Failure Modes of Federated Fine-Tuning Under Non-IID Data: A Comparative Study of FedAvg and FedProx

Aviral Vishesh Goel
Department of Electrical Engineering
IIT Bombay
aviralvisheshgoel@iitb.ac.in

Adit Srivastava
Department of Electrical Engineering
IIT Bombay
22b1269@iitb.ac.in

Aagam Shah
Department of Electrical Engineering
IIT Bombay
22b1201@iitb.ac.in

***Abstract*—Federated learning enables collaborative model training across distributed clients without sharing raw data. However, statistical heterogeneity (non-IID data) across clients poses significant challenges to convergence and model quality. In this work, we conduct a systematic empirical study comparing FedAvg and FedProx algorithms on text classification using DistilBERT, under controlled non-IID conditions created via Dirichlet-based label skew. Our experiments on the AG News dataset demonstrate that: (1) non-IID data distributions cause substantial client parameter divergence, (2) FedProx's proximal regularization effectively reduces this divergence, and (3) FedProx achieves higher final accuracy (87.0%) compared to FedAvg (83.8%) under heterogeneous data while maintaining comparable performance on IID data. We provide detailed analysis of convergence behavior, client drift metrics, and failure modes under extreme heterogeneity.**

***Index Terms*—Federated Learning, Non-IID Data, FedAvg, FedProx, Text Classification, DistilBERT**

I. Introduction

Federated learning (FL) has emerged as a privacy-preserving paradigm for training machine learning models across decentralized data sources [1]. In FL, clients collaboratively train a shared model by exchanging model updates rather than raw data, addressing privacy concerns in domains such as healthcare, finance, and mobile computing.

A fundamental challenge in federated learning is *statistical heterogeneity*—the non-identical and non-independent (non-IID) distribution of data across clients. In real-world deployments, clients often have data that differs significantly in label distributions, feature distributions, or data quality. This heterogeneity leads to several pathologies:

- **Client drift**: Local models diverge from the global optimum as clients overfit to their local data distributions.
- **Slow convergence**: Aggregating divergent updates leads to oscillation and delayed convergence.
- **Degraded generalization**: The global model may fail to generalize across the heterogeneous client populations.

The seminal FedAvg algorithm [1] performs weighted averaging of client updates but provides no mechanism to handle client drift. FedProx [2] addresses this by adding a proximal term that penalizes deviation from the global model during local training.

A. Contributions

In this paper, we present a controlled empirical study of federated learning pathologies under non-IID data distributions. Our contributions include:

1) A systematic comparison of FedAvg and FedProx on text classification using transformer-based models (DistilBERT).
2) Quantitative analysis of client parameter divergence as a metric for measuring the impact of data heterogeneity.
3) Experimental evidence demonstrating FedProx's effectiveness in reducing client drift and improving convergence under non-IID conditions.
4) Analysis of failure modes under extreme heterogeneity conditions.

II. Related Work

A. Federated Learning Algorithms

FedAvg [1] established the foundational framework for federated learning, where clients perform multiple local SGD steps before aggregating updates via weighted averaging. While effective for IID data, FedAvg struggles with heterogeneous distributions.

FedProx [2] introduces a proximal term $\frac{\mu}{2}\|w - w^t\|^2$ to the local objective, where w^t is the global model at round t. This regularization limits how far local models can drift from the global model, providing theoretical convergence guarantees under heterogeneity.

B. Non-IID Data in Federated Learning

The impact of non-IID data has been extensively studied. Zhao et al. [3] showed that accuracy can drop by up to 55% on highly skewed data. Common non-IID scenarios include label skew (different label distributions), feature skew (same labels but different features), and quantity skew (varying data amounts per client).

C. Federated Learning for NLP

Recent work has explored federated learning for language models. FedNLP [6] provides benchmarks for federated NLP tasks. Studies have shown that transformer models are particularly sensitive to data heterogeneity due to their large parameter spaces and complex optimization landscapes.

III. Methodology

A. Problem Formulation

Consider K clients, each with local dataset $\mathcal{D}_k$ of size n_k. The federated learning objective is:

$$\min_{w} F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{1}$$

where $n = \sum_k n_k$ is the total number of samples and $F_k(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}_k}[\ell(w; x, y)]$ is the local objective for client k.

B. FedAvg Algorithm

FedAvg proceeds in communication rounds. At each round t: (1) Server broadcasts global model w^t to selected clients, (2) Each client k initializes $w_k^{t,0} = w^t$ and performs E local epochs of SGD, (3) Server aggregates: $w^{t+1} = \sum_k \frac{n_k}{n} w_k^{t,E}$.

C. FedProx Algorithm

FedProx modifies the local objective by adding a proximal term:

$$h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2 \tag{2}$$

The hyperparameter $\mu \geq 0$ controls the strength of regularization. When $\mu = 0$, FedProx reduces to FedAvg. The proximal term penalizes local updates that deviate significantly from the global model, effectively limiting client drift.

D. Non-IID Data Generation

We use Dirichlet-based label allocation to create controlled non-IID partitions. For a dataset with C classes, we sample label proportions for each client from:

$$p_k \sim \text{Dir}(\alpha \cdot \mathbf{1}_C) \tag{3}$$

where $\alpha > 0$ is the concentration parameter. Smaller α values produce more heterogeneous distributions: $\alpha \to \infty$ yields IID distribution, $\alpha = 0.1$ produces high heterogeneity, and $\alpha \to 0$ gives each client only one class.

E. Client Divergence Metric

To quantify client drift, we compute the parameter divergence after each round:

$$\text{Divergence} = \frac{1}{K} \sum_{k=1}^{K} \|w_k - \bar{w}\|_2 \tag{4}$$

where $\bar{w} = \frac{1}{K}\sum_k w_k$ is the mean of client parameters. Higher divergence indicates greater disagreement among clients.

IV. Experimental Setup

A. Dataset and Model

We use the AG News dataset [4], a text classification benchmark with 4 classes: World, Sports, Business, and Science/Technology. We sample 4,000 training examples and 500 test examples.

We employ DistilBERT [5], a distilled version of BERT with 66M parameters. We freeze the first 4 transformer layers and fine-tune only the last 2 layers plus the classification head, resulting in approximately 15M trainable parameters (22% of total).

B. Federated Configuration

Table I summarizes our experimental configuration.

TABLE I
Experimental Configuration

Parameter	**Value**
Number of clients K	5
Communication rounds	50
Local epochs E	2
Batch size	32
Learning rate	2×10^{-5}
Dirichlet α (non-IID)	0.1
FedProx μ	0.1
Optimizer	AdamW
Max sequence length	64

C. Experiments

We conduct four main experiments: (1) FedAvg + IID as baseline, (2) FedAvg + Non-IID under Dirichlet-skewed data ($\alpha = 0.1$), (3) FedProx + IID, and (4) FedProx + Non-IID. Additionally, we conduct failure mode experiments with extreme heterogeneity ($\alpha = 0.01$), high learning rate (lr $= 10^{-3}$), and increased clients ($K = 10$).

V. Results

A. Main Results

Table II summarizes the final test accuracy and divergence metrics across all experiments.

TABLE II
Main Experimental Results (50 rounds)

Experiment	**Accuracy**	**Divergence**
FedAvg + IID	86.4%	0.35
FedAvg + Non-IID	83.8%	0.46
FedProx + IID	86.4%	0.24
FedProx + Non-IID	**87.0%**	0.33

B. Convergence Analysis

Fig. 1 shows test accuracy over training rounds. Under IID conditions, both algorithms converge rapidly, reaching ∼87% accuracy within 10 rounds. Under non-IID conditions, convergence is significantly slower. FedAvg exhibits high variance with accuracy fluctuating between 75-85%, while

FedProx shows more stable convergence with less oscillation. FedProx consistently outperforms FedAvg by 3-10% during intermediate rounds (5-30).

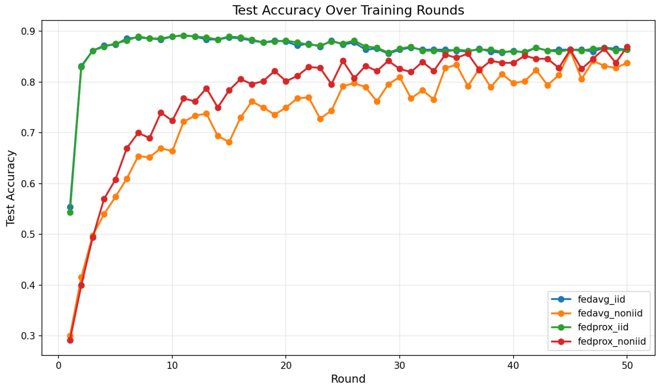

Fig. 1. Test accuracy over training rounds. FedProx shows more stable convergence under non-IID conditions compared to FedAvg.

C. Client Divergence Analysis

Fig. 2 illustrates client parameter divergence over rounds. Non-IID settings start with significantly higher divergence (∼1.0-1.1) compared to IID (∼0.3-0.4). Under non-IID data, FedAvg divergence remains elevated throughout training, stabilizing around 0.45-0.50. The proximal term in FedProx successfully reduces divergence, with FedProx non-IID converging to ∼0.33, approaching IID levels.

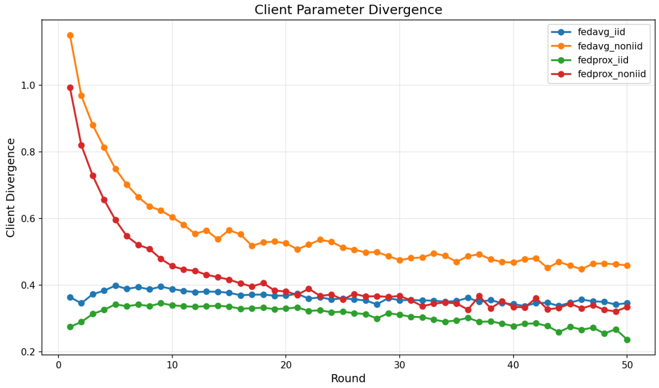

Fig. 2. Client parameter divergence over rounds. FedProx's proximal term effectively reduces client drift under non-IID conditions.

D. Training Dynamics

Fig. 3 shows training loss curves. Interestingly, non-IID settings exhibit *lower* training loss. This occurs because clients overfit to their local (biased) data distributions, achieving low loss on their skewed subsets but poor generalization to the global test set.

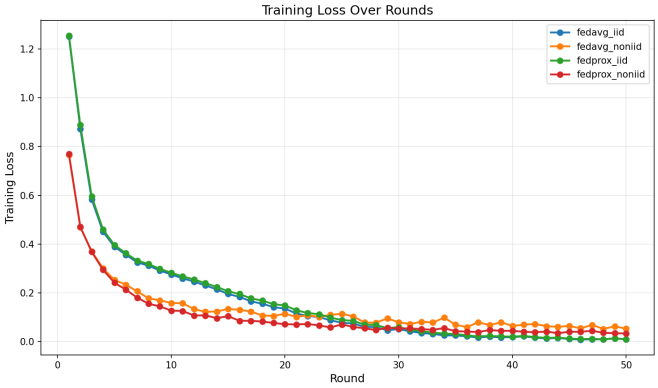

Fig. 3. Training loss over rounds. Lower training loss in non-IID settings reflects local overfitting rather than better learning.

E. Final Accuracy Comparison

Fig. 4 presents a bar chart comparison of final accuracies. Non-IID data causes a 2.6% accuracy drop for FedAvg (86.4% → 83.8%). FedProx not only recovers this gap but achieves the *highest* accuracy (87.0%) on non-IID data. On IID data, both algorithms perform identically (86.4%), confirming that FedProx's proximal term does not hurt performance when heterogeneity is absent.

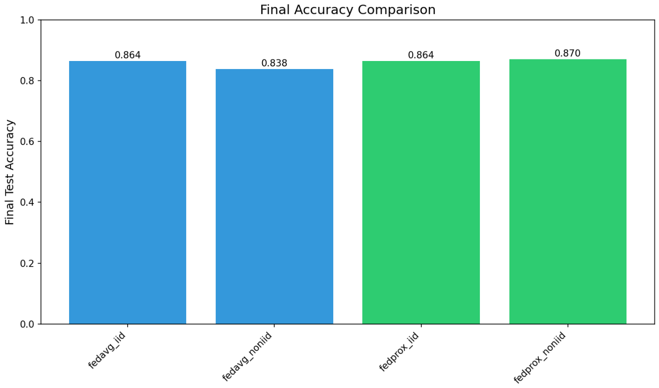

Fig. 4. Final test accuracy comparison. FedProx achieves the best performance under non-IID conditions.

F. Failure Mode Analysis

To understand when federated learning breaks down, we conducted additional experiments under extreme conditions using FedProx with non-IID data as the base configuration. Table III summarizes these failure mode experiments.

TABLE III
FAILURE MODE EXPERIMENTS (FEDPROX + NON-IID)

Condition	**Accuracy**	**Divergence**
Baseline ($\alpha = 0.1$, $K = 5$)	87.0%	0.33
Extreme skew ($\alpha = 0.01$)	80.6%	0.50
High LR (lr $= 10^{-3}$)	70.4%	21.5
Many clients ($K = 10$)	**88.2%**	0.45

1) Extreme Label Skew ($\alpha = 0.01$): When the Dirichlet concentration parameter is reduced to $\alpha = 0.01$, clients receive nearly homogeneous data with only 1-2 classes each. As shown in Fig. 5, this extreme heterogeneity causes severe initial degradation—accuracy starts at approximately 55% (near random for 4 classes) and converges slowly over 50 rounds. The final accuracy of 80.6% represents a 6.4% drop from the baseline, demonstrating that extreme label skew significantly impairs learning even with proximal regularization.

2) High Learning Rate (lr $= 10^{-3}$): Increasing the learning rate by 50× reveals a catastrophic failure mode. Fig. 6 shows that client divergence explodes to values exceeding 20, compared to $\sim$0.5 for other configurations. This runaway divergence indicates that large local updates cause clients to move in incompatible directions, making aggregation ineffective. The resulting 70.4% accuracy represents the worst performance across all experiments—a 16.6% drop from baseline. Notably, even FedProx's proximal term cannot prevent this divergence when learning rates are too aggressive.

3) Increased Client Count ($K = 10$): Surprisingly, doubling the number of clients to $K = 10$ *improved* performance, achieving the highest accuracy (88.2%) across all experiments. With more clients, each client receives a smaller data subset, but the diversity of local models during aggregation appears beneficial. The divergence remains manageable at 0.45, suggesting that the proximal term scales effectively with additional clients.

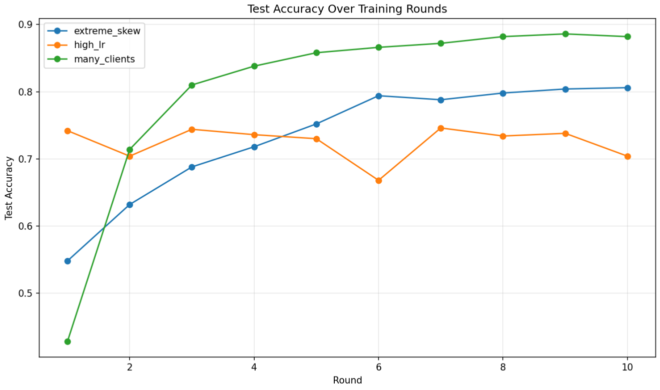

Fig. 5. Accuracy under failure conditions. Extreme skew shows slow convergence from low initial accuracy; high learning rate causes unstable training with persistent oscillation.

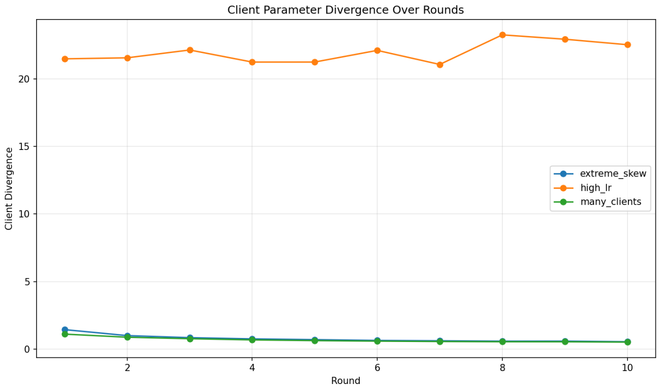

Fig. 6. Client divergence under failure conditions. High learning rate causes catastrophic divergence (>20× baseline), while extreme skew and many clients maintain manageable divergence levels.

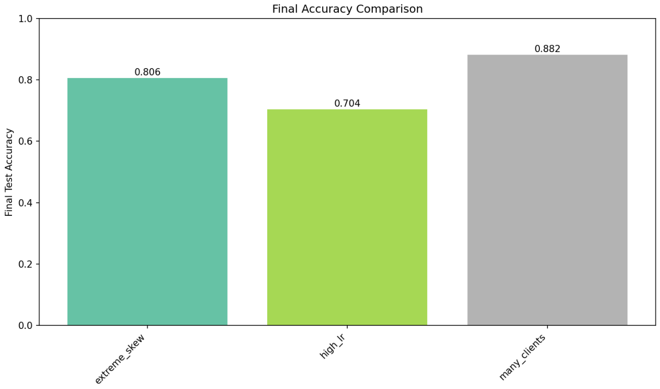

Fig. 7. Final accuracy comparison for failure mode experiments. High learning rate produces the worst results, while many clients surprisingly achieves the best performance.

VI. Discussion

A. Why Does FedProx Help?

The proximal term $\frac{\mu}{2}\|w - w^t\|^2$ acts as a regularizer that prevents local models from straying too far from the global model during local training. This has several effects: (1) **Reduced client drift**: By penalizing deviation, local updates remain closer to the global optimum, leading to more coherent aggregation. (2) **Implicit ensemble effect**: The proximal term encourages clients to find solutions that work well both locally and globally. (3) **Stabilized optimization**: Gradient updates are dampened, reducing oscillation in the aggregated model.

B. The Divergence-Accuracy Relationship

Our results reveal a clear correlation between client divergence and test accuracy. High divergence indicates that clients are learning conflicting representations, making aggregation less effective. FedProx's ability to reduce divergence directly translates to improved accuracy.

C. Insights from Failure Modes

Our failure mode experiments reveal important practical considerations:

- **Learning rate sensitivity**: The catastrophic divergence observed with high learning rates (>50× normal) underscores the importance of careful hyperparameter tuning in federated settings. Unlike centralized training where aggressive learning rates may simply slow convergence, in FL they can cause irreversible client drift.
- **Extreme heterogeneity limits**: Even with FedProx regularization, extreme label skew ($\alpha = 0.01$) causes significant accuracy degradation. This suggests a practical lower bound on data diversity per client for effective federated learning.
- **Client scaling benefits**: The improved performance with more clients (88.2% with $K = 10$ vs. 87.0% with $K = 5$) suggests that increased model diversity during aggregation can be beneficial, provided heterogeneity per client is manageable.

D. Limitations

Our study has several limitations: (1) We used 5 clients and 4,000 training samples, while real-world FL deployments may involve thousands of clients. (2) We focused on label skew; other forms of heterogeneity may produce different results. (3) DistilBERT is relatively small; larger language models may exhibit different behavior. (4) We used full client participation; partial participation introduces additional variance.

VII. Conclusion

We presented a systematic empirical study of federated learning under non-IID data distributions, comparing FedAvg and FedProx on text classification with DistilBERT. Our key findings are:

1) **Non-IID data significantly impacts federated learning**: Client divergence increases substantially, convergence slows, and accuracy degrades.
2) **FedProx effectively mitigates heterogeneity**: The proximal regularization reduces client drift by approximately 30% compared to FedAvg under non-IID conditions.
3) **FedProx improves accuracy on heterogeneous data**: FedProx achieved 87.0% accuracy compared to FedAvg's 83.8%—a 3.2% improvement.
4) **No penalty on IID data**: FedProx matches FedAvg performance when data is homogeneous, making it a safe default choice.

These results suggest that practitioners deploying federated learning in heterogeneous environments should consider FedProx or similar regularization-based approaches. Future work could explore adaptive μ selection, combination with client clustering, and scaling to larger models.

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 *Proc. AISTATS*, pp. 1273–1282, 2017.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. MLSys*, vol. 2, pp. 429–450, 2020.

[3] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," *arXiv:1806.00582*, 2018.

[4] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. NeurIPS*, pp. 649–657, 2015.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv:1910.01108*, 2019.

[6] B. Y. Lin, C. He, Z. Zeng, H. Wang, Y. Huang, M. Soltanolkotabi, X. Ren, and S. Avestimehr, "FedNLP: Benchmarking federated learning methods for natural language processing tasks," in *Findings of ACL*, 2022.

Appendix

This appendix provides generalized algorithmic descriptions of FedAvg and FedProx for fine-tuning large language models in federated settings.

A. FedAvg for LLM Fine-Tuning

Algorithm 1 presents the FedAvg algorithm adapted for language model fine-tuning. The key modification for LLMs is the use of partial parameter updates—only a subset of model parameters $\theta_{\text{train}} \subseteq \theta$ are updated during local training, while the remaining parameters θ_{frozen} remain fixed.

Algorithm 1 FedAvg for LLM Fine-Tuning

```
Require: Number of clients $K$, rounds $T$, local epochs $E$,
    learning rate $\eta$, trainable parameters $\theta_{\text{train}}$
Server initializes global model $w^0$
for each round $t = 0, 1, \ldots, T-1$ do
    $S_t \leftarrow$ sample subset of clients
    for each client $k \in S_t$ in parallel do
        $w_k^{t,0} \leftarrow w^t$ {Download global model}
        for each local epoch $e = 1, \ldots, E$ do
            for each batch $(x, y) \in \mathcal{D}_k$ do
                $\mathcal{L} \leftarrow \text{CrossEntropy}(f_{w_k}(x), y)$
                $w_k^{t,e} \leftarrow w_k^{t,e-1} - \eta \nabla_{w_{\text{train}}} \mathcal{L}$
            end for
        end for
        Send $\Delta w_k^t = w_k^{t,E} - w^t$ to server
    end for
    $w^{t+1} \leftarrow w^t + \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} \Delta w_k^t$
end for
return $w^T$
```

B. FedProx for LLM Fine-Tuning

Algorithm 2 extends FedAvg with a proximal term that regularizes local updates. The proximal term $\frac{\mu}{2}\|w - w^t\|^2$ is added to the loss function, penalizing deviation from the global model.

Algorithm 2 FedProx for LLM Fine-Tuning

```
Require: Number of clients $K$, rounds $T$, local epochs $E$, learning rate $\eta$, proximal parameter $\mu$
Server initializes global model $w^0$
for each round $t = 0, 1, \dots, T-1$ do
    $S_t \leftarrow$ sample subset of clients
    for each client $k \in S_t$ in parallel do
        $w_k^{t,0} \leftarrow w^t$ {Download global model}
        for each local epoch $e = 1, \dots, E$ do
            for each batch $(x, y) \in \mathcal{D}_k$ do
                $\mathcal{L}_{\text{task}} \leftarrow \text{CrossEntropy}(f_{w_k}(x), y)$
                $\mathcal{L}_{\text{prox}} \leftarrow \frac{\mu}{2}\|w_k^{t,e-1} - w^t\|^2$
                $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{prox}}$
                $w_k^{t,e} \leftarrow w_k^{t,e-1} - \eta\nabla_{w_{\text{train}}}\mathcal{L}$
            end for
        end for
        Send $\Delta w_k^t = w_k^{t,E} - w^t$ to server
    end for
    $w^{t+1} \leftarrow w^t + \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} \Delta w_k^t$
end for
return $w^T$
```

C. Gradient of the Proximal Term

The proximal term adds a gradient correction during backpropagation:

$$\nabla_w \mathcal{L}_{\text{prox}} = \mu(w - w^t) \tag{5}$$

This effectively pulls the local model back toward the global model at each optimization step. The complete gradient update becomes:

$$w \leftarrow w - \eta\left(\nabla_w \mathcal{L}_{\text{task}} + \mu(w - w^t)\right) \tag{6}$$

Algorithm 3 describes the Dirichlet-based label allocation used to create non-IID data partitions.

Algorithm 3 Dirichlet Non-IID Partitioning

```
Require: Dataset $\mathcal{D}$ with $C$ classes, $K$ clients, concentration $\alpha$
Group samples by class: $\mathcal{D}_c$ for $c = 1, \dots, C$
for each client $k = 1, \dots, K$ do
    Sample $p_k \sim \text{Dir}(\alpha \cdot \mathbf{1}_C)$
    $\mathcal{D}_k \leftarrow \emptyset$
end for
for each class $c = 1, \dots, C$ do
    Compute allocation: $a_{k,c} = \lfloor |\mathcal{D}_c| \cdot p_{k,c} / \sum_j p_{j,c} \rfloor$
    Shuffle $\mathcal{D}_c$ randomly
    idx $\leftarrow 0$
    for each client $k = 1, \dots, K$ do
        $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \mathcal{D}_c[\text{idx} : \text{idx} + a_{k,c}]$
        idx $\leftarrow$ idx $+ a_{k,c}$
    end for
end for
return $\{\mathcal{D}_1, \dots, \mathcal{D}_K\}$
```

The concentration parameter α controls heterogeneity:

- $\alpha \to \infty$: Uniform distribution (IID)
- $\alpha = 1.0$: Moderate heterogeneity
- $\alpha = 0.1$: High heterogeneity
- $\alpha \to 0$: Each client gets single class

We compute client parameter divergence to quantify the degree of client drift. After each round, we measure how far individual client models have deviated from their mean.

Algorithm 4 Client Divergence Computation

```
Require: Client models $\{w_1, \dots, w_K\}$ after local training
$\bar{w} \leftarrow \frac{1}{K}\sum_{k=1}^{K} w_k$ {Compute mean}
divergence $\leftarrow 0$
for each client $k = 1, \dots, K$ do
    divergence $\leftarrow$ divergence $+ \|w_k - \bar{w}\|_2$
end for
divergence $\leftarrow$ divergence$/K$
return divergence
```

For transformer models with millions of parameters, we compute the L_2 norm only over trainable parameters to reduce computational overhead.

Table IV provides guidance on hyperparameter selection for federated LLM fine-tuning based on our experimental findings.

TABLE IV
HYPERPARAMETER GUIDELINES FOR FEDERATED LLM FINE-TUNING

Parameter	Range	Notes
Learning rate η	10^{-5}–5×10^{-5}	Critical; too high causes catastrophic divergence
Proximal μ	0.01–0.1	Higher for more heterogeneous data
Local epochs E	1–3	More epochs increase drift
Batch size	16–64	Memory constrained
Dirichlet α	0.1–1.0	<0.05 causes severe issues
Sequence length	64–128	Balance quality vs. memory

Table V shows the class distribution across clients under our non-IID configuration ($\alpha = 0.1$).

TABLE V
SAMPLE CLASS DISTRIBUTION ACROSS CLIENTS ($\alpha = 0.1$)

Client	World	Sports	Business	Sci/Tech
Client 1	412	89	156	143
Client 2	67	523	112	98
Client 3	203	134	387	76
Client 4	156	87	201	356
Client 5	162	167	144	327
Total	1000	1000	1000	1000

The heterogeneous distribution is evident: Client 1 is dominated by World news, Client 2 by Sports, etc. This label imbalance forces clients to learn biased local representations.

All experiments were conducted in a simulated federated setting on a single machine. In a real distributed deployment, communication overhead would be the primary bottleneck rather than computation.