When Aggregation Stops Collaborating: Layer-wise Inertia in Low-Data Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Federated learning (FL) enables collaborative model training across decentralized clients while preserving data privacy, leveraging aggregated updates to build robust global models. However, this training paradigm faces significant challenges due to data heterogeneity, and each client has access to only scarce local training data, which often impedes effective collaboration. In such scenarios, we reveal that the collaboration bottleneck is closely tied to the *Layer-wise Inertia Phenomenon* in FL, where intermediate layers of the global model rapidly become stagnant after early communication rounds, ultimately weakening the effectiveness of global aggregation. We demonstrate the presence of this phenomenon across a wide range of federated settings, spanning diverse datasets and architectures. To address this issue, we propose LIPS (Layer-wise Inertia Phenomenon with Sparsity), a simple yet effective method that periodically introduces *transient sparsity* to stimulate meaningful updates and empower global aggregation. Experiments demonstrate that LIPS effectively mitigates layer-wise inertia, enhances aggregation effectiveness, and improves overall performance in various FL scenarios. This work not only deepens the understanding of layer-wise learning dynamics in FL but also paves the way for more effective collaboration strategies in resource-constrained environments.

1 Introduction

Federated learning (FL) (McMahan et al., 2016; Yang et al., 2019) enables collaborative training of machine learning models across decentralized clients while keeping the raw data local, making it a widely used solution for addressing privacy concerns and data access limitations in domains such as healthcare (Rieke et al., 2020; Sadilek et al., 2021), finance (Long et al., 2020), and personalized services (Long et al., 2020; Wen et al., 2023). This paradigm is particularly important in low-data scenarios, where each client may only possess a limited number of labeled samples due to privacy restrictions, annotation costs, or the rarity of certain conditions (Nguyen et al., 2022; Rieke et al., 2020).

However, a significant challenge in FL lies in the non-independent and identically distributed (non-IID) nature of data across clients (Zhao et al., 2018; Li et al., 2022). Since clients may collect data from different populations, devices, or environments, their local updates can diverge substantially, making it difficult to train a single global model that generalizes well across all clients. This challenge becomes even more severe in low-data regimes, where the limited number of local samples makes client models more prone to overfitting and further amplifies the instability of local updates.

While personalized and layer-wise aggregation strategies have been proposed to mitigate client heterogeneity (Li et al., 2020; Tan et al., 2022; Wu et al., 2023; Tamirisa et al., 2024), their behavior in low-data regimes remains insufficiently understood. For example, FedBN keeps batch normalization layers local to handle feature distribution shifts (Li et al., 2021c), while FedRep separates shared representations from personalized classifiers (Collins et al., 2021). These methods implicitly assume that the layers selected for global aggregation remain effective carriers of collaborative knowledge throughout training. However, this assumption becomes questionable in low-data regimes, where limited local supervision may weaken the reliability of client updates and, consequently, the collaborative value of the aggregated layers.

In this paper, we systematically investigate layer-wise aggregation behaviors in low-data FL. Our investigation reveals a counterintuitive phenomenon: instead of continuously benefiting from aggregation, many intermediate layers of the global model quickly become inert after the early communication rounds. We refer to this behavior as the *Layer-wise Inertia Phenomenon*. This finding indicates that aggregation does not necessarily imply effective collaboration; some shared layers may remain globally aggregated in form, yet contribute little meaningful learning in practice. What's worse, the layer-wise inertia phenomenon intensifies in deeper models and persists across diverse data distributions and client numbers, undermining FL's core promise of effective collaboration. We refer to this limitation as the *collaboration dilemma* in low-data FL.

To address this collaboration dilemma, we propose **Layer-wise Inertia Phenomenon with Sparsity (LIPS)**, a novel strategy for revitalizing ineffective layer-wise aggregation in low-data FL. The key idea of LIPS is to disrupt inert learning dynamics by periodically introducing *transient sparsity* after communication rounds. Specifically, LIPS temporarily deactivates low-sensitivity parameters to prevent the model from repeatedly relying on stagnant weights and to encourage subsequent local training to produce more effective updates after aggregation. By reshaping layer-wise update dynamics, LIPS promotes more effective global updates and improves the ability of the aggregated model to generalize across heterogeneous clients. We evaluate LIPS on diverse datasets and architectures, including CIFAR-10 (Krizhevsky et al., 2010), CIFAR-100 (Krizhevsky et al., 2009), and TinyImageNet (Le & Yang, 2015), with VGG (Simonyan & Zisserman, 2015) and ResNet-based (He et al., 2016) models. Extensive experiments show that LIPS consistently improves performance under low-data and heterogeneous FL settings, demonstrating its effectiveness in mitigating layer-wise inertia and enhancing cross-client collaboration.

Our contributions in this work can be summarized as follows:

- We identify and demonstrate the existence of the layer-wise inertia phenomenon in low-data federated learning, where certain layers exhibit early stagnation.

- We analyze the underlying causes of this phenomenon and highlight its negative impact on learning dynamics and its constraint on the overall performance of federated learning.

- We propose LIPS, a simple yet effective method that introduces transient sparsity to mitigate the inertia phenomenon and enhance the aggregation during federated training.

- We validate the effectiveness of LIPS in enhancing model performance and promoting more effective collaboration across a variety of federated learning scenarios.

2 Preliminary

Federated learning. Federated learning (FL) aims to train a shared model across multiple decentralized clients without directly accessing their private local data. Let $\mathcal{D}_i$ denote the local dataset of client i, and let $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ be the collection of all client data. In standard FL, the server maintains a global model parameterized by $\mathbf{w}$, while each client updates a local copy using its own dataset. A representative algorithm is FedAvg (McMahan et al., 2017), where clients perform local optimization and periodically send their updated model parameters to the server for aggregation.

The standard FL objective can be written as:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{i=1}^{n} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} F_i(\mathbf{w}), \tag{1}$$

where $F_i(\mathbf{w}) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \mathcal{L}(f(\mathbf{w}; x), y)$ is the empirical loss on client i. Here, $f(\mathbf{w}; x)$ denotes the model prediction for input x, and $\mathcal{L}$ is the training loss, such as cross-entropy.

Although FedAvg provides a simple and effective mechanism for cross-client collaboration, its effectiveness depends on whether local updates contain useful and compatible information for global aggregation. This becomes challenging under non-IID data distributions, where clients may optimize toward different local objectives and produce divergent updates. The challenge is further amplified in low-data regimes, where each client has access to only a limited number of samples. In such cases, local training can quickly overfit to small client-specific datasets, making the resulting updates weak, biased, or unstable. Consequently, the aggregated global model may not fully benefit from the intended collaborative effect of FL.

To mitigate client heterogeneity, recent methods selectively localize or aggregate different model components (Collins et al., 2021; Li et al., 2021c; Zhang et al., 2023). For example, some approaches keep task- or distribution-sensitive layers client-specific, while aggregating the remaining layers globally. These methods suggest that different layers play different roles in federated optimization. However, they also raise an important question central to this work: once a layer is selected for global aggregation, does it continue to receive meaningful collaborative updates throughout training, especially when local data are scarce? This motivates us to examine the layer-wise evolution of the global model under low-data FL.

Layer-wise cosine similarity (global model). To study whether globally aggregated layers continue to evolve during federated training, we measure the layer-wise change of the global model across communication rounds. Specifically, we use cosine similarity between the parameters of each layer at a given round and those at an earlier reference round. Cosine similarity has been widely used to quantify weight stability, representation drift, and model evolution over training (Chen et al., 2025; Gromov et al., 2025; Min & Wang, 2025; Jiang et al., 2025). In our setting, it provides a direct way to characterize whether a layer remains actively updated or gradually becomes inert.

For the l-th layer of the global model, we define the layer-wise cosine similarity at communication round t with respect to a reference round t_0 as:

$$C_l^t = \frac{\mathbf{w}_l^t \cdot \mathbf{w}_l^{t_0}}{\|\mathbf{w}_l^t\| \left\|\mathbf{w}_l^{t_0}\right\|}, \tag{2}$$

where $\mathbf{w}_l^t$ and $\mathbf{w}_l^{t_0}$ denote the vectorized parameters of the l-th layer in the global model at rounds t and t_0, respectively. A value of C_l^t close to 1 indicates that the layer parameters remain highly similar to their earlier state, suggesting limited layer-wise evolution. In contrast, a smaller value indicates more substantial parameter changes over communication rounds. Therefore, persistently high cosine similarity after early training rounds can serve as evidence that a globally aggregated layer has entered an inert state, where aggregation continues procedurally but contributes little effective learning.

3 Layer-wise Inertia Phenomenon in Low-Data FL

To better understand how different layers of the global model behave during training in low-data FL, we conduct experiments on the CIFAR-10, CIFAR-100, and TinyImageNet datasets using standard FedAvg algorithm, modified to keep batch normalization (BN) layers local to each client. This adjustment, inspired by (Li et al., 2021c), mitigates the negative effects of distribution shifts and improves the stability of the model in heterogeneous settings. To simulate low-data and heterogeneous FL, we partition the training data across clients in a non-IID manner using a Dirichlet distribution with concentration parameter $\alpha = 0.1$, while limiting each client to 100 training samples. Further implementation details are provided in Section 5.

We track the layer-wise cosine similarity of the global model after each aggregation step, using its state at the second communication round as the reference, i.e., $t_0 = 2$.[1] A high cosine similarity indicates that the corresponding layer remains close to its early state, suggesting limited parameter evolution. This measurement allows us to directly investigate a central question of this work: whether the layers aggregation continue to carry effective collaborative signals throughout federated training.

① **In low-data FL, the global model exhibits the Layer-wise Inertia Phenomenon, wherein updates stagnate early and hinder further adaptation.** As shown in Figure 1, we uncover an intriguing phenomenon: most layers, particularly the middle layers, exhibit minimal changes during the training process,

[1] We select an early post-aggregation state as the reference because it captures the model after cross-client aggregation has begun, while avoiding the large transition from random initialization to the first aggregated model.

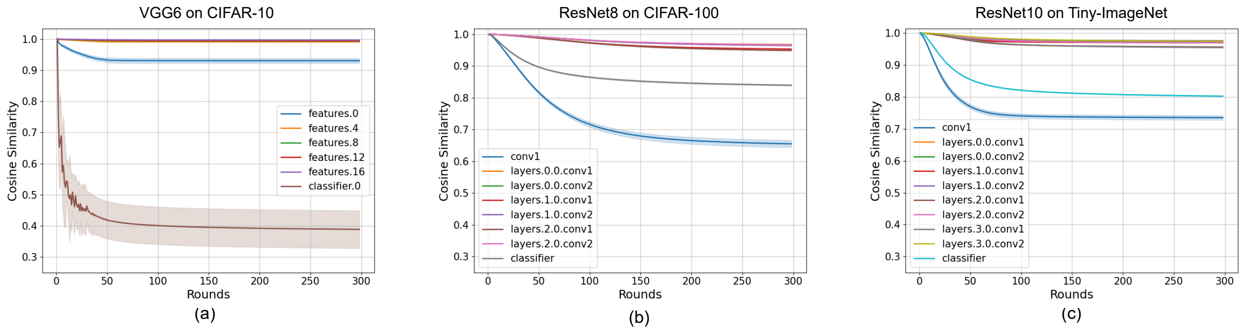

Figure 1: Layer-wise cosine similarity of global model throughout the training process, computed with respect to an early-round reference state. Results are reported across CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets using VGG6, ResNet-8, and ResNet-10 architectures, respectively, as shown in (a), (b), and (c). The number of clients is set to 100, with each client having 100 training samples for CIFAR-10 and CIFAR-100, and 300 training samples for Tiny-ImageNet, under a Dirichlet data distribution with Dir(α=0.1). The legend indicates the layer names for each architecture.

after the early phases of training. This is evidenced by the consistently high cosine similarity values (above 0.95) overall of the middle layers' weights when compared to their states in the early training phase. Notably, this behavior is consistently observed across different datasets and architectures. Additional results for other settings can be found in Appendix 10.

We term this phenomenon **Layer-wise Inertia** in FL, describing the tendency of certain layers to experience stagnation, with minimal or slow updates during aggregation. This stagnation implies that the majority of layers contribute little to the overall model updates after the early training phases, leaving only a few layers (e.g., the first and last) as the primary drivers of collaboration. These observations expose a key tension in low-data FL. Federated aggregation may fail to sustain effective learning dynamics in many globally shared layers. This motivates a closer examination of whether inert layers continue to contribute meaningfully to global model performance.

② **Layer-wise Inertia limits the collaborative effectiveness of aggregation in the global model.** We might ask: *How does layer-wise inertia in the middle layers impact aggregation effectiveness?* To investigate this, we conduct an experiment where the first and last layers of the model are aggregated throughout training, while the weights of middle layers remain fixed on each client after certain communication rounds. For simplicity, we set this point at round 50 for all datasets, based on observations in Figure 1, where most layers tend to exhibit minimal updates within clients beyond this point. We term this approach `w/.fix`, while the standard approach, where the entire model is aggregated throughout training, is termed `w/o.fix`.

As shown in Table 1, the performance under the `w/.fix` approach is surprisingly comparable to that achieved with `w/o.fix`. This suggests that middle layer aggregation contributes minimally after the early stages of communication. This can be attributed to the minimal updates in the middle layers, which significantly restrict collaborative updates from clients and diminish the overall effectiveness of aggregation. These findings suggest that mitigating layer-wise inertia, especially in the middle layers, may be key to unlocking more effective collaboration and improving model performance in low-data FL.

Table 1: Comparison of test accuracy on CIFAR-10, CIFAR-100, and TinyImageNet in federated learning, with and without fixing middle layer aggregation after the early stages of communication.

Dataset	CIFAR-10	CIFAR-100	Tiny ImageNet
`w/o.fix`	85.83 ± 1.28	43.08 ± 0.45	36.83 ± 0.22
`w/.fix`	85.15 ± 1.59	43.07 ± 0.53	36.86 ± 0.08

③ **Client diversity across data distributions and client numbers struggles to solve the Layer-wise Inertia Phenomenon.** To investigate whether standard configuration adjustments in FL can mitigate this stagnation, we examine the layer-wise cosine similarity C_l^T under varying numbers of clients and different levels of data heterogeneity (via Dirichlet distributions), as shown in Figure 2.

Across all settings, we consistently observe higher cosine similarity values, indicating that the final states of the middle layers in the global model remain closer to their early states, regardless of the data distribution or the number of clients. These results suggest that increasing client heterogeneity, either by altering the number of clients or the non-IID level, does not meaningfully mitigate the stagnation of specific layers. This highlights a fundamental limitation in low-data FL, where standard sources of variability are insufficient to promote more effective collaboration across all layers. This underscores the importance of explicitly addressing layer-wise inertia to revitalize stale layers and enhance the overall effectiveness of collaboration through aggregation in low-data FL.

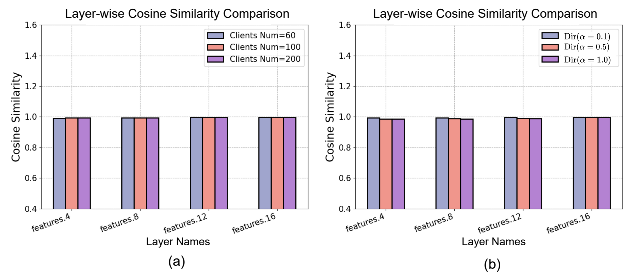

Figure 2: Comparison of the global model's layer-wise cosine similarity after training on CIFAR-10. (a) Varying number of clients and (b) under different data distributions.

4 LIPS: Reactivating Inert Layers via Transient Sparsity

Section 3 shows that many aggregated layers, quickly become inert in low-data FL. To understand its underlying cause, we examine how layer-wise inertia varies with model capacity and data availability.

As shown in Figure 3(a) and (b), deeper models exhibit stronger inertia: compared with ResNet6, ResNet10 shows higher cosine similarity across more intermediate layers. Meanwhile, Figure 3(c) and (d) show that reducing the number of samples per client from 500 to 300 and 100 consistently intensifies this stagnation. These results suggest that layer-wise inertia is closely related to the capacity–data imbalance in low-data FL: when model capacity is large but local supervision is scarce, clients can rapidly overfit to limited local samples and produce insufficiently diverse updates for aggregation.

This observation motivates us to introduce an explicit mechanism to regularize local training dynamics and reactivate stagnant layers. To this end, we propose **LIPS** (**L**ayer-wise **I**nertia **P**henomenon with **S**parsity), a simple and model-agnostic method that introduces sparsity to mitigate layer-wise inertia in low-data FL. By explicit introducing sparsity, LIPS disrupts over-reliance on saturated weights, and encourages alternative update, thereby restoring more effective collaborative updates during FL.

4.1 Overview of LIPS

LIPS mitigates layer-wise inertia through two coordinated steps: *Sensitivity-Guided Parameter Selection* and *Periodic Transient Sparsity*. In Sensitivity-Guided Parameter Selection, LIPS first selects low-sensitivity parameters in inertia-prone intermediate layers, which are likely to contribute little to current learning dynamics. In Periodic Transient Sparsity, LIPS periodically masks these selected parameters to reduce the model's repeated reliance on stagnant weights. After aggregation, it helps to explore alternative update pathways and stimulate stagnant layer-wise dynamics. The masked parameters remain trainable during subsequent dense local training, allowing LIPS to reactivate inert layers without permanently reducing model capacity. Algorithm 1 summarizes the overall procedure.

4.2 Sensitivity-Guided Parameter Selection

In LIPS, a crucial step is determining which parameters should be temporarily suppressed after aggregation. The goal is not to permanently remove unimportant parameters, but to identify weights that are likely to contribute to stagnant learning dynamics and reduce their influence during aggregation. Intuitively, inertia-inducing weights tend to be weakly involved in optimization: they have small magnitudes and receive negligible updates, suggesting that they contribute little to the current representation and provide limited useful signals for collaboration.

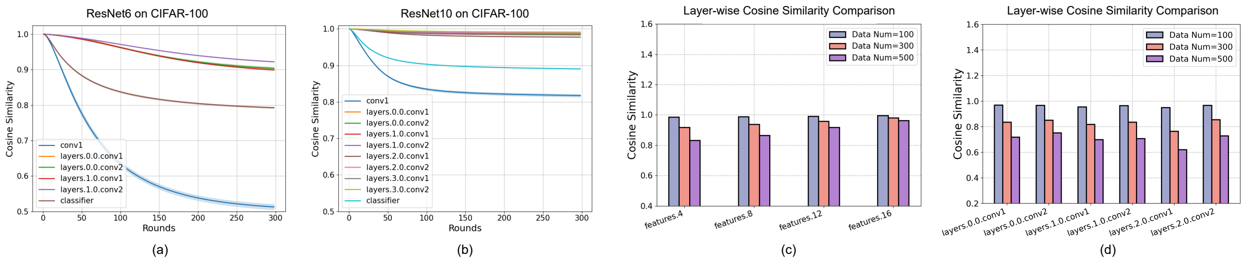

Figure 3: Layer-wise cosine similarity of the global model after aggregation. (a) and (b): Results on CIFAR-100 with 100 clients using ResNet6 and ResNet10 under Dir(α=0.1). (c) and (d): Comparison under different data volumes per client (100, 300, 500 samples) on CIFAR-10 and CIFAR-100.

To this end, LIPS employs a **sensitivity-based criterion** to identify low-contribution parameters at the client level, as discussed in prior research (Lee et al., 2019; Wu et al., 2023; Nowak et al., 2024). In low-data FL, parameters with low sensitivity may correspond to weights that are both small in magnitude and weakly updated, indicating that they are unlikely to drive meaningful optimization and may instead preserve stagnant layer-wise dynamics. Temporarily suppressing these parameters reduces the model's reliance on stagnant weights, thereby providing a way to reshape its learning dynamics. After global aggregation, the resulting model encourages subsequent local training to move beyond stagnant update patterns and produce more effective updates.

Algorithm 1 LIPS

Input: Initial client models $\{\mathbf{w}_i^0\}_{i=1}^n$; number of clients n; total communication rounds T; local epochs E; initial sparsity ratio τ_0; sparsification interval k.
Output: Final client models $\{\mathbf{w}_i^T\}_{i=1}^n$.
for $t = 1$ to T **do**
  Client-side:
  if $t \bmod k = 0$ **then**
    Compute the sparsity ratio $\tau(t; \tau_0, T)$.
    Estimate parameter sensitivity by Eq. 3.
    Construct M_i^t for low-sensitivity parameters by Eq. 4.
    Apply transient sparsity: $\mathbf{w}_i^t \leftarrow M_i^t \odot \mathbf{w}_i^t$ by Eq. 5.
  end if
  for $i = 1$ to n **in parallel do**
    Train $\mathbf{w}_i^t$ for E local epochs.
    Send updated $\mathbf{w}_i^t$ to the server.
  end for
  Server-side:
  Aggregate client models to obtain $\overline{\mathbf{w}}^t$.
  Send $\overline{\mathbf{w}}^t$ to each client.
  Client-side:
  for $i = 1$ to n **in parallel do**
    Initialize $\mathbf{w}_i^{t+1}$ with $\overline{\mathbf{w}}^t$, excluding BN layers.
  end for
end for

Specifically, the sensitivity of the j-th parameter $w_{i,j}^t$ in client i at communication round t is computed as:

$$s_{i,j}^t = \left|\Delta w_{i,j}^t \cdot w_{i,j}^t\right|, \tag{3}$$

where $\Delta w_{i,j}^t$ denotes the update of parameter $w_{i,j}^t$ during local training at round t.

This metric provides a lightweight estimate of how actively a parameter participates in current optimization. Parameters with lower sensitivity $s_{i,j}^t$ are prioritized for temporary masking, as they are more likely to correspond to inactive weights that sustain stagnant layer-wise dynamics. A detailed derivation of this criterion is provided in Appendix 9, with further discussion and comparison to other criteria presented in Section 5.2.

4.3 Periodic Transient Sparsity

After evaluating parameter sensitivity using Eq. 3, LIPS temporarily zeros out a fraction τ of parameters with the lowest sensitivity in each selected layer for client i. This operation is implemented using a binary mask M_i^t, which determines whether each parameter is retained or suppressed at communication round t. The mask is defined as:

$$m_{i,j}^t = \begin{cases} 0, & \text{if } s_{i,j}^t \text{ is among the lowest } \tau \text{ fraction in its layer,} \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

The sparsified model is then obtained by:

$$\mathbf{w}_i^t = M_i^t \odot \mathbf{w}_i^t, \tag{5}$$

which serves as a perturbed initialization for the next local training stage. The sparsity ratio τ is decayed over time, allowing LIPS to apply stronger capacity reallocation in earlier rounds and promote convergence stability in later rounds:

$$\tau(t; \tau_0, T) = \tau_0 \left(1 - \frac{t}{T}\right), \tag{6}$$

where τ_0 is the initial sparsity ratio and T is the total number of communication rounds.

The mask is applied periodically every k communication rounds to avoid excessive disruption to optimization. Importantly, this sparsity is *transient*: when sparsification is triggered, the mask is applied only once after aggregation, and all parameters remain trainable. Thus, LIPS reshapes the starting point of local training without reducing local model capacity.

By periodically introducing transient sparsity, LIPS encourages the model to redistribute its learning capacity, stimulates underutilized parameters, and mitigates stagnant layer-wise dynamics. We compare this design with maintaining sparsity during local training in Appendix 15. Based on our observations, layer-wise inertia is most pronounced in intermediate layers. Therefore, LIPS applies transient sparsity only to the middle layers, while excluding the first and last layers from sparsification.

5 Experiments

Dataset and architectures. We conduct experiments on three datasets, including CIFAR-10 (Krizhevsky et al., 2010), CIFAR-100 (Krizhevsky et al., 2009), and TinyImageNet (Le & Yang, 2015). To evaluate the effectiveness of our method in different scenarios, we use the commonly adopted Dirichlet non-IID setting (Hsu et al., 2019; Lin et al., 2020; Wu et al., 2023), where each client's data is sampled from a Dirichlet distribution $q \sim \text{Dir}(\alpha p)$. Here, p is the class prior, and α controls the degree of non-IID. Smaller α induces stronger label-distribution heterogeneity across clients, whereas larger α yields more balanced local label distributions and reduces inter-client distribution shifts. This approach effectively captures diverse and complex non-IID scenarios, making it a robust evaluation method. We study three different architectures: specifically, VGG6 for CIFAR-10, ResNet-8 for CIFAR100, and ResNet-10 for TinyImageNet as in (Wu et al., 2023).

We evaluate LIPS against several baselines, including FedAvg (McMahan et al., 2017), FedRep (Collins et al., 2021), FedBN (Li et al., 2021c), pFedSD (Jin et al., 2022), FedDrop (Wen et al., 2022) and FedCAC (Wu et al., 2023). Additionally, we include a baseline, termed as "Separate", where models are trained locally on client data without any federated aggregation. To highlight the effectiveness of client collaboration in the low-data regime, we assign a limited amount of data to each client in our main experiments. Specifically, for CIFAR-10 and CIFAR-100, each client is assigned 100 training samples, while for TinyImageNet, each client is assigned 300 training samples along with 400 test samples. Each task involves 100 clients, and the test data follows the same distribution as the training data to ensure consistent evaluation. Additionally, we extend our exploration to scenarios with 200 and 300 clients and varying training sample sizes of 60, 300, 500, and 700 per client. We provide more details of the implementation in Appendix 8.

5.1 Overall Performance

In this section, we will thoughtfully evaluate the effectiveness of the proposed LIPS across CIFAR-10, CIFAR-100, and TinyImageNet datasets. Our evaluation considers a range of scenarios, including varying data distributions, different numbers of clients, and diverse numbers of data per client.

5.1.1 Performance Across Datasets.

We evaluate our approach under Dirichlet non-IID scenarios with α values of 0.1, 0.5, 1.0 for CIFAR-10 and 0.01, 0.1, 0.5 for CIFAR-100 and TinyImageNet, as the same settings in (Wu et al., 2023). As shown

Table 2: Performance comparison on CIFAR-10, CIFAR-100, and TinyImageNet using different architectures (VGG6, ResNet-8, and ResNet-10, respectively), with 100 clients, each holding 100 samples, under varying values of α.

Method	CIFAR-10			CIFAR-100			TinyImageNet		
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0.5$
Separate	77.65±2.57	51.01±0.74	41.20±0.29	78.12±0.93	34.96±0.14	14.35±0.51	64.23±0.29	22.89±0.15	8.00±0.54
FedAvg	60.45±2.63	67.76±1.11	68.31±0.62	13.96±0.12	24.51±0.87	26.34±1.56	7.93±0.57	11.75±0.36	12.63±0.37
FedRep	81.09±1.94	59.76±1.11	51.47±0.42	76.96±1.90	34.31±0.16	13.71±0.38	60.24±1.53	18.31±0.12	8.09±0.15
pFedSD	80.82±1.87	62.04±0.91	60.60±0.65	73.22±1.11	38.62±0.36	21.84±0.57	53.29±0.62	28.12±0.43	16.23±0.35
FedBN	85.83±1.28	72.69±0.75	68.66±0.56	79.28±0.72	43.08±0.45	24.45±0.61	72.55±0.63	36.83±0.22	20.73±1.23
FedDrop	84.54±1.32	73.24±0.54	70.64±0.35	80.13±0.76	47.14±0.77	26.38±1.04	73.45±0.14	36.56±0.62	21.34±0.45
FedCAC	84.74±1.46	70.79±0.61	66.93±0.65	78.87±0.50	43.96±0.54	25.31±0.54	68.93±0.56	32.50±0.32	18.50±0.53
LIPS	**86.21±1.66**	**74.78±0.68**	**72.07±0.38**	**81.39±0.43**	**47.84±0.47**	**28.97±1.02**	**74.83±0.08**	**40.61±0.54**	**23.53±0.15**

in Table 2, most baselines achieve significant improvements over the "Separate" approach in the Dirichlet non-IID scenario, particularly at higher α values, indicating that federated learning can effectively enhance performance. However, the experimental results reveal notable differences in performance across datasets. For CIFAR-100 and TinyImageNet, the increased number of classes complicates local tasks and increases the risk of overfitting, which diminishes the effectiveness of methods such as FedRep and FedCAC.

Additionally, FedDrop performs competitively against other baselines, indicating that dropout in FL can partially alleviate the challenges of low-data regime. This supports our motivation that sparsity is useful for improving stagnant learning dynamics. However, LIPS consistently outperforms FedDrop across datasets and α settings, showing that generic sparsity is not enough. By applying sensitivity-guided transient sparsity to inertia-prone layers, LIPS more effectively improves collaborative aggregation.

5.1.2 Impact on Data Distribution.

As shown in Table 2, LIPS consistently improves performance across various degrees of client data heterogeneity. It is particularly effective in more challenging scenarios involving highly diverse local data distributions, such as Dir(α=1.0) for CIFAR-10 and Dir(α=0.5) for CIFAR-100 and TinyImageNet. In these settings, LIPS achieves performance gains of 3–4% over strong baselines. By explicitly accounting for differences in client data and local tasks, LIPS enables more effective global aggregation and fosters collaboration even under significant distribution shifts. This leads to consistent performance gains across diverse FL scenarios.

5.1.3 Varying Numbers of Data per Client.

As discussed in Section 3, smaller training datasets within clients exacerbate the layer-wise inertia phenomenon. To evaluate the effectiveness of the proposed LIPS method under different data regimes, we conduct experiments on CIFAR-10 and CIFAR-100 in non-IID settings with Dirichlet parameters of 0.1. Each client is assigned 60, 100, 300, 500, and 700 training samples, respectively.

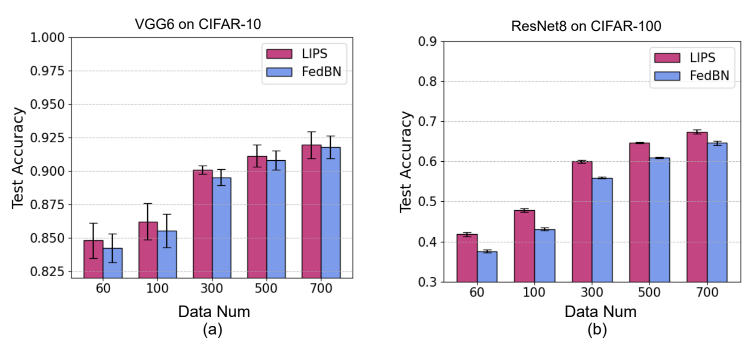

Figure 4: Performance comparison across varying numbers of training samples (Data Num) per client (100, 300, 500, and 700 samples) on (a) CIFAR-10 and (b) CIFAR-100 datasets under a data distribution with Dir(α=0.1).

The results, presented in Figure 4, demonstrate that LIPS achieves greater performance improvement under low-data settings compared with baseline. This can be attributed to the more pronounced layer-wise inertia phenomenon when client data is limited, which restricts the effectiveness of aggregation in FL. On the other hand, as the number of data increased per client, this effect diminishes intermediate layers receive richer updates, alleviating stagnation (as shown in Figure 3), and enabling more effective learning even with standard FL methods. Consequently,

Table 3: Comparison of LIPS and FedBN performance on CIFAR-10 with Dir(α=0.5) and CIFAR-100 with Dir(α=0.1), under different number of clients.

Method	CIFAR-10			CIFAR-100		
	$N = 100$	$N = 200$	$N = 300$	$N = 100$	$N = 200$	$N = 300$
FedBN	72.69	74.45	75.13	43.08	44.26	44.84
LIPS	74.78(2.09↑)	77.56(3.11↑)	78.61(3.48↑)	47.84(4.76↑)	49.58(5.32↑)	50.36(5.52↑)

standard FL methods become more effective, while the additional benefit of LIPS becomes less pronounced. These results demonstrate that LIPS is particularly effective in low-data regimes, where federated collaboration is most constrained by layer-wise inertia.

5.1.4 Impact on the Number of Clients.

To examine the impact of varying client numbers on the performance of LIPS, we conduct experiments on CIFAR-10 and CIFAR-100 under non-IID settings with Dir(α=0.5) and Dir(α=0.1), respectively. The number of clients is set to 100, 200 and 300, with each client having 100 training samples. As shown in Table 3, compared to FedBN, which is identical except for the absence of sparsity, the performance improvement achieved by LIPS becomes more pronounced as the number of clients increases. This is likely because a larger number of clients exacerbates the heterogeneity in data distributions, making it more challenging for traditional aggregation methods to converge effectively. LIPS redistributes learning capacity and fosters more meaningful updates, thereby mitigating the adverse effects of increased client diversity. These results highlight the scalability of LIPS and its effectiveness in various settings.

5.2 Ablation Study

Table 4: Performance comparison of various weight selection methods on CIFAR-10/100 with different values of α.

Method	CIFAR-10		CIFAR-100	
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.01$	$\alpha = 0.1$
FedBN	85.83±1.28	72.69±0.75	79.28±0.72	43.08±0.45
$\texttt{LIPS}_{\texttt{Magnitude}}$	86.01±1.18	72.56±0.49	79.40±0.91	43.31±0.04
$\texttt{LIPS}_{\texttt{Sensitivity}}$	**86.21±1.66**	**74.78±0.68**	**81.39±0.43**	47.84±0.47

Effect of weights selection methods. We compare our sensitivity-based criterion with an alternative weight selection method, Magnitude, which zeros out weights based on their magnitude values. Experiments are conducted under non-IID settings with different Dirichlet parameters on the CIFAR-10 and CIFAR-100 datasets. As shown in Table 4, the Magnitude method generally achieves accuracy comparable to the baseline without sparsity. In contrast, the Sensitivity-based method delivers consistently stronger performance across both datasets and various non-IID settings. This indicates that magnitude alone is insufficient for identifying weights associated with layer-wise inertia, whereas our sensitivity-based criterion more effectively selects stagnant weights for transient sparsification.

Effect of frequency k and initial sparsity ratio τ_0. We further investigate the impact of two key hyperparameters in LIPS: sparsification frequency k and the initial sparsity ratio τ_0, as detailed in Appendix 13. We find that introduce sparsity every 5 communication rounds ($k = 5$) strikes a good balance between model performance and adaptability across all datasets. This setting avoids overly frequent structural changes that may destabilize training while ensuring sufficient exploration of the parameter space. Regarding τ_0, we adopt $\tau_0 = 0.5$ for CIFAR-10 and $\tau_0 = 0.7$ for CIFAR-100 and TinyImageNet, as these configurations consistently yield better performance. These results suggest that both the timing and degree of sparsification play important roles in enabling LIPS to effectively counteract overfitting and promote more meaningful global aggregation.

Effect of weights initialization. When introducing sparsity in federated learning, an important design choice is how to initialize the weights that are zeroed out during sparsification for subsequent local training. We compare two strategies: (1) initializing these weights to zero ($\texttt{LIPS}_{\texttt{w/zero}}$) and (2) restoring them to

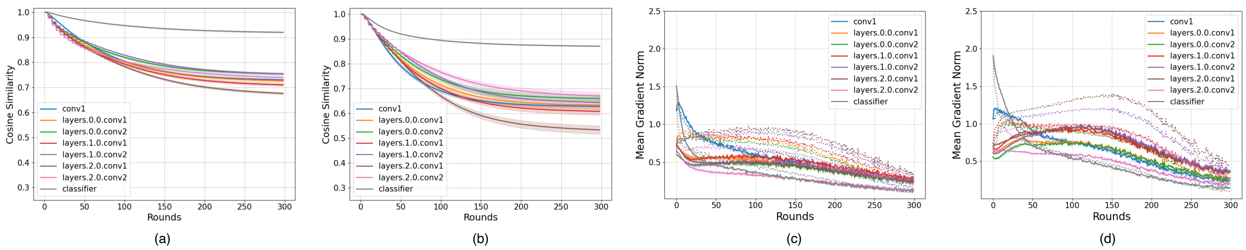

Figure 5: Layer-wise cosine similarity throughout the training process for LIPS on CIFAR-100 datasets under Dir(α=0.01) and Dir(α=0.1), shown in (a) and (b), respectively. A comparison of layer-wise mean gradient norms across all clients between FedBN (solid line) and LIPS (dotted line) after each training round under Dir(α=0.01) and Dir(α=0.1) in (c) and (d), respectively.

their original initialization values used before sparsification ($\texttt{LIPS}_{\texttt{w/init}}$). As shown in Table 5, in general, $\texttt{LIPS}_{\texttt{w/init}}$ achieves slightly better performance than zero initialization in most cases.

Although $\texttt{LIPS}_{\texttt{w/init}}$ achieves promising performance, it requires retaining the original initialization throughout training. For simplicity, we adopt zero initialization as the default setting in all experiments. That said, using the original initialization for reset weights presents a promising avenue for future research aimed at enhancing model performance.

Table 5: Performance comparison of weights initialization on CIFAR-10/100 with different values of α.

Method	CIFAR-10		CIFAR-100	
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.01$	$\alpha = 0.1$
FedBN	85.83±1.28	72.69±0.75	79.28±0.72	43.08±0.45
$\texttt{LIPS}_{\texttt{w/zero}}$	86.21±1.66	74.78±0.68	81.39±0.43	**47.84±0.47**
$\texttt{LIPS}_{\texttt{w/init}}$	**87.10±1.38**	**75.31±0.67**	**81.45±0.82**	47.80±0.62

5.3 Visualizing the Impact of LIPS on Layer Dynamics

Layer-wise cosine similarity. For LIPS, we track the global model's layer-wise cosine similarity relative to the 2nd round states across the training process on CIFAR-100 datasets, using ResNet-8 architectures under Dir(α=0.01) and Dir(α=0.1). The experiments are conducted following the same configuration as in Figure 1. As illustrated in Figure 5 (b), introducing sparsity during training significantly reduces the cosine similarity in the middle layers compared to their counterparts in Figure 1 (b). This reduction suggests that sparsity effectively mitigates the Layer-wise Inertia phenomenon by facilitating more substantial updates in these layers, thereby enhancing collaboration during global aggregation.

Layer-wise mean gradient norm. Under the same experimental settings, we further analyze the layer-wise mean gradient norm across all clients after each round of training, comparing FedBN and LIPS in Figure 5 (c) and (d). Our results reveal that, in contrast to FedBN, which does not utilize sparsity, LIPS increases the gradient norm during local client training. This increase in gradient norm is a crucial factor for activating weight updates, particularly for layers that typically stagnate under traditional FL settings. By fostering more active weight updates, LIPS improves the overall effectiveness of aggregation in federated learning, leading to a more robust global model that performs better across diverse client scenarios.

Additional results on other datasets and architectures are provided in Appendix 12.

6 Related Work

6.1 Personalization in Federated Learning

Personalized federated learning aims to address the issue of data heterogeneity by adapting clients to their local data distribution (Tan et al., 2022). Common approaches include multitask learning (Smith et al., 2017; Agarwal et al., 2020), clustering (Duan et al., 2021; Ghosh et al., 2020; Mansour et al., 2020), transfer learning (Yu et al., 2020; Zhang et al., 2021), meta learning (Singhal et al., 2021; Jiang et al., 2019), etc. Recently, partial model personalization has gained attention for improving client model performance by adapting specific components of the model to local tasks. For instance, approaches like FedRep (Collins et al., 2021) and FedPer (Arivazhagan et al., 2019) focus on personalizing certain layers, such as classifier layers, while preserving globally shared representations. Furthermore, advanced aggregation methods, such as FedProx (Li et al., 2020) and Scaffold (Karimireddy et al., 2020), incorporate regularization terms or variance reduction techniques to balance local and global learning objectives effectively. While these approaches tackle specific aspects of model adaptation and aggregation, they often overlook the layer-wise dynamics that critically influence aggregation effectiveness.

6.2 Federated Learning with Limited Data

Federated learning in low-data settings presents a unique set of challenges, where individual clients possess only a small number of samples. This amplifies the risks of overfitting and limits the representativeness of local updates, undermining the benefits of collaboration. Prior works such as FedMix (Yoon et al., 2021), FedGEN (Venkateswaran et al., 2023), and FedNTD (Lee et al., 2022) explore data augmentation and generative modeling to alleviate data scarcity. Other efforts focus on regularization-based methods (Li et al., 2020; Jeon et al., 2023; Yuan et al., 2022; Li et al., 2021b) to improve model generalization under limited supervision. Despite these advances, few works systematically examine how limited data influences layer-wise training dynamics during federated optimization. Our work addresses this gap by identifying the Layer-wise Inertia Phenomenon in low-data FL and proposing a sparsity-driven solution to enhance model adaptation and aggregation efficacy in such constrained environments.

6.3 Sparsity in Federated Learning

Sparsity has been extensively explored in FL, primarily as a strategy to enhance communication efficiency by reducing the amount of information exchanged between clients and the server. For instance, gradient sparsification techniques selectively transmit only the most significant gradients (Mitra et al., 2021; Han et al., 2020), while model pruning methods reduce the model size by eliminating redundant weights during communication rounds (Babakniya et al., 2023; Li et al., 2021a; Jiang et al., 2022; Chen et al., 2023). Additionally, recent advancements extend the benefits of sparsity beyond communication efficiency to improve local training efficiency (Bibikar et al., 2022; Huang et al., 2022; Dai et al., 2022; Kuo et al., 2024) through sparse training methods (Mocanu et al., 2018; Evci et al., 2020; Xiao et al., 2022; Liu et al., 2023; Wu et al., 2025). While these approaches effectively address bandwidth constraints and computational costs, their primary focus on optimizing resource usage. In contrast, our work aims to enhance aggregation in FL by dynamically introducing transient sparsity during training. This approach fosters more meaningful updates across layers, enabling more effective aggregation and improving the performance of the global model in heterogeneous federated learning scenarios.

7 Conclusion

In this work, we uncovered the Layer-wise Inertia Phenomenon in low-data federated learning (FL), where middle layers of the global model exhibit stagnation after early training rounds, severely impairing the exchange of meaningful updates across clients. To address this issue, we proposed a simple yet effective method, Layer-wise Inertia with Sparsity (LIPS), which periodically introduces transient sparsity during training to stimulate meaningful updates and mitigate the layer-wise inertia. Through extensive experiments, we showed that LIPS improves model performance across diverse non-IID settings by activating underutilized layers and enhancing the quality of aggregation. We also include a discussion of future directions and limitations in Appendix 17.

References

Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *arXiv preprint arXiv:2003.12880*, 2020.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Sara Babakniya, Souvik Kundu, Saurav Prakash, Yue Niu, and Salman Avestimehr. Revisiting sparsity hunting in federated learning: Why does sparsity consensus matter? *Transactions on Machine Learning Research*, 2023.

Sameer Bibikar, Haris Vikalo, Zhangyang Wang, and Xiaohan Chen. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6080–6088, 2022.

Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. In *International Conference on Machine Learning*, pp. 5234–5256. PMLR, 2023.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=IC5RJvRoMp`.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In *International Conference on Machine Learning*, pp. 4587–4604. PMLR, 2022.

Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International conference on machine learning*, pp. 2943–2952. PMLR, 2020.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Dan Roberts. The unreasonable ineffectiveness of the deeper layers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=ngmEcEer8a`.

Pengchao Han, Shiqiang Wang, and Kin K Leung. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In *2020 IEEE 40th international conference on distributed computing systems (ICDCS)*, pp. 300–310. IEEE, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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.

Tiansheng Huang, Shiwei Liu, Li Shen, Fengxiang He, Weiwei Lin, and Dacheng Tao. Achieving personalized federated learning with sparse local models. *arXiv preprint arXiv:2201.11380*, 2022.

Insu Jeon, Minui Hong, Junhyeog Yun, and Gunhee Kim. Federated learning via meta-variational dropout. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=VNyKBipt91`.

Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. Tracing representation progression: Analyzing and enhancing layer-wise similarity. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=vVxeFSR4fU`.

Xiaopeng Jiang and Cristian Borcea. Complement sparsification: Low-overhead model pruning for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8087–8095, 2023.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10374–10386, 2022.

Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):567–580, 2022.

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*, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010.

Kevin Kuo, Arian Raje, Kousik Rajesh, and Virginia Smith. Federated lora with sparse communication. *arXiv preprint arXiv:2406.05233*, 2024.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=qw3MZb1Juo`.

N Lee, T Ajanthan, and P Torr. Snip: single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*. Open Review, 2019.

Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 68–79. IEEE, 2021a.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 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.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pp. 6357–6368. PMLR, 2021b.

Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021c.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Constantin Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. In *ICLR*, 2023. URL `https://openreview.net/pdf?id=bXNl-myZkJl`.

Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated learning: privacy and incentive*, pp. 240–254. Springer, 2020.

Rongwei Lu, Yutong Jiang, Yinan Mao, Chen Tang, Bin Chen, Laizhong Cui, and Zhi Wang. Data-aware gradient compression for fl in communication-constrained mobile computing. *IEEE Transactions on Mobile Computing*, 2024.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

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*, pp. 1273–1282. PMLR, 2017.

H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2(2), 2016.

Zeping Min and Xinshang Wang. DOCS: Quantifying weight similarity for deeper insights into large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=XBHoaHlGQM`.

Aritra Mitra, Rayana Jaafar, George J Pappas, and Hamed Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34: 14606–14619, 2021.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37, 2022.

Aleksandra Nowak, Bram Grooten, Decebal Constantin Mocanu, and Jacek Tabor. Fantastic weights and how to find them: Where to prune in dynamic sparse training. *Advances in Neural Information Processing Systems*, 36, 2024.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Adam Sadilek, Luyang Liu, Dung Nguyen, Methun Kamruzzaman, Stylianos Serghiou, Benjamin Rader, Alex Ingerman, Stefan Mellem, Peter Kairouz, Elaine O Nsoesie, et al. Privacy-first health research with federated learning. *NPJ digital medicine*, 4(1):132, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR*, 2015.

Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34: 11220–11232, 2021.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Zhou, Ron Arel, and Aviv Shamsian. Fedselect: Personalized federated learning with customized selection of parameters for fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23985–23994, 2024.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

Praveen Venkateswaran, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. Fedgen: Generalizable federated learning for sequential data. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*, pp. 308–318. IEEE, 2023.

Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Dingzhu Wen, Ki-Jun Jeon, and Kaibin Huang. Federated dropout—a simple approach for enabling federated learning on resource constrained devices. *IEEE wireless communications letters*, 11(5):923–927, 2022.

Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2): 513–535, 2023.

Boqian Wu, Qiao Xiao, Shunxin Wang, Nicola Strisciuglio, Mykola Pechenizkiy, Maurice van Keulen, Decebal Constantin Mocanu, and Elena Mocanu. Dynamic sparse training versus dense training: The unexpected winner in image corruption robustness. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=daUQ7vmGap`.

Xinghao Wu, Xuefeng Liu, Jianwei Niu, Guogang Zhu, and Shaojie Tang. Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19375–19384, 2023.

Qiao Xiao, Boqian Wu, Yu Zhang, Shiwei Liu, Mykola Pechenizkiy, Elena Mocanu, and Decebal Constantin Mocanu. Dynamic sparse network for time series classification: Learning what to "see". In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=ZxOO5jfqSYw`.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Ogga20D2HO-`.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Honglin Yuan, Warren Richard Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=VimqQq-i_Q`.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11237–11244, 2023.

Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34: 10092–10104, 2021.

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

Appendix

Table of Contents

8 Implementation Details

In our methods, we adopt the SGD optimizer with a learning rate of 0.1. We set the number of local epochs $E = 5$, and batch size equals 100, consistent with the setup in (Wu et al., 2023). The maximum communication rounds T is set to 300 to ensure full convergence. The same settings are applied to other baseline methods for fair comparison. In each run, we evaluate the uniform averaging test accuracy across all clients in each communication round and select the final accuracy as the final result. Each experiment is repeated three times under different seeds, and the mean and standard deviation are reported.

9 Sensitivity-based Criterion Derivation

In this section, we derive the sensitivity criterion used in LIPS for evaluating the importance of each parameter during federated training.

Definition of sensitivity. Given a model $\mathbf{w}_i^t$ at client i during communication round t, the parameter set is expressed as:

$$\Theta = \{w_{i,1}^t, w_{i,2}^t, \ldots, w_{i,j}^t, \ldots, w_{i,n}^t\}.$$

The sensitivity of the j-th parameter $w_{i,j}^t$ is defined as the absolute difference in the loss function $\mathcal{L}$ when the parameter $w_{i,j}^t$ is zeroed out:

$$s_{i,j}^t = \left|\mathcal{L}(\Theta) - \mathcal{L}(w_{i,1}^t, \ldots, w_{i,j-1}^t, 0, w_{i,j+1}^t, \ldots, w_{i,n}^t)\right|, \tag{7}$$

where $\mathcal{L}$ is the task-specific loss function. This sensitivity metric quantify the importance of the parameter $w_{i,j}^t$ to the local task, with higher sensitivity values indicating parameters that are more critical to the model's performance.

Taylor approximation. Directly computing $s_{i,j}^t$ for each parameter requires additional forward passes, significantly increasing computational costs. To overcome this, we use a first-order Taylor approximation to approximate the sensitivity:

$$s_{i,j}^t = \left|\mathcal{L}(\Theta) - \mathcal{L}(w_{i,1}^t, \ldots, 0, \ldots, w_{i,n}^t)\right| \approx \left|\nabla_{w_{i,j}}\mathcal{L}(\Theta) \cdot w_{i,j}^t + R_1(\Theta)\right|, \tag{8}$$

where $R_1(\Theta)$ represents higher-order terms that are neglected due to the linear approximation. Thus, the sensitivity can be approximated as:

$$s_{i,j}^t \approx \left|\nabla_{w_{i,j}}\mathcal{L}(\Theta) \cdot w_{i,j}^t\right|. \tag{9}$$

This approximation requires only a single backpropagation pass to compute the gradients, significantly reducing computational overhead.

Incorporating local updates in federated learning. In federated learning, each client performs multiple local training epochs before sending updates to the server. To incorporate this, we replace $\nabla_{w_{i,j}}\mathcal{L}$ with the variation in the parameter $w_{i,j}^t$ over local training epochs. Specifically:

$$\Delta w_{i,j}^t = w_{i,j}^t - w_{i,j}^{t'}, \tag{10}$$

where $w_{i,j}^{t'}$ and $w_{i,j}^t$ denote the parameter values before and after the local training update, respectively. Substituting this into the sensitivity definition, we obtain:

$$s_{i,j}^t = \left|\Delta w_{i,j}^t \cdot w_{i,j}^t\right|. \tag{11}$$

Sensitivity-based metric The final sensitivity metric for parameter $w_{i,j}^t$ at client i during round t is given by:

$$s_{i,j}^t = \left|\Delta w_{i,j}^t \cdot w_{i,j}^t\right|, \tag{12}$$

where $\Delta w_{i,j}^t = w_{i,j}^t - w_{i,j}^{t'}$.

This metric captures both the magnitude of local updates ($\Delta w_{i,j}^t$) and the parameter's final state ($w_{i,j}^t$), making it well-suited for identifying less critical parameters that can be zeroed out to enforce sparsity while preserving model performance.

10 Additional Layer-wise Cosine Similarity Visualization

In Figure 6, we present additional results on the layer-wise cosine similarity of the global model after aggregation throughout training under different federated learning settings for CIFAR-10 and CIFAR-100. Consistently, we observe that most layers, particularly the middle layers, exhibit minimal changes during the training process after the early stages of training. This is evidenced by the consistently high cosine similarity values (above 0.95) for the middle layers' weights when compared to their states in the early training phase. These results confirm that this behavior is consistently observed across different datasets and architectures.

Additionally, we provide layer-wise cosine similarity results for the global model throughout training on CIFAR-10 and CIFAR-100 under a scenario with 300 training samples per client. As highlighted in our main claim, fewer training samples exacerbate the layer-wise inertia phenomenon. From Figure 7 (a) and (b), we observe that with more training samples per client, the layer-wise inertia phenomenon is notably weakened compared to the scenario with 100 training samples per client, as shown in Figure 6 (a) and (c), respectively. These observations further support our claim in Section 3.

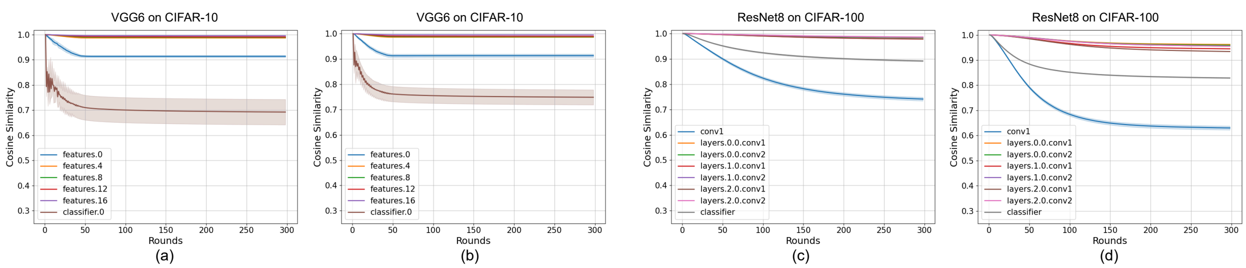

Figure 6: Layer-wise cosine similarity during the training process. We track the global model's layer-wise cosine similarity relative to the 2nd communication round states after aggregation throughout training on CIFAR-10 and CIFAR-100 datasets using VGG6 and ResNet-8 architectures, respectively. The experiments are conducted with 100 clients, each having 100 training samples. For CIFAR-10, results are shown for (a) Dir(α=0.5) and (b) Dir(α=1.0), while for CIFAR-100, results are shown for (c) Dir(α=0.01) and (d) Dir(α=0.5). The legend indicates the layer names for each architecture.

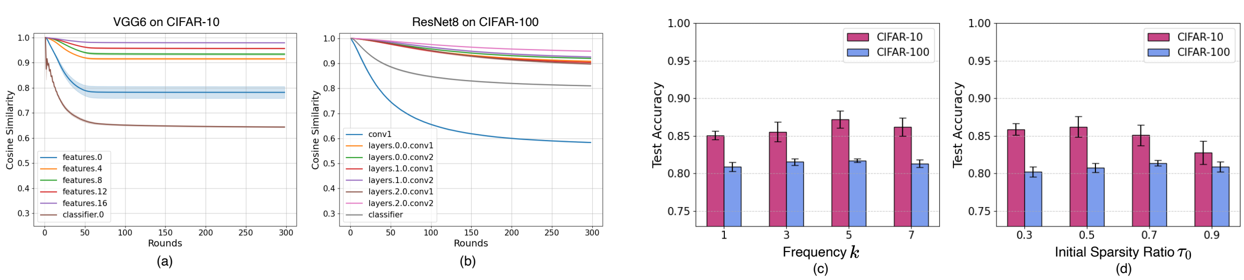

Figure 7: Layer-wise cosine similarity during the training process. (a) and (b) show the global model's layer-wise cosine similarity relative to the 2nd communication round for CIFAR-10 and CIFAR-100 datasets with 300 training samples per client, under Dir(α=0.5) and Dir(α=0.01), respectively. (c) illustrates the effect of different sparsification intervals k, showing how the frequency of introducing sparsity impacts performance. (d) highlights the impact of varying initial sparsity ratios τ_0 on the performance of CIFAR-10 (Dir(α=0.1) and CIFAR-10 (Dir(α=0.01)).

11 Layer-wise Learning Behavior in Low-Data Centralized Training

To further understand the origins of the Layer-wise Inertia Phenomenon, we conduct layer-wise learning behavior in low-data centralized training settings, as shown in Figure 8. We can observe that inertia is also evident in centralized training when data is scarce, suggesting that this phenomenon primarily arises from overfitting due to insufficient training data, which results in minimal parameter updates in certain layers.

Interestingly, we observe that FL offers partial mitigation. Specifically, the first and last layers, typically more sensitive to input variation and task-specific supervision, show lower cosine similarity across communication rounds in FL compared to centralized training, indicating more effective adaptation, compared to Figure 1. However, for intermediate layers, the cosine similarity remains consistently high in both settings. While FL provides slight improvements, it still struggles to significantly alter the stagnant behavior of these layers, highlighting a shared challenge across both training paradigms. These findings underscore the need for targeted solutions, such as our proposed sparsity-based strategy, to revitalize underutilized layers in low-data regimes.

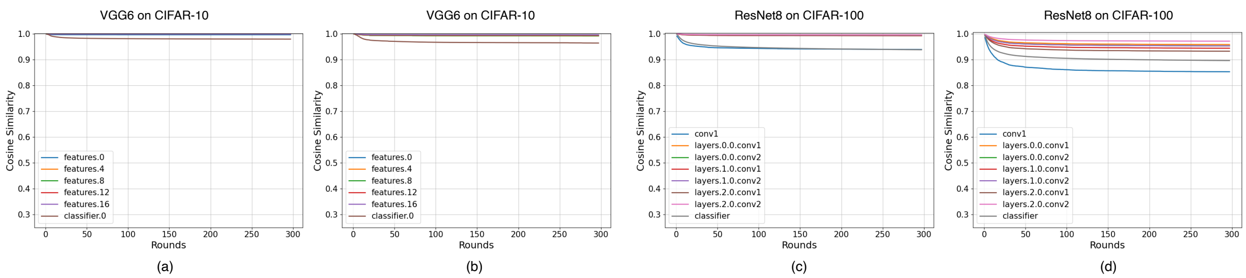

Figure 8: Layer-wise cosine similarity of models under centralized training throughout the training process. We track the cosine similarity of each layer in the global model relative to its state at the 2nd communication round, using CIFAR-10 and CIFAR-100 datasets with VGG6 and ResNet-8 architectures, respectively. Experiments are conducted with 100 clients under Dir(α=0.1), where each client holds 100 training samples for subfigures (a) and (c), and 300 samples for (b) and (d). The legend indicates the corresponding layer names for each architecture.

12 Additional Visualization of the Impact of LIPS on Layer Dynamics

To further validate the effectiveness of LIPS in addressing the Layer-wise Inertia Phenomenon, we provide additional visualizations across different datasets and model architectures beyond those presented in the main text.

In Figure 9, we extend our analysis of layer-wise cosine similarity and mean gradient norm to CIFAR-10 and TinyImageNet datasets using the VGG6 and ResNet-10 architectures, respectively. Specifically, subfigures (a) and (b) show the global model's layer-wise cosine similarity relative to its 2nd round checkpoint under Dirichlet data distributions with $\alpha = 0.1$ and $\alpha = 0.01$ for CIFAR-10 and TinyImageNet datasets, respectively. These results confirm that LIPS reduces cosine similarity in intermediate layers over the course of training, indicating more dynamic updates and better mitigation of stagnation.

Subfigures (c) and (d) further compare the mean gradient norm per layer between FedBN and LIPS across training rounds for CIFAR-10 and TinyImageNet datasets. LIPS consistently increases the gradient magnitudes, particularly in the middle layers, thereby promoting more effective local updates. This sustained update activity contributes to better and more meaningful collaboration through aggregation in FL environments.

These visualizations reinforce the generality of LIPS across diverse federated learning scenarios and demonstrate its robustness in enhancing learning dynamics across multiple datasets and model backbones.

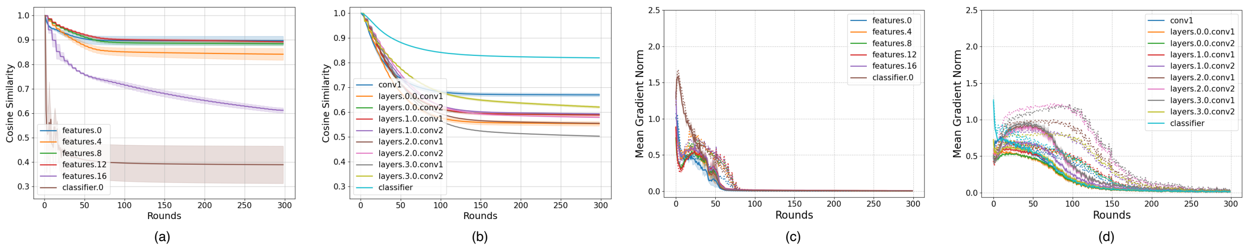

Figure 9: Layer-wise cosine similarity and gradient norm analysis during the training process. (a) and (b) show the global model's layer-wise cosine similarity relative to the 2nd communication round for CIFAR-10 and TinyImageNet datasets, with Dir(α=0.1) and Dir(α=0.01), respectively. (c) and (d) show comparison of layer-wise mean gradient norms across all clients between FedBN (solid line) and LIPS (dotted line) after each training round for CIFAR-10 and TinyImageNet datasets, with Dir(α=0.1) and Dir(α=0.01), respectively.

13 Ablation Study

In this section, we perform a comprehensive ablation study to investigate the impact of two key hyperparameters in our method: the sparsification frequency k and the initial sparsity ratio τ_0. Below, we detail the effect of these parameters and highlight the configurations that yield optimal performance in our approach.

13.1 Effect of frequency k

In Figure 7 (c), we evaluate the performance of our method under varying sparsification intervals k, which determine how frequently sparsity is introduced during communication rounds. The highest accuracies are achieved when sparsification is performed every 5 rounds in non-IID settings with Dir(α=0.1) for CIFAR-10 and Dir(α=0.01) for CIFAR-100, a setting adopted as the default in our main experiments. The findings suggest that overly frequent sparsification can disrupt training convergence by introducing excessive instability in the model's sparse topology. On the other hand, infrequent sparsification reduces its effectiveness in addressing the Layer-wise Inertia phenomenon, as the model fails to sufficiently adapt its sparse topology during training. Striking the right balance in sparsification frequency is therefore essential for optimizing performance in our settings.

13.2 Effect of initial sparsity ratio τ_0

To assess the impact of LIPS under different initial sparsity ratios, which determine the proportion of weights pruned from specific layers in each round, we conducted experiments illustrated in Figure 7 (d). The results reveal that the optimal sparsity ratio varies across datasets, as tasks with differing complexities and models of varying scales exhibit distinct levels of weight redundancy. In our experiments, an initial sparsity ratio of 0.5 was most effective for CIFAR-10 with the VGG6 architecture. For CIFAR-100 with ResNet models, a higher initial sparsity ratio of 0.7 yielded better results. This can be attributed to the deeper architecture and residual connections of ResNet, which might provide greater representational power and allow the model to maintain performance even under higher sparsity levels.

14 Performance Comparison with Smaller Models

As analyzed in Section 3, having more layers exacerbates the Layer-wise Inertia phenomenon. Conversely, fewer layers help mitigate this issue. This raises an intuitive question: *why not simply use smaller models in such cases?*

To address this concern, we compare the final performance of smaller models with models having more layers in Table 6. Specifically, we evaluate ResNet6, ResNet8, and ResNet10 on CIFAR-100 with 100 training samples per client under a Dirichlet data distribution Dir(α=0.1).

The results show that while ResNet6 reduces the Layer-wise Inertia phenomenon, it performs worse overall compared to ResNet8 and ResNet10. The primary reason is that smaller models lack the learning capacity required to handle the complexity of the task, resulting in inferior performance. Therefore, addressing the Layer-wise Inertia phenomenon is crucial to unlocking the full potential of larger models, enabling them to perform optimally in federated learning scenarios.

Table 6: Performance comparison of FedBN and LIPS on CIFAR-100 using various ResNet-based architectures: ResNet6, ResNet8, and ResNet10, under data distribution Dir(α=0.1) with 100 clients, each having 100 training samples.

Method	ResNet6	ResNet8	ResNet10
FedBN	40.44±0.41	43.08±0.45	42.84±0.19
LIPS	44.56±0.38	47.84±0.47	47.54±0.32

15 Performance Comparison with Local Sparse Training

In this section, we compare the performance of LIPS with and without maintaining sparsity throughout local training, as shown in Table 7. The `w/. local sparsity` setting enforces sparsity during both the global aggregation phase and local updates, while the `w/o. local sparsity` configuration uses transient sparsity—applying sparsity only after aggregation and lifting it before local training.

The results show that maintaining sparsity throughout training (`w/. local sparsity`) leads to slightly lower accuracy compared to the transient sparsity variant (`w/o. local sparsity`); however, the performance difference is modest and both settings still outperform FedBN, which does not incorporate sparsity at all. For example, on CIFAR-100 with $\alpha = 0.5$, accuracy drops from 28.31% to 26.64%. However, this comes with a substantial benefit: applying sparsity only at aggregation (transient sparsity) reduces training computation by nearly 2×, since dense updates are used locally. This highlights a practical trade-off between training efficiency and performance. While full sparse training can further reduce computation cost, the transient sparsity mechanism used in LIPS retains dense local training, which preserves model quality more effectively.

Table 7: Performance comparison on CIFAR-100, and TinyImageNet using different architectures (ResNet-8, and ResNet-10, respectively), with 100 clients under varying values of α. We report both accuracy(%) (↑) and training FLOPs(10^{12}) (↓) under initial sparsity ratio $\tau_0 = 0.7$.

Method	CIFAR-100			TinyImageNet		
	$\alpha = 0.1$	$\alpha = 0.5$	FLOPs	$\alpha = 0.1$	$\alpha = 0.5$	FLOPs
FedBN	43.08±0.45	24.45±0.61	14.9	36.83±0.22	20.73±1.23	219.4
w/o. local sparsity	47.96±0.54	28.31±0.54	1.0×	40.50±0.32	23.50±0.53	1.0×
w/. local sparsity	46.26±0.46	26.64±0.62	0.6×	38.92±0.45	22.18±0.43	0.6×

16 Discussion

16.1 Dropout and Other Regularization Techniques

While our method shares the high-level objective of mitigating overfitting with techniques like dropout, it differs significantly in motivation, mechanism, and effectiveness in federated learning (FL). Dropout typically deactivates neurons randomly during training to prevent co-adaptation, resetting at every iteration. In contrast, our approach introduces structured sparsity by selectively pruning weights based on sensitivity, maintaining these sparsity patterns throughout training.

Importantly, our method is grounded in the observed Layer-wise Inertia phenomenon, targeting layers that suffer from stagnation due to overfitting—something dropout does not explicitly address. Additionally, our pruning mechanism is more flexible: it can be combined with various initialization strategies (such as original weight reinitialization), offering further performance benefits, as shown in our ablation studies 5.2.

While dropout has been applied in FL (e.g., for personalization or regularization purposes (Wen et al., 2022; Jeon et al., 2023)), it does not explicitly aim to tackle the challenge of persistent parameter inactivity in intermediate layers of global models. In contrast, our approach is driven by a novel empirical observation—the Layer-wise Inertia Phenomenon—which reveals the stagnation of certain layers during training in low-data regimes. By directly targeting this issue, our method moves beyond generic regularization by introducing sensitivity-guided sparsity that selectively reactivates and stimulates underutilized layers. This design makes our approach not only more principled and interpretable, but also particularly effective in enhancing aggregation and collaboration in federated learning, especially when training data is limited.

16.2 Existing Sparsity Techniques in FL

Sparsity has primarily been used in FL to reduce communication overhead, such as through gradient sparsity (Lu et al., 2024; Wangni et al., 2018) or model pruning (Jiang et al., 2022; Jiang & Borcea, 2023). These techniques target communication or storage efficiency, whereas LIPS is designed to address a fundamentally different challenge: improving learning dynamics and collaboration in low-data regimes. Recent works have considered dynamic sparse training in centralized settings (Mocanu et al., 2018; Evci et al., 2020), and some efforts extend such ideas to FL (Bibikar et al., 2022; Jiang et al., 2022; Chen et al., 2023). However, they often focus on training efficiency or latency rather than tackling representation stagnation due to limited updates in global aggregation. Our method complements these efforts by offering a simple yet effective way to reinvigorate stagnant layers without additional communication overhead.

17 Limitations and Future Work

While LIPS targets the middle layers to address inertia, extending the sparsification to other layers, including the first and last layers, could be beneficial under certain conditions. This would require exploring layer-specific sensitivities to sparsity and determining their impact on both model performance and stability during global aggregation.

The simplicity of LIPS also makes it a strong candidate for integration into existing FL frameworks with minimal changes. Future enhancements could involve combining LIPS with more sophisticated aggregation techniques, such as weighted averaging based on client contributions, or integrating personalization strategies that allow clients to maintain task-specific models while benefiting from global knowledge. These combinations have the potential to further improve collaboration and enhance performance in heterogeneous federated environments.

Lastly, the current evaluation of LIPS has been limited to specific datasets and architectures. Expanding the scope to include more diverse datasets in other domains, larger-scale models, and real-world federated learning applications would provide a broader understanding of its effectiveness and generalizability. These future directions highlight the potential for LIPS to evolve into a more comprehensive and scalable solution for efficient and effective federated learning.

18 Impact Statements

This paper contributes to the understanding and optimization of federated learning by analyzing layer-wise learning dynamics. Our findings provide valuable insights into federated optimization, potentially improving real-world applications where communication efficiency and model adaptability are essential. Although our contributions do not inherently lead to negative societal impacts, we encourage the community to remain mindful of potential implications when extending our research.