

ADAPTIVE LOCAL STEP TUNING FOR FEDERATED LEARNING UNDER DATA HETEROGENEITY

Huynh Thanh Loc*

Ho Chi Minh City University of Foreign Languages - Information Technology, Vietnam

*Corresponding author: locht@huflit.edu.vn

(Received: May 05, 2026; Revised: June 18, 2026; Accepted: June 22, 2026)

DOI: 10.31130/ud-jst.2026.24(9).369E

Abstract - Federated Learning (FL) enables collaborative model training across decentralized clients while preserving data privacy. However, non-IID data often induces client drift and unstable convergence due to mismatched local training dynamics across clients. Standard FL methods typically utilize fixed local training budgets, leading to computational inefficiency in heterogeneous environments. This paper proposes Adaptive Local Step Tuning (ALST), a lightweight client-autonomous framework that dynamically adjusts local training steps based on an empirical drift proxy estimated from a single probe update. The framework operates entirely on the client side without introducing additional communication overhead or metadata sharing. Experimental results on EMNIST and CIFAR-100 demonstrate that ALST maintains stable convergence while improving the efficiency-performance trade-off under heterogeneous settings. While EMNIST validates the stability of the proposed adaptive mechanism, results on the more challenging CIFAR-100 benchmark show that ALST reduces total local computation by 23.09% while maintaining predictive performance within 0.14% of fixed-work baselines.

Key words - Federated Learning; Non-IID Data; Client Drift; Adaptive Local Step Tuning

1. Introduction

Federated Learning (FL) has emerged as an important paradigm for decentralized machine learning, enabling multiple clients to collaboratively train a shared global model without exchanging raw data [1], [2]. By preserving data locality, FL provides an effective solution for privacy-sensitive applications such as mobile services, healthcare, and edge intelligence [3]. Recent studies have also highlighted the importance of lightweight deep learning for resource-constrained edge devices [4]. Among existing FL algorithms, Federated Averaging (FedAvg) remains the most widely adopted optimization framework due to its simplicity and communication efficiency [5]. However, the performance of FedAvg is highly sensitive to statistical heterogeneity, commonly referred to as non-IID data, where local client distributions differ significantly from the overall population distribution.

Under heterogeneous data distributions, local clients optimize toward inconsistent objectives, leading to local model updates that deviate from the global optimization direction. This phenomenon, widely known as client drift, degrades convergence stability and reduces global model performance [5] - [8]. Existing studies categorize statistical heterogeneity into several forms, including label skew, feature skew, and quantity skew [2], [8]. In practical FL deployments, such heterogeneity is unavoidable because

user-generated data naturally varies across devices and environments. Furthermore, standard FL protocols generally employ a fixed local training budget for all clients regardless of their optimization difficulty. This fixed-work strategy is inherently inefficient. For clients with high statistical skew, excessive local iterations do not proportionally contribute to global utility but instead accelerate weight divergence, thereby destabilizing convergence and wasting computational resources [7], [10].

Several approaches have been proposed to mitigate client drift. Variance-reduction methods, such as SCAFFOLD, introduce control variates to stabilize local optimization [8], while proximal regularization methods, such as FedProx, constrain local updates via additional optimization terms [11]. Other methods, including FedNova, address aggregation bias caused by heterogeneous local training lengths across clients [10]. Although effective, many of these approaches require additional communication, server-side coordination, or complex hyperparameter tuning, which may reduce practicality in resource-constrained edge environments.

To address these limitations, this paper introduces Adaptive Local Step Tuning (ALST), a lightweight client-autonomous framework that adjusts local training steps dynamically based on an empirical drift proxy estimated from a single probe update. Instead of applying a uniform local workload across all clients, ALST enables each client to independently adapt its local optimization effort based on observed optimization behavior. The proposed mechanism operates entirely on the client side without requiring additional metadata exchange or communication overhead, thereby maintaining communication efficiency and privacy.

The main contributions of this work are summarized as follows:

- We propose a lightweight adaptive local training mechanism that dynamically adjusts the number of local training steps based on client-specific optimization behavior under heterogeneous data distributions.
- We design a client-autonomous adjustment strategy based on an empirical drift proxy estimated from a single probe update, enabling adaptive local optimization without additional communication overhead.
- We establish a controlled experimental framework on EMNIST and CIFAR-100 to evaluate the effectiveness of ALST under heterogeneous federated settings.

- Experimental results demonstrate that the proposed method maintains stable convergence while reducing unnecessary local computation compared with fixed-work federated optimization baselines.

2. Related Work

FL under heterogeneous data distributions has attracted significant research attention due to the instability caused by client drift. FedAvg remains the foundational optimization algorithm for FL because of its simplicity and communication efficiency [1]. However, FedAvg assumes relatively consistent local optimization behavior across clients, which becomes problematic under strongly non-IID settings.

Several methods have been proposed to improve convergence stability in heterogeneous FL environments. FedProx introduces a proximal regularization term to constrain local updates and reduce divergence between local and global models [11]. SCAFFOLD further mitigates client drift through control variates that correct local update directions [9]. FedNova addresses objective inconsistency caused by heterogeneous local training lengths via normalized aggregation [10]. Although these methods improve optimization stability, they often introduce additional communication overhead, auxiliary variables, or increased optimization complexity.

Recent studies have also explored adaptive local training strategies to improve computational efficiency in FL systems. These approaches regulate local optimization effort based on client-specific training behavior or resource availability [11] - [14]. However, many existing approaches rely on centralized coordination or on additional metadata exchange between clients and the server.

In contrast to existing approaches such as FedProx, SCAFFOLD, and FedNova, the proposed ALST framework performs its adaptive local step adjustment entirely on the client side using a lightweight probe update to estimate local optimization behavior. Unlike methods requiring proximal regularization, control variates, or aggregation normalization, ALST regulates local optimization effort without introducing auxiliary optimization variables or additional communication overhead. This lightweight client-autonomous design preserves communication neutrality while minimizing implementation complexity under heterogeneous federated settings.

3. Proposed Method

3.1. Federated Learning Formulation

Consider a FL system consisting of K clients. Each client k possesses a local dataset D_k with data distribution $P_k(x, y)$, which may vary significantly across clients under heterogeneous settings. The global optimization objective is defined as:

$$\min_w F(w) = \sum_{k=1}^K p_k F_k(w) \quad (1)$$

where, $F_k(w)$ and p_k denote the local objective and aggregation weight of the client k , respectively.

In standard FedAvg, all participating clients perform a fixed number of local training epochs E_{base} during each communication round before transmitting updated model parameters to the server. Under non-IID conditions, this fixed-work strategy may lead to excessive local optimization on highly skewed clients, thereby increasing divergence among local updates and destabilizing global convergence.

3.2. Probe-Based Drift Estimation

To address this issue, the proposed ALST framework dynamically adjusts local training steps based on an empirical drift proxy estimated from local optimization behavior.

At the beginning of each communication round, client k performs a lightweight probe update and computes the variation in local loss:

$$D_k = |L_k^{(1)} - L_k^{(0)}| \quad (2)$$

where $L_k^{(0)}$ and $L_k^{(1)}$ denote the mean empirical training loss computed on the same first local mini-batch before and after a single SGD probe update, respectively. Using the same mini-batch ensures that the observed loss variation primarily reflects the effect of the probe update rather than differences in data sampling. The standard cross-entropy loss is adopted as the local objective for both EMNIST and CIFAR-100. The local loss is computed as follows:

$$L_k = \frac{1}{|B_k|} \sum_{(x_i, y_i) \in B_k} \ell(f(x_i; w), y_i) \quad (3)$$

where B_k is the first local mini-batch of client k , $|B_k|$ denotes the mini-batch size, $f(x_i; w)$ denotes the model prediction with parameters w , y_i is the corresponding ground-truth label, and $\ell(\cdot)$ denotes the cross-entropy loss function.

The probe update is included in the local training budget to avoid additional computational overhead during adaptive step estimation.

The quantity D_k serves as an empirical indicator of local optimization difficulty and drift intensity. The absolute value is used because the magnitude of local loss variation, regardless of direction, provides a lightweight proxy for local optimization instability under heterogeneous conditions. Larger D_k values indicate stronger divergence tendencies between local and global update directions, suggesting that excessive local training may negatively affect convergence stability.

3.3. Adaptive Local Step Tuning

Based on the estimated drift intensity, each client dynamically adjusts its local training budget as:

$$E_k = \max(E_{min}, \text{round}(E_{base} - \alpha D_k)) \quad (4)$$

where, E_k is the adjusted local training budget for client k , E_{base} is the default local training budget used in standard Federated Averaging, E_{min} is the minimum allowable local training budget ($E_{min} \geq 1$) to prevent zero-progress communication rounds, and α is the adaptation coefficient controlling the adjustment sensitivity.

This linear adjustment strategy reduces local computation for clients exhibiting stronger drift intensity while preserving sufficient local optimization progress. Consequently, ALST constrains the magnitudes of unstable updates under heterogeneous conditions without introducing additional communication overhead.

3.4. ALST Training Procedure

The overall ALST procedure during each communication round is summarized as follows:

1. The server broadcasts the current global model to participating clients.
2. Each client computes the initial local loss.
3. A single probe update is performed locally.
4. The client estimates drift intensity D_k .
5. The adaptive local epoch budget E_k is computed.
6. The client performs local training using the adjusted epoch budget.
7. The server aggregates updated local models using standard federated averaging.

Since all adaptation operations are performed entirely on the client side, ALST preserves metadata privacy and does not require additional information exchange between clients and the server.

4. Experimental Setup

4.1. Datasets

Experiments are conducted on two benchmark FL datasets provided by TensorFlow Federated (TFF) [16].

- **EMNIST:** A naturally partitioned handwritten character recognition dataset consisting of decentralized client-specific data partitions. The dataset exhibits moderate statistical heterogeneity and is commonly used to evaluate convergence stability in federated optimization settings.

- **CIFAR-100:** A federated image classification benchmark containing 100 object categories, which exhibit significantly higher optimization difficulty and stronger heterogeneous characteristics than EMNIST. This dataset is used to evaluate the effectiveness of ALST under more challenging non-IID conditions.

4.2. Compared Methods

The proposed ALST framework is compared against the following baselines:

- **FedAvg:** Standard federated averaging with fixed local training budget across all participating clients.
- **Fixed-E:** A fixed-work implementation sharing the same training infrastructure and hyperparameter configuration as ALST, but without its adaptive local step adjustment mechanism.
- **ALST:** The proposed client-autonomous local step adjustment strategy based on the linear adjustment policy described in Section 3.3.

All methods use identical model architectures, optimization settings, and communication protocols to ensure fair comparison.

4.3. Training Configuration

All experiments are implemented using TFF [16]. To ensure reproducibility and fair comparison, all methods share identical optimization settings unless otherwise specified.

For EMNIST experiments, training is conducted over 100 communication rounds; for CIFAR-100 experiments, 200 communication rounds are utilized due to the heightened optimization challenges presented by the dataset.

The local batch size is fixed at 20, and 50 clients are sampled during each communication round. All local optimization procedures use stochastic gradient descent (SGD) as the client optimizer.

To improve statistical reliability, each experiment is repeated with three random seeds: 42, 123, and 999. Final results are reported as the mean and standard deviation across all runs.

For ALST, the adaptive local training budget is constrained within predefined lower and upper bounds to maintain stable optimization behavior and prevent excessively small local updates.

4.4. Evaluation Metrics

The proposed method is evaluated using the following metrics:

- **Global Test Accuracy:** The overall performance of the global model is evaluated on the test dataset.
- **Cumulative Local Training Steps:** The total number of local optimization steps performed throughout the entire training process.
- **Compute Saving:** Relative decrease in total local computation compared to the FedAvg baseline.
- **Convergence Stability:** Consistency of optimization behavior across communication rounds and different random seeds.

Besides quantitative performance comparison, this study examines convergence behavior and efficiency-performance trade-offs through convergence curves and cumulative local computation metrics.

5. Result and Discussion

5.1. CIFAR-100 Results

Table 1. Experimental results on CIFAR-100

Method	Test Accuracy	Total Local Steps	Compute Saving
FedAvg	0.2258 ± 0.0052	50,000.00 ± 0.00	0.00%
Fixed-E	0.2265 ± 0.0041	50,000.00 ± 0.00	0.00%
ALST	0.2244 ± 0.0039	38,454.33 ± 125.96	23.09%

Table 1 summarizes the final experimental results on CIFAR-100. Compared with fixed-work federated optimization baselines, the proposed ALST framework achieves comparable predictive performance while significantly reducing total local computation.

In particular, ALST reduces total local training steps by 23.09% compared with FedAvg while maintaining test

accuracy within approximately 0.14% of the baseline. These results suggest that ALST becomes increasingly beneficial under more challenging heterogeneous optimization conditions.

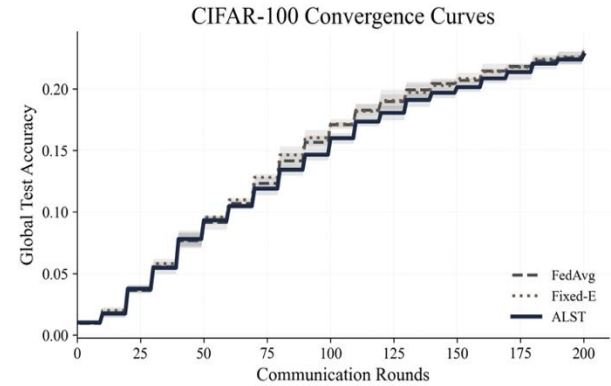


Figure 1. Global test accuracy over communication rounds on CIFAR-100. ALST maintains stable convergence behavior comparable to fixed-work federated optimization baselines

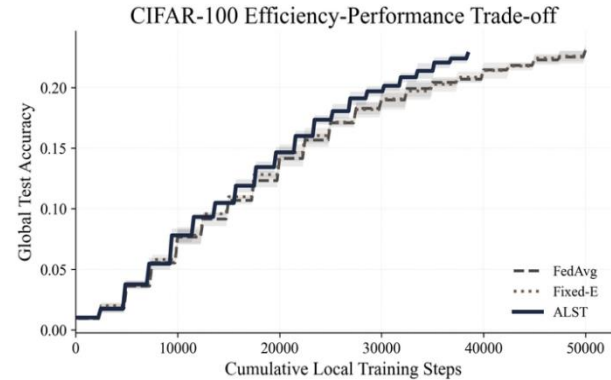


Figure 2. Efficiency–performance trade-off on CIFAR-100 measured by global test accuracy versus cumulative local training steps. ALST achieves competitive predictive performance with substantially reduced local computation

As shown in Figure 1, ALST maintains stable convergence behavior throughout training despite dynamically adjusting local training budgets. Although the final test accuracy is slightly lower than FedAvg and Fixed-E, the convergence trajectories remain smooth and consistent across communication rounds without exhibiting unstable optimization behavior.

Moreover, the convergence curves indicate that substantial reductions in local computation can be achieved without significantly sacrificing predictive performance, highlighting the effectiveness of ALST under highly heterogeneous federated settings.

5.2. Efficiency–Performance Trade-off

To further evaluate computational efficiency, cumulative local computation and predictive performance are analyzed on the CIFAR-100 dataset. The results indicate that ALST substantially reduces unnecessary local training effort while maintaining predictive performance comparable to fixed-work baselines.

As illustrated in Figure 2, ALST achieves comparable accuracy with significantly fewer accumulated local training steps than FedAvg and Fixed-E. The relatively

small performance gap between the compared methods suggests that ALST effectively improves computational efficiency without noticeably degrading convergence quality.

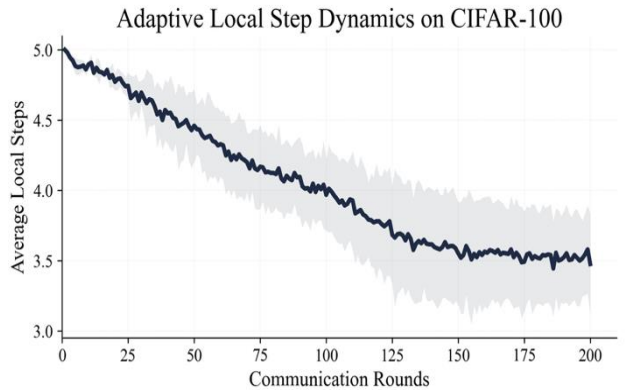


Figure 3. Average local training steps of ALST on CIFAR-100 across communication rounds. The adaptive tuning strategy progressively reduces local computation under heterogeneous optimization conditions

These observations further indicate that ALST achieves a favorable efficiency-performance Pareto trade-off, making the proposed framework suitable for resource-constrained decentralized learning environments.

5.3. Adaptive Local Step Dynamics

To further examine the adaptive behavior of ALST, the average local training budget is analyzed throughout CIFAR-100 training. The results show that ALST progressively reduces local computation as optimization difficulty increases under heterogeneous federated conditions.

As illustrated in Figure 3, the average local training budget gradually decreases from the initial configuration toward a more conservative optimization regime during training. This behavior indicates that ALST autonomously limits excessive local optimization effort while preserving sufficient local training progress.

These observations suggest that ALST helps mitigate unstable local updates under heterogeneous settings, thereby improving both the efficiency and stability of federated optimization.

5.4. EMNIST Results

Table 2. Experimental results on EMNIST

Method	Test Accuracy	Total Local Steps	Compute Saving
FedAvg	0.9332 ± 0.0049	250.00 ± 0.00	0.00%
Fixed-E	0.9376 ± 0.0047	250.00 ± 0.00	0.00%
ALST	0.9374 ± 0.0048	249.73 ± 0.15	0.11%

Table 2 summarizes the experimental results on EMNIST under moderate heterogeneous conditions. Compared with FedAvg and Fixed-E, the proposed ALST framework achieves comparable predictive performance while maintaining nearly identical local computation.

In particular, ALST slightly improves final test accuracy compared with FedAvg while introducing negligible differences in accumulated local training steps.

These results suggest that the proposed adaptive mechanism behaves conservatively under moderate optimization difficulty, preserving stable convergence without unnecessary local computation.

The quantitative results indicate that the adaptive adjustment mechanism is activated only when necessary. Since the optimization process on EMNIST remains relatively stable throughout training, ALST performs nearly the same amount of local computation as the fixed-work baselines while maintaining comparable predictive performance. This behavior demonstrates that the proposed framework avoids unnecessary adaptation when local optimization difficulty is limited.

As shown in Figure 4, the convergence curves on EMNIST remain closely aligned across all methods throughout training. Unlike CIFAR-100, no significant divergence behavior is observed under this relatively less challenging federated setting.

Moreover, the convergence trajectories suggest that ALST does not destabilize federated optimization when local optimization difficulty is relatively low. The minimal performance differences among the compared methods further indicate that ALST adaptively reduces its intervention under moderate statistical heterogeneity, thereby maintaining robustness and convergence stability across different FL scenarios.

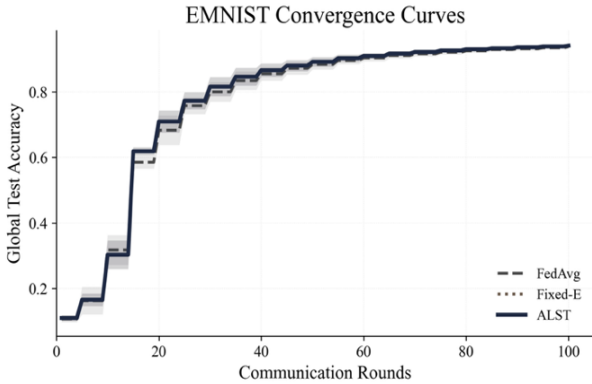


Figure 4. Convergence behavior on EMNIST. *ALST maintains stable convergence dynamics comparable to fixed-work federated optimization baselines under moderate heterogeneous conditions*

5.5. Discussion

The experimental results suggest that ALST is particularly beneficial under challenging heterogeneous federated settings such as CIFAR-100, where excessive local training can amplify client drift and computational inefficiency. By dynamically reducing unnecessary local optimization effort, ALST improves computational efficiency while preserving stable convergence behavior. In contrast, under relatively moderate heterogeneous conditions, such as EMNIST, the proposed mechanism behaves conservatively and maintains training dynamics comparable to fixed-work baselines.

Another important advantage of ALST is its lightweight client-side design. Unlike methods requiring auxiliary optimization variables or additional

communication rounds, ALST operates without introducing extra communication overhead, making it suitable for resource-constrained edge and IoT environments. In addition, the drift indicators remain entirely on the client side and are never transmitted to the server, thereby preserving metadata privacy during federated optimization.

Despite these promising results, several limitations remain. First, the proposed adjustment strategy relies on an empirical drift proxy estimated from local loss variation rather than a formal theoretical measure of optimization divergence. Although the proposed adjustment strategy is heuristic, the experimental results demonstrate consistent convergence behavior across both moderate and highly heterogeneous federated benchmarks. Compared with gradient-based divergence estimation, the proposed loss-based drift proxy incurs minimal computational overhead because it requires only scalar loss evaluation instead of explicit gradient norm computation, making it more suitable for resource-constrained edge environments.

Second, the current study focuses primarily on image classification benchmarks and does not evaluate deployment on real edge devices. Although EMNIST and CIFAR-100 represent two federated benchmarks with different optimization characteristics, additional evaluation on larger-scale datasets and federated systems with more participating clients would further strengthen the generality of the proposed framework.

Future work will therefore investigate theoretical convergence guarantees, adaptive parameter calibration strategies, and broader evaluation on larger-scale federated systems and real-world edge deployment scenarios.

6. Conclusion

This paper proposed ALST, a lightweight client-autonomous framework for regulating local optimization effort in FL under heterogeneous data distributions. The proposed method dynamically adjusts local training steps based on an empirical drift proxy estimated from a lightweight probe update while preserving communication neutrality and metadata privacy.

Experimental results on EMNIST and CIFAR-100 demonstrate that ALST maintains stable convergence while reducing unnecessary local computation under heterogeneous federated settings. In particular, ALST achieves a 23.09% reduction in total local computation on CIFAR-100 while maintaining predictive performance within approximately 0.14% of fixed-work baselines.

The experimental analysis further indicates that ALST becomes increasingly beneficial under more challenging heterogeneous optimization conditions. While ALST behaves conservatively on moderate heterogeneous benchmarks such as EMNIST, the proposed framework demonstrates a favorable efficiency-performance trade-off on more difficult federated optimization tasks.

Overall, these findings suggest that ALST is an effective and practical strategy for improving the computational efficiency and practicality of FL systems. Future work will investigate theoretical convergence analysis, adaptive parameter calibration, and deployment in realistic edge computing environments.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, PMLR, 2017, pp. 1273–1282.
- [2] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021. <https://doi.org/10.1561/22000000083>
- [3] D. M. J. Gutierrez *et al.*, "Non-IID Data in Federated Learning: A Survey with Taxonomy, Metrics, Methods, Frameworks and Future Directions," *arXiv preprint*, arXiv:2411.12377, 2024. [Online]. Available: <https://arxiv.org/abs/2411.12377> [Accessed May 28, 2026].
- [4] Q. N. Pham and D. K. Ninh, "Lightweight fall detection on edge devices via knowledge distillation," *The University of Danang - Journal of Science and Technology*, vol. 23, no. 9D, pp. 74–78, Sep. 2025. [https://doi.org/10.31130/ud-jst.2025.23\(9D\).568E](https://doi.org/10.31130/ud-jst.2025.23(9D).568E)
- [5] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [6] M. Sherkati, W. Shi, and M. Amini, "A comprehensive evaluation of metrics on their ability to capture the degrees of Non-IIDness with label skew in federated learning," *Computers and Electrical Engineering*, vol. 129, p. 110743, 2026. <https://doi.org/10.1016/j.compeleceng.2025.110743>
- [7] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021. <https://doi.org/10.1016/j.neucom.2021.07.098>
- [8] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-art and Research Challenges," *ACM Computing Surveys*, vol. 56, no. 3, Oct. 2023. <https://doi.org/10.1145/3625558>
- [9] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning*, Virtual, PMLR, 2020, pp. 5132–5143.
- [10] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Virtual, Curran Associates, Inc., 2020, pp. 7611–7623.
- [11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.
- [12] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019. <https://doi.org/10.1109/JSAC.2019.2904348>
- [13] J. Wang and G. Joshi, "Adaptive Communication Strategies to Achieve the Best Error-Runtime Trade-off in Local-Update SGD," in *Proceedings of Machine Learning and Systems*, vol. 1, pp. 212–229, 2019.
- [14] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proceedings of the 37th International Conference on Machine Learning*, Virtual, PMLR, 2020, pp. 4387–4398.
- [15] S. Reddi *et al.*, "Adaptive Federated Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [16] The TensorFlow Federated Authors, "TensorFlow Federated: Machine Learning on Decentralized Data", *TensorFlow.org*. [Online]. Available: <https://www.tensorflow.org/federated> [Accessed May 28, 2026].