Federated Learning Under the Lens of Model Editing

Giovanna Brod Zamojska (s337350) Florian Magnan (s347801)
Réda Ouzzane (s347803) Niloofar Vazirpanah (s340784)

Abstract

Federated Learning (FL) enables decentralized model training without data sharing, but suffers from client drift and poor convergence under data heterogeneity. This work explores FL on CIFAR-100 using a ViT-S/16 backbone pretrained with DINO, comparing centralized training to FedAvg under both IID and non-IID settings. To mitigate heterogeneity and reduce communication costs, we introduce model editing through sparse fine-tuning, guided by sensitivity-based gradient masks.

As a personal contribution, we evaluate alternative calibration strategies for sparse updates, selecting parameters based on sensitivity, magnitude, or random sampling. Results show that the choice of calibration rule strongly impacts convergence and generalization, revealing trade-offs between parameter importance and editing stability. The code is available here.

1 Introduction

Federated Learning (FL) allows collaborative training across distributed clients while preserving data privacy. However, statistical heterogeneity and limited communication severely impact the performance of standard algorithms such as FedAvg. These issues lead to unstable updates and degraded generalization.

Recent approaches have introduced sparse training as a model editing technique to selectively update a subset of parameters, reducing drift and communication costs. This is typically done using importance scores such as gradient magnitude or Fisher Information.

We compare centralized and federated setups and study various parameter masking strategies— sensitivity-based, magnitude-based, and random—under IID and non-IID conditions. Our analysis emphasizes the critical role of calibration and the interplay between sparsity and robustness.

2 Related Work

Federated Learning (FL) was first introduced by McMahan et al. [9] as a method to train models across decentralized data silos without sharing raw inputs. Subsequent work has addressed challenges such as non-IID distributions, communication bottlenecks, and partial client participation [2, 7]

To mitigate statistical heterogeneity, strategies such as client re-weighting and personalized models have been proposed [3]. More recently, model editing approaches like sparse fine-tuning have been explored to reduce parameter conflicts and improve communication efficiency. Iurada et al. [5] proposed updating only a subset of model parameters, selected using Fisher-based sensitivity scores computed through iterative mask calibration.

This project builds on these ideas by applying sparse fine-tuning within FL and comparing alternative parameter selection strategies for gradient masking.

3 Methodology

This section outlines the methods used in the project. To ensure fair comparisons across all experiments, we fixed the random seed to 42, ensuring consistent data splits and reproducibility.

Across all experiments, we use Stochastic Gradient Descent (SGD) as the optimization algorithm for gradient updates. The learning rate is scheduled using a cosine annealing scheduler, and model training is supervised using the cross-entropy loss function.

Input images are augmented with standard techniques including random resized cropping to 224×224 and horizontal flipping, followed by normalization using ImageNet statistics.

3.1 Centralized Training

As a baseline, we fine-tuned the ViT-S/16 model by freezing the backbone and training a linear classification head on top.

This setup serves as a reference to evaluate the performance of the proposed sparse model editing and federated approaches.

3.2 Federated Learning Framework

We adopted the Federated Averaging (FedAvg) algorithm. A central server broadcasts the current global model to a randomly selected fraction of clients in each communication round. Each selected client trains locally on its private dataset and returns updated parameters, which are aggregated by the server according to:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_t^{(k)}$$

where $w_t^{(k)}$ is the model from client k, n_k is the number of local samples on client k, and $n = \sum_k n_k$. Local updates are performed using Stochastic Gradient Descent with Momentum (SGDM), following:

$$v_t = \mu v_{t-1} + \nabla_\theta \mathcal{L}(\theta_t), \quad \theta_{t+1} = \theta_t - \eta v_t$$

where μ is the momentum and η the learning rate.

3.3 Data Partitioning: IID and Non-IID

We used two partitioning schemes to simulate different data distribution scenarios. In the IID setup, the CIFAR-100 dataset is uniformly split among clients to ensure balanced class distributions. In contrast, the non-IID setting introduces label skew by assigning each client data from only $N_c \in \{1, 5, 10, 50\}$ classes. Lower values of N_c result in higher heterogeneity, simulating real-world decentralization where clients observe different data distributions. This partitioning design allows us to systematically study the effect of data heterogeneity on training dynamics.

3.4 Sparse Fine-Tuning via Model Editing

To mitigate update conflicts and reduce communication overhead, we apply sparse fine-tuning, a model editing strategy that restricts updates to a subset of the model's weights. Instead of updating all parameters, each client computes a binary mask $M \in \{0, 1\}^d$, where d is the total number of model parameters, selecting only a subset of parameters for update.

The mask is constructed using parameter sensitivity scores estimated from a diagonal approximation of the Fisher Information Matrix (FIM). In practice, this diagonal is approximated by averaging the squared gradients of the loss over batches, as follows:

$$F_i \approx \frac{1}{N} \sum_{b=1}^{N} \left(\frac{\partial L_b(\theta)}{\partial \theta_i} \right)^2 ,$$

where θ_i denotes the i-th parameter, $L_b(\theta)$ is the average loss over batch b, and N is the number of batches used for estimation. Note that this approximation squares the gradient of the averaged batch loss rather than the per-sample gradients, introducing bias that tends to underestimate the true Fisher diagonal. Despite this, the approach remains computationally efficient and suitable for large models and datasets.

The resulting mask M is applied to the gradients during training as:

$$\tilde{\nabla}_i = M_i \cdot \frac{\partial L}{\partial \theta_i},$$

where $\tilde{\nabla}_i$ is the masked gradient of parameter θ_i. Only parameters with $M_i = 1$ are updated, enforcing sparsity. This method is implemented via a custom optimizer, `SparseSGD`, which integrates masking into standard momentum-based SGD.

3.5 Personal Contribution: Mask Selection Strategies

The default masking strategy in sparse fine-tuning updates the parameters with the lowest sensitivity scores, as measured by the diagonal of the Fisher Information Matrix. In this project, several alternative calibration strategies were implemented to investigate how different criteria for parameter selection affect federated learning performance. These included selecting the most sensitive parameters (i.e., those with the highest Fisher scores), selecting parameters based on their absolute magnitude—either the largest or the smallest—and a baseline strategy where parameters were selected uniformly at random.

The objective was to evaluate how the nature of the selected subset influences convergence speed, model generalization, and training stability, especially under varying data heterogeneity in IID and non-IID client distributions.

4 Experiments and Results

Model performance across all experiments was evaluated using top-1 test accuracy. Centralized training relied on a standard fully supervised setup, while federated learning followed the FedAvg protocol. Hyperparameters were tuned based on validation accuracy, and final results were reported on a held-out test set.

4.1 Centralized Training Baseline

Initial experiments were conducted to assess how different hyperparameters affect learning dynamics, using a fixed random seed and 5 training epochs. We first investigated the impact of momentum and weight decay across various learning rates. Results showed that weight decay had minimal influence on accuracy and was therefore fixed at 5×10^{-4} for subsequent runs. Momentum had a small but consistent effect, with 0.9 yielding slightly better performance across all learning rates. Based on these insights, we performed a full grid search varying learning rates and training epochs. The best configuration achieved a validation accuracy of 74.59% and a test accuracy of 74.62% using a batch size of 128, learning rate of 0.01, weight decay of 5×10^{-4}, momentum of 0.9, and 20 training epochs. Reducing epochs to 10 led to a marginal drop in validation accuracy. It is worth noting that our setup used only basic data augmentation (random horizontal flip), without more advanced techniques like RandAugment or MixUp, in order to keep preprocessing simple and avoid added data loading overhead.

Figure 1 illustrates the impact of training epochs on validation accuracy. While increasing the number of epochs leads to improved accuracy, the performance tends to saturate early, with only marginal gains observed beyond 10 epochs. This supports the choice of limiting training duration for efficiency when the computational budget is constrained.

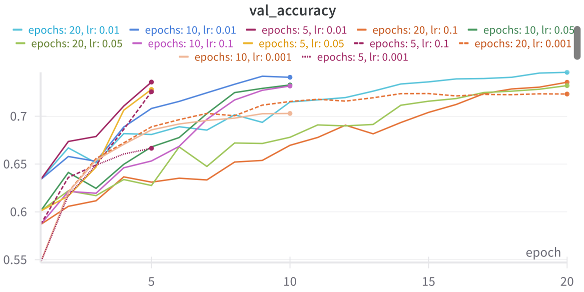

Figure 1: Validation accuracy trend across different numbers of training epochs. Accuracy improves with more epochs but tends to saturate early, showing limited gains beyond 10 epochs.

4.2 Sparse Fine-Tuning on Centralized Model

We applied sparse fine-tuning (see Section 3.4) to the best centralized baseline. The setup used the approximated FIM and SparseSGD optimizer.

Following the TA-LoS procedure [5], we calibrated parameter masks over multiple rounds rather than computing them in a single step. Due to the faster convergence observed in sparse models—and the negligible gain between 10 and 20 epochs in the dense baseline—we fixed training to 10 epochs.

By replacing the standard SGD optimizer with `SparseSGD`, we then fine-tuned and experimented with different sparsity levels $\{0.99, 0.95, 0.90\}$ and calibration rounds $\{1, 3, 5\}$. Results, shown in Figure 2, indicate that increasing the number of calibration rounds had minimal impact on final validation accuracy, in contrast to findings in [5], which used exact FIM. On the other hand, decreasing sparsity consistently improved performance, with the best configuration (sparsity 0.90, rounds 3) reaching 84.41% validation accuracy and 84.31% test accuracy—significantly surpassing the dense baseline.

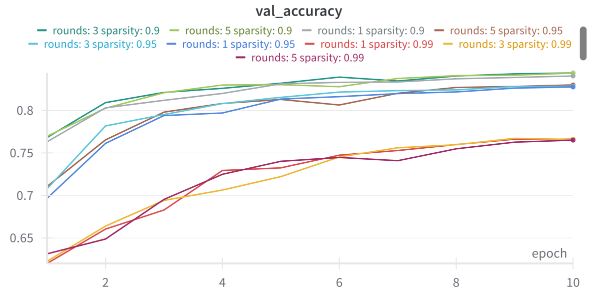

Figure 2: Validation accuracy across training epochs for different sparsity levels and calibration rounds during sparse finetuning of centralized baseline.

4.3 Federated Averaging under IID and Non-IID Settings

We conducted extensive hyperparameter tuning for the FedAvg algorithm under both IID and non-IID client distributions.

Validation results across J, Nc configurations and with fixed Momentum = 0.8 and Learning Rate = 0.001 are summarized in Table 4.1. The combination of learning rate 0.001 and momentum 0.8 consistently outperformed the others, achieving a validation accuracy of 20.22% after 20 communication rounds ($n_{\text{rounds}} = 20$). In contrast, higher learning rates—particularly 0.1—resulted in unstable training and poor convergence, highlighting the sensitivity of the optimization process to overly aggressive updates in IID settings.

For the non-IID setup, we retained the same learning rate and momentum values and tuned additional parameters: the number of classes per client N_c, the number of communication rounds n_{rounds},and the number of local update steps per round J.To ensure comparability across configurations, we kept the product $J \cdot n_{\text{rounds}} = 200$ fixed. The optimal configuration, $N_c = 50$, $n_{\text{rounds}} = 50$, and $J = 4$, achieved a validation accuracy of 38.84% (it achieved 15.49% after $n_{\text{rounds}} = 20$). The IID setting consistently outperformed all non-IID configurations, even the most homogeneous one (e.g., $N_c = 50$), which still exhibited lower accuracy and slower convergence.

To assess the impact of data heterogeneity on model performance, we varied the number of classes per client $N_c \in \{1, 5, 10, 50\}$ while keeping other training parameters fixed. We observed that increasing N_c, which reduces statistical heterogeneity, systematically improves validation accuracy across all local step configurations. In particular, the best performance was consistently obtained for $N_c = 50$, a setting close to IID. Conversely, configurations with lower N_c values (1, 5, 10) resulted in significantly lower accuracy, highlighting the sensitivity of FedAvg to label imbalance and distribution skewness. These results confirm the well-known challenges of learning under non-IID conditions in federated settings [6].

Table 4.1: FedAvg (Classical) – Validation Accuracy Across Hyperparameters with learning rate = 0.001, momentum = 0.8 and J = 4.

Setting	Rounds	Nc	Val Accuracy
IID	20	-	20.22%
Non-IID	20	50	15.49%
Non-IID	50	1	1.19%
Non-IID	50	5	4.52%
Non-IID	50	10	6.75%
Non-IID	50	50	38.84%

4.4 Comparison of IID and non IID strategies

To assess the impact of the local training duration on global model performance, we varied the number of local update steps per client $J \in \{4, 8, 16\}$, while keeping the total number of updates $J \cdot n_{\text{rounds}} = 200$ fixed.

In the IID setting, reducing J and increasing synchronization frequency significantly improved performance: the configuration with $J = 8$ and $n_{\text{rounds}} = 25$ outperformed $J = 16$, indicating that more frequent model averaging facilitates convergence when client data distributions are similar.

In the non-IID setting, the baseline configuration

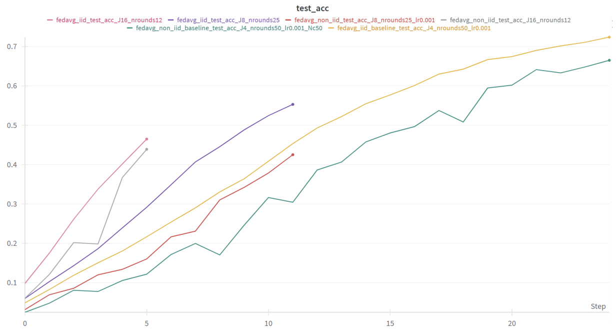

Figure 3: Impact of Local Update Steps (J) on Test Accuracy in IID and Non-IID FedAvg Configurations.

with $J = 4$ and $n_{\text{rounds}} = 50$ achieved the highest accuracy. Both higher values of J ($J = 8$ and $J = 16$) led to degraded performance, confirming that excessive local updates exacerbate client drift under data heterogeneity. These findings highlight the importance of communication frequency in federated optimization, particularly in non-IID scenarios where synchronization plays a key role in stabilizing learning. This trend is also supported by the analysis of implicit regularizers in FedAvg, which shows that increased local update steps amplify gradient variance and hinder convergence under non-IID distributions [8].

4.5 Sparse Model Editing under Federated Learning

We tuned the hyperparameters for model editing by varying the sparsity ratio ({0.85, 0.90, 0.95}) and the number of calibration rounds ({1, 3, 5}) in both IID and non-IID federated learning settings. For the IID setup, we reused the best learning rate and momentum from the FedAvg IID experiments. In the non-IID setting, we adopted the best FedAvg non-IID hyperparameters.Validation accuracy across sparsity and calibration configurations is reported in Table 4.2. In both cases, increasing the number of calibration rounds generally improved convergence speed and stability. We also observed that higher sparsity levels tended to degrade final accuracy unless sufficient calibration was applied, confirming the need for careful balance between parameter pruning and information retention. Additionally, configurations with fewer calibration steps often led to underfitting or stagnation.

Table 4.2: Model Editing FedAvg – Validation Accuracy Across Sparsity and Calibration rounds on 20 commununication rounds for IID and 50 for Non-IID

C.R	Sparsity	IID	Non-IID
1	0.85	14.47%	9.2%
5	0.85	31.80%	56.19%
5	0.90	30.23%	63.59%
5	0.95	23.85%	48.69%

4.6 Comparison of Global Strategies

To compare the different global strategies, we evaluated the final test accuracy of each configuration using the best hyperparameters found during tuning. The results are visualized in Figure 4, and summarized as follows: the centralized FedAvg baseline under IID conditions achieved the highest performance, reaching 72.41% accuracy. The non-IID FedAvg baseline followed with 66.51%. The sparse fine-tuning approach in the non-IID setup obtained 65.69%, while its IID counterpart reached 63.30%.

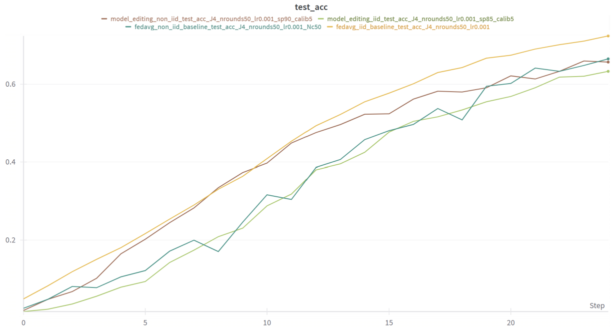

Figure 4: Test accuracy comparison of global training strategies (FedAvg baseline and model editing) under IID and non-IID settings.

Despite operating under non-IID constraints, the sparse fine-tuning strategy closely matches the FedAvg baseline performance, suggesting its robustness to heterogeneity. This is especially significant given that sparse fine-tuning updates only 10% of the most important parameters based on approximated Fisher Information, demonstrating the efficiency of targeted model editing.

However, in IID settings, sparse fine-tuning underperforms compared to standard FedAvg. This gap may be attributed to the lack of strong regularization benefits that sparsity brings under data heterogeneity, where overfitting is a greater concern. Additionally, although convergence was slightly faster for the sparse models, final generalization was limited in the IID case.

Overall, while FedAvg remains superior in IID scenario, model editing proves to be a competitive alternative under non-IID settings, balancing communication efficiency with competitive accuracy.

In Table 4.3 the impact of sparse finetuning is compared across the centralized baseline and FedAvg under both IID and non-IID settings.

Table 4.3: Test Accuracy – Dense vs Model Editing Across All Settings

Method	**Sparsity**	**C.R**	**TestAcc**
Centr	—	—	**74.62%**
CentrEdit	0.90	3	**84.31%**
FedAvgIID	—	—	**72.41%**
FedAvgNonIID	—	—	**66.51%**
FedAvgIIDEdit	0.85	5	**63.30%**
FedAvgNonIIDEdit	0.90	5	**65.69%**

4.7 Analysis of Alternative Mask Selection Strategies

To extend our investigation beyond the standard model editing procedure, we explored five alternative mask selection strategies, as outlined in the Methodology section: `train_least_important`, `train_most_important`, `random`, `magnitude_most`, and `magnitude_least`. These strategies were evaluated under both IID and non-IID settings, using the sparse fine-tuning framework.

For the `train_least_important` and `train_most_important` strategies, we performed a full grid search on both the sparsity ratio ({0.85, 0.90, 0.95}) and the number of calibration rounds ({1, 3, 5}) , as these approaches still rely on the Fisher Information Matrix to guide parameter updates. In contrast, for `random`, `magnitude_most`, and `magnitude_least`, only the sparsity ratio ({0.85, 0.90, 0.95}) was tuned. The calibration step was deemed irrelevant in these cases because the mask is no longer dependent on an estimated importance score: it is either random or directly derived from weight magnitudes, making calibration ineffective or redundant.

4.8 Evaluation of Personalized Masking Strategies

We evaluated the impact of different mask selection strategies on model performance under both IID and non-IID federated learning settings with number of communication rounds equal to 20. The goal was to assess how the choice of parameters to update—based on sensitivity, magnitude, or random selection—affects convergence and generalization in sparse model editing.

In the IID setting, the best performing strategy was magnitude_most (18.61%), where the parameters with the highest absolute weights were selected for training. This approach likely benefits from refining the most influential weights while maintaining overall model coherence, which is easier to preserve when clients are exposed to similar data distributions. The random (16.53%) and magnitude_least (15.38%) strategies followed, showing moderate performance. While random selection introduces stochasticity, it can occasionally preserve essential structure by chance. In contrast, magnitude_least, which updates the smallest weights, showed limited capacity to drive meaningful updates. Surprisingly, train_least_important, which freezes highly sensitive parameters, underperformed in the IID context (12.04%), likely because it restricts learning from modifying impactful components when client updates are already aligned. Finally, train_most_important led to the weakest results (1.83%), possibly due to the disruption of critical internal representations.

In contrast, the non-IID scenario revealed a markedly different ranking. The train_least_important strategy achieved the highest accuracy (25,6%), significantly outperforming all

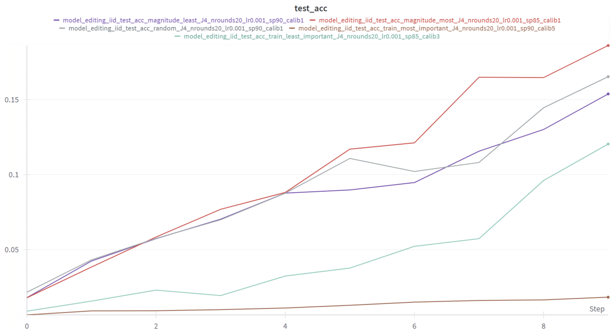

Figure 5: Test accuracy comparison of personalized model editing strategies IID settings across 20 communication rounds.

other methods. By preserving the most sensitive parameters—often encoding shared knowledge—this approach mitigates client drift and enhances global stability despite data heterogeneity. The magnitude_most strategy remained competitive (14,24%), although slightly less effective, as modifying dominant weights can sometimes increase inter-client divergence. Interestingly, train_most_important showed improved behavior in the non-IID setting relative to IID (11,28% vs 1,83%), though its aggressive editing still proved suboptimal. The magnitude_least strategy performed weakly (10,5%), and random selection resulted in the lowest accuracy (8,54%), likely due to incoherent updates across clients and the absence of any structural guidance.

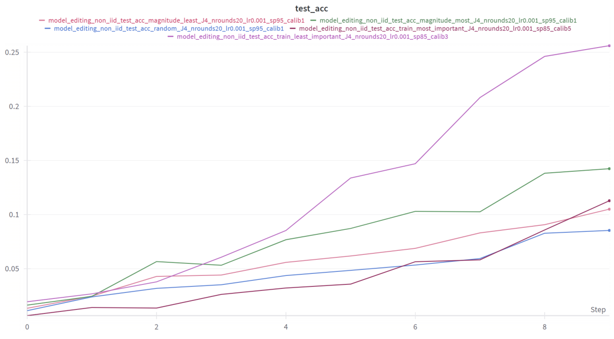

Figure 6: Test accuracy comparison of personalized model editing strategies non-IID settings across 20 communication rounds.

5 Discussion

5.1 Impact of Data Heterogeneity on FedAvg

Numerous studies have shown that FedAvg suffers significantly in the presence of statistical heterogeneity across clients [10]. In particular, decreasing the number of classes per client N_c tends to worsen the alignment between local and global objectives, thereby hindering convergence. Our preliminary experiments confirmed this trend: when $N_c \in \{1, 5, 10\}$, test accuracy remained low regardless of the number of local steps J or communication rounds. As a result, our subsequent comparisons focus on the case $N_c = 50$, which approximates an IID configuration.

To further examine the role of local training duration, we explored different values of $J \in \{4, 8, 16\}$ under the constraint of a fixed computational budget $J \cdot n_{\text{rounds}}$. The underlying goal was to evaluate how the frequency of global synchronization influences convergence in both IID and non-IID regimes.

These analyses align with theoretical insights suggesting that large values of J increase the divergence between client models in heterogeneous settings [?]. While more local steps may speed up local convergence, they exacerbate client drift and degrade overall generalization. Conversely, frequent aggregation tends to stabilize learning, particularly when data distributions are non-uniform.

5.2 Model Editing and Masking Strategies in Non-IID Settings

Sparse fine-tuning, or model editing, demonstrates strong performance in non-IID federated learning. By freezing the most sensitive backbone parameters—identified via Fisher Information or magnitude scores—and updating only the least critical ones, strategies like `train_least_important` effectively reduce client drift and improve global stability. This targeted approach enables slight but meaningful updates to the backbone, enhancing generalization across heterogeneous clients. Our experiments show that masking strategies must be adapted to the data distribution: `train_least_important` outperforms others in non-IID settings, while `random` and `magnitude_most` perform better under IID but suffer in non-IID contexts. Conversely, `train_most_important`, which focuses updates on highly sensitive parameters, tends to disrupt shared representations and degrade convergence. These findings are consistent with prior research on sparse training and drift mitigation in FL [1, 4], highlighting the importance of distribution-aware parameter selection.

5.3 Limitations and Future Work

This study has mainly focused on a quasi-IID setting ($N_c = 50$), limiting the assessment of robustness in more heterogeneous scenarios. Future evaluations should include lower values of N_c (e.g., 1, 5, 10) to better reflect real-world non-IID conditions.

To address client-server drift in such settings, combining **model editing** with corrective methods like **FedProx** or **SCAFFOLD** may offer enhanced stability. While FedProx introduces a local regularization term, its impact is limited under strong heterogeneity. SCAFFOLD provides better theoretical guarantees but at a higher communication cost.

Further directions include testing robustness to adversarial behaviors (e.g., corrupted clients) and adding privacy-preserving mechanisms like differential privacy to support deployment in real-world FL applications.

References

[1] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. 8

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

[3] Peter Kairouz, H Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021. 1

[4] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations (ICLR)*, 2019. 8

[5] Tatiana Tommasi Leonardo Iurada, Marco Ciccone. Efficient model editing with task-localized sparse fine-tuning. *ICLR*, 2025. 1, 3

[6] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations (ICLR) Workshop*, 2019. arXiv preprint arXiv:1907.02189. 4

[7] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020. 1

[8] Jinwoo Lim, Suhyun Kim, and Soo-Mook Moon. Convergence analysis of federated learning methods using backward error analysis. *arXiv preprint arXiv:2503.03139*, 2025. To appear in AAAI 2025. 5

[9] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1273–1282, 2017. 1

[10] Yao Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 7