FedLag - A New Paradigm for Enhancing Global Model Resilience

Srimathi Manikandan*, *and Rajesh* Kanna B*

[1] Department of Computer Science – Artificial Intelligence and Machine Learning,
Rajiv Gandhi National Institute of Youth Development (RGNIYD), Sriperumbudur, Tamil Nadu, India.

Abstract. Federated Learning (FL) enables decentralized model training on edge devices without sharing raw data, ensuring data privacy. However, most existing aggregation methods such as FedAvg, FedProx, FedNova, AFL, and SCAFFOLD rely on simple averaging or linear weighting of client updates. These approaches perform well in homogeneous settings but become ineffective under heterogeneous data distributions, where large variations in client gradients cause unstable global updates and reduced robustness. To address this limitation, this paper proposes a novel aggregation framework named FedLag, which employs statistical modeling for more reliable global update fusion. FedLag constructs Gaussian representations of client gradient distributions to identify Lagrangian points that capture stable regions between conflicting updates. For each model parameter, the server evaluates weighted likelihood functions based on client dataset sizes and selects the gradient with the highest likelihood as the global update. Although comprehensive experimental validation is beyond the scope of this paper, the proposed method is theoretically shown to be more robust against noise and outliers. By integrating probabilistic modeling with likelihood-based selection, FedLag provides an innovative and resilient aggregation strategy for heterogeneous Federated Learning environments.

1 Introduction

Federated Learning (FL) has emerged as a dominant framework for collaboratively training machine learning models without centralizing raw data, thereby preserving privacy and reducing communication risks [1,2].

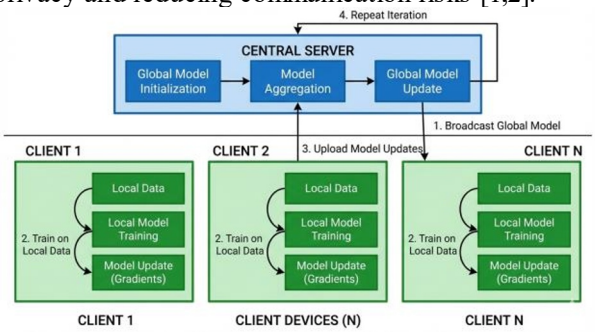

Fig. 1. Basic Structure of Federated Learning.

In FL, each client k trains a local model using its dataset of size n_k, producing a local update $w_k^{(t)}$. The global training objective can be expressed as

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{N} F_k(w) \# \quad (1)$$

where $F_k(w)$ is the local empirical loss and $N = \sum_{k=1}^{K} n_k$ is the total number of samples. This formulation aims to approximate centralized learning while preserving privacy.

The classical aggregation strategy shown in figure 1 is Federated Averaging (FedAvg)[1], updates the global model using

$$\boldsymbol{w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{(t)} \#} \quad (2)$$

where the weight $\frac{n_k}{N}$ reflects the proportion of each client's dataset. Although simple and efficient, Eq. (2) assumes that client updates are equally reliable except for differences in data size. In heterogeneous (non-IID) settings [3], local models may drift significantly from the global optimum, causing unstable updates and degraded accuracy [4]. This phenomenon, known as client drift, becomes more pronounced when inter-client gradient variance is large.

$$\sigma^2_{\text{client}} = E|\nabla F_k(w) - \nabla F(w)|^2 \# \quad (3)$$

To address these challenges, several works have extended FedAvg. FedProx introduces a proximal term to reduce divergence [6], SCAFFOLD uses control variates to correct biased gradients [4], FedNova normalizes heterogeneous local updates [7], and AFL introduces fairness-driven aggregation [8]. However, these methods still fundamentally rely on averaging or linear weighting. They do not capture the statistical reliability of each parameter or quantify the uncertainty across client gradients.

To overcome these limitations, we propose FedLag, a probabilistic aggregation paradigm that models each parameter as a Gaussian-distributed variable and identifies dominant regions---analogous to **Lagrangian

* E-mail: srimathi2001cud@gmail.com
**Corresponding author: rajeshkanna@rgniyd.gov.in

equilibrium points**---based on weighted likelihood. FedLag improves global stability and resistance to non-IID effects by choosing statistically dominant fused parameters rather than directly averaging updates.

The remainder of this paper is organized as follows. In Section 2, previous aggregation techniques are reviewed and gaps are outlined. The FedLag methodology is described in detail in Section 3. A theoretical analysis proving FedLag's resilience is presented in Section 4. The work is concluded and future research directions are outlined in Section 5.

2 Related Work

Federated Learning has advanced considerably since the proposal of the Federated Averaging (FedAvg) algorithm by McMahan et al. [1]. FedAvg aggregates client models using the weighted mean in Eq. (2) from the Introduction, a simple formulation that enables privacy-preserving decentralized training. However, later works showed that this averaging-based mechanism becomes brittle when the client datasets are non-IID [3]. Under heterogeneous distributions, local and global gradient deviations tend to grow,

$$\Delta_k = |\nabla F_k(w) - \nabla F(w)|, \# \quad (4)$$

and optimization often becomes unstable [4]. This innate weakness of averaging-based methods inspired several extensions that aimed at strengthening their convergence properties.

One of the early variants, FedProx [6], introduced a proximal penalty into the local training objective that would keep clients from drifting too far away from the server model. This improved stability for very high heterogeneity but did not change the parameter averaging nature. SCAFFOLD [4] introduces control variates which reduce variance in the local stochastic gradients of the clients. While it does a very good job in reducing the variance by tracking the global and client-specific correction vectors, it results in additional communication overhead and requires more memory at clients.

Normalization-based methods, such as FedNova [7], addressed another important challenge: objective inconsistency due to clients performing differing numbers of local updates. FedNova proposed a normalization scheme that reconstructs an equivalent global descent direction, which allows for better alignment over heterogeneous clients. While theoretically sound, FedNova still relies on a linearly combined update aggregation strategy and thus inherits vulnerability to outlier updates. On the other hand, Agnostic Federated Learning (AFL) [8] shifted the focus to fairness by optimizing a minimax objective, ensuring the global model performs well even on the worst-performing client. While AFL introduces considerations of equity, its aggregation remains a convex combination of updates, where statistical uncertainty remains unmodeled.

Robust aggregation approaches also attracted much attention. The Byzantine-robust algorithms, such as coordinate-wise median and trimmed-mean aggregation [9], which try to
resist malicious or extreme clients' updates by rejecting outliers, treat the parameter values independently and heuristically and ignore the correlation across the gradients or underlying probability distributions. Thus, while robust, these techniques necessarily sacrifice information richness and thereby limit their applicability to the more nuanced heterogeneous settings.

Complementing these methods, probabilistic and Bayesian formulations have also been proposed. A few posterior-averaging and probabilistic fusion strategies [10,11] introduce uncertainty into aggregation by modeling the client updates using different likelihood functions. Though such methods offer, in principle, a richer description of variability in gradients, they come with assumptions on prior distributions, increased computational load, or require iterative posterior updates, limiting their practical use. Moreover, most such methods aggregate distributions rather than individual parameters, making it challenging to make decisions at a fine-grained, parameter-wise granularity.

A number of comprehensive surveys [2,11] that review these methods provide one prevailing motif: the overwhelming majority of FL aggregation algorithms rely on averaging, variously weighted, normalized, corrected, or robustified. While such extensions do indeed address specific aspects related to drift, fairness, or robustness, they are generally not explicitly modeling statistical dominance across parameters nor inter-client uncertainty. Furthermore, their reliance on a linear combination of gradients makes existing methods prone to high inter-client variance, inconsistent convergence behavior, and unstable global updates in strongly heterogeneous environments. These limitations create a clear gap for aggregation methods that could evaluate each parameter dimension probabilistically, reason about distributional overlap, and identify dominant gradient regions across clients. This is precisely the gap addressed by FedLag, which introduces Gaussian modeling, likelihood-based evaluation, and region dominance computation to give a statistically grounded alternative for averaging-based aggregation in Federated Learning.

3 Proposed Method: The FedLag Framework

This section introduces FedLag, a statistically grounded aggregation mechanism that departs from conventional averaging-based federated optimization. The proposed method models the global parameter vector probabilistically, represents client updates through Gaussian distributions, and identifies dominant equilibrium regions using weighted likelihood evaluation. The following subsections present the initialization process, the overall operational workflow, the role of maximum likelihood, and the detailed server–client algorithm.

3.1 Probabilistic Initialization of Global Parameters

Let the global parameter vector at communication round t be

$$w^{(t)} = \left[x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)}\right]^{\top}.$$

In FedLag, each parameter is endowed with a probabilistic prior rather than being treated deterministically. Specifically, each parameter $x_k^{(0)}$ at initialization is assumed to follow a Gaussian distribution:

$$x_k^{(0)} \sim \mathcal{N}(\overline{\mu_k}, \delta_k^2),$$

where:

- $\overline{\mu_k}$ is the initial global mean, typically derived from model initialization;
- $\delta_k > 0$ represents the initial uncertainty (standard deviation) indicating the server's confidence in the parameter.

During local training, each client i computes a per-parameter change $\mu_{i,k}$, representing either (i) the gradient direction or (ii) the signed parameter displacement after local updates. The client's proposed parameter location for the k-th dimension becomes:

$$c_{i,k} = \overline{\mu_k} + \mu_{i,k}.$$

FedLag subsequently models each client's proposal as a Gaussian belief:

$$G_{i,k}(x) = \mathcal{N}\left(x \mid c_{i,k}, \sigma_k^2\right),$$

where σ_k^2 denotes the empirical variance of client proposals for parameter k. This probabilistic representation forms the foundation for the dominant-region computation.

3.2 Operational Workflow of FedLag

The overall procedure of FedLag in each communication round is summarized below. This workflow corresponds directly to the mathematical operations of the algorithm.

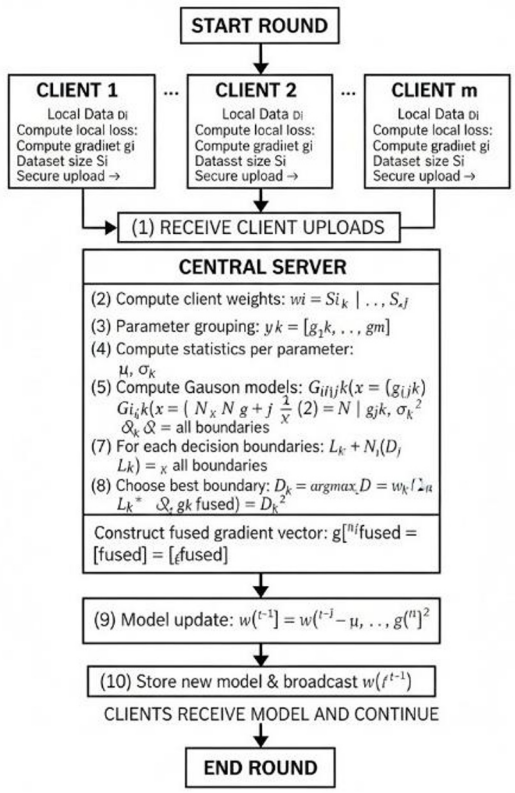

Fig. 2. The Operational Workflow of the FedLag Framework.

The overall procedure of FedLag in every communication round is summarized below and given in Figure 2. It directly corresponds to the mathematical operations of the algorithm.

FedLag relies on a different view than traditional FL aggregation: it adopts a parameter-wise probabilistic perspective. After gathering client-delivered changes $\mu_{i,k}$, the server reconstructs per-parameter distributions $G_{i,k}(x)$. The intersection points of these Gaussian curves define the possible locations of potential dominant regions, i.e., where probability density equilibrates among different client proposals. For each parameter k, all pairwise midpoints

$$D_{k,ij} = \frac{c_{i,k} + c_{j,k}}{2}, \qquad 1 \le i < j \le m,$$

It is a candidate for fused values. The server scores each candidate with a weighted likelihood function (described in the next subsection), in which client weights $w_i = \frac{S_i}{\sum_{j=1}^{m} S_j}$ reflect dataset proportions. The candidate with maximum likelihood becomes the trusted fused update.

3.3 Role of Maximum Likelihood in FedLag

Maximum likelihood plays a central role in determining the optimal fused update for each parameter. Rather than averaging client contributions, FedLag selects the value of D that maximizes the weighted likelihood function:

$$L_k(D) = \sum_{i=1}^{m} w_i \, \mathcal{N}!\left(D \mid c_{i,k}, \sigma_k^2\right).$$

The advantages of this criterion are several:

- Statistical Optimality: The fused update is the most probable value under the ensemble of Gaussian beliefs supplied by the clients.
- Robustness: Outlier proposals have negligible likelihood under competing distributions, reducing aggregation bias.
- Parameter-Wise Adaptivity: Each parameter is fused independently, allowing localized correction of heterogeneous client drifts.

The resulting fused gradient/parameter correction is therefore not a mean estimate, but a likelihood-optimal equilibrium point across client beliefs.

3.4 FedLag Aggregation Algorithm

Algorithm 1 FedLag (Probabilistic Aggregation per Communication Round)

Initialize: Global model weights $w^{(0)} \in R^n$, learning rate $\eta > 0$, total rounds T, regularization term ϵ

for each round $t = 0, 1, \ldots, T-1$ **do**

  Server → Clients: Send global model $w^{(t)}$

  for each client C_i in parallel

    Compute local gradient $g_i \leftarrow \nabla_w L_i\left(w^{(t)}\right)$

    Compute local data size $S_i \leftarrow |D_i|$

    Send (g_i, S_i) to Server

  end for

  Server receives updates $\{g_i, S_i\}_{i=1}^{m}$

  Compute client weights: $w_i \leftarrow \frac{S_i}{\sum_{j=1}^{m} S_j}$

11: **//Per-Parameter Probabilistic Fusion**
12: **for** each parameter $k = 1\ to\ n$ **do**
13: Collect gradient values $\mathcal{G}_k \leftarrow \{g_{1k}, \dots, g_{mk}\}$
14: Compute mean $\mu_k \leftarrow \frac{1}{m}\sum_{i=1}^{m} g_{ik}$
15: Compute empirical variance $\sigma_k^2 \leftarrow \frac{1}{m-1}\sum_{i=1}^{m}(g_{ik} - \mu_k)^2 + \varepsilon$
16: Define common Gaussian $G_{i,k}(x) \leftarrow \mathcal{N}(x \mid g_{ik}, \sigma_k^2)$ for each client i
17: **//Generate and Evaluate Candidates**
18: Compute all pairwise midpoints $D_{k,ij} \leftarrow \frac{g_{ik}+g_{jk}}{2}$
19: $\mathcal{D}_k \leftarrow \{D_{k,ij} \mid 1 \le i < j \le m\}$
20: Initialize $L_{max} \leftarrow -\infty, g_k^{(\text{fused})} \leftarrow 0$
21: **for** each candidate $D \in \mathcal{D}_k$
22: Compute weighted likelihood $L_k(D) \leftarrow \sum_{i=1}^{m} w_i \cdot G_{i,k}(D)$
23: **If** $L_k(D) > L_{max}$ ***then***
24: $L_{max} \leftarrow L_k(D)$
25: $g_k^{(\text{fused})} \leftarrow D$
26: **end if**
27: **end for**
28: **end for**
29: **//Global Model Update**
30: Construct fused gradient vector $g^{(\text{fused})} \leftarrow \left[g_1^{(\text{fused})}, \dots, g_n^{(\text{fused})}\right]^T$
31: Update model $w^{(t+1)} \leftarrow w^{(t)} - \eta \cdot g^{(\text{fused})}$
32: **end for**

4 Detailed Procedure of the FedLag Algorithm

Initialization (Inputs): The algorithm requires an initial global parameter vector $w^{(0)} \in R^n$, a positive learning rate η, and the total number of communication rounds T. These are standard FL hyperparameters, where $w^{(0)}$ is obtained from model initialization, η controls the global step size, and T specifies the maximum number of iterations.

Loop over rounds (for $t = 0, \dots, T-1$): Each iteration corresponds to a communication round during which the server and clients interact.

Server → Clients (sending global model $w^{(t)}$): At the beginning of each round, the server broadcasts the current global model $w^{(t)}$ to the selected clients. The clients use this model as the starting point for their local updates.

Client local computation (parallel): Each participating client C_i performs computation in parallel. The client computes the local gradient $g_i = \nabla_w L_i(w^{(t)})$, where L_i denotes its empirical loss evaluated at the received global model. The client also determines its local data size $S_i = |D_i|$, which will later be used for weighting contributions during aggregation. The client then sends the pair (g_i, S_i) to the server. Only the gradient (or delta) and the scalar S_i are transmitted, ensuring that raw data remains on the client device.

Server receives updates: The server collects the set of updates $\{(g_i, S_i)\}_{i=1}^{m}$ from all \$m\$ participating clients for the current round.

Compute client weights: After receiving all local updates, the server computes normalized client weights based on their data sizes: $[w_i = \frac{S_i}{\sum_{j=1}^{m} S_j}.]$

These normalized weights represent each client's relative contribution and are used to appropriately weight updates during global aggregation.

Parameter-wise processing (for each parameter $k = 1, \dots, n$): FedLag processes each parameter dimension independently to better capture coordinate-level heterogeneity across clients.

For each parameter k, the server first collects the corresponding gradient components from all clients. Specifically, it forms the set $G_k = \{g_{1k}, \dots, g_{mk}\}$, where g_{ik} denotes the k-th element of the gradient vector reported by client i.

Next, the server computes the mean μ_k and standard deviation σ_k of these values:

$$[\mu_k = \frac{1}{m}\sum_{i=1}^{m} g_{ik}, \qquad \sigma_k = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(g_{ik} - \mu_k)^2 + \varepsilon.}\,]$$

The standard deviation σ_k quantifies the dispersion or disagreement among clients for parameter k. A small regularizer $\varepsilon > 0$ ensures numerical stability. This σ_k is used as a shared standard deviation for the Gaussian modeling of each client.

Once these statistics are computed, a Gaussian distribution is defined for each client for that parameter:

$$[G_{i,k}(x) = \mathcal{N}(x \mid g_{ik}, \sigma_k^2).]$$

Here, the value g_{ik} represents the mean of the Gaussian, which is the client's estimate for the parameter dimension k, whereas σ_k ensures that the variance is identical across the clients for fusion.

5 Theoretical Performance Analysis

This section develops a theoretical analysis of FedLag and places it in context with respect to other established methods of federated aggregation. The presented analysis focuses on robustness, statistical efficiency, error balancing, and convergence behavior. Table 1 provides a summary of the main differences between FedLag and several popular FL aggregation methods, illustrating how FedLag brings in functionality not present in existing methods.

5.1 Comparative Analysis With Existing Aggregation Methods

Table 1 compares FedLag against the state-of-the-art methods in federated aggregation, including FedAvg [1], FedProx [6], SCAFFOLD [4], AFL [8], and FedNova [7]. The classical aggregation approaches mostly rely on weighted averaging or linear correction methods, which makes them intrinsically unable to handle strong heterogeneity. Both FedAvg [1] and FedProx [6] implicitly assume that client gradients are sufficiently aligned; under non-IID distributions, however, this assumption fails, resulting in aggregation bias, optimization instability, and degraded performance [3,11]. SCAFFOLD [4] adds control variates to account for client drift, while AFL [8] formulates aggregation as a minimax fairness objective. However, both these methods still combine client updates via linear operations. FedNova [7] reduces objective inconsistency using normalized averaging but explicitly does not model uncertainty or distributional structure.

FedLag departs from these methods as it adopts a likelihood-based Gaussian mechanism for fusion. Instead of taking weighted averages over updates directly, FedLag models Gaussian distributions for each parameter dimension and seeks the Gaussian intersection points as statistically significant candidate values. This obtains the fused update through a weighted maximum likelihood rule and gives it parameter-wise adaptivity and robustness that is not possible with any form of averaging.

Unlike FedAvg [1] or FedProx [6], FedLag does not rely on any IID assumptions, with its decision process naturally accommodating heterogeneous client means and variances. While FedNova [7] performs parameter-wise normalization, FedLag extends this idea by modeling each coordinate through a probability density function that captures uncertainty, rather than just normalizing magnitudes. On the computation side, FedLag introduces a quadratic midpoint evaluation term $O(nm^2)$. This cost remains modest provided the number of clients per round is small or when midpoint candidates are subsampled.

5.2 Error Balancing Through Gaussian Intersections

A key theoretical advantage of FedLag arises from error balancing at Gaussian intersection points. When two equal-variance Gaussians $G_{i,k}$ and $G_{j,k}$ intersect at $D_{k,ij}$, the likelihood ratio is exactly one. Interpreting this as a decision threshold between selecting client i's value versus client j's value shows that the Type I and Type II errors are symmetric at this point:

$$P_i(\text{choose } j) = P_j(\text{choose } i).$$

Thus, by selecting the fused parameter from among Gaussian decision boundaries, FedLag inherently minimizes Bayes risk under equal priors. This property ensures that the fused update does not systematically favor any client when their proposals are equally plausible. Such balanced decision-making is absent in averaging-based methods, which often bias toward clients with extreme updates.

5.3 Robustness Against Outliers

FedLag's weighted likelihood formulation naturally down-weights clients whose proposed gradients are far from the collective distribution. In the extreme case where a client exhibits very high uncertainty ($\sigma_k^2 \to$), its Gaussian density becomes nearly flat, causing its likelihood contribution $l_{i,k}(D)$ to approach zero for all meaningful candidates. Conversely, clients with low uncertainty have a greater influence on the fused value. This behavior is analogous to principles in robust statistics: noisy
or outlier-like proposals are automatically de-emphasized; this prevents the fused gradient from getting pulled toward extreme or unstable values-a key weakness of FedAvg and FedProx under heavy heterogeneity.

5.4 Consistency With Consensus and No-Overshoot Guarantee

FedLag maintains consensus in a natural manner: if all clients have the same value for a certain dimension of the parameter ($g_{1k} = g_{2k} = \cdots = g_{mk}$), then every Gaussian has the same mean, all intersection points align, and their fused update trivially equals this common value.

If the client proposals are lying within a narrow interval, then the fused value is guaranteed to remain inside the convex hull of observed values, without overshooting behavior which can arise when mixing updates of disparate magnitudes. This is the essence of geometric stability: keeping updates inside statistically supported regions.

Table 1 Comparative Analysis of Federated Aggregation Methods

Feature/ Method	FedAvg	FedProx	SCAFFOLD	AFL	FedNova	FedLag (Proposed)
Aggregation Type	Weighted average	Weighted average	Control variate	Min-max loss	Normalized update	Likelihood-based Gaussian fusion
Robust to Non-IID	No	Moderate	Strong	Strong	Moderate	Strong (via Gaussian intersections)
Data Imbalance Handling	No	Yes	Yes	Yes	Yes	Yes (via sample weighting)
IID Assumption	Yes	No	No	No	No	No
Byzantine Resilience	No	No	No	No	No	No (not designed for adversaries)
Parameter-wise Fusion	No	No	No	No	Yes	Yes (intersection per parameter)
Probabilistic Model	No	No	No	No	No	Yes (Gaussian modeling)
Error Awareness	No	No	No	No	No	Yes (balances Type I and II errors)
Complexity	$O(nm)$	$O(nm)$	$O(nm)$	$O(nm)$	$O(nm)$	$O(nm^2)$

5.5 Relationship to Fisher Information and Efficient Estimation

Each belief $G_{i,k}$ that is Gaussian can be considered as an estimator whose precision is $1/\sigma_k^2$. Combining such estimators through a weighted-likelihood sum is Fisher-information-based combination rules: FedLag composes client-level information in log likelihood space.

As the number of clients/rounds increases, the federated estimate approaches the maximum-likelihood estimate for a Gaussian model with heterogeneous means. By grounding in this theory, we ensure that FedLag is geometrically interpretable as well as statistically efficient.

5.6 Convergence Considerations

While formal convergence proofs are sensitive to the optimization setup, learning rate schedules, and so forth, FedLag's performance is consistent with recent analyses that connect gradient variance with data drift.[3] FedLag helps in contract client proposals by removing high-variance components and scaling the statitically significant updates.

Such a contraction reduces the dispersion of parameter update over rounds enhancing the conditioning of aggregated gradient and leading to more stable convergence by naive averaging. Therefore, despite that FedLag is proposed at theoretical level, its design principles naturally imply that it has a better convergence stability in non-IID environments than both FedAvg and FedProx.

6 Conclusion and Future Work

This paper proposes FedLag, a new aggregation paradigm for federated learning that replaces traditional averaging-based fusion with a probabilistically grounded, likelihood-driven mechanism. By modeling each client's parameter update as a Gaussian distribution and computing the candidate fused values from their intersection points, FedLag casts global aggregation as a dominant region selection problem that is solved using weighted maximum likelihood estimation. This design allows the algorithm to incorporate uncertainty, balance decision errors, and adapt parameter-wise to the statistical structure of heterogeneous client updates-each a capability not offered by existing methods, such as FedAvg, FedProx, SCAFFOLD, AFL, and FedNova.

We theoretically analyzed and showed that FedLag naturally balances Type I and Type II errors at Gaussian intersection points, is robust to outlier or high-variance gradients, keeps consensus when clients agree, and is aligned with Fisher-information-based–based principles of efficient estimation. Altogether, these properties support the claim that FedLag provides a more resilient, stable, and interpretable aggregation procedure under non-IID conditions. While this is a purely theoretical study and empirical results are not presented, the mathematical properties of FedLag provide strong evidence of its potential to improve global optimization behavior and mitigate client drift in practical deployments. In the future, FedLag will be extended in several promising ways.

- Empirical Evaluation: We strive to perform extensive empirical evaluations across standard FL benchmarks to quantify improvements in accuracy, communication efficiency, and convergence speed.
- Formal Convergence Metrics: Future extensions will include detailed convergence metrics that formally characterize how FedLag contracts gradient variance and stabilizes update trajectories.
- Resilience Assessment: We will measure resilience properties, including robustness under noisy, or adversarial client updates, to better understand the performance of FedLag in real-world multi-device ecosystems.
- Application Compatibility: It will be necessary to assess the compatibility of the prediction-that is, how FedLag influences the performance in downstream tasks, generalization behavior, and model calibration-in healthcare, IoT, and edge intelligence applications where its deployment will be applied.
- Methodological Extensions: Examining scalable midpoint selection, Gaussian mixture generalizations, and secure or Byzantine-resilient variants will further enhance practicality and robustness of the proposed method.

FedLag provides a statistically principled and geometrically meaningful framework for federated optimization that opens various ways of advancing theory and practice in designing resilient global aggregation rules.

7 Acknowledgement

Thanks to the TTDF Scheme by the Department of Telecommunications for Accelerated Research on the 6G ecosystem, with grant title "FL-Auth: A Federated Learning-Based Secure Authentication Protocol for 6G Networks" vide no.TTDF/6G/57.

References

1. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, Communication-efficient learning of deep networks from decentralized data. *Proc. AISTATS,* **54**, 1273–1282 (2017). https://doi.org/10.48550/arXiv.1602.05629
2. P. Kairouz, et al., Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **14**, 1–210 (2021). https://doi.org/10.1561/2200000083
3. Y. Zhao, M. Li, K. Lai, M. Song, Federated learning with non-IID data. *arXiv* **1806.00582** (2018). https://doi.org/10.48550/arXiv.1806.00582
4. S. P. Karimireddy, S. Kale, M. Mohri, et al., SCAFFOLD: Stochastic controlled averaging for federated learning. *Proc. ICML,* **119**, 5132–5143 (2020). https://doi.org/10.48550/arXiv.1910.06378

5. J. Wang, Q. Liu, H. Liang, et al., Tackling client variability in federated learning. *Proc. NeurIPS,* **33**, 3513–3523 (2020).
6. T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks. *Proc. MLSys,* **2**, 429–450 (2020).
7. J. Wang, V. Tantia, A. Guha, O. Hinder, Federated learning with normalized averaging. *Proc. NeurIPS,* **33**, 19578–19591 (2020).
8. M. Mohri, G. Sivek, A. T. Suresh, Agnostic federated learning. *Proc. ICML,* **97**, 4615–4625 (2019).
9. D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates. *Proc. ICML,* **80**, 5650–5659 (2018).
10. A. Al-Shedivat, et al., Federated learning via posterior averaging: A Bayesian approach. *arXiv,* **2010.05273** (2020). https://doi.org/10.48550/arXiv.2010.05273
11. F. Hanzely, P. Richtárik, Federated learning of a mixture of global and local models. *Proc. NeurIPS,* **33**, 21073–21084 (2020).
12. Q. Li, Z. Wen, B. He, A survey on federated learning systems: Vision, hype, and reality. *IEEE Trans. Knowl. Data Eng.* **35**, 1061–1083 (2023). https://doi.org/10.48550/arXiv.1907.09693