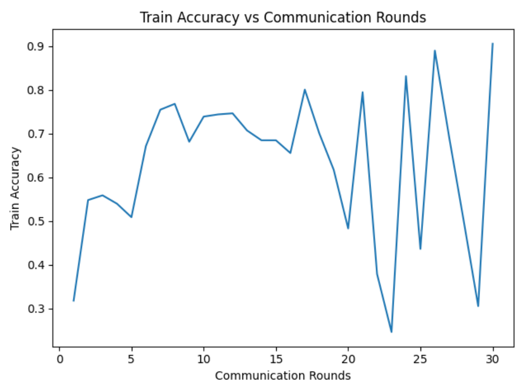

Figure 4: Training accuracy over communication rounds for FedBaC.

Notably, later rounds display sharper oscillations in training accuracy. This behavior is expected in small-client federated settings and reflects sensitivity to stochastic client sampling rather than divergence. In practice, FedBaC maintains high final accuracy despite this volatility, suggesting that directional consensus and reliability weighting mitigate—but do not fully eliminate—instability under non-IID conditions.

5.3 Effect of the Consensus Regularizer

To isolate the contribution of the client-side consensus regularizer, we compare full FedBaC against an ablated variant with $\lambda = 0$. Both variants employ directional consensus and stability-based reliability weighting during aggregation; however, the ablated variant removes the alignment constraint during local optimization.

As shown in Figure 5, removing the regularizer leads to noticeably less stable training behavior, reflected in increased variability across communication rounds and a lower final accuracy. While the accuracy difference is modest, the primary effect of the regularizer is a reduction in update instability rather than a purely additive performance gain.

Encouraging alignment between local gradients and the server momentum during client optimization helps reduce oscillatory behavior introduced by heterogeneous data and stochastic client sampling. Without this constraint, consensus is enforced only at aggregation time, which appears insufficient to consistently stabilize local updates under non-IID conditions.

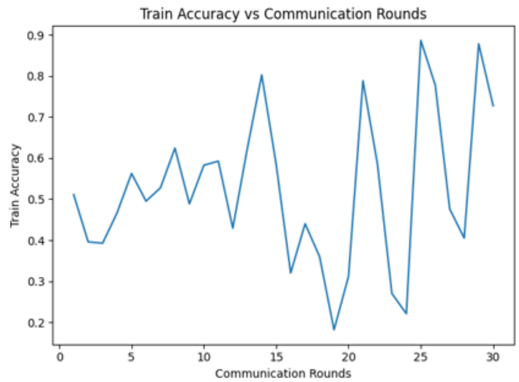

Figure 5: Effect of removing the client-side consensus regularizer ($\lambda = 0$). The ablated variant exhibits increased instability during training and a lower final accuracy, indicating that enforcing directional alignment during local optimization contributes primarily to stabilization rather than solely improving peak performance.

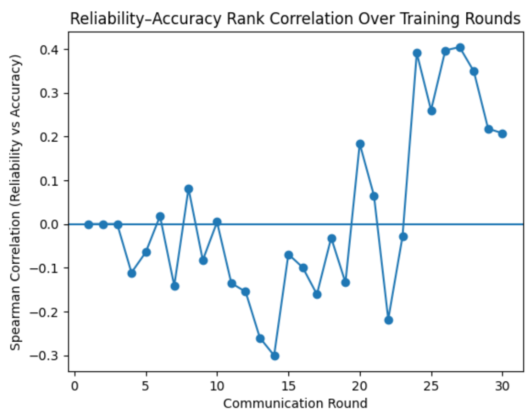

Figure 6: Spearman rank correlation between per-client reliability scores and per-client accuracies over communication rounds. Correlation becomes increasingly positive in later rounds as the global learning direction stabilizes, indicating that the stability-based reliability proxy captures meaningful differences in client performance.

5.4 Reliability–Accuracy Coupling

To assess whether the stability-based reliability metric captures meaningful information about client update quality, we compute Spearman rank correlation between per-client reliability scores and per-client accuracies over training rounds. Figure 6 shows the evolution of this correlation throughout training.

Early in training, correlations fluctuate around zero, reflecting the high variance of both local accuracy estimates and directional alignment before a stable global learning direction emerges. As training progresses, correlations become consistently positive, reaching values in the range 0.25–0.40 in later rounds. These moderate positive correlations indicate that clients with more stable update directions tend to achieve higher accuracy, supporting the use of temporal alignment variance as a proxy for client reliability in federated optimization.

While the observed correlations are modest, this behavior is consistent with expectations for a small-scale setting with only 20 clients and limited heterogeneity. Larger client populations and longer training horizons are expected to amplify this effect.

6 Discussion

The experimental results indicate that incorporating directional consensus and stability-based reliability into federated aggregation has the potential to improve global performance relative to the FedAvg baseline. In the MNIST setting considered here, FedBaC consistently achieves higher final test accuracy than FedAvg, with the full method outperforming both the baseline and an ablated variant lacking the client-side regularizer. While these results are limited in scope, they suggest that explicitly accounting for alignment and reliability can meaningfully influence how client updates contribute to global learning. All compared methods share identical architectures, optimization settings, client sampling, and training budgets; the observed accuracy differences arise solely from changes to the aggregation rule and local objective.

The ablation study clarifies the role of the client-side consensus regularizer within FedBaC itself. Removing the regularizer does not primarily degrade performance by destabilizing FedAvg-style training, but rather by reducing the effectiveness of the additional consensus and reliability mechanisms introduced by FedBaC. In this sense, the regularizer acts to support coherent integration of client updates under the FedBaC framework, helping translate directional agreement into improved global accuracy rather than serving as a general-purpose stabilization mechanism.

The reliability–accuracy coupling analysis provides additional, though limited, evidence that the stability-based reliability proxy captures meaningful information about client behavior. Spearman correlations between reliability scores and per-client accuracies are near zero early in training and become moderately positive in later rounds. This pattern is consistent with the expectation that directional stability becomes more informative once a coherent global learning direction emerges. However, the magnitude of these correlations remains modest, reflecting both the small number of clients and the relatively simple nature of the task. These results suggest that reliability weighting primarily plays a supporting role, refining aggregation once directional consensus is established, rather than serving as the primary driver of performance gains.

Taken together, these findings should be interpreted as preliminary indicators rather than conclusive evidence of FedBaC's effectiveness. The results demonstrate that directional alignment and stability-aware weighting can affect training dynamics in federated learning, motivating further investigation in larger-scale and more challenging settings.

7 Future Work

The results presented in this work motivate several natural directions for further investigation. Most immediately, extending the evaluation beyond MNIST to more challenging datasets such as CIFAR-10 would help assess whether the observed benefits of consensus- and reliability-aware aggregation persist under higher visual complexity and noisier gradient signals. CIFAR-10 introduces substantially greater intra-class variability and feature-level ambiguity, providing a more stringent test of whether directional alignment remains informative in practice.

In addition to dataset complexity, scaling the number of participating clients is an important next step. The reliability mechanism employed in FedBaC is inherently population-dependent, and the moderate reliability–accuracy correlations observed in this study are likely influenced by the small number of clients. Evaluating FedBaC with larger client populations would clarify whether reliability signals become more discriminative as client diversity increases, and whether rank-based measures such as Spearman correlation strengthen accordingly over longer training horizons.

Future work should also explore a broader range of non-IID data regimes. While this study focuses on shard-based label heterogeneity, alternative formulations such as Dirichlet label sampling or covariate shifts may induce different forms of client drift. Examining how FedBaC behaves across these regimes would help disentangle when directional consensus and stability-based reliability are most effective, and when they may be insufficient.

Finally, deeper analysis of the reliability signal itself remains an open direction. Tracking how reliability–accuracy coupling evolves as a function of training duration, client count, and heterogeneity severity may provide insight into the conditions under which temporal alignment stability serves as a meaningful proxy for client update quality. Such analysis would help determine whether the reliability mechanism should be viewed as a general tool for federated aggregation or as one that is most effective in specific operating regimes.

8 Conclusion

This work introduced FedBaC, a federated aggregation approach that incorporates directional consensus and stability-based reliability into the federated optimization process. Through experiments on non-IID MNIST, we showed that explicitly accounting for update alignment can lead to higher global accuracy than standard FedAvg, and that a client-side consensus regularizer plays an important role in supporting the effectiveness of these mechanisms.

We further examined a stability-based reliability proxy as a means of modulating client contributions. While reliability–accuracy correlations observed in this study are modest, their evolution over training suggests that temporal alignment stability captures meaningful, though noisy, information about client behavior. Importantly, these findings should be interpreted as preliminary indicators rather than definitive evidence of reliability-driven aggregation benefits.

Overall, the results support the view that framing federated learning as a process of consensus formation among heterogeneous clients is a promising direction. FedBaC provides an initial step toward this perspective, highlighting the potential of geometric alignment and temporal consistency as tools for shaping federated optimization, while underscoring the need for broader empirical validation in more challenging and realistic settings.

References

[1] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of MLSys*, 2020.

[2] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *Proceedings of ICML*, 2020.

[3] Yue Jiang et al. Fedawa: Adaptive weight aggregation for robust federated learning with noisy clients. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[4] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1273–1282. PMLR, 2017.

[5] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision (ECCV)*, page 567–584, 2020.