Federated Learning Under the Lens of Task Arithmetic: Mitigating Interference via Sparse Fine-Tuning

Ferdinando Del Vecchio (s347426)
s347426@studenti.polito.it

Giacomo Di Palma (s347125)
s347125@studenti.polito.it

Marco Donatucci (s337624)
s337624@studenti.polito.it

Emanuele Romano (s346325)
s346325@studenti.polito.it

Abstract

Federated Learning (FL) enables collaborative model training across distributed clients without sharing raw data, preserving privacy by design. However, statistical heterogeneity, where clients have non-identically distributed data, remains a major challenge, causing interference during model aggregation and degrading global performance. In this work, we investigate the application of sparse fine-tuning techniques to address data heterogeneity in Federated Learning. Using a pretrained Vision Transformer backbone on the CIFAR-100 dataset, we implement a federated framework following the FedAvg protocol and introduce a gradient masking mechanism based on Fisher Information sensitivity scores. This approach restricts local updates to parameters that are least sensitive to pretrained knowledge, aiming to reduce interference between clients. We conduct extensive experiments comparing centralized and federated baselines under various data heterogeneity conditions, and evaluate five different masking strategies. Our ablation studies examine the effects of calibration rounds and sparsity levels on model convergence. Results demonstrate that sparse fine-tuning can improve generalization under severe data heterogeneity, providing insights into the trade-offs between sparsity and model robustness in federated settings.

1. Introduction

The proliferation of edge devices and growing privacy concerns have driven the adoption of Federated Learning (FL) as a paradigm for training machine learning models on decentralized data [9]. By keeping data local and sharing only model updates, FL enables collaborative learning while respecting user privacy. However, this distributed setting introduces a fundamental challenge: client data is rarely independent and identically distributed (IID). In practice, users generate data that reflects their individual behaviours, preferences, and environments, leading to significant statistical heterogeneity across the federation [5].

This heterogeneity has profound implications for model training. When clients optimize on their local distributions, their updates can point in conflicting directions, a phenomenon known as "client drift" [6]. Upon aggregation, these divergent updates interfere with one another, slowing convergence and degrading the quality of the global model. The more heterogeneous the data, the more severe this interference becomes, often rendering standard federated algorithms ineffective in real-world deployments [2].

Recent advances in model editing and task arithmetic have revealed a promising direction for addressing such interference [4]. The key insight is that not all parameters contribute equally to a model's knowledge: some are highly specialized to specific tasks, while others are more general. By identifying and selectively updating only certain parameters during fine-tuning, it becomes possible to adapt models to new tasks while preserving their original capabilities. This principle of *sparse fine-tuning* has shown remarkable success in mitigating interference when composing knowledge from multiple sources.

In this work, we bring these insights from model editing into the Federated Learning domain. We propose that the interference caused by heterogeneous client updates can be viewed through the lens of task composition: each client essentially fine-tunes the global model on their local "task." By restricting updates to parameters that are least sensitive to the pretrained knowledge, we aim to reduce the conflicts that arise during aggregation. This approach offers a principled way to leverage powerful pretrained representations while maintaining the collaborative benefits of federated learning.

We evaluate our approach on the CIFAR-100 image classification benchmark using a pretrained Vision Transformer backbone. Through comprehensive experiments, we compare our sparse fine-tuning strategy against standard federated baselines under varying degrees of data heterogeneity.

We further explore different criteria for selecting which parameters to update, providing insights into the relationship between parameter sensitivity and client drift mitigation.

2. Related Works

Federated Learning (FL) has emerged as a critical paradigm for decentralized machine learning, enabling the training of models across distributed devices without sharing raw data, thereby embodying principles of data minimization [9]. However, practical FL deployments face significant hurdles regarding statistical heterogeneity (non-IID data) and system heterogeneity (varying computational resources). This section reviews the foundational algorithms, optimization strategies, and architectural considerations developed to address these challenges.

2.1. Foundations and Heterogeneity Challenges

The standard algorithm for FL is Federated Averaging (FedAvg), introduced by McMahan *et al*. [9]. FedAvg aggregates locally computed updates (via Stochastic Gradient Descent) from participating clients to refine a shared global model. While FedAvg has proven robust in various settings, its performance degrades significantly when client data is not independent and identically distributed (non-IID) or when data quantities are unbalanced [9].

Kairouz *et al*. [5] provide a comprehensive taxonomy of these challenges, identifying non-IID data and system constraints (such as bandwidth and device availability) as primary obstacles to FL scalability. To address heterogeneity, Li *et al*. [7] proposed FedProx. This framework generalizes FedAvg by introducing a proximal term to the local objective function. This term restricts the deviation of local updates from the global model, thereby improving convergence stability in scenarios characterized by both statistical and systems heterogeneity.

2.2. Advanced Optimization Algorithms

A major issue caused by statistical heterogeneity is "client-drift," where local updates move towards local optima rather than the global optimum. To mitigate this, Karimireddy *et al*. [6] introduced SCAFFOLD (Stochastic Controlled Averaging). SCAFFOLD utilizes control variates (variance reduction) to estimate the update direction of the server and clients, correcting the drift in local updates. Theoretical analysis and experiments demonstrate that SCAFFOLD requires significantly fewer communication rounds compared to FedAvg and is resilient to client sampling.

In parallel, Reddi *et al*. [12] proposed a framework for Adaptive Federated Optimization. While FedAvg effectively uses SGD on the server side, adaptive optimizers like Adam, Adagrad, and Yogi can be adapted for the server update step. Algorithms such as FedAdam, FedYogi, and FedAdagrad apply adaptive learning rates to the aggregated server updates, which helps in settings with sparse gradients or heavy-tail noise distributions, common in language and text tasks.

2.3. Handling Feature Shifts and Architecture Design

While optimization algorithms tackle weight updates, other approaches focus on the model architecture's response to distribution shifts. For scenarios where data heterogeneity manifests as feature shifts (*e.g*., different visual appearances due to different acquisition devices), Li *et al*. [8] proposed FedBN. FedBN keeps Batch Normalization (BN) parameters local to each client rather than aggregating them on the server. This strategy mitigates feature shifts and has been shown to outperform FedAvg and FedProx in non-IID benchmarks.

Recent work by Qu *et al*. [11] investigates the impact of the backbone architecture itself, comparing Convolutional Neural Networks (CNNs) with Vision Transformers (ViTs) in federated settings. Their findings suggest that self-attention-based architectures are inherently more robust to distribution shifts and severe occlusions than CNNs. Consequently, replacing CNNs with Transformers can reduce catastrophic forgetting and accelerate convergence in heterogeneous environments.

2.4. Data Distribution and Realistic Benchmarks

Understanding the nature of non-IID data is crucial. Hsu *et al*. [2] analyzed the effects of non-identical distributions by synthesizing datasets using Dirichlet distributions to control heterogeneity. They observed that as data becomes more non-identical, the accuracy of FedAvg degrades, and proposed FedAvgM (FedAvg with Server Momentum) as a mitigation strategy.

Further extending this to real-world scenarios, Hsu *et al*. [3] introduced large-scale benchmarks derived from datasets like iNaturalist and Google Landmarks, which feature natural user-based partitions. They proposed two algorithms to handle the long-tailed class distributions and data imbalance found in the wild: FedIR (Importance Reweighting), which applies importance weights to local objectives, and FedVC (Virtual Clients), which resamples and splits clients to normalize resource utilization.

2.5. Privacy and System Constraints

Finally, the implementation of FL must respect privacy and fairness constraints. Chen *et al*. [1] highlight the trade-offs between privacy and fairness, noting that while FL protects data locality, the resulting models must be carefully tuned to avoid bias against specific demographic groups. Additionally, Pfeiffer *et al*. [10] survey the challenges of computationally constrained devices, categorizing heterogeneity into hard constraints (*e.g*., memory) and soft constraints

(*e.g.*, training speed). They emphasize that dealing with stragglers and heterogeneous device capabilities often requires asynchronous aggregation or specialized model splitting techniques like Split Learning.

2.6. Task-Localized Sparse Fine-tuning

Recent advances in efficient model editing have shown that sparse fine-tuning can effectively update pretrained models while preserving knowledge. Iurada *et al*. [4] proposed a task-localized sparse fine-tuning approach that identifies and updates only a subset of parameters most relevant to a given task. This paradigm leverages Fisher Information scores to identify parameter sensitivity, enabling efficient adaptation with minimal interference to existing knowledge. Such techniques are particularly relevant for federated learning settings, where communication efficiency and computational constraints are paramount concerns.

3. Methodology

This section details the technical implementation of our experiments. We describe the baseline setups, data partitioning strategies, the sparse fine-tuning pipeline, and our extension comparing different parameter selection rules.

3.1. Experimental Setup

3.1.1. Centralized Baseline

As an upper bound for federated performance, we implement centralized training where all data is available on a single machine. We use a pretrained DINO ViT-S/16 backbone and optimize a CIFAR-100 classifier using cross-entropy loss.

To mirror the federated protocol, we adopt a two-stage training procedure:

1. **Head pre-training.** We freeze the DINO backbone and train only a linear classification head for 20 epochs using SGDM with Cosine Annealing. The best validation checkpoint is saved.
2. **Backbone fine-tuning.** Starting from the pretrained head, we freeze the classifier and fine-tune the entire backbone for 40 epochs using SGDM with a lower learning rate and Cosine Annealing.

3.1.2. Federated Learning Simulation

To simulate distributed training on a single GPU, we implement sequential client training. In each communication round:

1. The server broadcasts the global model weights to selected clients.
2. Each client trains independently on its local data partition for J local steps (minibatch gradient updates).
3. The server collects and aggregates client updates via weighted averaging based on each client's sample count.

We adopt Federated Averaging (FedAvg) as our aggregation strategy. We fix $K = 100$ total clients and a participation fraction of $C = 0.1$, meaning 10 clients are sampled per round.

3.2. Dataset Partitioning

3.2.1. Train/Validation Split

We use CIFAR-100, which contains 50,000 training images across 100 classes. Images are resized to 224×224 pixels and normalized with ImageNet statistics; during training we apply standard augmentations (random crop, horizontal flip, and RandAugment). Before client partitioning, we reserve 10% of the training data (5,000 images) as a global validation set for model selection. The remaining 45,000 images are distributed among clients.

3.2.2. IID Sharding

For IID (independent and identically distributed) partitioning, the training set is shuffled and uniformly divided among all clients. Each client receives approximately equal representation of all 100 classes, mimicking the global distribution.

3.2.3. Non-IID Sharding

To simulate label distribution skew, we restrict each client to samples from only N_c distinct classes. The parameter N_c controls heterogeneity severity:

- $N_c = 1$: Each client sees only one class (extreme heterogeneity).
- $N_c = 5, 10, 50$: Intermediate levels of heterogeneity.
- $N_c = 100$ (IID): Each client sees all classes.

This procedure yields label-skewed client datasets while keeping the overall sample counts approximately balanced.

3.2.4. Scaled Rounds

When increasing local steps J, we proportionally reduce communication rounds to maintain constant total computation. We fix a reference budget at $J = 4$ local steps and 200 rounds (800 total steps), and scale accordingly for $J \in \{4, 8, 16\}$. This enables fair comparison across different values of J.

3.3. Sparse Fine-Tuning Pipeline

To mitigate interference between heterogeneous clients, we restrict model updates to a subset of parameters during federated training. Our pipeline consists of mask calibration followed by masked local training.

3.3.1. Parameter Sensitivity via Fisher Information

We identify parameter importance using the diagonal of the empirical Fisher Information Matrix. This metric estimates how much each parameter affects the model's predictions. High Fisher values indicate parameters critical to the pretrained knowledge, while low values indicate parameters

that can be modified with minimal impact on existing capabilities.

3.3.2. Multi-Round Mask Calibration

We compute Fisher scores over multiple calibration rounds rather than a single pass. Each calibration round samples a subset of clients and estimates Fisher scores on each; we then average estimates across clients and accumulate across rounds. This multi-round approach yields more stable sensitivity estimates. After calibration, we apply global thresholding to create a binary mask at the desired sparsity ratio.

3.3.3. SparseSGDM Optimizer

We implement SparseSGDM, extending standard SGD with momentum to incorporate gradient masking. The optimizer applies the binary mask to both gradients and the weight decay term, ensuring that masked parameters receive no updates and remain frozen throughout training. In the federated setting, all clients share the same global mask, ensuring they update identical parameter subsets. This coordination maintains compatibility during aggregation even when local data distributions differ substantially.

3.4. Extension: Alternative Masking Rules

The core of our extension is a systematic comparison of different criteria for selecting which parameters to update. While the standard approach uses Fisher Information to select the least-sensitive parameters, we investigate whether alternative selection rules might be effective in the federated setting.

3.4.1. Masking Strategies

Given a sparsity ratio (fraction of parameters to freeze), we compare five selection rules:

Least-Sensitive (Baseline). Select parameters with the lowest Fisher scores. Rationale: these parameters contribute least to the pretrained knowledge and can be safely modified without catastrophic forgetting.

Most-Sensitive. Select parameters with the highest Fisher scores. Rationale: these parameters are most task-relevant and updating them may yield faster adaptation.

Lowest-Magnitude. Select parameters with the smallest absolute weight values. Rationale: near-zero weights may be redundant and modifying them could add capacity without disrupting existing representations.

Highest-Magnitude. Select parameters with the largest absolute weight values. Rationale: large weights dominate forward passes; updating them may have the strongest effect on model behavior.

Random. Select parameters uniformly at random. This serves as a baseline to evaluate whether intelligent parameter selection provides meaningful benefits over naive approaches.

3.4.2. Implementation Details

For Fisher-based methods, scores are computed using the multi-round calibration procedure described above. For magnitude-based methods, we use the absolute values of the pretrained weights. For all methods, we apply global thresholding: we collect all scores, sort them, and select parameters according to the rule's criterion. This produces a binary mask that is distributed to all clients before federated training begins.

4. Experiments

In this section, we present the experimental evaluation of our proposed methods. We first detail the experimental setup, including the dataset and model architecture. We then establish strong centralized and federated baselines to benchmark performance. Finally, we analyze the impact of our sparse fine-tuning strategy through ablation studies and a comparative analysis of different gradient masking rules.

4.1. Experimental Setup

Dataset and Model. We conduct all experiments on the **CIFAR-100** dataset, which consists of 60,000 32×32 color images in 100 classes, with 500 training images and 100 test images per class. We use a pre-trained **DINO ViT-S/16** backbone with a linear classification head. We adopted a two-stage training strategy: first, we froze the backbone and trained only the linear head; subsequently, we froze the trained head and fine-tuned the entire backbone. All models are trained using the SGD optimizer with momentum (SGDM).

Federated Setting. We simulate a federated environment with $K = 100$ clients and a participation rate of $C = 0.1$ (10 clients per round). We explore both IID and non-IID data partitions. For non-IID settings, we restrict each client to hold data from only N_c classes, creating statistical heterogeneity.

4.2. Centralized Baseline

To establish an upper bound on performance, we evaluated the model in a centralized setting. We compared four learning rate schedulers: Cosine Annealing, ReduceLROnPlateau, Exponential, and StepLR. The results are summarized in Table 1.

As shown in Table 1, the **Cosine Annealing** scheduler achieved the highest final test accuracy. We observed that the model converges rapidly. Consequently, we adopt Cosine Annealing for subsequent experiments.

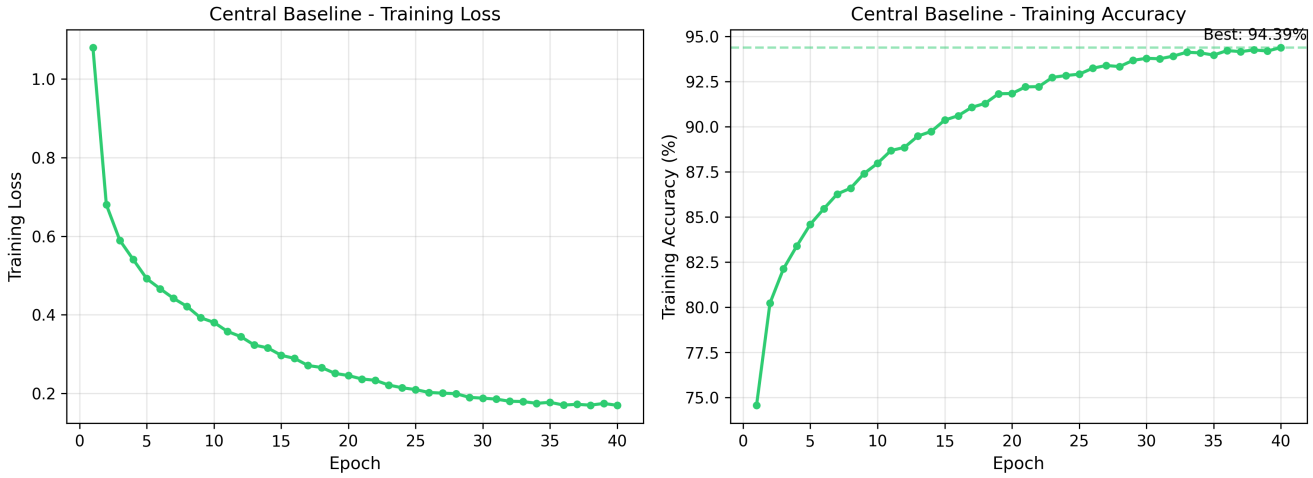

Figure 1. Centralized baseline training curves comparing different learning rate schedulers.

Table 1. Centralized Baseline Scheduler Comparison

Scheduler	Test Accuracy (%)
Cosine	86.18
Plateau	85.87
Exponential	85.96
Step	85.68

4.3. Federated Learning Baselines

We next evaluate the standard FedAvg algorithm under various conditions to understand the impact of data heterogeneity and local computation.

We varied the number of classes per client $N_c \in \{1, 5, 10, 50\}$ to simulate varying degrees of non-IID data and investigated the effect of the number of local update steps $J \in \{4, 8, 16\}$. Figure 2 provides a comprehensive summary of these results.

We observe a significant performance degradation as heterogeneity increases (N_c decreases), while increasing local steps generally benefits performance in IID or near-IID settings but can be detrimental in highly heterogeneous environments due to client drift.

4.4. Sparse Fine-Tuning Optimizations

4.4.1. Ablation Studies

We conducted ablation studies to determine the optimal hyperparameters for our sparse fine-tuning method, specifically the number of Fisher calibration rounds and the sparsity ratio.

As shown in Table 2 and Figure 3, calculating Fisher Information over **3 rounds** is sufficient for stable masking. Regarding sparsity, we found that maintaining **20% sparsity** provided the optimal balance between plasticity and stability.

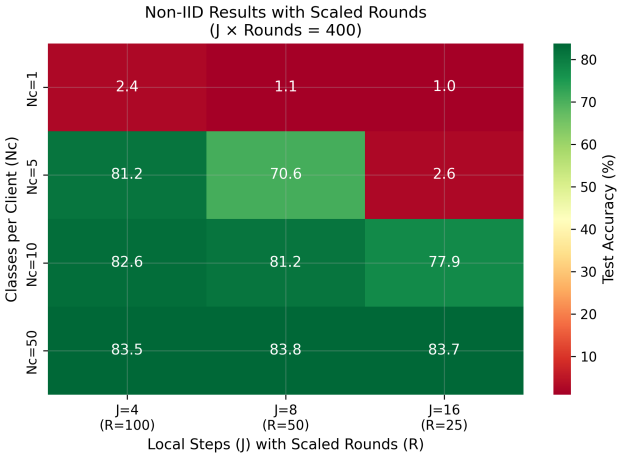

Figure 2. Impact of statistical heterogeneity (N_c) and local steps (J) on test accuracy.

Table 2. Sparse Ablation: Sparsity Levels and Calibration Rounds

Type	Value	Test Accuracy (%)
Calibration Rounds	1	77.87
Calibration Rounds	3	77.91
Calibration Rounds	5	77.91
Calibration Rounds	10	77.90
Sparsity	20%	80.10
Sparsity	50%	77.92
Sparsity	90%	73.81

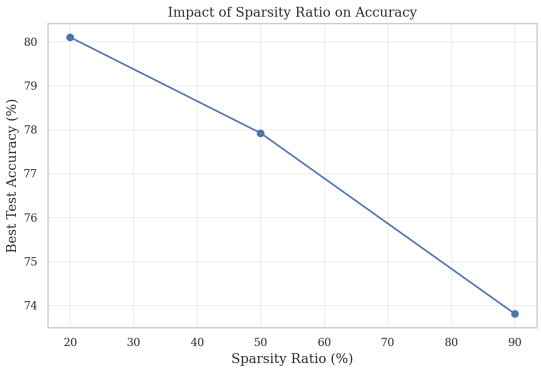

Figure 3. Ablation study results for calibration rounds and sparsity levels.

4.5. Extension: Comparison of Masking Rules

Finally, we evaluated five different strategies for selecting which parameters to update (Masking Rules) in a non-IID setting. We compared:

- **Least Sensitive:** Updates parameters with the smallest Fisher Information.
- **Most Sensitive:** Updates parameters with the largest Fisher Information.
- **Lowest Magnitude:** Updates parameters with the smallest absolute values.
- **Highest Magnitude:** Updates parameters with the largest absolute values.
- **Random:** Selects parameters randomly.

Table 3 and Figure 4 present the comparative results.

Table 3. Comparison of Masking Rules in Non-IID Settings

N_c	J	Masking Rule	Test Accuracy (%)
5	4	Highest Magnitude	81.16
5	4	Least Sensitive	79.32
5	4	Lowest Magnitude	80.53
5	4	Most Sensitive	80.18
5	4	Random	80.61
5	8	Highest Magnitude	80.14
5	8	Least Sensitive	79.62
5	8	Lowest Magnitude	77.17
5	8	Most Sensitive	78.40
5	8	Random	78.95
10	8	Highest Magnitude	82.09
10	8	Least Sensitive	80.60
10	8	Lowest Magnitude	80.00
10	8	Most Sensitive	79.93
10	8	Random	81.17

Across the considered configurations, the **Highest Magnitude** rule generally achieved performance comparable to, and in several cases slightly higher than, the other strategies, including Fisher-based sensitivity rules. However, differences between methods are often within the variability expected from single-seed training, and should therefore be interpreted with caution.

5. Conclusion

In this work, we investigated the problem of interference in Federated Learning under statistical heterogeneity through the lens of task composition. By interpreting each client update as a local task-specific adaptation of a shared pretrained model, we reframed client interference as the uncontrolled composition of heterogeneous adaptations. This perspective naturally motivated the use of sparse fine-tuning as a structural mechanism to limit conflicting updates during aggregation.

Through extensive experiments on CIFAR-100 using a pretrained DINO ViT-S/16 backbone, we showed that restricting updates to a globally coordinated subset of parameters can improve robustness in strongly non-IID settings. In particular, sparse fine-tuning consistently mitigated the performance degradation observed in standard FedAvg as data heterogeneity increased, highlighting a clear trade-off between plasticity and stability. Our ablation studies further indicated that moderate sparsity levels and a small number of Fisher calibration rounds are sufficient to obtain stable performance, suggesting that sensitivity-based masking can be implemented with limited overhead.

Beyond performance improvements, our results provide insight into the nature of interference in federated optimization. The comparable performance observed across different masking rules suggests that the key factor in reducing interference is not solely the precise criterion used to select parameters, but rather the imposition of a shared structural constraint on model adaptation. From this perspective, global sparsity acts as a coordination mechanism among clients, ensuring that local updates remain compatible during aggregation despite heterogeneous data distributions.

This study has several limitations. Our evaluation is restricted to a single dataset, backbone architecture, and single-seed experiments, which limits the statistical strength and generality of our conclusions. Additionally, the relative effectiveness of different masking strategies warrants further investigation under broader experimental conditions.

Future work will focus on extending this analysis to larger-scale backbones, multi-seed evaluations, and more realistic federated benchmarks. Exploring adaptive or client-aware sparsity mechanisms and analyzing communication-efficiency trade-offs also represent promising directions. Overall, our findings suggest that mitigating interference in Federated Learning may benefit less from

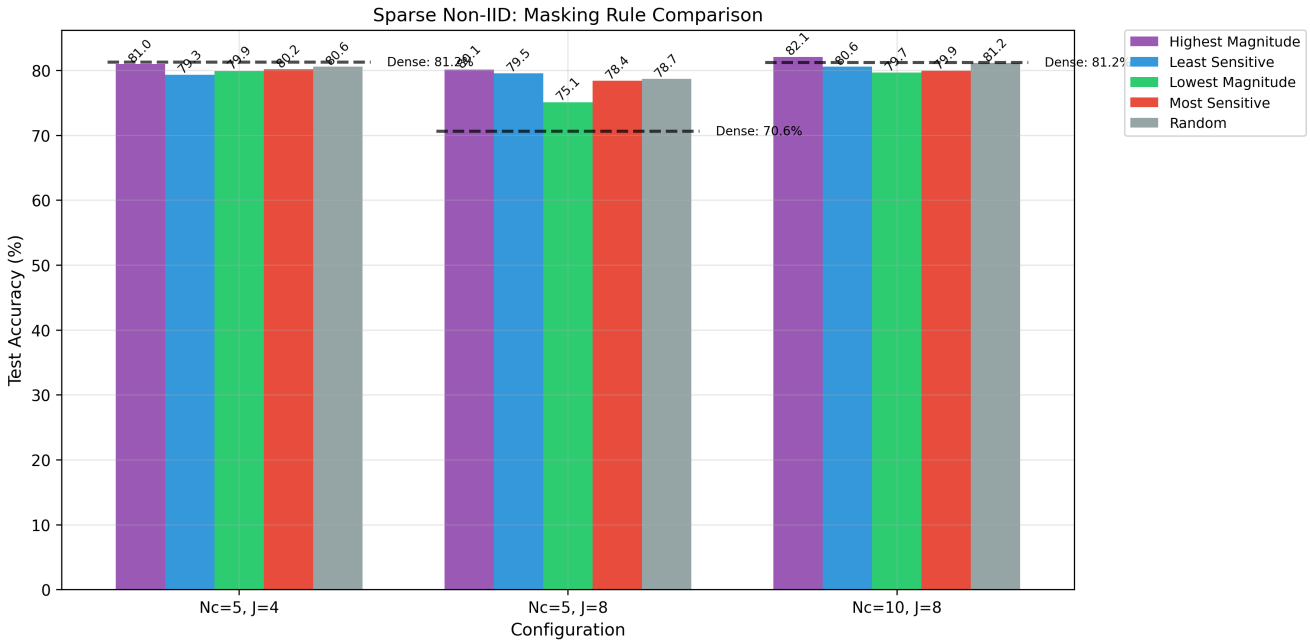

Figure 4. Performance comparison of different gradient masking rules across various non-IID settings.

increasingly complex optimization schemes and more from controlling which parts of a pretrained model are allowed to adapt in heterogeneous federated environments.

References

[1] Yongxin Chen et al. Privacy and fairness in federated learning: On the perspective of tradeoff. *ACM Computing Surveys*, 55(6):1–36, 2023. 2

[2] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 1, 2

[3] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Eur. Conf. Comput. Vis.*, pages 76–92. Springer, 2020. 2

[4] Leonardo Iurada, Marco Ciccone, and Tatiana Tommasi. Efficient model editing with task-localized sparse fine-tuning. In *Int. Conf. Learn. Represent.*, 2025. 1, 3

[5] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1, 2

[6] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, pages 5132–5143. PMLR, 2020. 1, 2

[7] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, pages 429–450, 2020. 2

[8] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *Int. Conf. Learn. Represent.*, 2021. 2

[9] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017. 1, 2

[10] Jonas Pfeiffer et al. Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Computing Surveys*, 55(4):1–38, 2023. 2

[11] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10061–10071, 2022. 2

[12] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *Int. Conf. Learn. Represent.*, 2021. 2