PLAYER-FL: A PRINCIPLED APPROACH TO PERSONALIZED LAYER-WISE CROSS-SILO FEDERATED LEARNING

Ahmed Elhussein [1 2] **Florent Pollet** [1 2] **Gamze Gürsoy** [3 1 2 4]

ABSTRACT

Federated learning (FL) with non-IID data often degrades client performance below local training baselines. Partial FL addresses this by federating only early layers that learn transferable features, but existing methods rely on ad-hoc, architecture-specific heuristics. We first conduct a systematic analysis of layer-wise generalization dynamics in FL, revealing an early-emerging transition between generalizable (safe-to-federate) and task-specific (should-remain-local) layers. Building on this, we introduce Principled Layer-wise Federated Learning (PLayer-FL), which aims to deliver the benefits of federation more robustly. PLayer-FL computes a novel federation-sensitivity metric efficiently after a single training epoch to choose the optimal split point for a given task. Inspired by model pruning, the metric quantifies each layer's robustness to aggregation and highlights where federation shifts from beneficial to detrimental. We show that this metric correlates strongly with established generalization measures across diverse architectures. Crucially, experiments demonstrate that PLayer-FL achieves consistently competitive performance across a wide range of tasks while distributing gains more equitably and reducing client-side regressions relative to baselines.

SOFTWARE AND DATA

Code for reproducing this paper can be found at https://github.com/G2Lab/PLayer-FL.

1 INTRODUCTION

Federated Learning (FL) is a distributed learning framework that enables collaborative model training without data-sharing. Early FL algorithms such as Federated Averaging (FedAvg) focused on developing a unified global model (McMahan et al., 2017). However, when client data is non-independently identically distributed (non-IID), global model performance degrades. Personalized FL methods address this by allowing client models to diverge from the global model, yet they do not consistently outperform FedAvg nor distribute benefits equitably across all participants (Zhao et al., 2018; Li et al., 2022; Divi et al., 2021).

Partial FL is a form of personalized FL that selectively federates only a subset of model layers while keeping the remainder local. In practice, early layers are federated and later layers are retained for local training (Arivazhagan et al., 2019; Oh et al., 2021; Collins et al., 2021). This approach is motivated by insights from representation learning: early layers occupy flatter loss regions and generalize across tasks, while later layers specialize (Yosinski et al., 2014; Chaudhari et al., 2019; Jiang et al., 2019). However, existing partial FL methods suffer from two gaps: (1) it has not been systematically established that these layer-wise generalization dynamics hold under federated, non-IID training; and (2) the choice of which layers to federate relies on architecture-specific heuristics, which limits generalizability.

We propose that parameter importance metrics from model pruning, using computationally simple, architecture-agnostic first-order approximations (Molchanov et al., 2019), can inspire the design of measures to identify layers robust to federation. Building on this, we introduce Principled Layer-wise FL (PLayer-FL), which computes a novel federation sensitivity metric after one training epoch to determine the optimal split point. We validate our approach across CNNs, Transformers, and MLPs on seven non-IID datasets and demonstrate consistent performance, fairness, and incentivization improvements essential for sustainable collaboration.

[1]Department of Biomedical Informatics, Columbia University, NYC, USA [2]New York Genome Center, NYC, USA [3]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK [4]Department of Computer Science, Columbia University, NYC, USA. Correspondence to: Gamze Gürsoy <gg584@cam.ac.uk>.

Proceedings of the 9th MLSys Conference, Bellevue, WA, USA, 2026. Copyright 2026 by the author(s).

2 CONTRIBUTION AND SIGNIFICANCE

In this study, we make several contributions toward robust, equitable FL with non-IID data:

- **Early identification of layer-wise generalization patterns in federated training.** We show that models with *identical initialization but independent training on non-IID data* exhibit consistent generalization patterns across corresponding layers, even after just one epoch. This enables early, reliable identification of which layers benefit from federation versus which would suffer from negative transfer, minimizing exposure to harmful training dynamics.
- **Federation sensitivity: An efficient metric for robust collaboration.** We introduce federation sensitivity, a novel metric that estimates each layer's robustness to aggregation and *identifies a transition point where federation shifts from beneficial to potentially harmful*. This metric is architecture-agnostic, correlates strongly with established generalization measures, emerges from the first epoch, and imposes minimal computational overhead – enabling principled deployment without requiring extensive tuning.
- **PLayer-FL: Principled partial FL.** We develop PLayer-FL, which uses federation sensitivity to determine which layers to federate after only one epoch of training. PLayer-FL achieves *stable and competitive* performance across diverse non-IID datasets: it achieves the best performance, fairness and incentivization rankings across benchmarks, ensuring equitable benefits and protecting clients from performance degradation relative to local training.

3 BACKGROUND

3.1 Federated Learning

In each training round k, the server broadcasts global model parameters Θ^{k-1} to clients $c \in C$. These clients update their local model to produce Θ_c^k which are sent back to the server. The central server then aggregates these parameters and re-broadcasts the updated global model. This is repeated until convergence. In FedAvg, the aggregation is defined as $\Theta^k = \sum_{c=1}^{n} \alpha_c \Theta_c^k$, where α_c is the weight of each client determined by sample size.

3.1.1 Cross-Silo FL

FL has two main settings, cross-silo and cross-device, each with distinct use cases and challenges (Kairouz et al., 2021). This work concentrates on cross-silo FL, a setting common in sectors such as healthcare to overcome data sharing constraints. In healthcare, cross-silo FL has advanced research on poorly understood diseases by enabling training on larger datasets (Pati et al., 2022; Dayan et al., 2021). Cross-silo FL involves a few clients with medium to large datasets. A key challenge is non-IID data as institutions often have different populations, data collection practices, and operational protocols (Pati et al., 2021; Terrail et al., 2022). Further, since clients frequently possess enough data to train adequately performant models on their own, FL must surpass local training to motivate participation (Huang et al., 2022). In contrast, cross-device FL manages numerous edge devices with limited data, prioritizing communication efficiency. While most FL methods target cross-device scenarios, fewer address cross-silo FL's unique challenges (Huang et al., 2022; Terrail et al., 2022), highlighting the need for specialized approaches.

3.2 Model pruning

Model pruning reduces model size by removing parameters with limited impact on the output (Reed, 1993). Effective metrics include weight magnitude (Han et al., 2015; Liu et al., 2017) and first-order approximations of the change in loss following parameter removal (LeCun et al., 1989; Molchanov et al., 2019). The importance of a parameter w_s can be estimated as $\mathcal{I}_s(w) = (g_s w_s)^2$, where g_s is the gradient with respect to w_s. These methods are reliably effective across diverse architectures, a property we exploit for federation decisions.

3.3 Model generalization

Generalization can be characterized via the loss landscape: parameters in flatter regions exhibit reduced sensitivity to perturbations and greater generalizability (Yosinski et al., 2014). Gradient variance, even after one epoch, correlates with layer generalizability (Jiang et al., 2019), and the Hessian eigenspectrum reflects loss surface curvature (Chaudhari et al., 2019). Complementarily, representational similarity methods such as Centered Kernel Alignment (CKA) (Kornblith et al., 2019) compare how different models transform the same inputs at each layer, revealing that generalizing networks produce similar representations while overfit networks do not (Morcos et al., 2018).

We apply these techniques to a question specific to FL: *to what extent do identically initialized models trained on non-IID data resemble each other?* Direct weight comparison via metrics such as L2-norm or cosine similarity is insufficient: these are not invariant to linear transformations and weight differences may not emerge early in training. Gradient-based and representation-based metrics overcome both limitations.

4 Related works

Global FL. FedAvg (McMahan et al., 2017) trains a single global model via weighted averaging of local updates, but struggles with non-IID data (Li et al., 2020). Approaches to mitigate this include data sharing (Zhao et al., 2018), data augmentation (Duan et al., 2019), and regularization to keep local updates close to the global model (Li et al., 2020), but none produce client specific models.

Personalized FL. Methods include client clustering (Briggs et al., 2020; Mansour et al., 2020), multi-task learning (Smith et al., 2017; Dinh et al., 2021), regularization-based approaches (Li et al., 2021; T Dinh et al., 2020), and local adaptation of the converged global model (Yu et al., 2020). These methods introduce additional hyperparameters and do not consistently outperform FedAvg across non-IID settings.

Partial FL. FedPer (Arivazhagan et al., 2019), FedBABU (Oh et al., 2021), and FedRep (Collins et al., 2021) federate early layers while keeping later layers local, but select which layers to share via architecture-specific heuristics. FedLP (Zhu et al., 2023) and FedLAMA (Lee et al., 2023) select layers dynamically but target communication efficiency rather than non-IID robustness. pFedLA (Ma et al., 2022) learns client- and layer-specific aggregation weights via server-side hypernetworks, adding considerable complexity. pFedHR (Wang et al., 2024) requires sharing data across clients, which is often infeasible in cross-silo FL. PLayer-FL differs from all of these by providing a principled, architecture-agnostic metric that identifies the split point after a single epoch without additional hyperparameters or data sharing.

5 Layer-wise generalization dynamics in Federated Learning

A fundamental challenge in partial FL is determining which layers benefit from federation versus which would be harmed by aggregation across non-IID data. To address this, we investigate whether identically initialized models trained on non-IID data exhibit predictable, layer-specific patterns in their generalization capabilities.

We examine two training scenarios: (1) independent training, where each client trains completely locally (representing maximum divergence), and (2) federated training, where models undergo periodic aggregation (representing typical FL divergence). By comparing these scenarios, we can identify the extent to which federation either mitigates or exacerbates client divergence at different layers. Figures 1–4 present results from the first epoch. Figures A.3–A.11 present final model results, demonstrating that the layer-wise generalization patterns observed early in training persist throughout the entire training process. Definitions for all metrics are provided in Section A.1.

Key finding. Prior work shows that randomly initialized models on IID data follow similar training trajectories (Wang et al., 2018). We study the complementary setting: identically initialized models trained on non-IID client subsets. A layer-wise analysis reveals a stable pattern across architectures and datasets: **early layers converge to similar, broadly transferable solutions despite non-IID data, whereas later layers diverge markedly**. This echoes transfer-learning behavior but is established here specifically for FL.

Identifying a transition point. Crucially, we observe a consistent *transition point* in each architecture, a boundary that separates layers functioning as generalizable feature extractors (suitable for federation) from layers serving as task-specific feature utilizers (requiring local training). This transition marks where the benefits of federation shift to potential harm through negative transfer. Remarkably, this transition point emerges after just one epoch and remains stable throughout training, enabling early, reliable decisions about which layers to federate without prolonged exposure to harmful aggregation effects.

This empirical finding motivates our federation sensitivity metric: if layer-wise generalization patterns are consistent and emerge early, we can develop a principled, data-driven approach to identify robust collaboration boundaries for each task rather than relying on architecture-specific heuristics.

5.1 Layer generalization

Early layers are robust to perturbation – a protective property for federation. Figures 1 and 2 report the gradient variance (Eq. A.1) and the sum of Hessian eigenvalues (Eq. A.2) after one epoch across four datasets (note that independent and FL training are equivalent after one epoch, prior to any aggregation). To account for layer size, we normalize both metrics by the number of parameters in each layer. Extended analyses at the end of independent and FL training are given in Figures A.2–A.7.

A consistent pattern emerges: both metrics rise sharply in the final layers (see log scale), indicating that early layers lie in flatter regions of the loss landscape while later layers reside in sharper, more sensitive regions. This has direct implications for robust federation: layers in flat regions have lower sensitivity to perturbation and greater generalization capacity, whereas layers in sharp regions are more vulnerable and task-specific.

Identifying layers for robust collaboration. Interpreting federated averaging as a perturbation to locally optimized weights reveals a *transition point*. Early layers remain in locally flat basins even after aggregation, so federating them should not harm client performance. By contrast, later lay-

ers exhibit high perturbation sensitivity; aggregating them across non-IID clients risks disrupting task-specific optima and degrading performance.

This interpretation rests on a mild assumption: that, for early layers, the global and local optima are proximate – *i.e.,* aggregation does not induce a parameter shift large enough to exit the local basin of the loss. Under this condition, averaging early-layer weights is a small perturbation within a flat region and should, at minimum, avoid worsening performance. For later layers, even small perturbations can be detrimental due to sharper minima. We empirically validate the proximity assumption via representational similarity analysis in the next section, showing that early layers converge to similar solutions across clients despite non-IID data.

5.2 Validating the transition point: Representational similarity analysis

Having established that early layers occupy flat loss regions, we now validate the assumption: **do early layers actually converge to similar solutions across clients despite non-IID training?** If true, this confirms that federating early layers should at least not harm performance.

We measure layer-wise representational similarity across all client models using Centered Kernel Alignment (CKA, Eq. A.3) (Kornblith et al., 2019), which quantifies how similarly different models transform the same input samples at each layer. Figure 3 shows the average representational similarity for each model compared to all other models after one epoch of training. Figures A.8 and A.9 provide comparable analyses at the conclusion of training across all datasets.

The results validate our hypothesis: early layers exhibit high representational similarity across all clients (indicating convergence to similar solutions), while later layers show dramatically increasing dissimilarity (Figure 3). This transition aligns precisely with the gradient-based metrics from the previous section, confirming that:

- **Early layers:** Converge to similar solutions and occupy flat regions and so are *safe to federate without harming clients*
- **Later layers:** Diverge to client-specific solutions and occupy sharp regions so *federation would disrupt optimization and degrade performance*

Critically, this transition point emerges after just one training epoch and becomes more pronounced with continued training, enabling early identification without prolonged exposure to harmful aggregation dynamics.

These findings provide empirical support for our central claim:

Hypothesis 1 (Early-layer robustness to federation under non-IID FL). In non-IID settings, federating early layers – which converge to similar, generalizable solutions – does not harm client performance. Conversely, federating later layers is likely to degrade performance by disrupting task-specific optimization achieved locally.

5.3 Federation sensitivity: Efficiently identifying the transition point

Having empirically validated that a transition point exists we now introduce a principled metric to automatically identify this boundary for any architecture and dataset. This addresses a critical gap in partial FL: existing methods rely on architecture-specific heuristics that create deployment risks, while our approach provides a data-driven, parameter-light method for robust federation decisions.

Key insight from model pruning. Our approach builds on the observation that layers with strong cross-client generalization occupy flatter loss regions, a property quantifiable using parameter importance metrics from model pruning. In pruning, parameter importance $\mathcal{I}_p(\theta) = (\theta_p \nabla\theta_p)^2$ estimates each parameter's contribution to the loss using a first-order approximation (Molchanov et al., 2019). We adapt this concept to the federation context: *if a layer's non-bias parameters have low importance (indicating flat loss geometry and robustness to perturbation), that layer is suitable for federation*. Conversely, high parameter importance signals vulnerability to the perturbation induced by aggregation.

Federation sensitivity metric. We define the federation sensitivity of layer l, denoted $\mathcal{F}_l(\Theta)$, as:

$$\mathcal{F}_l(\Theta) \triangleq \sum_{k=1}^{l} \frac{1}{n'_k} \sum_{p=1}^{n'_k} \mathcal{I}_p(\theta) = \sum_{k=1}^{l} \frac{1}{n'_k} \sum_{p=1}^{n'_k} (\theta_p \nabla\theta_p)^2 \quad (1)$$

where Θ denotes the full model parameters, θ_p is the pth non-bias parameter in layer k, $\nabla\theta_p$ is its gradient, and n'_k is the number of non-bias parameters in layer k.

This metric incorporates two key design choices that reflect the constraints and goals of robust partial federation:

1. **Layer-wise normalization.** We divide by n'_k to enable fair comparison across layers of different sizes, ensuring that large layers do not artificially dominate the metric. This provides architectural robustness across diverse model designs.
2. **Cumulative aggregation.** We sum contributions from all layers up to l because partial federation has an inher-

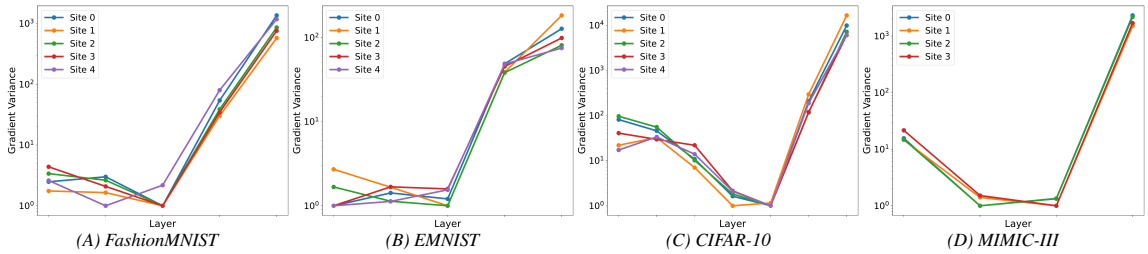


Figure 1. **Layer gradient variance after one epoch**. All models identically initialized and independently trained on non-IID data.

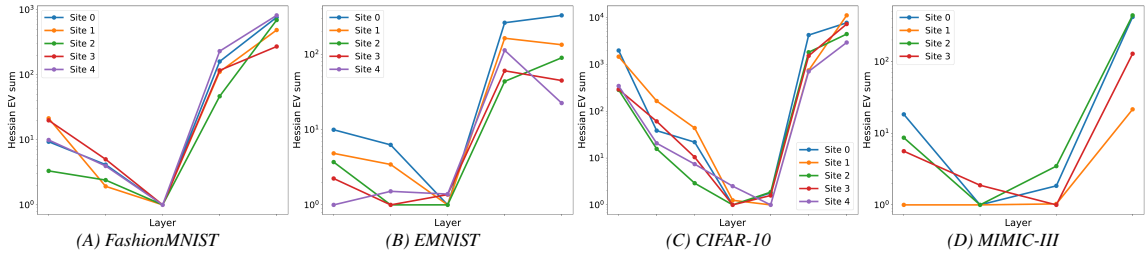


Figure 2. **Hessian eigenvalue sum after one epoch**. Each model was identically initialized and trained on respective non-IID datasets.

ent constraint: selecting any layer for federation necessarily requires including all previous layers. This cumulative formulation naturally identifies the transition point where adding another layer to federation would become harmful.

Interpretation: Federation sensitivity quantifies each layer's vulnerability to aggregation. Low values indicate robustness (suitable for federation); high values indicate vulnerability (would be harmed by federation).

Empirical validation of the transition point. Figure 4 shows federation sensitivity after one epoch across all architectures (values normalized as percentage of first-layer sensitivity). Figures A.10 and A.11 show results after full training. A characteristic pattern emerges: **federation sensitivity remains low in early layers, then spikes sharply in later layers**.

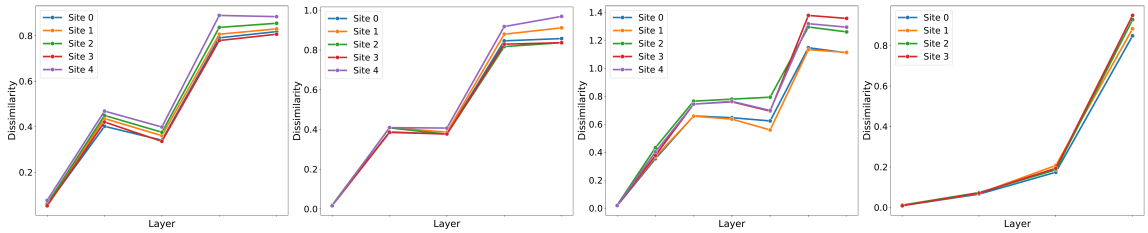
Critically, this spike aligns with the transitions observed in our generalization metrics (gradient variance, Hessian eigenvalues) and representational similarity analysis (Section 5). This convergence of multiple independent metrics provides strong empirical validation that federation sensitivity successfully identifies the transition point between layers suitable for federation and those that would suffer negative transfer.

Protective properties enabling deployment. Federation sensitivity offers several advantages critical for real-world deployment:

- **Stable:** Requires minimal hyperparameter tuning, eliminating a major source of deployment risk and ensuring consistent behavior across settings.
- **Computationally efficient:** Uses weights and gradients already computed during training, adding negligible overhead.
- **Architecture-agnostic:** Shows consistent patterns across Fully-Connected Networks, CNNs, and Transformers, enabling broad applicability without architecture-specific modifications.
- **Early emergence:** Identifies the optimal split point after just one epoch, minimizing exposure to harmful training dynamics while enabling rapid, reliable federation decisions.

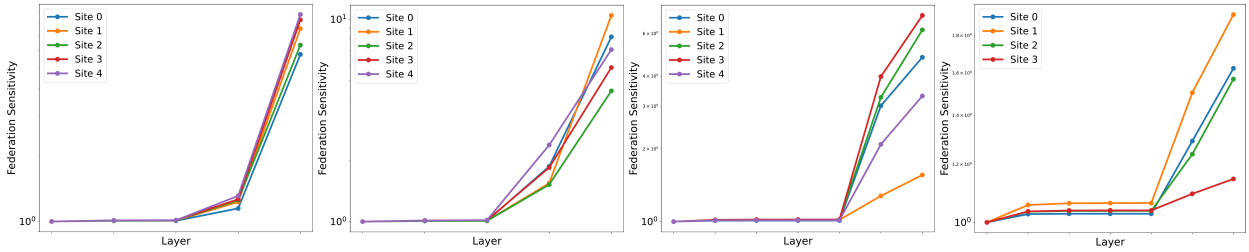
6 Proposed Framework

We propose Principled Layer-wise-FL (PLayer-FL), a partial FL approach that uses federation sensitivity to dynamically determine which layers to federate (Algorithm 1). A key advantage of our method is its computational efficiency – federation sensitivity is calculated during the first epoch of training using weights and gradients that are already computed as part of the standard training process. To identify the transition point between federated and local layers, we compare the relative federation sensitivity between consecutive layers against a threshold t (eqn. 2). While this threshold controls how aggressively the algorithm federates layers, with higher values leading to more federated layers, we

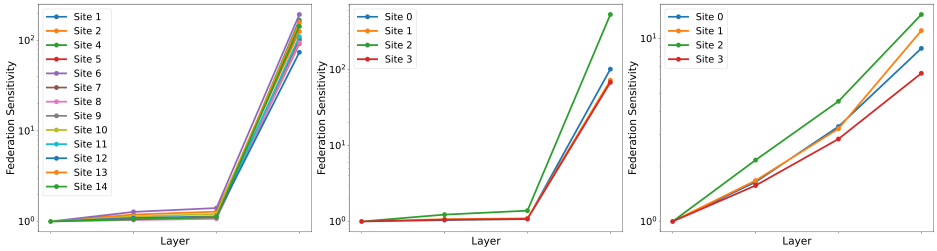


(A) FashionMNIST *(B) EMNIST* *(C) CIFAR-10* *(D) MIMIC-III*

Figure 3. **Model representation similarity by layer**. Models identically initialized and independently trained on non-IID data.



(A) FashionMNIST *(B) EMNIST* *(C) CIFAR-10* *(D) ISIC-2019*



(E) Sent-140 *(F) MIMIC-III* *(G) Fed-Heart-Disease*

Figure 4. **Federation sensitivity after one epoch**. All models identically initialized and trained on non-IID data subsets. Values expressed as % of first layer value

found in our experiments that the transition point remains stable across a wide range of threshold values (Figure 4).

PLayer-FL stands out by offering a light-weight and principled approach for determining the layers to federate during training specifically to address the challenges of non-IID data. Algorithms 1 - 3 present the formal pseudocode. For brevity, we omit procedures CLIENTUPDATE and SERVERAGGREGATE that are identical to FedAvg. Note that PLayer-FL conducts simultaneous updates to both the federated and localized layers. However, only the federated layers are sent to the central server for aggregation. We use the same learning rate for both federated and localized training, but this can be altered. Additionally, as our algorithm's sole purpose is to delineate layers to be federated, it can be used in conjunction with other personalized FL algorithms.

PLayer-FL has the same computational complexity as FedAvg. This is because, compared to FedAvg, PLayer-FL introduces two procedures, both executed once during training: CALCULATEFEDSENSITIVITY (Algorithm 2) and LAYERSPLIT (Algorithm 3) with computational complexity

Algorithm 1 PLayer-FL: Main Algorithm

Input: Number of rounds N, set of clients C, threshold t
Output: Model parameters Θ_c for $c \in C$
Initialize global model parameters Θ^0
for all clients $c \in C$ in parallel **do**
  $\Theta_c^1 \leftarrow$ CLIENTUPDATE(Θ^0)
  Calculate layer federation sensitivity (Algorithm 2):
  $\mathcal{F}(\Theta_c) \leftarrow$ CALCULATEFEDSENSITIVITY(Θ_c)
end for
Determine federated and local layers (Algorithm 3):
  $\Theta_{fed}, \Theta_{local} \leftarrow$ LAYERSPLIT($\Theta, \{\mathcal{F}(\Theta_c)\}, t$)
Execute FL on selected layers:
for $r = 1$ **to** N **do**
  for all clients $c \in C$ **do**
    $\Theta_c^r \leftarrow$ CLIENTUPDATE(Θ^r)
  end for
  $\Theta_{fed}^{r+1} \leftarrow$ SERVERAGGREGATE($\{\Theta_{c,fed}^r\}, C$)
end for

Algorithm 2 PLayer-FL: CALCULATEFEDSENSITIVITY

Input: Client model parameters Θ_c
Output: Layer-wise federation sensitivity array $\mathcal{F}(\Theta_c)$
Initialize $\mathcal{F}(\Theta_c)$ as an empty array
Let L be the set of all layers in Θ_c
for all layers $l \in L$ **do**
  Calculate federation sensitivity, $\mathcal{F}_l(\Theta_c)$, for layer l as in equation 1
  Append $\mathcal{F}_l(\Theta_c)$ to $\mathcal{F}(\Theta_c)$
end for
Return $\mathcal{F}(\Theta_c)$

Algorithm 3 PLayer-FL: LAYERSPLIT

Input: Model parameters Θ, client sensitivities $\{\mathcal{F}(\Theta_c)\}$, threshold t
Output: Federated layers Θ_{fed}, local layers Θ_{local}
Aggregate client federation sensitivities:
  $\mathcal{F}(\Theta) \leftarrow \sum_{c=1}^{C} \mathcal{F}(\Theta_c)$
Find transition point p where relative sensitivity exceeds threshold t:

$$p \leftarrow \min \left\{ p \in \{1, \ldots, L-1\} : \frac{\mathcal{F}_{p+1}(\Theta)}{\mathcal{F}_p(\Theta)} > t \right\} \quad (2)$$

Partition model at point p:
  $\Theta_{fed} \leftarrow \Theta_{l \leq p}, \Theta_{local} \leftarrow \Theta_{l > p}$
Return $\Theta_{fed}, \Theta_{local}$

of $O(P)$, where P is the number of parameters, and $O(L)$, where L is the number of layers, respectively.

7 EVALUATION

7.1 Methods

Our setup focuses on **cross-silo FL**: we use *highly* non-IID datasets from a limited number of clients, all of whom have sufficient data to train a local model and are available for each round of training (see Section 3.1.1 for justification).

7.1.1 Datasets and architectures

We evaluate performance on FashionMNIST, ExtendedMNIST, CIFAR-10, ISIC2019, Fed-Heart-Disease, Sentiment-140 and MIMIC-III. The datasets cover tabular, imaging, and natural language modalities. The first three datasets do not have a natural partition, thus we introduce label skew using a Dirichlet distribution ($\alpha = 0.5$) following Li et al. (2022). The other datasets have natural partitions:

- Fed-ISIC-2019 (Terrail et al., 2022): Dermoscopy skin lesion classification (4 classes); partitioned by hospital.
- Fed-Heart-Disease (Terrail et al., 2022): Patient heart disease classification (5 classes); partitioned by hospital.
- Sent-140 (Caldas et al., 2018): Classification of tweet sentiments; partitioned by users, selecting the top 15 users.
- MIMIC-III (Johnson et al., 2016): Mortality prediction using hospital admission notes; partitioned by diagnosis.

Table 1 also reports the model architecture and the transition point identified by the federation-sensitivity metric for each dataset.

7.1.2 Baselines

We evaluate our method against three categories of baselines: (1) standard approaches – local site training and FedAvg (McMahan et al., 2017), (2) established personalized FL algorithms – FedProx (Li et al., 2020), pFedMe (T Dinh et al., 2020), Ditto (Li et al., 2021), and Local Adaptation (Yu et al., 2020), and (3) existing partial FL methods – FedBABU (Oh et al., 2021), FedLP (Zhu et al., 2023), FedLAMA (Lee et al., 2023), and pFedLA (Ma et al., 2022). We also evaluate PLayer-FL-Random, which federates up to a randomly selected layer. This serves as proxy for **non-principled** partial FL approaches. Full training and evaluation details are in Section A.4.

7.1.3 Metrics

We evaluate performance using three metrics. Our primary metric is the macro-averaged F1 score, which weights all classes equally and is robust to class imbalance: $\text{F1}_{\text{macro}} = \frac{1}{N}\sum_{i=1}^{N} 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i}$, where N is the number of classes. We also report accuracy and test loss (Sections A.6.2 and A.6.3).

For personalized FL algorithms, we study per-site metrics to assess whether benefits are equitably distributed (Divi et al., 2021). We measure *fairness* as the variance in client performance: Fairness $= \frac{1}{C}\sum_{c=1}^{C}(P_c - \overline{P_c})^2$, where P_c is the performance of client c and $\overline{P_c}$ is the mean across clients. We measure *incentivization* as the fraction of clients that outperform both local training and FedAvg (Cho et al., 2022): Incentivization $= \frac{1}{C}\sum_{c=1}^{C} \mathbb{I}\{P_c > \max(S_c, G_c)\}$, where S_c and G_c are the local and FedAvg performances for client c. To draw conclusions across datasets, we use the Friedman test (Demšar, 2006).

7.2 Results

PLayer-FL achieves robust and equitable FL collaboration. Our results demonstrate that PLayer-FL delivers consistently competitive performance while ensuring equitable outcomes and strong participation incentives.

Consistent and competitive performance across diverse settings. Table 2 presents F1 scores and average ranks across all datasets (accuracy and loss in Tables A.7 and A.10 show consistent conclusions). PLayer-FL achieves the high-

Table 1. Datasets, model architectures, and transition points identified by federation sensitivity. CNN = Convolutional Neural Network, MLP = Multi-Layer Perceptron, FC = Fully Connected layer. Transition point indicates the last federated layer; all subsequent layers are trained locally.

Dataset	Modality	Classes	Partition	Type	Architecture	Transition Point
FashionMNIST	Image	10	Dir$\sim$(0.5)	CNN	{3$\times$Conv – 2$\times$FC}	Conv_3
EMNIST	Image	62	Dir$\sim$(0.5)	CNN	{3$\times$Conv – 2$\times$FC}	Conv_3
CIFAR-10	Image	10	Dir$\sim$(0.5)	CNN	{5$\times$Conv – 2$\times$FC}	Conv_5
ISIC-2019	Image	4	Hospital	CNN	{5$\times$Conv – 2$\times$FC}	Conv_5
Fed-Heart	Tabular	5	Hospital	MLP	{4$\times$FC}	FC_2
Sent-140	NLP	2	User	Transformer	{Pos$\oplus$Tok Emb – Attn$\oplus$Res – FC$\oplus$Proj – FC}	$\text{FC}_1\oplus$Proj
MIMIC-III	NLP	2	Diagnosis	Transformer	{Pos$\oplus$Tok Emb – Attn$\oplus$Res – FC$\oplus$Proj – FC}	$\text{FC}_1\oplus$Proj

Table 2. Macro-averaged F1 Score and average rank. In **bold** is top-performing model. Friedman rank test p-value $< 5 \times 10^{-3}$

Algorithm	FMNIST	EMNIST	CIFAR	ISIC	Heart	Sentiment	mimic	Mean Rank
Local	75.9 $\pm$ 1.6	56.9 $\pm$ 1.4	61.6 $\pm$ 3.2	**53.1 $\pm$ 1.3**	**42.0 $\pm$ 0.8**	57.4 $\pm$ 1.0	58.6 $\pm$ 1.5	6.9
FedAvg	75.6 $\pm$ 1.8	64.2 $\pm$ 1.1	65.4 $\pm$ 4.5	43.3 $\pm$ 0.7	40.4 $\pm$ 1.0	52.6 $\pm$ 0.3	63.0 $\pm$ 1.7	6.0
FedProx	76.4 $\pm$ 1.8	65.3 $\pm$ 1.9	65.8 $\pm$ 4.9	40.6 $\pm$ 2.6	37.2 $\pm$ 1.6	52.6 $\pm$ 0.3	63.2 $\pm$ 0.8	6.3
pFedMe	77.3 $\pm$ 1.8	66.2 $\pm$ 0.7	48.9 $\pm$ 2.5	44.9 $\pm$ 2.6	42.5 $\pm$ 0.8	58.9 $\pm$ 0.4	60.9 $\pm$ 0.4	5.3
Ditto	77.8 $\pm$ 1.2	61.6 $\pm$ 0.8	48.7 $\pm$ 1.9	43.4 $\pm$ 0.7	40.7 $\pm$ 1.0	58.9 $\pm$ 0.9	61.0 $\pm$ 0.5	6.0
LocalAdaptation	76.9 $\pm$ 1.5	63.8 $\pm$ 1.8	65.5 $\pm$ 3.3	43.5 $\pm$ 1.9	39.8 $\pm$ 0.9	53.2 $\pm$ 0.6	62.6 $\pm$ 1.6	6.4
FedBABU	77.2 $\pm$ 1.0	**66.1 $\pm$ 0.2**	**67.7 $\pm$ 4.0**	49.1 $\pm$ 3.5	40.3 $\pm$ 0.8	58.3 $\pm$ 0.2	62.7 $\pm$ 0.4	4.1
FedLP	77.2 $\pm$ 1.3	64.1 $\pm$ 0.9	63.7 $\pm$ 5.1	40.2 $\pm$ 2.1	40.1 $\pm$ 1.0	54.4 $\pm$ 1.1	62.1 $\pm$ 1.3	6.7
FedLama	71.6 $\pm$ 1.7	56.9 $\pm$ 1.3	41.8 $\pm$ 3.0	36.0 $\pm$ 1.1	39.7 $\pm$ 0.8	50.3 $\pm$ 0.4	**66.7 $\pm$ 1.0**	10.1
pFedLA	47.1 $\pm$ 0.9	07.1 $\pm$ 0.0	10.5 $\pm$ 1.6	28.4 $\pm$ 1.9	40.1 $\pm$ 0.8	50.3 $\pm$ 0.4	65.0 $\pm$ 2.6	11.4
PLayer-FL	**79.3 $\pm$ 1.4**	62.2 $\pm$ 1.2	67.1 $\pm$ 3.7	52.1 $\pm$ 1.5	41.9 $\pm$ 1.0	**59.6 $\pm$ 1.2**	63.0 $\pm$ 0.7	**2.6**
PLayer-FL-Random	76.5 $\pm$ 2.3	57.1 $\pm$ 1.1	65.7 $\pm$ 2.3	52.1 $\pm$ 1.5	41.3 $\pm$ 0.9	55.7 $\pm$ 0.9	61.0 $\pm$ 1.4	6.1

est mean rank with statistical significance ($p < 5 \times 10^{-3}$), demonstrating robust performance across various architectures and realistic non-IID conditions. Notably, in line with typical observations in cross-silo FL settings where data is highly non-IID (Huang et al., 2022), local site training is competitive when compared to FL algorithms, as each site can effectively train its own model.

Importantly, while PLayer-FL does not always achieve the top individual score, it consistently ranks in the top three across all benchmarks, a pattern that reflects a critical protective property: *stability without hyperparameter tuning*. Unlike optimization-based personalized FL methods that require careful tuning of regularization weights or learning rates (which can harm performance when misconfigured), PLayer-FL's parameter-light design eliminates this deployment risk. By relying on generalization dynamics observable in the first epoch rather than task-specific hyperparameters, PLayer-FL provides *stable, predictable performance across diverse settings*.

Fairness: Equitable benefits across all clients. Table 3 shows that PLayer-FL achieves the *best average fairness rank* across all benchmarks (full results in Table A.5), with statistical significance ($p < 5 \times 10^{-3}$). This demonstrates that PLayer-FL distributes performance improvements equitably rather than benefiting only a subset of clients, a property that ensures no client is systematically disadvantaged. This fairness is not incidental but emerges directly from our principled approach: by federating only layers where clients naturally converge, we avoid the harmful averaging of divergent, client-specific representations that causes inequitable outcomes in standard FL.

Incentivization: Protecting clients from performance degradation. Beyond fairness, PLayer-FL achieves the *best average incentivization rank* (Table 3, full results in Table A.6, $p = 0.043$), meaning that clients are protected from performing worse than local training. This is essential for sustainable collaboration: institutions will only participate if federation does not harm their individual performance. By identifying and respecting the transition point, PLayer-FL ensures that collaboration provides value to all participants – addressing a fundamental barrier to FL adoption in cross-silo settings.

Validation of the transition point approach. These results provide comprehensive empirical validation of our core hypothesis: *federating early, generalizable layers while keeping later, task-specific layers local provides robust protective guarantees*. The superiority of PLayer-FL over PLayer-FL-Random further demonstrates that our federation sensitivity metric successfully identifies the transition point – achieving advantages beyond those of arbitrary partial federation. Together, these findings establish that principled identification of a transition point enables FL that is simultaneously effective, equitable, and protective.

Threshold robustness and outlier resistance. PLayer-FL uses a single hyperparameter, the threshold t, to determine the split point. Figure A.1 shows that the selected split layer is stable across a wide range of t values: the split chosen at the default t=2 persists from roughly t=1.2 to t=10 in five of seven benchmarks, reflecting the sharp transition in federation sensitivity (Figure 4). Across 10 independent runs at t=2, the standard deviation of the selected split point is below 0.25 for all datasets (Table A.2). We also evaluate a Median aggregation variant that replaces the default summation of per-client sensitivity scores, providing robustness to anomalous clients. Both strategies select the same split layer at t=2 for all datasets, confirming that no single client dominates the split decision. Full details are in Appendix A.3.

Communication and computational overhead. A detailed system-level analysis is provided in Appendix A.7.1; we summarize the key findings here. Each round, PLayer-FL communicates only the federated layers (layers 1 to p) rather than the full model, reducing per-round communication volume in proportion to the fraction of parameters kept local. The magnitude of savings is architecture-dependent: 1–8% for the CNN benchmarks, where compact fully-connected heads mean most parameters precede the split point, and 13–25% for the Transformer and MLP architectures (Sentiment, MIMIC-III, Heart), where a larger share of parameters falls after the transition point (Table A.13). Computationally, PLayer-FL has the same per-round cost as FedAvg, since both train the full model locally, the split affects only which parameters are sent to the server. The sole additional overhead is the one-time metric computation after epoch 1 (Algorithm 2), which reuses gradients already available from training and completes in under 90 ms on CPU even for 17M-parameter models. The one-time communication cost of transmitting the per-layer sensitivity scores is under 500 bytes for all datasets, and memory requirements are identical to FedAvg.

8 Conclusion

In this work, we present PLayer-FL, a principled partial FL algorithm that uses a novel federation sensitivity metric to determine which layers to federate and which to locally train. We show that this metric correlates with known measures of generalizability and model similarity and that it emerges after only one epoch of training. As such, it can be readily incorporated into the training algorithm with minimal computational overhead. Over a set of rigorous experiments that include benchmark and real-world datasets, we show that PLayer-FL outperforms existing algorithms in cross-silo settings. We also show that it produces fairer outcomes and is more likely to incentivize participation in FL.

Table 3. Fairness and incentivization average ranks, Friedman rank test p-value $< 5 \times 10^{-3}$ and $= 0.043$, respectively.

Algorithm	Fairness	Incentivization
FedProx	7.7	5.4
pFedMe	4.3	5.1
Ditto	5.0	5.6
LocalAdaptation	6.4	5.2
FedBABU	4.4	4.4
FedLP	6.1	6.2
FedLAMA	5.6	7.3
pFedLA	5.1	7.7
PLayer-FL	**3.8**	**3.4**
PLayer-FL-Random	6.5	4.8

Limitations: Our work demonstrates strong empirical evidence for the federation sensitivity metric across diverse model architectures but currently lacks theoretical guarantees. However, the success of similar metrics in related fields and the established understanding of generalizable versus task-specific layers reinforces the validity and utility of our approach. Developing formal theoretical foundations presents a promising direction for future research. Additionally, our study primarily focuses on cross-silo FL, where clients have sufficient data to calculate federation sensitivity, rather than the cross-device setting. However, this focus is deliberate, as cross-silo FL represents a significant sub-field with critical applications in industries like healthcare. These settings face unique challenges, making the development of tailored method essential (Terrail et al., 2022; Huang et al., 2022; Kairouz et al., 2021; Pati et al., 2021).

ACKNOWLEDGMENTS

This work has been funded by the National Institutes of Health (NIH) under grant number R01LM014380.

REFERENCES

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Briggs, C., Fan, Z., and Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2020.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Cho, Y. J., Jhunjhunwala, D., Li, T., Smith, V., and Joshi, G. To federate or not to federate: Incentivizing client participation in federated learning. *arXiv preprint arXiv:2205.14840*, 2022.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Dayan, I., Roth, H. R., Zhong, A., Harouni, A., Gentili, A., Abidin, A. Z., Liu, A., Costa, A. B., Wood, B. J., Tsai, C.-S., et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, 27 (10):1735–1743, 2021.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

Dinh, C. T., Vu, T. T., Tran, N. H., Dao, M. N., and Zhang, H. Fedu: A unified framework for federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 400, 2021.

Divi, S., Lin, Y.-S., Farrukh, H., and Celik, Z. B. New metrics to evaluate the performance and fairness of personalized federated learning. *arXiv preprint arXiv:2107.13173*, 2021.

Duan, M., Liu, D., Chen, X., Tan, Y., Ren, J., Qiao, L., and Liang, L. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th international conference on computer design (ICCD)*, pp. 246–254. IEEE, 2019.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Huang, C., Huang, J., and Liu, X. Cross-silo federated learning: Challenges and opportunities. *arXiv preprint arXiv:2206.12949*, 2022.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Lee, S., Zhang, T., and Avestimehr, A. S. Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8491–8499, 2023.

Li, Q., Diao, Y., Chen, Q., and He, B. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978. IEEE, 2022.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Ma, X., Zhang, J., Guo, S., and Xu, W. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10092–10101, 2022.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., and Kautz, J. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11264–11272, 2019.

Morcos, A., Raghu, M., and Bengio, S. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018.

Oh, J., Kim, S., and Yun, S.-Y. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

Pati, S., Baid, U., Zenk, M., Edwards, B., Sheller, M., Reina, G. A., Foley, P., Gruzdev, A., Martin, J., Albarqouni, S., et al. The federated tumor segmentation (fets) challenge. *arXiv preprint arXiv:2105.05874*, 2021.

Pati, S., Baid, U., Edwards, B., Sheller, M., Wang, S.-H., Reina, G. A., Foley, P., Gruzdev, A., Karkada, D., Davatzikos, C., et al. Federated learning enables big data for rare cancer boundary detection. *Nature communications*, 13(1):7346, 2022.

Reed, R. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

T Dinh, C., Tran, N., and Nguyen, J. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Terrail, J. O. d., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *arXiv preprint arXiv:2210.04620*, 2022.

Wang, J., Yang, X., Cui, S., Che, L., Lyu, L., Xu, D. D., and Ma, F. Towards personalized federated learning via heterogeneous model reassembly. *Advances in Neural Information Processing Systems*, 36, 2024.

Wang, L., Hu, L., Gu, J., Hu, Z., Wu, Y., He, K., and Hopcroft, J. Towards understanding learning representations: To what extent do different neural networks learn the same representation. *Advances in neural information processing systems*, 31, 2018.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Yu, T., Bagdasaryan, E., and Shmatikov, V. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhu, Z., Shi, Y., Luo, J., Wang, F., Peng, C., Fan, P., and Letaief, K. B. Fedlp: Layer-wise pruning mechanism for communication-computation efficient federated learning. In *ICC 2023-IEEE International Conference on Communications*, pp. 1250–1255. IEEE, 2023.