Federated Learning under Data Heterogeneity: Implementation and Comparative Analysis

Mohammad Salman, Abdul Samad, Rabia Aslam

***Abstract*—Federated Learning (FL) enables multiple clients to collaboratively train a shared global model without sharing local data, offering both privacy and scalability advantages. In this work, we study core FL algorithms and their behavior under data heterogeneity. We first verify the theoretical equivalence of Federated SGD (FedSGD) and centralized SGD, confirming that with full client participation and a single local update, both yield nearly identical model trajectories. We then implement Federated Averaging (FedAvg) and analyze how local update frequency and client sampling affect performance and communication efficiency. Using Dirichlet-based data partitions, we observe that higher non-IIDness significantly degrades accuracy and convergence stability. To address this, we extend FedAvg with four methods i.e. FedProx, SCAFFOLD, Gradient Harmonization (FedGH), and Sharpness-Aware Minimization (FedSAM). Our experiments show that FedProx improves stability, SCAFFOLD effectively reduces client drift, FedGH aligns conflicting gradients, and FedSAM enhances generalization. Overall, these approaches demonstrate meaningful improvements over FedAvg, especially in highly heterogeneous settings, highlighting the importance of drift-mitigation and regularization techniques in practical federated systems.**

***Index Terms*—Federated Learning, Non-IID Data, FedAvg, FedProx, SCAFFOLD, FedSAM, Gradient Harmonization, Client Drift**

I. Introduction

Federated Learning (FL) is an emerging paradigm that enables multiple clients to collaboratively train a shared global model without exchanging their private data. This decentralized approach addresses privacy and data ownership concerns while maintaining learning performance across distributed devices. However, FL faces key challenges such as communication overhead and statistical heterogeneity among clients, which can hinder convergence and model accuracy. This study investigates the fundamental algorithms of FL, including FedSGD and FedAvg, and evaluates several advanced methods, FedProx, SCAFFOLD, FedGH, and FedSAM to analyze their effectiveness in mitigating non-IID data issues and improving global model performance.

II. Methodology

A. Model Architecture

A lightweight Convolutional Neural Network (CNN) is employed consistently across all experiments to ensure fair comparability. The model is implemented in PyTorch using double-precision floating-point arithmetic (float64) for numerical stability across federated and centralized settings.

The architecture follows a conventional feature extraction and classification pipeline suited for CIFAR-10 images of size $3 \times 32 \times 32$. It includes two convolutional layers with ReLU activation and max-pooling, followed by two fully connected layers for classification. Table I summarizes the complete network configuration.

TABLE I
SimpleCNN architecture (CIFAR-10 input: $3 \times 32 \times 32$).

Layer	Type / Kernel	In $\rightarrow$ Out	Params
Conv1	Conv2D, 3×3	$3 \rightarrow 32$	896
MaxPool1	2×2 Pool	—	—
Conv2	Conv2D, 3×3	$32 \rightarrow 64$	18,496
MaxPool2	2×2 Pool	—	—
FC1	Linear	$4096 \rightarrow 128$	524,416
FC2	Linear	$128 \rightarrow 10$	1,290
Total	—	—	**545,098**

This architecture maintains a compact design while preserving sufficient expressive power for CIFAR-10 classification tasks. Its lightweight structure ensures that any observed performance variations arise primarily from the training strategies (FedSGD, FedAvg, or data heterogeneity) rather than differences in model complexity.

III. FedSGD vs. Centralized SGD

This experiment aims to verify the theoretical equivalence between Federated Stochastic Gradient Descent (FedSGD) and centralized SGD under independent and identically distributed (IID) data conditions. Both training schemes utilize an identical SimpleCNN architecture on the CIFAR-10 dataset. We have experimented with a learning rate of $\eta = 0.01$ and $\eta = 0.1$ having uniform weight initialization..

In the FedSGD setting, the global dataset is evenly partitioned among three clients (IID distribution) . Each client computes gradients locally on its subset and contributes to one synchronized global update per communication round ($K = 1$). Model parameters are aggregated via weighted averaging of the local gradients, ensuring global consistency across clients.

The centralized SGD baseline, in contrast, uses the complete dataset as a single batch per iteration, effectively mimicking a single-client scenario.

IV. Results and Discussions

Both models are trained for 1000 communication rounds with early stopping based on test accuracy improvements with a patience level 0f 200 and min_delta change of 0.1. Performance is compared in terms of test accuracy, loss trajectory, and parameter divergence. Results confirm that FedSGD achieves convergence behavior identical to centralized SGD

when all clients participate in every round, thus validating their theoretical equivalence under IID data distribution. We see this equivalence because although FedSGD distributes the gradient computation across clients, the weighted averaging mathematically reconstructs the exact same gradient as computing over the full centralized dataset. The observed 10^{-12} divergence arises solely from floating-point rounding in different computation orders (sequential in centralized vs. distributed aggregation in FedSGD). Detailed JSON results are uploaded on github.

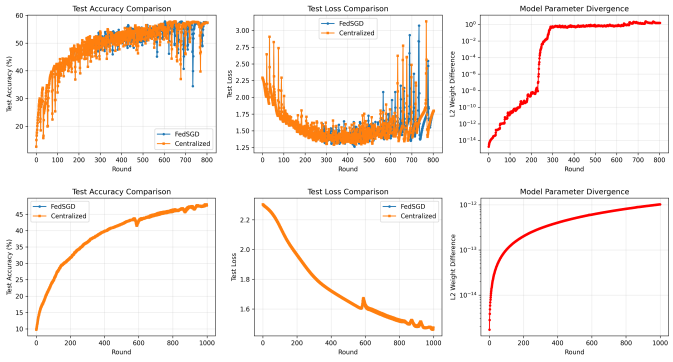

Fig. 1. Comparison between **FedSGD** and **Centralized SGD** under two learning rates. The upper plot uses a learning rate of 0.1, achieving higher final accuracies of 57.3% (FedSGD) and 57.41% (Centralized), but exhibits a small drift due to the larger step size and floating-point aggregation effects reflected in the gradual rise of model divergence up to weight_diff = 1.5084 after 800 rounds as we have used early stopping with patience of 200 and min_delta of 0.1. The lower plot corresponds to a smaller learning rate of 0.01, resulting in smoother convergence and zero divergence (0.0) throughout, but a lower final accuracy of 47.66% for both methods. The lightweight **SimpleCNN** architecture is intentionally employed so that observed behaviors stem from the federated learning process rather than architectural complexity.

V. Federated Averaging (FedAvg)

In this task, the Federated Averaging (FedAvg) algorithm is implemented to investigate the trade-off between local computation and communication frequency. The experiment is conducted on the CIFAR-10 dataset using ten clients, each trained with Stochastic Gradient Descent (SGD) at a learning rate of $\eta = 0.01$ and batch size of 64. A shared SimpleCNN model is initialized identically across all clients.

Two separate studies are performed to analyze the effects of local update and participation rate:

1) Local Epoch Variation: The number of local training epochs per communication round is varied as $K \in \{1, 5, 10, 20\}$ while keeping all ten clients active in each round. Metrics such as test accuracy, loss, client drift (divergence between local and global model parameters) and regret are recorded. This setup highlights how increasing K enhances local computation but can amplify client drift and delay global convergence.

2) Client Sampling: To explore communication efficiency, the fraction of participating clients per round is varied as $f \in \{1.0, 0.5, 0.2, 0.1\}$. Each run measures test accuracy, communication cost (in MB per round), and convergence rate. Results show that smaller f values reduce communication overhead but may cause slower convergence due to partial client participation. We see that With all clients participating

TABLE II
Per-round Communication Cost for Different Client Fractions (f)

The model size per transmission is 4.16 see equation 1 for details.

f	fN	**Uploads** (C→S)	**Download** (S→C)	**Total / Round (MB)**
1.0	10	$10 \times 4.16 = 41.6$	4.16	**45.76**
0.5	5	$5 \times 4.16 = 20.8$	4.16	**24.96**
0.2	2	$2 \times 4.16 = 8.32$	4.16	**12.48**

(f=1.0), communication cost is about 45.8 MB per round. At half participation (f=0.5), the cost falls to 25 MB/round, and at one-fifth participation (f=0.2), to roughly 12.5 MB/round. Lowering f reduces total communication but may slow convergence due to fewer aggregated gradients.

Overall, the experiment demonstrates that larger local epochs (K) or lower client fractions (f) improve communication efficiency at the expense of model synchronization and convergence stability.

VI. Impact of Data Heterogeneity

We investigate the impact of data heterogeneity on Federated Averaging (FedAvg) performance using CIFAR-10 dataset. The experimental configuration consists of 5 clients trained over 100 communication rounds with $K = 5$ local epochs per round. We employ same SimpleCNN architecture with learning rate $\eta = 0.01$.

To systematically control label heterogeneity, we partition the dataset using Dirichlet distribution with concentration parameter $\alpha \in \{100, 1.0, 0.2, 0.05\}$. Large α values (e.g., $\alpha = 100$) approximate IID conditions where each client receives approximately uniform class distribution, while small α values (e.g., $\alpha = 0.05$) create highly skewed distributions where clients observe predominantly distinct classes. All experiments use identical initialization (seed=42) to isolate the effect of data distribution.

VII. Results and Discussion

Figure 2 (top row) visualizes the data distribution across clients for different α values. As α decreases, the class distribution becomes increasingly skewed, with clients specializing in specific classes. The accuracy curves (second row) reveal a clear inverse relationship between heterogeneity and model performance:

- $\alpha = 100$ (IID): Achieves highest accuracy of ∼71.53%, with rapid convergence within 10 rounds and stable performance thereafter.
- $\alpha = 1.0$ (Moderate Non-IID): Reaches ∼69.98% accuracy, showing slight degradation but maintaining stable convergence.
- $\alpha = 0.2$ (High Non-IID): Performance drops to ∼67.07%, with slower convergence rate.
- $\alpha = 0.05$ (Extreme Non-IID): Significant performance degradation to ∼62.18%, with unstable convergence characterized by oscillations throughout training.

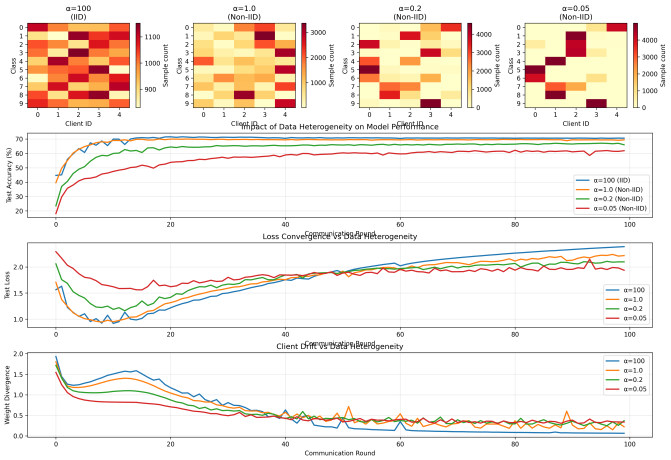

Fig. 2. Impact of **data heterogeneity** (Dirichlet α) on federated model convergence. The top row visualizes the class-wise sample distribution across five clients, The subsequent plots illustrate the corresponding test accuracy, loss, and parameter divergence over 100 rounds.

A. Loss Convergence and Client Drift

The loss curves (third row) demonstrate that more heterogeneous settings exhibit higher final loss values and slower convergence rates. Notably, the $\alpha = 0.05$ case shows persistent high loss ($\sim$2.0) even after 100 rounds, indicating poor generalization to the global test distribution.

Client drift analysis (bottom row) quantifies model divergence using weight divergence metrics. Initial divergence is highest across all settings ($\sim$2.0) but decreases as training progresses. Critically, heterogeneous settings ($\alpha \leq 1.0$) maintain elevated divergence ($\sim$0.3-0.5) even after convergence, while the IID case ($\alpha = 100$) converges to near-zero divergence ($\sim$0.1). Also more skewed runs show higher gap between training accuracy and global test accuracy. The persistent drift in non-IID settings indicates that local updates consistently push models in conflicting directions, preventing effective aggregation.

B. Key Observations

Our experiments confirm that FedAvg performance degrades monotonically with increasing data heterogeneity. For moderate alpha, here 1.0, we see that accuracy is not much lower than alpha 100. The performance gap between IID and extreme non-IID settings exceeds nearly 10 percentage points (71.5% vs. 62.1%), validating known limitations of vanilla FedAvg under label skew. The weight divergence metric effectively captures client drift, showing strong correlation with accuracy degradation. These results underscore the need for heterogeneity-aware aggregation strategies in practical federated learning deployments.

Results show that as α decreases, clients experience higher model divergence and slower convergence, leading to overall performance degradation. This confirms that severe data heterogeneity negatively impacts global model accuracy and stability, highlighting the fundamental challenge of federated learning under non-IID data conditions.

TABLE III
EFFECT OF DATA HETEROGENEITY ON FEDAVG PERFORMANCE (CIFAR-10, SIMPLECNN)

α	Type	Final Acc. (%)	Best Acc. (%)	Avg Div.
100.00	IID	70.58	71.53	0.5371
1.00	Non-IID	69.40	69.98	0.6012
0.20	Non-IID	65.99	67.07	0.5654
0.05	Non-IID	61.87	62.18	0.5060

TABLE IV
COMMUNICATION AND COMPUTATIONAL SUMMARY PER ROUND

Metric	**Task 1**	**Task 2**	**Task 3**
Experiment	FedSGD vs. Centralized	FedAvg Efficiency	Data Heterogeneity
Clients (N)	3	10	5
Parameters	545,098 (SimpleCNN)		
Local Epochs (K)	$K = 1$	$K \in \{1, 5, 10, 20\}$	$K = 5$
Client Fraction (f)	$f = 1.0$	$f \in \{1.0, 0.5, 0.2\}$	$f = 1.0$
Data Split	IID	IID	Dirichlet $\alpha \in \{100, 1, 0.2, 0.05\}$
Uploads/Round	3	fN	5
Downloads/ Round	3	fN	5
Comm. Volume (MB/round)	$3 \times 4.16 \times 2 = 25.0$	$(fN + 1) \times 4.16$	$6 \times 4.16 = 25.0$
Focus	Equivalence FedSGD-SGD	Comp.-Comm. Trade-off	Non-IID Impact

C. Conclusion

Table IV summarizes the communication characteristics across FedSGD, Centralized, FedAvg and data heterogeneity. All experiments employ the SimpleCNN model with $P = 545{,}098$ trainable parameters in double precision (`float64`), where each parameter occupies 8 bytes. The model size per transmission is:

$$\text{Model Size} = \frac{P \times 8}{1024^2} = \frac{545{,}098 \times 8}{1{,}048{,}576} \approx 4.16\,\text{MB}. \quad (1)$$

The FedSGD experiment (Task 1) demonstrates gradient-level equivalence with centralized training when all clients participate with $K = 1$ local epoch. Task 2 explores the efficiency trade-off by varying local epochs K and client participation fraction f, revealing that increased local computation reduces communication rounds but may increase client drift. Task 3 investigates non-IID data effects using Dirichlet partitioning with concentration parameter α, where smaller α values create more heterogeneous distributions.

The communication volume per round depends on the number of participating clients and bidirectional model exchange. Reducing client participation ($f < 1$) or increasing local epochs ($K > 1$) decreases communication frequency but may compromise convergence quality under heterogeneous conditions. These metrics quantify the fundamental efficiency-stability trade-off in federated learning.

REFERENCES