Revisiting Weighted Aggregation in Federated Learning with Neural Networks

Zexi Li [1] **Tao Lin** [2] **Xinyi Shang** [3] **Chao Wu** [1]

Abstract

In federated learning (FL), weighted aggregation of local models is conducted to generate a global model, and the aggregation weights are normalized (the sum of weights is 1) and proportional to the local data sizes. In this paper, we revisit the weighted aggregation process and gain new insights into the training dynamics of FL. First, we find that the sum of weights can be smaller than 1, causing *global weight shrinking* effect (analogous to weight decay) and improving generalization. We explore how the optimal shrinking factor is affected by clients' data heterogeneity and local epochs. Second, we dive into the relative aggregation weights among clients to depict the clients' importance. We develop *client coherence* to study the learning dynamics and find a critical point that exists. Before entering the critical point, more coherent clients play more essential roles in generalization. Based on the above insights, we propose an effective method for **Fed**erated Learning with **L**earnable **A**ggregation **W**eights, named as **FEDLAW** (source code). Extensive experiments verify that our method can improve the generalization of the global model by a large margin on different datasets and models.

1. Introduction

Federated learning (FL) (McMahan et al., 2017; Li et al., 2020a; Wang et al., 2021; Lin et al., 2020; Li et al., 2022c) is a promising distributed optimization paradigm where clients' data are kept local, and a central server aggregates clients' local gradients for collaborative training. In FL, weighted aggregation of local models is conducted to generate a global model. In FL, when aggregating local models, it is a common practice that the aggregation weights should be normalized (the sum of weights, i.e. the l_1 norm, notated as γ, is equal to 1) and proportional to the local data sizes. However, due to the non-convexity (Allen-Zhu et al., 2019; Li et al., 2018), over-parameterization (Allen-Zhu et al., 2019; Zou & Gu, 2019), scale invariance (Li et al., 2018; Dinh et al., 2017; Kwon et al., 2021), and other unique properties of deep neural networks (DNNs), there is a gap between theory and empirical practice when the models are DNNs. An intuitive example is shown in Figure 1, we find that smaller γ may be beneficial to generalization, which challenges the previous convention in theory that aggregation weights should be normalized as 1. But what is the mechanism behind and what is the optimal γ under different FL environments? It requires further investigation.

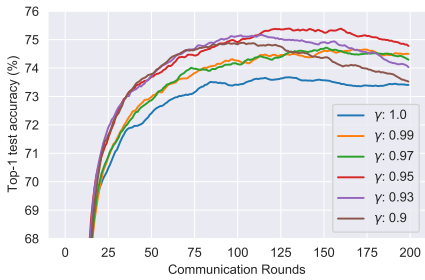

Figure 1. Test accuracy curves with different l_1 norms of aggregation weights (γ). CIFAR-10 with 20 clients, AlexNet.

Thus, in this paper, we revisit and rethink the weighted aggregation process to understand the training dynamics of FL and gain some intriguing insights.

How can the aggregation weights be assigned to generate a global DNN model with better generalization?

Towards this question, we find two aspects that matter most:
(1) the l_1 norm of aggregation weights (γ);
(2) the relative weights within the sampled clients ($\boldsymbol{\lambda}$).
To gain insights, we leverage the advantage of the server in FL that we learn the aggregation weights on a global-objective-consistent proxy dataset by gradient descent. The learned weights are the optimal weight candidates at each round and can reflect the training dynamics.
For (1), we identify the *global weight shrinking* effect in FL when γ is smaller than 1, which is analogous to weight decay regularization (Loshchilov & Hutter, 2018; Lewkowycz & Gur-Ari, 2020; Xie et al., 2020) in centralized training. However, a small value of γ—as stated in Figure 1—will

[1]Zhejiang University, China. Zexi Li <zexi.li@zju.edu.cn>. [2]Research Center for Industries of the Future, Westlake University, China. [3]Xiamen University, China. Xinyi Shang <shangxinyi@stu.xmu.edu.cn>. Work was done during Xinyi's visit to Westlake University. Correspondence to: Chao Wu <chao.wu@zju.edu.cn>, Tao Lin <lintao@westlake.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

cause negative effects; therefore, there exists an optimal γ that balances the regularization and optimization. We fix $\boldsymbol{\lambda}$ and learn γ (cf. section 4) on the proxy dataset to explore how the optimal shrinking factor (i.e. the l_1 norm of γ) is affected by clients' heterogeneity and local epochs.

For (2), we study how $\boldsymbol{\lambda}$ should be assigned to the local models to obtain a more generalized global model and how $\boldsymbol{\lambda}$ can reflect clients' importance in training dynamics. We fix γ and learn $\boldsymbol{\lambda}$ (cf. section 5) on the proxy dataset to study *client coherence* that includes: (i) *local gradient coherence*, the importance of clients in different learning periods; (ii) *heterogeneity coherence*, the consistency between the sum objective of sampled clients and the global objective.

Based on the insights, we propose an effective method for **Fed**erated Learning with **L**earnable **A**ggregation **W**eights, named as **FEDLAW** (cf. section 6). Extensive experiments verify that our method can improve the generalization of the global model by a large margin on different datasets and models. Moreover, it is validated that FEDLAW is still robust when the proxy dataset is small or shifted from the global distribution and corrupted clients exist.

Specifically, our contributions are two-folded.

- As our main contribution, we revisit and rethink the weighted aggregation in FL with DNNs and identify some interesting findings (see below take-aways). Especially, we find that smaller l_1 norms of aggregation weights may be beneficial to generalization, which challenges the previous normalized convention. This is also the first paper that introduces global regularization in FL, and we explore how to adaptively control such regularization.
- We showcase the applicability of these insights, and devise a simple yet effective method FEDLAW, which largely boosts the generalization of global models. The effectiveness and robustness of FEDLAW are validated by extensive experiments.

We summarize our key take-away messages of the understandings as follows.

- *Global weight shrinking* regularization effectively improves the generalization performance.
  - The magnitude of the global gradient (i.e. uniform average of local updates) determines the optimal weight shrinking factor. A larger norm of the global gradient requires stronger regularization, in the cases when (i) the number of local epochs is larger; (ii) the clients' data are more IID; (iii) during training before the global model is near convergence.
  - The effectiveness of global weight shrinking is stemmed from flatter loss landscapes of the global model as well as the improved local gradient coherence after the critical point.[1]
- Our novel concept of *client coherence* depicts the training dynamics of FL, from the aspects of *local gradient coherence* and *heterogeneity coherence*.
  - Local gradient coherence refers to the averaged cosine similarities of clients' local gradients. A critical point (from positive to negative) exists in the curves of local gradient coherence during the training. Generalization can benefit when the local gradient coherence is positive and more dominant.
  - Heterogeneity coherence refers to the distribution consistency between the global data and the sampled one (i.e. data distribution of a cohort of sampled clients) in each round. Increasing the heterogeneity coherence by reweighting the sampled clients could also improve the training performance.

[1]Different from the latter observations (w/o affecting the training dynamics), applying global weight shrinking results in a positive local gradient coherence after the critical point and the learning can benefit from it.

2. Related Works

Model aggregation in FL. There are previous works that try to learn the aggregation weights on given datasets by gradient descent. AUTO-FEDAVG (Xia et al., 2021) learns aggregation weights on different institutional medical data to realize personalized medicine, while L2C matches similar peers in decentralized FL (Li et al., 2022a) by learning aggregation weights on local datasets. These works all adopt the normalized aggregation weights (γ=1) without discovering the global weight shrinking effect, and they focus on *personalization* while we focus on *generalization*. Besides, they fail to understand the FL's dynamics from the learned weights for further insights, e.g., identifying the significance of client coherence. Ensemble distillation methods are used to improve the generalization of global models after weighted aggregation. FEDDF (Lin et al., 2020) uses the local models as teachers and finetune the global model via ensemble distillation; while in FEDBE (Chen & Chao, 2021), Bayesian ensemble distillation is further introduced. Since they also require a proxy dataset on the server, we will compare them with our proposed FEDLAW in section 6. Additionally, server-side stochastic weight averaging and client-side sharpness-aware minimization are incorporated to make the global model converge to a flatter minimum (Caldarola et al., 2022); distributionally robust optimization is also introduced to realize more robust federated averaging (Deng et al., 2020; Wu et al., 2022); but these works are orthogonal to our paper.

Training dynamics of DNNs in centralized learning. Our insights into global weight shrinking and client coherence in FL are analogous to weight decay and gradient coherence in centralized learning. *Weight decay:* The optimal weight decay factor is approximately inverse to the number of epochs, and the importance of applying weight decay diminishes when the training epochs are relatively long (Loshchilov & Hutter, 2018; Lewkowycz & Gur-Ari, 2020; Xie et al., 2020). The effectiveness of weight decay may be explained by the caused (i) larger effective learning rate (Zhang et al.,

2018; Wan et al., 2021), and (ii) flatter loss landscape (Lyu et al., 2022). *Gradient coherence:* Gradient coherence, or sample coherence, is a crucial technique for understanding the training dynamics of mini-batch SGD in centralized learning (Chatterjee, 2019; Zielinski et al., 2020; Chatterjee & Zielinski, 2020; Fort et al., 2019). The gradient coherence measures the pair-wise gradient similarity among samples. If they are highly similar, the overall gradient within a mini-batch will be stronger in certain directions, resulting in a dominantly faster loss reduction and better generalization (Chatterjee, 2019; Zielinski et al., 2020; Chatterjee & Zielinski, 2020). The critical period exists in mini-batch SGD, captured by the gradient coherence: the low coherence in the early training phase damages the final generalization performance, no matter the value of coherence controlled later (Chatterjee & Zielinski, 2020). In section 4 and section 5, we will show similar findings can be drawn in FL's dynamics, and some new insights are discovered.

Due to space limits, the detailed discussions about related works can be found in Appendix A.

3. Preliminary and Problem Setup

FL usually involves a server and n clients to jointly learn a global model without data sharing, which is originally proposed in (McMahan et al., 2017). Denote the set of clients by $\mathcal{S}$, the local dataset of client i by $\mathcal{D}_i = \{(x_j, y_j)\}_{j=1}^{N_i}$, the sum of clients' data by $\mathcal{D} = \bigcup_{i \in \mathcal{S}} \mathcal{D}_i$. The IID data distributions of clients refer to each client's distribution $\mathcal{D}_i$ is IID sampled from $\mathcal{D}$. However, in practical FL scenarios, *heterogeneity* exists among clients that their data are *NonIID* with each other. In this paper, we use Dirichlet sampling, which is widely used in FL literature (Lin et al., 2020; Li et al., 2020b; Acar et al., 2020), to synthesize ***client heterogeneity*** *(controlled by α, the smaller, the more NonIID)*. During FL training, clients iteratively conduct local training and communicate with the server for model updating. In the local training, ***the number of local epochs*** *is* E; when E is larger, the communication is more efficient but the updates are more asynchronous. Since α and E are the key factors affecting FL's training, in this paper, we study how α and E affect the training dynamics of FL from the perspective of weighted aggregation.

Denote the global model and the client i's local model in communication round t by $\mathbf{w}_g^t$ and $\mathbf{w}_i^t$. In each round, clients' local models are initialized as the global model that $\mathbf{w}_i^t \leftarrow \mathbf{w}_g^t$, and clients conduct local training in parallel. In each local training epoch, clients conduct SGD update with a local learning rate η_l, and each SGD iteration shows as

$$\mathbf{w}_i^t \leftarrow \mathbf{w}_i^t - \eta_l \nabla \ell(B_k, \mathbf{w}_i^t), \text{ for } k = 1, 2, \cdots, K, \quad (1)$$

where ℓ is the loss function and B_k is the mini-batch sampled from $\mathcal{D}_i$ at the k-th iteration. After the client local updates, the server samples m clients for aggregation. The client i's pseudo gradient of local updates is denoted as $\mathbf{g}_i^t = \mathbf{w}_g^t - \mathbf{w}_i^t$. Then, the server conducts weighted aggregation to merge the local models (or the pseudo gradients) into a new global model[2].

$$\mathbf{w}_g^{t+1} = \sum_{i=1}^{m} \mu_i \mathbf{w}_i^t = \|\boldsymbol{\mu}\|_1 \mathbf{w}_g^t - \eta_g \sum_{i=1}^{m} \mu_i \mathbf{g}_i^t, \text{ s.t. } \mu_i \geq 0, \quad (2)$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_m]$ is the aggregation weights, $\eta_g = 1$ is the global learning rate. For vanilla FEDAVG, it adopts a normalized weights proportional to the data sizes, $\mu_i = \frac{|\mathcal{D}_i|}{|\mathcal{D}|}, \mathcal{D} = \bigcup_{i \in \mathcal{S}} \mathcal{D}_i$. In this paper, we assume the aggregation weights are not normalized which means the l_1 norm is not necessarily equal to 1. We study the effects of the l_1 norm and relative weights independently by decouple $\boldsymbol{\mu}$ into $\{\gamma, \boldsymbol{\lambda}\}$, which satisfies $\gamma = \|\boldsymbol{\mu}\|_1, \lambda_i = \frac{\mu_i}{\|\boldsymbol{\mu}\|_1}$. Thus, Equation 2 can be reformulated into

$$\mathbf{w}_g^{t+1} = \gamma \sum_{i=1}^{m} \lambda_i \mathbf{w}_i^t, \text{ s.t. } \gamma > 0, \lambda_i \geq 0, \|\boldsymbol{\lambda}\|_1 = 1. \quad (3)$$

Vanilla FEDAVG is a special case where $\gamma = 1, \lambda_i = \frac{|\mathcal{D}_i|}{|\mathcal{D}|}, \forall i \in [m]$. When $\gamma < 1$, it will cause weight shrinking of the global model, so in this case, we also call γ the shrinking factor.

Clarification on the proxy dataset. We study global weight shrinking[3] (γ) in section 4 and client coherence ($\boldsymbol{\lambda}$) in section 5 through respectively learning γ and $\boldsymbol{\lambda}$ while fixing another on a server proxy dataset. The considered proxy dataset has the same distribution as the global learning objective (i.e. a class-balanced case in this paper; e.g. 2000 balanced samples in CIFAR-10), thus the learned aggregation weights $\{\gamma, \boldsymbol{\lambda}\}$ can reflect the contributions of clients and the optimal regularization factor towards this global objective. We note that this case of the proxy dataset is for understanding only, and we will validate the effectiveness and robustness of our proposed FEDLAW on tiny (e.g. 100 samples in CIFAR-10) or biased (e.g. long-tailed) proxy datasets. For concision, in section 4 and section 5, if not mentioned otherwise, we all use CIFAR-10 as the dataset and SimpleCNN as the model. Experiments on more datasets and models are shown in section 6 and Appendix.

4. Global Weight Shrinking

4.1. Global Weight Shrinking and Its Impacts on Optimization

Setting $\gamma < 1$ results in the global weight shrinking regularization. Table 1 and Figure 1 report the results on CIFAR-10

[2]As in Equation 2, FL's aggregation can be formulated into the aggregation of clients' local models (left) or clients' pseudo gradients (right). The two kinds of the formulation are equal, while we adopt the aggregation of models here for brevity.

[3]We use the word "shrink" instead of "decay" as it shrinks the global model rather than decaying the model by subtracting a decay term (used in traditional weight decay). Similar "shrink" can be found in (Li et al., 2020c).

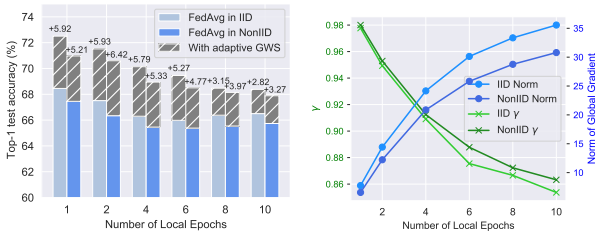

Figure 2. **Left:** Test accuracy gains of adaptive GWS. **Right:** The optimal γ and the norm of global gradient. $\alpha \in \{100, 1\}$.

Table 1. Impact of fixed γ across different architectures in both IID ($\alpha = 100$) and NonIID ($\alpha = 1$) settings ($E = 2$).

	γ	1.0	0.99	0.97	0.95	0.93	0.9
IID	SimpleCNN	65.53	67.60	69.20	69.52	**70.16**	69.83
	AlexNet	74.16	74.80	**75.54**	75.24	75.25	75.03
	ResNet8	75.51	76.64	76.80	**77.87**	76.80	76.74
NonIID	SimpleCNN	65.58	67.04	68.36	68.66	**69.28**	68.93
	AlexNet	73.56	73.83	74.37	**74.45**	74.40	74.24
	ResNet8	75.02	76.06	75.73	**77.00**	75.04	75.31

with different γ. It can be observed that the *global weight shrinking may improve generalization, depending on the choice of* γ. The smaller γ, the stronger regularization effect. Given a setting, there exists an optimal γ that balances the regularization and optimization, and deviation from this value, whether smaller or larger, may result in inferior performance. More results about the fixed γ can be found in Table 9 in Appendix.

4.2. Adaptive Global Weight Shrinking and Training Dynamics

We discover how to set an appropriate γ to balance regularization and optimization. We first expand the right of Equation 2 as follows.

$$\mathbf{w}_g^{t+1} = \gamma(\mathbf{w}_g^t - \eta_g \mathbf{g}_g^t) = \mathbf{w}_g^t - \gamma\eta_g \mathbf{g}_g^t - (1-\gamma)\mathbf{w}_g^t. \quad (4)$$

We refer $(1-\gamma)\mathbf{w}_g^t$ as the pseudo gradient of global weight shrinking (*regularization term*) and $\gamma\eta_g\mathbf{g}_g^t$ is the global averaged gradient (*optimization term*). We reckon that a larger optimization term requires a larger regularization term, which means the magnitude of the global pseudo gradient $\mathbf{g}_g^t$ determines the optimal shrinking factor γ in the way that larger $\mathbf{g}_g^t$, smaller γ (stronger regularization).

To verify our hypothesis, we achieve adaptive global weight shrinking (adaptive GWS) on the proxy dataset, which learns an optimal γ. Adaptive GWS adopts the update in Equation 3 and uses $\{\gamma = \gamma^*, \lambda_i = \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\}$ where

$$\gamma^* = \arg\min_{\gamma} \mathcal{L}_{proxy}(\gamma \cdot \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\mathbf{w}_i^t), \text{ s.t. } \gamma > 0. \quad (5)$$

Adaptive GWS largely improves the generalization. From the left of Figure 2, adaptive GWS can improve the performance of FEDAVG by a large margin in both IID and NonIID settings. Furthermore, adaptive GWS is more beneficial when the number of local epochs is small.

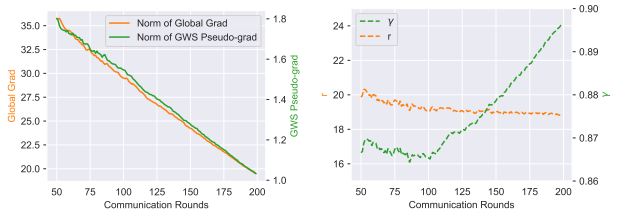

Figure 3. **Left:** Norm of two gradients in adaptive GWS. **Right:** The optimal γ and r in adaptive GWS, where r is the ratio of the global gradient and the regularization pseudo gradient.

1) Understanding the balance between optimization and regularization. Further, through the learned optimal γ, we verify the balance between optimization and regularization from the right of Figure 2 and Figure 3. A larger norm of the global gradient requires stronger regularization, in the cases when (i) the number of local epochs is larger; (ii) the clients' data are more IID; (iii) during training before the global model is near convergence (on the contrary, when the model is near convergence, smaller regularization is needed).

- As shown in the right blue Y-axis of right Figure 2, the norm of global gradient $\|\gamma\eta_g\mathbf{g}_g^t\|$ increases when the number of local epochs increases and data become IID. As a result, the optimal value of γ (shown in the left green Y-axis) becomes smaller in order to produce a larger weight shrinking pseudo gradient $\|(1-\gamma)\mathbf{w}_g^t\|$ to regularize the optimization. More results regarding how heterogeneity affects the optimal γ can be found in Figure 9 in Appendix.
- In Figure 1, GWS with smaller fixed γ will cause performance degradation in the late training. This is due to the conflicts of decaying global pseudo gradient and non-decaying regularization pseudo gradient. In Figure 3, while the norm of the global gradient is decaying, adaptive GWS learns a rising optimal γ to keep the GWS pseudo gradient decay proportionally. As a result, the ratio of two gradient terms remains steady at around 19 to maintain the balance between optimization and regularization.

2) The mechanisms behind adaptive GWS. We provide an in-depth general understanding of how adaptive GWS works and why it can improve generalization.

- **General understanding.**
  - **Scale invariance.** Adaptive GWS learns a dynamic shrinking factor γ in each round to shrink the global model's parameter. The method is effective due to the scale invariance property of DNNs (Li et al., 2018; Dinh et al., 2017; Kwon et al., 2021), which states that the function of a DNN remains similar or the same even when a factor rescales the model weights due to the non-linearity of activation functions or the normalization layer in DNNs. We show an intuitive understanding of scale invariance on the left figure of Figure 4, where the final models are rescaled by γ, and the loss function of the adaptive GWS's final model remains similar while the FEDAVG's final model even

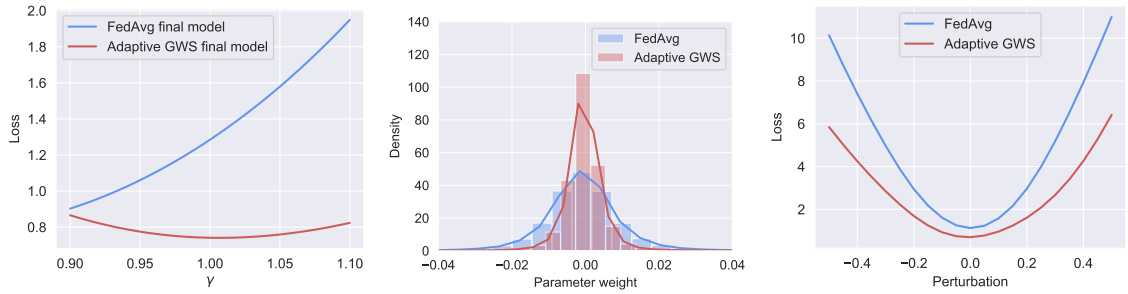

Figure 4. **General understanding of adaptive GWS. Left:** Scale invariance property of DNNs indicates that if the network is rescaled by γ, the function of the model remains similar. **Middle:** The histogram of final models' parameters shows that adaptive GWS makes more model parameters close to zero, nearly twice as many as FEDAVG. **Right:** The loss landscape is perturbed based on the Top-1 Hessian eigenvector of the final models, which shows that the model with adaptive GWS has flatter curvature and smaller loss.

has a smaller loss when $\gamma < 1$.

  - **Small model parameters.** The shrinking effect in each round can result in smaller model parameters of final global models, which is similar to weight decay. The parameter weight histogram is demonstrated in the middle figure of Figure 4. The final model of adaptive GWS has more model parameters close to zero, nearly twice as many as FEDAVG.

- **Why adaptive GWS can improve generalization.**
  - **Flatter loss landscapes.** One perspective of explaining the generalization of DNNs is through the flatness of the loss landscape. Previous works have shown that flatter curvature in loss landscape can indicate better generalization (Fort & Jastrzebski, 2019; Foret et al., 2020; Li et al., 2018). (Lyu et al., 2022) shows that weight decay of mini-batch SGD can result in flatter landscapes in DNNs with normalization layers. We also observe the similar phenomenon that *adaptive GWS improves generalization by seeking flatter minima in FL*, as shown in the right figure of Figure 4. Other metrics of flatness also demonstrate similar results (Figure 10 in Appendix).

3) The relation between adaptive GWS and local weight decay. Our proposed adaptive GWS can provide weight regularization from the global perspective, which is analogous to weight decay in mini-batch SGD. Importantly, GWS has a unique sparse regularization frequency that only changes the model weight in each round, resulting in stronger regularization. In GWS, $1-\gamma$ is near 0.1, whereas the factor of weight decay is often around 10^{-4}. Notably, the two methods are not conflicted in FL, and we conduct experiments on implementing weight decay in the local SGD solver and global weight shrinking on the server simultaneously. As shown in Table 2, *adaptive GWS is compatible with local weight decay and can further improve performance.* Unlike local weight decay, adaptive GWS is hyperparameter-free and effective. It can adaptively set γ to maximize the benefit of weight regularization. As the local weight decay becomes stronger, the learned γ is larger, resulting in weaker GWS regularization. More analysis about global weight shrinking can be found in subsection B.1 in Appendix.

Table 2. Adaptive GWS with different local weight decay factors ($E = 2$). IID ($\alpha = 100$), NonIID ($\alpha = 1$).

	Local weight decay	0	5e-5	1e-4	5e-4	1e-3
IID	FEDAVG	66.43	66.20	66.45	67.51	68.66
	Adaptive GWS	**71.47**	**71.36**	**71.35**	**71.44**	**71.54**
	γ of Adaptive GWS	0.9472	0.9477	0.948	0.9493	0.953
NonIID	FEDAVG	65.35	65.19	65.77	66.37	67.4
	Adaptive GWS	**70.31**	**69.93**	**70.44**	**70.47**	**69.99**
	γ of Adaptive GWS	0.9492	0.9499	0.9505	0.9529	0.9561

4) Insights from FL's adaptive GWS to mini-batch SGD. FL's adaptive GWS leverages the advantage of the server that learns an adaptive shrinking factor globally. It is promising that similar ideas can be adopted in mini-batch SGD by learning the hyperparameter of weight decay on a small proportion of training data. This may realize hyperparameter-free optimization, and we leave it for future works.

5. Client coherence

5.1. Basic Concept and Formulation

Inspired by gradient coherence in mini-batch SGD (Chatterjee, 2019; Zielinski et al., 2020; Chatterjee & Zielinski, 2020), we study *client coherence* in FL through weighted aggregation, which indicates how clients strengthen and complement each other to achieve better generalization. There are two aspects, the *local gradient coherence* of clients' model updates and the *heterogeneity coherence*.

Local Gradient Coherence. The gradient coherence in mini-batch SGD is at the data sample level. In FL, the concept of gradient coherence is extended to the level of clients, where we refer to it as "client coherence". Specifically, we study the similarity of local gradients among clients, as it has been shown that aggregating similar gradients leads to stronger global gradients, thereby improving generalization. We deduce the gradient coherence in mini-batch SGD and local gradient coherence in FL under a unified equation below:

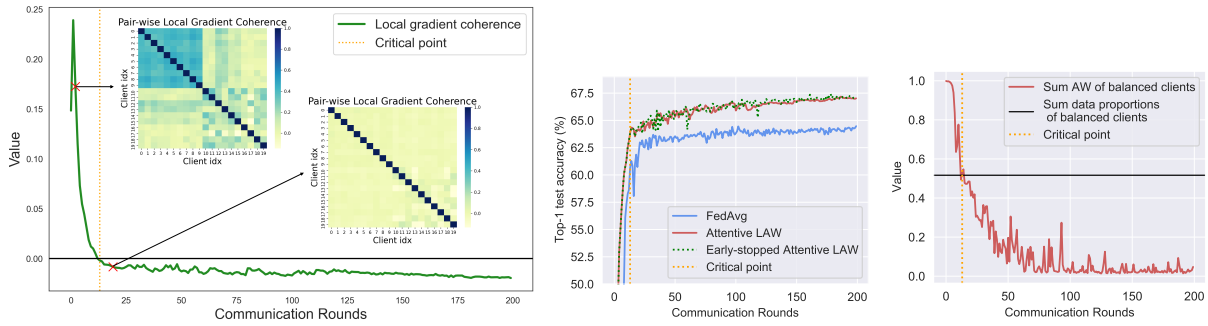

Figure 5. **Training dynamics of attentive LAW in terms of local gradient coherence.** Clients indexed 0-9 have balanced class distributions and 10-19 are imbalanced, $E = 3$. **Left:** Local gradient coherence. **Middle and Right:** The performance and aggregation weights of attentive LAW and early-stopped attentive LAW.

$$\begin{aligned}
\Delta\mathcal{L}^t &= \mathcal{L}(\mathbf{w}^t - \eta\mathbf{g}^t) - \mathcal{L}(\mathbf{w}^t) \approx -\eta \cdot \langle \mathbf{g}^t, \mathbf{g}^t \rangle \\
&= -\eta \cdot \langle \sum_{i=1}^{m} \lambda_i \mathbf{g}_i^t, \sum_{i=1}^{m} \lambda_i \mathbf{g}_i^t \rangle \\
&= -\eta \cdot (\sum_{i=1}^{m} \lambda_i^2 \|\mathbf{g}_i^t\|^2 + \sum_{i,j,i\neq j} \lambda_i \lambda_j \langle \mathbf{g}_i^t, \mathbf{g}_j^t \rangle) \\
&= -\eta \cdot (\sum_{i=1}^{m} \lambda_i^2 \|\mathbf{g}_i^t\|^2 + \sum_{i,j,i\neq j} \lambda_i \lambda_j \cos(\mathbf{g}_i^t, \mathbf{g}_j^t) \|\mathbf{g}_i^t\| \|\mathbf{g}_j^t\|) \,.
\end{aligned} \tag{6}$$

Equation 6 is a Taylor expansion of the loss function within one update. In mini-batch SGD, t is the iteration step, m is the batch size, and $\mathbf{g}_i^t$ is the gradient of a sample i at iteration t. Typically, there is no weighted averaging in a mini-batch, so $\forall i \in [m],\ \lambda_i = 1$. In FL, t is the communication round, $\mathbf{w}^t$ is the global model on the server at round t, m is the cohort size, $\mathbf{g}_i^t$ denotes the local gradient of client i at round t, and λ_i is the aggregation weight of client i. The term $\cos(\mathbf{g}_i^t, \mathbf{g}_j^t)$ means the cosine similarity between the gradients of clients i and j, defined as $\langle \mathbf{g}_i^t, \mathbf{g}_j^t \rangle / \|\mathbf{g}_i^t\| \|\mathbf{g}_j^t\|$. Assuming all gradients have bounded norms that $\forall i, \|\mathbf{g}_i^t\| \leq \epsilon$. The cosine similarity among gradients indicates the coherence: if the gradients have larger cosine similarity, it will have larger descent in the loss and improve the global generalization[4]. In this paper, we focus on the local gradient coherence among clients during FL training. We use the cosine stiffness definition (Fort et al., 2019) to quantify the local gradient coherence in FL.

Definition 5.1. The local gradient coherence of *two clients i and j* at round t is defined by the cosine similarity of their local updates sent to the server, as $c_{(i,j)}^t = \cos(\mathbf{g}_i^t, \mathbf{g}_j^t)$. The overall local gradient coherence of *a cohort of clients* at round t is defined by the weighted cosine similarity of all clients' local updates sent to the server, as $\boldsymbol{c}_{\boldsymbol{cohort}}^{\boldsymbol{t}} = \frac{1}{m} \sum_{i,j,i\neq j} \lambda_i \lambda_j \cos(\mathbf{g}_i^t, \mathbf{g}_j^t)$.

FL assumes multiple local epochs in each client, and clients usually have heterogeneous data. In this case, the local gradients of clients are usually almost orthogonal, which means that they have low coherence. This phenomenon is observed in (Charles et al., 2021), but it did not dig deeper to examine the training dynamics of FL. In this paper, we calculate the local gradient coherence in each round and find a *critical point* exists in the process (Figure 5 and Figure 6).

Heterogeneity Coherence. Heterogeneity coherence refers to the distribution consistency between the global data and the sampled one (i.e. data distribution of a cohort of sampled clients) in each round. The value of heterogeneity coherence is positively correlated with the IID-ness of clients as well as the client participation ratio; the higher, the better. We define heterogeneity coherence as follows.

Definition 5.2. Assuming there are n clients and the cohort size is m. For a given cohort of clients, the heterogeneity coherence is $\text{sim}(\mathcal{D}_{cohort}, \mathcal{D})$, where $\mathcal{D}_{cohort} = \sum_{i\in[m]} \lambda_i \mathcal{D}_i, \mathcal{D} = \sum_{j=1}^{n} \lambda_j \mathcal{D}_j$ and "sim" is the similarity of two data distributions.

5.2. Attentive Learnable Aggregation Weight and Training Dynamics

Vanilla FedAvg only considers data sizes as clients' aggregation weights $\boldsymbol{\lambda}$. However, clients with different heterogeneity degrees have different importance in client coherence, which can greatly affect the training dynamics. A three-node toy example is shown in Figure 12 in Appendix. The optimal $\boldsymbol{\lambda}$ is off the data-sized when clients have the same data size but different heterogeneity degrees. To study client coherence further, we propose attentive learnable aggregation weight (attentive LAW) to learn the optimal aggregation weights (i.e. $\boldsymbol{\lambda}$) on a proxy dataset. By connecting the optimal weights and the client coherence, we can know the roles of different clients in different learning periods. Attentive LAW conducts the model updates in Equation 3, where $\{\gamma = 1, \boldsymbol{\lambda} = \boldsymbol{\lambda}^*\}$,

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{L}_{proxy}(\sum_{i=1}^{m} \lambda_i \mathbf{w}_i^t), \text{ s.t. } \lambda_i \geq 0, \|\boldsymbol{\lambda}\|_1 = 1. \tag{7}$$

[4]The local gradient coherence is different from gradient diversity (Yin et al., 2018). A detailed discussion can be found in subsection B.2 in Appendix

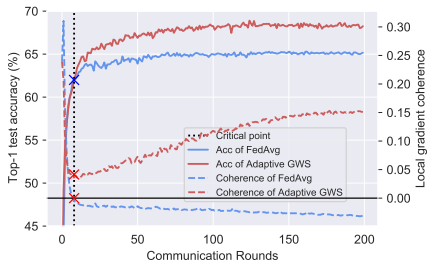

Figure 6. Training dynamics of adaptive GWS in terms of local gradient coherence. $E = 3$.

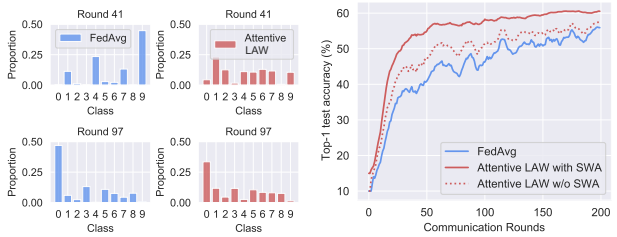

Figure 7. **Left:** Heterogeneity coherence of class distribution within a cohort. **Right:** Test accuracy curves. $E = 3$.

1) Generalization will benefit from positive local gradient coherence.

- **Critical point exists in terms of local gradient coherence.** To study the role of client data heterogeneity in local gradient coherence, we experiment on both balanced and imbalanced clients, whose distributions are shown in Figure 13 of Appendix. The results are demonstrated in Figure 5, which illustrate that *in the first couple of rounds, the coherence is dominant and positive, thus the test accuracy arises dramatically, and most generalization gains happen in this period*. The critical point is the round that the coherence is near zero. After the critical point, the test accuracy gain is marginal, and the coherence is kept negative but close to zero.
- **Assigning larger weights to clients with larger coherence before the critical point can improve overall performance.** From the left of Figure 5, it is clear that before the critical point, the coherence among balanced clients is much higher than that of imbalanced clients. This observation highlights the fact that clients with more balanced data have more coherent gradients [5]. To capitalize on this, according to Equation 6, we can assign larger weights to clients with more balanced data before the critical point to boost generalization. From the right of Figure 5, attentive LAW proves our hypothesis: it assigns larger weights to balanced clients in the early rounds, particularly in the first two rounds where it nearly assigns all weights to balanced clients. This may suggest that *the coherence of clients only matters before the critical point where the overall coherence is positive*. To verify this, we adopt early stopping near the critical point when conducting attentive LAW and use data-sized weights after the stopping round. Results in the middle of Figure 5 show that *the early-stopped attentive LAW has comparable performance after the critical point.* This insight can guide the design of effective algorithms for learning critically in early training stages.
- **GWS improves local gradient coherence to positive after the critical point.** Interestingly, we observe that *if we adopt adaptive GWS, the local gradient coherence remains positive after the critical point, allowing the model to continue benefiting from the coherent gradients.* As shown in Figure 6, before the critical point, both vanilla FEDAVG and adaptive GWS have high gradient coherence, resulting in similar increases in accuracy. After the critical point, the coherence of FEDAVG goes down below zero, resulting in marginal performance gains. In contrast, adaptive GWS maintains coherence above zero, allowing for further performance gains beyond FEDAVG.

2) Improving heterogeneity coherence within a cohort can boost performance. In scenarios with partial client participation in each round, the selected clients have an inconsistent sum objective with the global objective, resulting in low heterogeneity coherence (as defined in Definition 5.2). In theory, the expectation of the sum objective of the sampled clients is consistent with the global objective if the communication rounds (sampling times) are numerous. However, in practical FL, the number of rounds is limited, and from the perspective of learning dynamics, only the first few rounds matter most. Thus, the heterogeneity coherence problem brings a challenge, and sampling methods may not always help, because the availability of clients cannot be guaranteed and stragglers may exist (Li et al., 2020b).

To address the issue, reweighting the sampled clients in aggregation is quite essential. We find *attentive LAW improves heterogeneity coherence by dynamically adjusting the aggregation weights among clients*. We visualize the weighted class distributions within a cohort in Figure 7, which shows that attentive LAW learns weights to make the class distributions more balanced. The test accuracy curves demonstrate a dominant performance gain compared to FEDAVG, which showcases the significance of heterogeneity coherence. Additionally, we observe that attentive LAW with SWA[6] performs better by seeking a more generalized minimum in the aggregation weight hyperplane. More analysis about client coherence can be found in subsection B.2 of Appendix B.

[5] This also reveals why FL performs better in IID settings than NonIID: the clients' gradients in IID settings are more coherent, but the ones in the NonIID usually diverge.

[6] Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) is an effective technique to make simple averaging of multiple points along the trajectory of optimization with a cyclical learning rate, which leads to better generalization.

Table 3. **Top-1 test accuracy (%) achieved by comparing FL methods and FedLAW on three datasets with different model architectures** ($E = 3$). Blue/**bold** fonts highlight the best baseline/our approach.

Dataset	FashionMNIST				CIFAR-10				CIFAR-100			
NonIID (α)	100		0.1		100		0.1		100		0.1	
Model	MLP	LeNet	MLP	LeNet	CNN	ResNet	CNN	ResNet	CNN	ResNet	CNN	ResNet
FedAvg	89.29	90.54	85.11	88.08	65.78	74.57	60.13	46.04	25.74	27.49	27.74	24.92
FedProx	87.68	89.77	84.33	87.01	67.66	68.51	60.48	48.84	9.49	27.15	12.52	23.73
FedDyn	88.47	89.92	77.68	72.68	66.1	76.62	41.53	35.77	24.44	32.18	22.67	29.00
FedDF	86.16	89.09	78.48	85.90	69.60	77.36	57.38	54.09	28.52	27.42	24.52	23.10
FedBE	86.22	89.14	79.12	85.96	69.88	77.94	59.84	52.86	28.38	27.73	25.41	23.74
Server-FT	89.09	90.56	85.71	88.10	66.83	74.73	60.43	47.59	25.37	26.14	24.33	23.03
FedLAW	**88.51**	**90.66**	**86.30**	**88.26**	**70.17**	**80.46**	**62.46**	**52.83**	**32.51**	**33.17**	**32.30**	**24.84**
FedLAW (SWA)	**88.27**	**90.51**	**86.89**	**88.18**	**69.9**	**79.55**	**62.12**	**57.08**	**32.39**	**33.17**	**32.27**	**25.31**

Algorithm 1 FedLAW: Federated Learning with Learnable Aggregation Weights

Input: clients $\{1, \ldots, n\}$, server-side proxy dataset, communication round T, local epoch E, server epoch E_s, initial global model $\mathbf{w}_g^1$;
Output: final global model $\mathbf{w}_g^T$;

for each round $t = 1, \ldots, T$ **do**
  # Client updates
  for each client $i, i \in [n]$ **in parallel do**
    Set local model $\mathbf{w}_i^t \leftarrow \mathbf{w}_g^t$;
    Compute E epochs of client local training by Equation 1:
      $\mathbf{w}_i^t \leftarrow \mathbf{w}_i^t - \eta_l \nabla \mathcal{L}_i\left(\mathbf{w}_i^t\right)$;
  end for
  # Server updates
  The server samples m clients and receive their models $\{\mathbf{w}_i^t\}_{i=1}^m$;
  The server sets initial γ and $\boldsymbol{\lambda}$ as $\{\gamma = 1, \lambda_i = \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\}$;
  Compute E_s epochs of aggregation weight learning on the proxy dataset by Equation 8:
    $\{\gamma, \boldsymbol{\lambda}\} \leftarrow \{\gamma, \boldsymbol{\lambda}\} - \eta_s \nabla \mathcal{L}_{proxy}\left(\{\gamma, \boldsymbol{\lambda}\}\right)$;
  Obtain the optimal aggregation weights $\{\gamma^*, \boldsymbol{\lambda}^*\}$;
  Obtain the global model:
    $\mathbf{w}_g^{t+1} \leftarrow \gamma^* \cdot \left(\sum_{i=1}^m \lambda_i^* \mathbf{w}_i^t\right)$;
end for
Obtain the final global model $\mathbf{w}_g^T$.

6. FedLAW

6.1. Method

Based on the above understandings, we propose **Fed**erated Learning with **L**earnable **A**ggregation **W**eights algorithm (**FedLAW**) which combines the adaptive GWS and attentive LAW to optimize γ and $\boldsymbol{\lambda}$ simultaneously, defined as

$$\boldsymbol{\gamma}^*, \boldsymbol{\lambda}^* = \arg\min_{\gamma, \boldsymbol{\lambda}} \mathcal{L}_{proxy} \gamma \cdot \left(\sum_{i=1}^m \lambda_i \mathbf{w}_i^t\right), \tag{8}$$

$$\text{s.t. } \gamma > 0, \lambda_i \geq 0, \|\boldsymbol{\lambda}\|_1 = 1. \tag{9}$$

Table 4. Performance comparison under different numbers of clients. CIFAR-10, ResNet20, $E = 3$.

Setting	IID ($\alpha = 100$)		NonIID ($\alpha = 1$)	
Number of clients n	50	100	50	100
FedAvg	68.04	62.41	66.87	64.13
FedDF	48.24	38.66	38.70	22.51
Server-FT	67.77	62.30	66.73	64.63
FedLAW	**78.88**	**74.09**	**75.59**	**71.34**

Table 5. The performance of compared methods with different model architectures ($\alpha = 1,\ E = 1$).

Model	FedAvg	FedLAW	FedLAW(SWA)
ResNet20	74.11	78.72	78.64
ResNet56	74.22	78.93	79.08
ResNet110	74.50	78.11	79.19
WRN56_4	78.67	79.61	80.70
DenseNet121	85.13	86.50	87.06

The pseudo-code of FedLAW is shown in Algorithm 1.

With SWA (optional). We adopt an alternative two-stage strategy for SWA variant (implementing it in a reversed order also works), where we first fix $\boldsymbol{\lambda}$ and optimize γ, then we use the learned γ and fix it to optimize $\boldsymbol{\lambda}$ with SWA. In our experiments, we denote FedLAW with or without SWA as "FedLAW (SWA)" or "FedLAW".

6.2. Experiments

Baselines and Settings. We conduct experiments to verify the effectiveness of FedLAW. We mainly compare FedLAW with other server-side methods, i.e. FedDF (Lin et al., 2020) and FedBE (Chen & Chao, 2021), that also require a proxy dataset for additional computation. These two methods conduct ensemble distillation on the proxy data to transfer knowledge from clients' models to the global model. We add Server-FT as a baseline for simply finetuning global models on the proxy dataset. Besides, we implement client-side algorithms FedProx (Li et al., 2020b) and FedDyn (Acar et al., 2020) for comparison. If not mentioned otherwise, the number of clients is 20. More implementation details can be found in Appendix C and Appendix D.

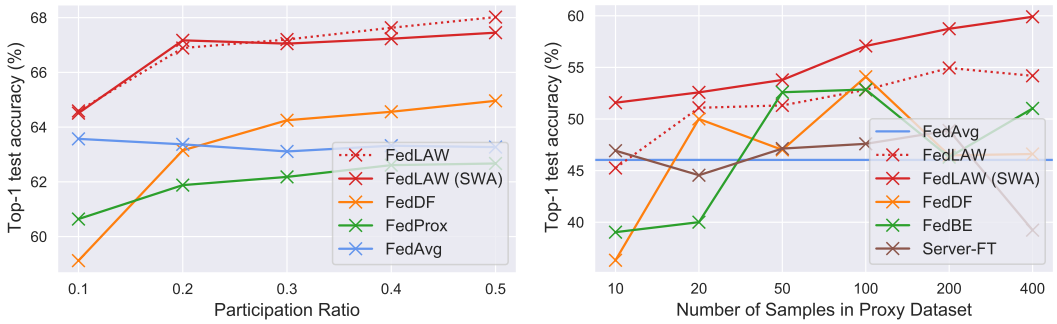

Figure 8. **Left:** The performance with different participation ratios ($\alpha = 1,\ E = 3$). **Right:** The performance with different sizes of the proxy dataset ($\alpha = 0.1,\ E = 3$).

Experimental results. ***Different datasets:*** As in Table 3, FEDLAW outperforms baselines on different datasets and models in both IID and NonIID settings. Compared with FEDDF, FEDBE and SERVER-FT, FEDLAW can better utilize the proxy dataset. ***Different numbers of clients:*** We implement experiments by scaling up the number of clients in Table 4, and it is shown that FEDLAW also surpasses the baselines by large margins. ***Different model architectures:*** We test FEDLAW across wider and deeper ResNet and other architecture, such as DenseNet (Huang et al., 2017), in the Table 5. It shows that FEDLAW is effective across different architectures, and it performs well even when the network goes deeper or wider. ***Different participation ratios:*** From the left of Figure 8, FEDLAW performs well under partial participation. ***Different sizes and distributions of proxy dataset:*** From the right of Figure 8, the server-side baselines are sensitive to the size of the proxy dataset that too small or too large proxy set will cause overfitting. However, FEDLAW is also effective under an extremely tiny proxy set and benefits more from a larger proxy set due to accurate aggregation weight optimization. We report the results of different distributions of the proxy dataset in Table 6 and Table 7, which show that FedLAW still works when there exists a distribution shift between the proxy dataset and the global data distribution of clients. ***Robustness against corrupted clients:*** Another advantage of FEDLAW is that it can filter out corrupted clients by assigning them lower weights. We generate corrupted clients by swapping two labels in their local training data. As in Table 8, FEDLAW is robust against corrupted clients, and it is as robust as the ensemble distillation methods, such as FEDDF, using the same proxy dataset.

More results. We present more results in the appendix. Specifically, the learning curves of test accuracy (Figures 15-17) and the server training process of FEDLAW (Figure 14).

Table 6. The performance on the distribution shift setting where the clients' data are overall balanced and the proxy data are long-tailed ($\rho = 10$).

Setting	IID ($\alpha = 100$)			NonIID ($\alpha = 1$)		
Type of proxy data	Balanced	Long-tailed		Balanced	Long-tailed	
Balanced Sampling	-	w/o	w	-	w/o	w
FEDAVG	75.24	75.24	75.24	73.46	73.46	**73.46**
FEDDF	76.20	74.04	73.31	74.39	73.99	73.14
FEDLAW	**79.40**	**77.14**	**78.56**	**76.70**	**76.78**	70.42

Table 7. The performance on the distribution shift setting where the clients' data are long-tailed ($\rho = 5$) and the proxy data are balanced.

Method	IID ($\alpha = 100$)	NonIID ($\alpha = 1$)
FEDAVG	61.12	59.82
FEDDF	39.03	40.68
FEDLAW	**67.61**	**66.48**

Table 8. The performance on different percentages of corrupted clients (IID, $E = 3$).

Corrupt percent.	25%	50%	75%
FEDAVG	63.40	61.14	58.21
FEDDF	68.73	66.94	66.07
SERVER-FT	63.61	61.24	58.36
FEDLAW	67.87	66.67	63.91
FEDLAW(SWA)	68.04	66.95	65.51

7. Conclusion

In this paper, we revisit and rethink the weighted aggregation in federated learning with neural networks and gain new insights into the training dynamics. First, we break the convention that the l_1 norm of aggregation weights should be normalized as 1 and identify the *global weight shrinking* phenomenon and its dynamics when the norm is smaller than 1. Second, we discover two aspects of *client coherence*, local gradient coherence and heterogeneity coherence, and study the dynamics during training. Based on the findings, we devise a simple but effective method FEDLAW. Extensive experiments verify that our method can improve the generalization of the global model by a large margin on different datasets and models.

Acknowledgments

This work was supported by the National Key Research and Development Project of China (Grant No.2021ZD0110505), the National Natural Science Foundation of China (Grant No.U19B2042), the Zhejiang Provincial Key Research and Development Project (Grant No.2022C01044), the University Synergy Innovation Program of Anhui Province (Grant No.GXXT-2021-004), the Academy Of Social Governance Zhejiang University, and the Fundamental Research Funds for the Central Universities (Grant No.226-2022-00064). This work was also supported in part by the Research Center for Industries of the Future (RCIF) at Westlake University, and Westlake Education Foundation.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Caldarola, D., Caputo, B., and Ciccone, M. Improving generalization in federated learning by seeking flat minima. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pp. 654–672. Springer, 2022.

Charles, Z., Garrett, Z., Huo, Z., Shmulyian, S., and Smith, V. On large-cohort training for federated learning. *Advances in neural information processing systems*, 34: 20461–20475, 2021.

Chatterjee, S. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2019.

Chatterjee, S. and Zielinski, P. Making coherence out of nothing at all: measuring the evolution of gradient alignment. *arXiv preprint arXiv:2008.01217*, 2020.

Chen, H. and Chao, W. Fedbe: Making bayesian model ensemble applicable to federated learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Deng, Y., Kamani, M. M., and Mahdavi, M. Distributionally robust federated averaging. *Advances in neural information processing systems*, 33:15111–15122, 2020.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Du, J., Yan, H., Feng, J., Zhou, J. T., Zhen, L., Goh, R. S. M., and Tan, V. Y. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Fort, S. and Jastrzebski, S. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

Fort, S., Nowak, P. K., Jastrzebski, S., and Narayanan, S. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pp. 1165–1173. PMLR, 2017.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Guo, P., Yang, D., Hatamizadeh, A., Xu, A., Xu, Z., Li, W., Zhao, C., Xu, D., Harmon, S., Turkbey, E., et al. Auto-fedrl: Federated hyperparameter optimization for multi-institutional medical image segmentation. *arXiv preprint arXiv:2203.06338*, 2022.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., and Zhang, Y. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7865–7873, 2021.

Izmailov, P., Wilson, A., Podoprikhin, D., Vetrov, D., and Garipov, T. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 876–885, 2018.

Jastrzębski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of dnn loss and the sgd step length. *arXiv preprint arXiv:1807.05031*, 2018.

Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., and Geras, K. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2019.

Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., and Geras, K. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020.

Jomaa, H. S., Grabocka, J., and Schmidt-Thieme, L. Hyp-rl: Hyperparameter optimization by reinforcement learning. *arXiv preprint arXiv:1906.11527*, 2019.

Kantorovich, L. V. On the translocation of masses. *Journal of mathematical sciences*, 133(4):1381–1382, 2006.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Keskar, N. S., Nocedal, J., Tang, P. T. P., Mudigere, D., and Smelyanskiy, M. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

Lewkowycz, A. and Gur-Ari, G. On the training dynamics of deep networks with l_2 regularization. *Advances in Neural Information Processing Systems*, 33:4790–4799, 2020.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Li, S., Zhou, T., Tian, X., and Tao, D. Learning to collaborate in decentralized learning of personalized models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9766–9775, 2022a.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.*, 37(3):50–60, 2020a. doi: 10.1109/MSP.2020.2975749. URL https://doi.org/10.1109/MSP.2020.2975749.

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, 2020b.

Li, X., Chen, S., and Yang, J. Understanding the disharmony between weight normalization family and weight decay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4715–4722, 2020c.

Li, Z., Lu, J., Luo, S., Zhu, D., Shao, Y., Li, Y., Zhang, Z., Wang, Y., and Wu, C. Towards effective clustered federated learning: A peer-to-peer framework with adaptive neighbor matching. *IEEE Transactions on Big Data*, 2022b.

Li, Z., Lu, J., Luo, S., Zhu, D., Shao, Y., Li, Y., Zhang, Z., and Wu, C. Mining latent relationships among clients: Peer-to-peer federated learning with adaptive neighbor matching. *arXiv preprint arXiv:2203.12285*, 2022c.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Lyu, K., Li, Z., and Arora, S. Understanding the generalization benefit of normalization layers: Sharpness reduction. *arXiv preprint arXiv:2206.07085*, 2022.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

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.

Mostafa, H. Robust federated learning through representation matching and adaptive hyper-parameters. *arXiv preprint arXiv:1912.13075*, 2019.

Singh, S. P. and Jaggi, M. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

Vlaar, T. and Frankle, J. What can linear interpolation of neural network loss landscapes tell us? *arXiv preprint arXiv:2106.16004*, 2021.

Wan, R., Zhu, Z., Zhang, X., and Sun, J. Spherical motion dynamics: Learning dynamics of normalized neural network using sgd and weight decay. *Advances in Neural Information Processing Systems*, 34:6380–6391, 2021.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020a.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Wu, B., Liang, Z., Han, Y., Bian, Y., Zhao, P., and Huang, J. Drflm: Distributionally robust federated learning with inter-client noise via local mixup. *arXiv preprint arXiv:2204.07742*, 2022.

Xia, Y., Yang, D., Li, W., Myronenko, A., Xu, D., Obinata, H., Mori, H., An, P., Harmon, S., Turkbey, E., et al. Auto-fedavg: learnable federated averaging for multi-institutional medical image segmentation. *arXiv preprint arXiv:2104.10195*, 2021.

Xie, Z., Sato, I., and Sugiyama, M. Understanding and scheduling weight decay. *arXiv preprint arXiv:2011.11152*, 2020.

Yan, G., Wang, H., and Li, J. Critical learning periods in federated learning. *arXiv preprint arXiv:2109.05613*, 2021.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. W. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pp. 581–590. IEEE, 2020.

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1998–2007. PMLR, 2018.

Zhang, G., Wang, C., Xu, B., and Grosse, R. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2018.

Zielinski, P., Krishnan, S., and Chatterjee, S. Weak and strong gradient directions: Explaining memorization, generalization, and hardness of examples at scale. *arXiv preprint arXiv:2003.07422*, 2020.

Zou, D. and Gu, Q. An improved analysis of training overparameterized deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Appendix

In this appendix, we provide details omitted in the main paper and more experimental results and analyses.

- Appendix A: more related works (cf. section 2 of the main paper).
- Appendix B: more experimental results and analyses (cf. section 4, section 5 and section 6 of the main paper).
- Appendix C: additional details of FEDLAW (cf. section 6 of the main paper).
- Appendix D: details of experimental setups (cf. section 4, section 5 and section 6 of the main paper).

A. More Related Works

A.1. Model Aggregation in Federated Learning

Model aggregation in federated learning. Model aggregation weights should be calibrated under asynchronous local updates. FEDNOVA (Wang et al., 2020b) is proposed to tackle the objective inconsistency problem caused by asynchronous updates; it theoretically shows that the convergence will be improved if the numbers of local iterations normalize the aggregation weights. However, it does not take the heterogeneity degree of clients into account, which is also a key factor that affects the generalization of the global model. In (Chen & Chao, 2021), the authors point out that due to heterogeneity, the best-performing model will shift away from FEDAVG, but they do not give insights on how to adjust aggregation weight to approximate the best model, they use Bayesian ensemble distillation method to prove the generalization of the global model instead. To solve the misalignment of neurons in FL with DNNs, FEDMA (Wang et al., 2020a) is proposed: FEDMA constructs the shared global model layer-wise by matching and averaging hidden elements with similar features extraction signatures. Besides, optimal transport (Kantorovich, 2006) can be adopted in layer-wise neuron alignment in the process of model fusion (Singh & Jaggi, 2020). These previous works improve the global model performance by layer-wise alignment, but they are complex and computation-expensive, and they can not be applied under the traditional weighted aggregation scheme. In this paper, we only focus on the convex combination of clients' local models by weighted aggregation, which is the most common and general way of model aggregation.

A.2. Generalization and Training Dynamics of Neural Networks

Loss landscape of neural networks and generalization. Deep neural networks (DNNs) are highly non-convex and over-parameterized, and visualizing the loss landscape of DNNs (Li et al., 2018; Vlaar & Frankle, 2021) helps understand the training process and the properties of minima. There are mainly two lines of works about the loss landscape of DNNs. The first one is the linear interpolation of neural network loss landscape (Vlaar & Frankle, 2021; Garipov et al., 2018; Draxler et al., 2018), it plots linear slices of the landscape between two networks. In linear interpolation loss landscape, mode connectivity (Draxler et al., 2018; Vlaar & Frankle, 2021; Entezari et al., 2022) is referred to as the phenomenon that there might be increasing loss on the linear path between two minima found by SGD, and the loss increase on the path between two minima is referred to as (energy) barrier. It is also found that there may exist barriers between the initial model and the trained model (Vlaar & Frankle, 2021). The second line concerns the loss landscape around a trained model's parameters (Li et al., 2018). It is shown that the flatness of loss landscape curvature can reflect the generalization (Foret et al., 2020; Izmailov et al., 2018) and top hessian eigenvalues can present flatness (Yao et al., 2020; Jastrzębski et al., 2018). Networks with small top hessian eigenvalues have flat curvature and generalize well. Previous works seek flatter minima for improving generalization by implicitly regularizing the hessian (Foret et al., 2020; Kwon et al., 2021; Du et al., 2021).

Critical learning period in training neural networks. (Jastrzebski et al., 2019) found that the early phase of training of deep neural networks is critical for their final performance. They show that a break-even point exists on the learning trajectory, beyond which SGD implicitly regularizes the curvature of the loss surface and noise in the gradient. They also found that using a large learning rate in the initial phase of training reduces the variance of the gradient and improves generalization. In FL, (Yan et al., 2021) discovers the early training period is also critical to federated learning. They reduce the quantity of training data in the first couple of rounds and then recover the training data, and it is found that no matter how much data are added in the late period, the models still cannot reach a better accuracy. However, it did not further study the role of client heterogeneity in the critical learning period while we examine it by local gradient coherence.

A.3. Federated Hyperparameter Optimization

Current federated learning methods struggle in cases with heterogeneous client-side data distributions which can quickly lead to divergent local models and a collapse in performance. Careful hyperparameter tuning is particularly important in these cases. Hyperparameters can be optimized using gradient descent to minimize the final validation loss (Maclaurin et al.,

Table 9. More results about fixed γ across different architectures in various NonIID settings.

γ	**1.0**	**0.99**	**0.97**	**0.95**	**0.93**	**0.9**
Model	$\alpha = 10$					
SimpleCNN	65.96	67.19	69.41	**69.81**	69.69	69.59
AlexNet	73.9	74.43	74.96	75.12	**75.33**	74.06
ResNet8	76.26	75.63	76.92	**77.23**	76.9	76.61
Model	$\alpha = 0.5$					
SimpleCNN	65.78	66.59	67.93	**68.85**	68.75	68.25
AlexNet	73.72	73.06	73.89	**73.98**	73.6	73.33
ResNet8	73.4	73.93	**75.39**	74.12	73.66	73.46
Model	$\alpha = 0.2$					
SimpleCNN	63.52	64.68	63.72	**65.82**	65.4	64.97
AlexNet	68.41	70.46	**70.87**	70.74	70.58	69.42
ResNet8	71.85	70.96	**72.76**	72.04	71.25	62.32
Model	$\alpha = 0.1$					
SimpleCNN	60.57	61.22	61.83	**62.05**	62.05	60.85
AlexNet	**66.18**	65.25	64.74	64.23	64.16	61.24
ResNet8	**63.89**	60.55	61.38	59.23	58.76	39.85

Table 10. The performance of adaptive GWS under different global learning rates.

	IID ($\alpha = 100$)			NonIID ($\alpha = 1$)		
Global learning rate	0.5	1	1.5	0.5	1	1.5
FedAvg	69.15	68.18	64.35	68.71	67.09	64.00
Adaptive GWS	71.45	71.98	71.13	69.65	71.02	71.04
γ of Adaptive GWS	0.986	0.974	0.963	0.991	0.979	0.967

2015; Franceschi et al., 2017). Moreover, hyperparameters can be optimized based on reinforcement learning methods (Guo et al., 2022; Jomaa et al., 2019; Mostafa, 2019). However, in this paper, optimizing aggregation weights is not our main novelty. Instead, we focus on leveraging this toolbox to examine the crucial training dynamics in FL in a principled way.

B. More Results and Analyses

B.1. Global Weight Shrinking

Fixed γ. We add more results about global weight shrinking experiments with fixed γ as in Table 9. It is found that when data are more NonIID, fixed γ will cause negative effects; this is more dominant when $\alpha = 0.1$ and the models are AlexNet or ResNet8.

Adaptive GWS with global learning rate. We conduct experiments with the adaptive GWS under different global learning rates for both IID and NonIID settings. We train SimpleCNN on CIFAR10 with 1 local epoch, and the results are reported in Table 10. It can be observed that in both IID and NonIID settings, a small global server learning rate can improve FEDAVG's performance. In contrast, the larger the global learning rate, the smaller the learned γ (stronger regularization). It is aligned with our insights in the main paper that larger pseudo gradients require stronger regularization. Moreover, adaptive GWS is robust to the choice of the global server learning rate, especially in the IID setting.

Adaptive GWS under various heterogeneity. We show adaptive GWS works under various heterogeneity and visualize γ and the norm of the global gradient in each setting, as in Figure 9. It demonstrates that adaptive GWS can boost performance under different NonIID settings, but it has a smaller benefit when the system is extremely NonIID (i.e., $\alpha = 0.1$). Additionally, according to the right figure of Figure 9, except for the outlier γ when $\alpha = 10$, the learned γ decreases when data become more IID, causing stronger weight shrinking effect. We think this is a result of a balance between optimization and regularization. The volumes of global gradients change when the heterogeneity changes. The norm of global gradient increases when data become more IID, and it requires smaller γ to cause stronger regularization.

More results of general understanding of adaptive GWS. First, we first visualize the norm of model parameter weight during training as in the left figure of Figure 10. Adaptive GWS results in a smaller model parameter during training. Second, we use two common metrics to measure the flatness of loss landscape during training as in the middle and right figures of

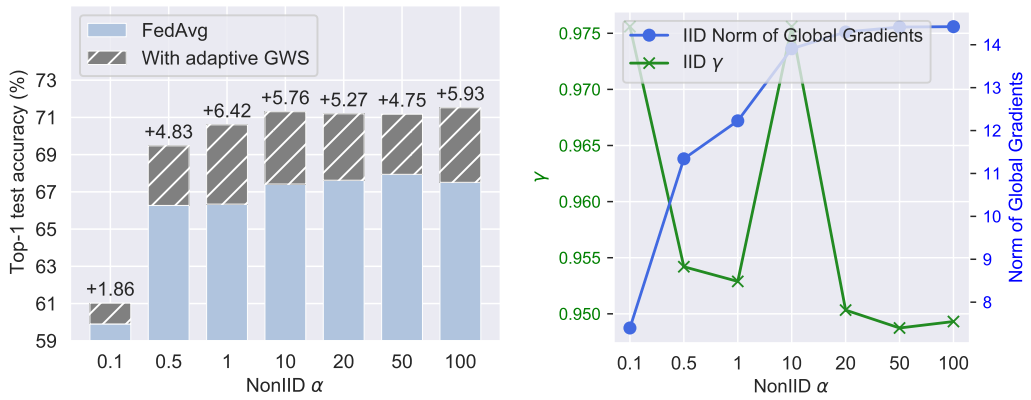

Figure 9. **Adaptive GWS under various heterogeneity. Left:** Test accuracy gains with adaptive GWS. In all settings, adaptive GWS can bring performance gains. **Right:** Learned γ of adaptive GWS in different settings. γ decreases when data become more IID, causing the stronger weight shrinking effect. This is due to the changes in the volumes of global gradients. The norm of global gradient increases when data become more IID, and it requires smaller γ to cause stronger regularization.

Figure 10, and they are the hessian eigenvalue-based metrics. The dominant hessian eigenvalue evaluates the worst-case loss landscape, which means the larger top 1 eigenvalue indicates the greater change in the loss along this direction and the sharper the minima (Keskar et al., 2017). We adopt the top 1 hessian eigenvalue and the ratio of top 1 and top 5, which are commonly used as a proxy for flatness (Jastrzebski et al., 2020; Fort & Jastrzebski, 2019). Usually, a smaller top 1 hessian eigenvalue and a smaller ratio of top 1 hessian eigenvalue and top 5 indicates flatter curvature of DNN. As in the figures, during the training, FEDAVG generates global models with sharp landscapes whereas adaptive GWS tends to generate more generalized models with flatter curvatures.

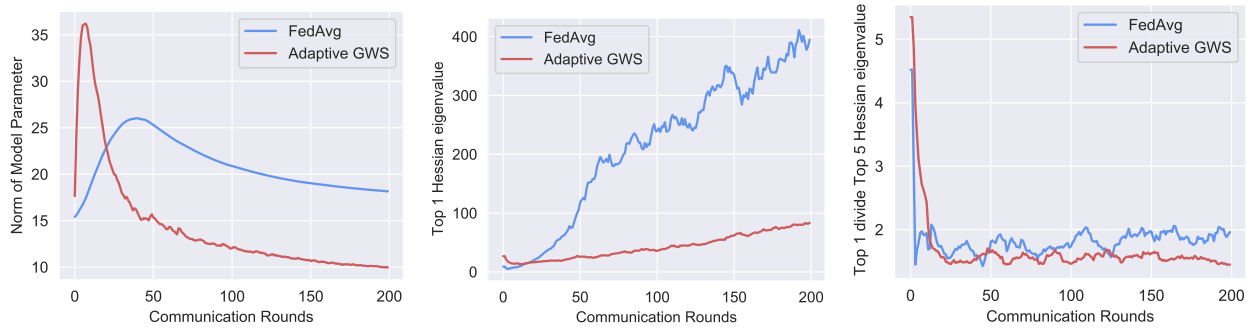

Figure 10. **More results of general understanding of adaptive GWS. Left:** Adaptive GWS results in a smaller model parameter during training. **Middle:** Smaller top 1 hessian eigenvalue indicates flatter curvature of DNNs. The result shows FEDAVG tends to generate sharper global models during training while adaptive GWS seeks flatter networks. **Right:** The ratio of the top 1 hessian eigenvalue and top 5 is another indicator; a smaller value means flatter minima.

The distribution of r. We visualize r (the ratio of the global gradient and the regularization pseudo gradient) values of all experiments in Figure 2 and Figure 9 as in Figure 11. It is found that the distribution of r can be approximated into a Gaussian distribution with its mean around 20.5.

B.2. Client Coherence

The relationship with gradient diversity. The conclusion of gradient diversity (Yin et al., 2018) is opposite to the one of gradient coherence. Gradient diversity argues that higher similarities between workers' gradients will degrade performance in distributed mini-batch SGD, while gradient coherence claims that higher similarities between the gradients of samples will boost generalization (Yin et al., 2018; Chatterjee, 2019). Moreover, gradient diversity is somewhat controversial. As argued in the line of works about gradient coherence (Chatterjee & Zielinski, 2020; Chatterjee, 2019), the manuscript of gradient diversity did not explicitly measure the gradient diversity in the experiments (or further study its properties): only

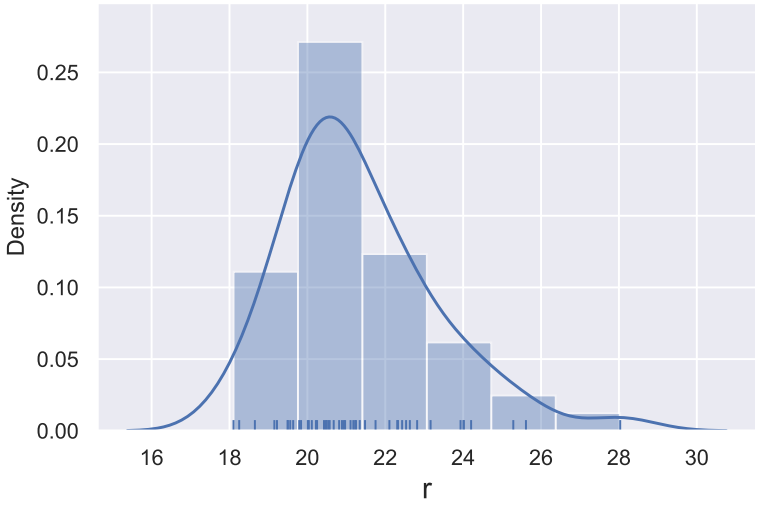

Figure 11. **Distribution of r.** We visualize r values of all experiments in Figure 2 and Figure 9 and find that the distribution of r can be approximated into a Gaussian distribution with its mean around 20.5.

experiments on CIFAR-10 can be found where they replicate $1/r$ of the dataset r times and show that greater the value of r less the effectiveness of mini-batching to speed up. Apart from this controversy, the strongly-convex assumption in the theorem of gradient diversity (Yin et al., 2018) may make it weaker to generalize its conclusions in neural networks while we are studying the empirical properties in FL with neural networks. Taking the above statements into consideration, gradient diversity may be infeasible in our settings.

The relationship with client similarity works in FL. There are some works (Karimireddy et al., 2020; Li et al., 2020b) taking the bounded gradient dissimilarity assumption to deduce theorems. In their assumptions, they bound the gradient sum or gradient norm, but we use the cosine similarity to study how the clients interplay with each other and contribute to the global. So the perspectives are quite different. Additionally, there are previous works in FL that use cosine similarity of clients' gradients to improve personalization (Huang et al., 2021; Li et al., 2022b); however, we focus on the training dynamics in generalization, and one of our novel findings is we discover a critical point exists and the periods that before or after this point play different roles in the global generalization.

Visualization of how heterogeneity affects the optimal aggregation weight. We set up a three-node toy example on CIFAR-10 by hybrid Dirichlet sampling as shown in Figure 12. We first sample client 0's data distribution by Dirichlet sampling according to α_1; then we sample data distributions for clients 1 and 2 on the remaining data with α_2. We set up three settings with different α_1, α_2 and illustrate the data distributions on the **Left column** in Figure 12. In the example, the aggregation weights (AWs) are $[\lambda_0, \lambda_1, \lambda_2]$, we regularize the weights as $\lambda_0 + \lambda_1 + \lambda_2 = 1$ which is a plane that can be visualized in 2-D. We uniformly sample points on the plane to obtain global models with different AW and compute the test loss, and then the loss landscapes on the plane can be visualized. We implement FEDAVG for 100 rounds and record the loss landscape and the optimal weight on the loss landscape in each round; then we illustrate the loss landscape of round 10 on the **Middle column** and the optimal weight trajectory on the **Right column** of Figure 12.

In these settings, clients have different heterogeneity degrees: in the first setting, client 0 has a balanced dataset while the data of clients 1 and 2 are complementary; in the second and third settings, clients 1 and 2 have the same data distribution, which differs from the client 0's. From Figure 12, it is evident that the weight of FEDAVG is biased from optimal weights when heterogeneity degrees vary in clients, we can draw the following conclusions: (1) optimal weight can be viewed as a Gaussian distribution in the aggregation weight hyperplane; (2) the mean of the Gaussian will drift towards to the directions where data are more inter-heterogeneous (for instance, in the third setting, client 0's major classes are 2, 3 and 8 while client 1 and 2 have rare data on these classes, so client 0's contribution is more dominant); (3) the variance of the Gaussian is larger in inter-homogeneous direction and is smaller in inter-heterogeneous direction (the variance along the client 1-client 2 direction is large in the second and third settings, because the two clients have inter-homogeneous data; opposite phenomenon is shown in the first setting, where client 1 and 2 have inter-heterogeneous data); (4) the flatness of loss landscape on aggregation weight hyperplane is consistent with the variance of the Gaussian, which means the directions with more significant variance will have flatter curvature in the landscape. From our analysis, it is clear that clients' contributions

to the global model should not be solely measured by dataset size, and the heterogeneity degree should also be taken into account. And we observe that in a more heterogeneous environment, the loss landscape is sharper, which means the bias from optimal weight will cause more generalization drop. In other words, in a heterogeneous environment, appropriate aggregation weight matters more.

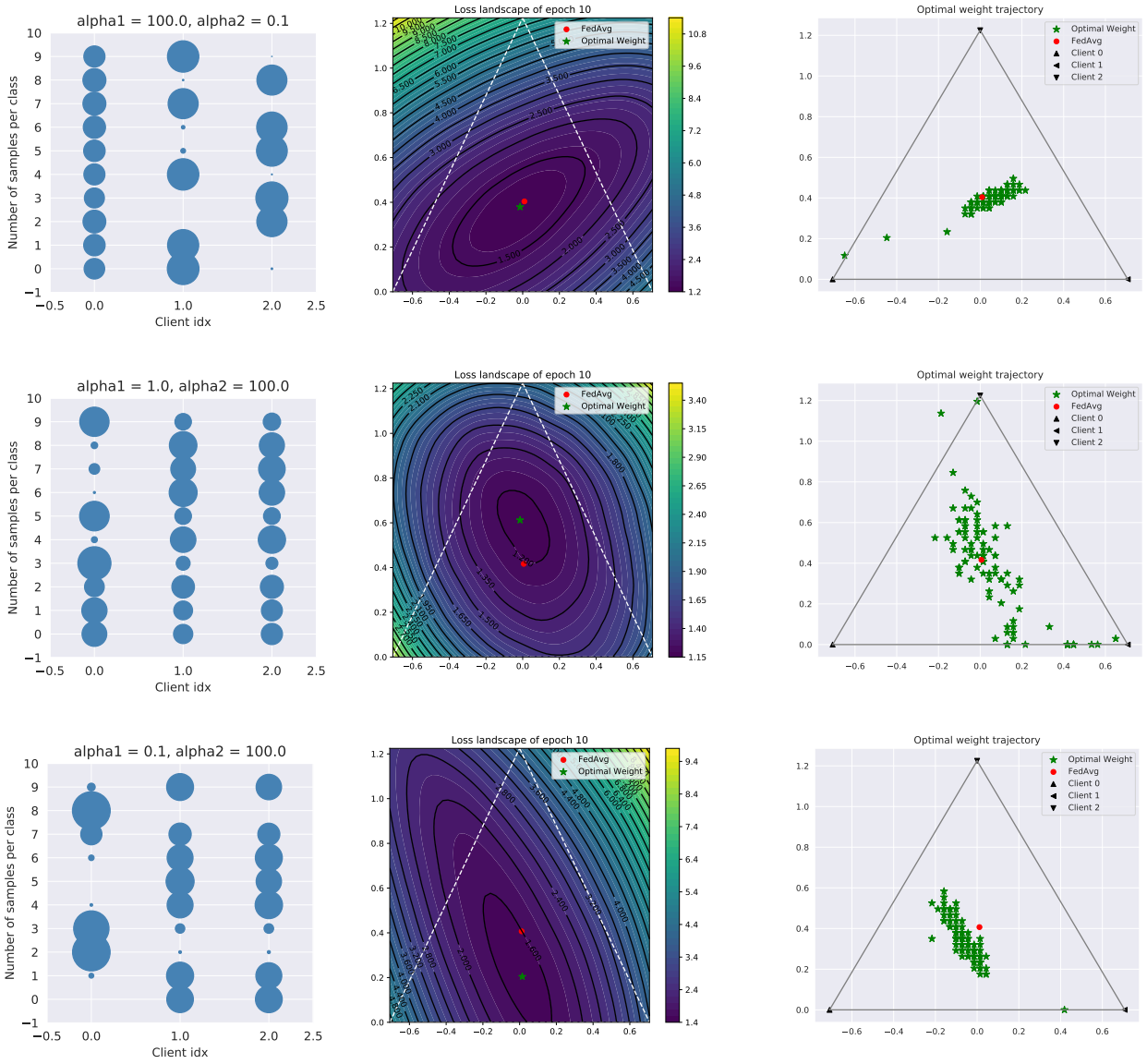

Figure 12. **Heterogeneity also affects the optimal aggregation weight.** A three-node toy example on CIFAR-10 is shown. **Left:** Data distribution of each client, note that each client has the same dataset size. **Middle:** Loss landscape on the plane of aggregation weight, it is noticed that FEDAVG is off the optimal and the landscape has various flatness in different directions. **Right:** optimal weight trajectory during training. We plot the optimal weights in each round (green dots) and find that the optimal weights are biased from FEDAVG.

Visualization of the hybrid NonIID setting of Figure 5. We visualize the hybrid NonIID setting of Figure 5 in Figure 13. We take $\alpha_1 = 10$ and $\alpha_2 = 0.1$, so the first 10 clients (indexed 0-9) have class-balanced data while the last 10 clients (indexed 10-19) have class-imbalanced data.

Data size or heterogeneity? A correlation analysis. Data size and heterogeneity all affect clients' contributions to the global model, but which affects it most? As in previous literature, the importance is depicted by the dataset size that clients with more data will be assigned larger weights. According to the analysis in Figure 12, the importance of weight may be

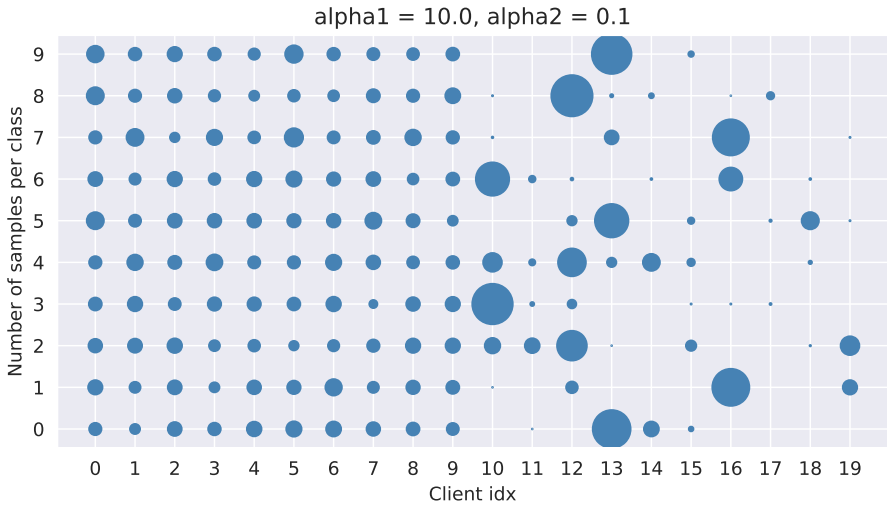

Figure 13. **Data distribution of Figure 5**

Table 11. **Pearson correlation coefficient analysis of AW.** Heterogeneity degree is calculated as the reciprocal of the variance of class distribution for each client. We take the accumulated weights during the training as clients' AW.

Factors	$E=1$	$E=5$	$E=1\&E=5$
Dataset size (DS)	-0.098	**0.21**	0.035
Heterogeneity degree (HD)	**0.41**	0.024	0.35
DS×HD	0.26	0.17	**0.31**

associated with the heterogeneity degrees of clients. To explore which factor is more dominant in the AW optimized by attentive LAW, we have made a Pearson correlation coefficient analysis in Table 11. Results show that dataset size is more dominant when the local epoch is large; otherwise, the heterogeneity degree. This phenomenon is intuitive: when the local epoch increases, clients with a larger dataset will have more local iterations than others (Wang et al., 2020b), so their updates are more dominant. In the cases where the local epoch is small, clients' updates are of similar volumes; here the updates' directions are much more important since balanced clients are prone to have stronger coherence, and their AWs are larger in model aggregation. We combine two factors by multiplication, and the result shows that the combined indicator is more dominant when the two cases are mixed.

B.3. Learning curves of FedLAW and baselines

We add the test accuracy curves to show the learning processes of the algorithms and visualize them in Figure 15 (FasionMNIST), Figure 16 (CIFAR-10), and Figure 17 (CIFAR-100). The curves are according to the results in Table 3. It shows that FEDLAW surpasses the baseline algorithms in most cases. Besides, FEDLAW is steady in the learning curves and it avoids over-fitting in the late training.

We also visualize the server training process of FedLAW in Figure 14. It is found that γ converges faster than λ. For γ, it converges to the optimal value in about 30 server epochs, while for λ, it needs 80 epochs to fully converge.

C. Additional Details of FedLAW

In FEDLAW, we optimize AW on the server as Equation 8, and there are constraints that $\lambda_i \geq 0, \|\boldsymbol{\lambda}\|_1 = 1$. To realize these constraints, we adopt base functions in λ, and there are two alternatives, the quadratic function and the exponential function.

$$\text{Quadratic: } \lambda_i = \frac{x_i^2}{\sum_j^m x_j^2}; \text{Exponential: } \lambda_i = \frac{e^{x_i}}{\sum_j^m e^{x_j}}. \tag{10}$$

$\boldsymbol{x}$ is the variable that determines the value of $\boldsymbol{\lambda}$. We compute the gradients of $\boldsymbol{x}$ to update $\boldsymbol{\lambda}$. By using the base functions, $\boldsymbol{\lambda}$ can meet the constraints of non-negativity and $l_1 = 1$. The exponential function is the same as the Softmax function and we find these two functions have similar performances overall, so we only adopt the exponential function in the experiments.

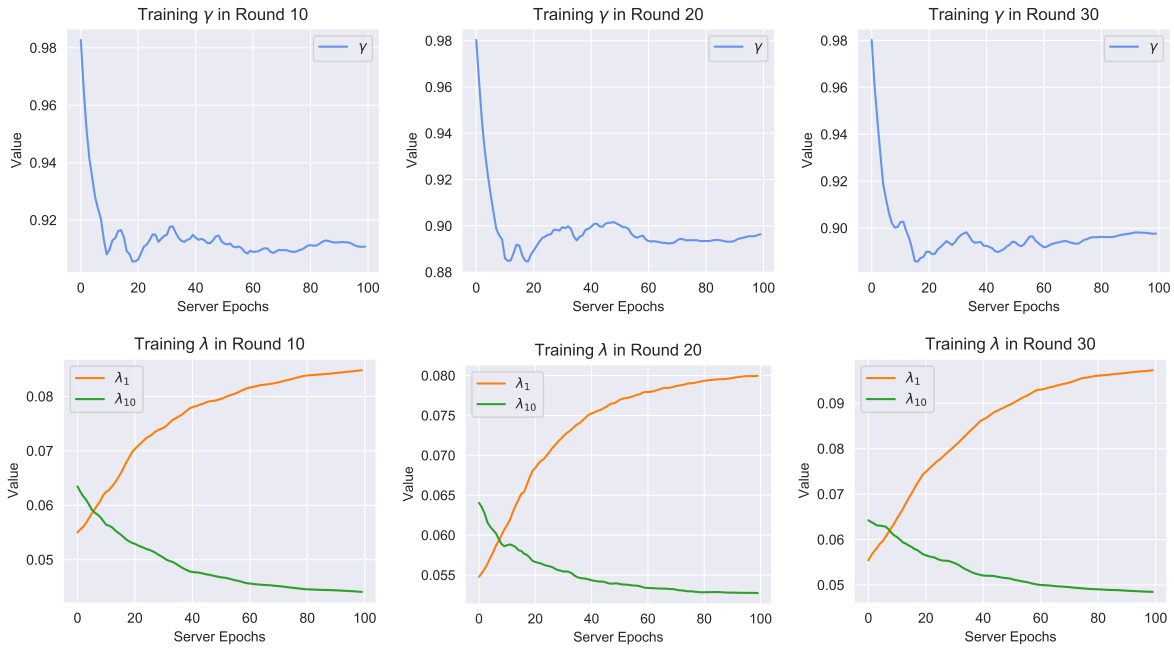

Figure 14. **Server training visualization of FEDLAW.** CIFAR10, $n = 20$, $E = 3$, NonIID $\alpha = 1.0$, ResNet20.

D. Implementation Details

D.1. Environment.

We conduct experiments under Python 3.8.5 and Pytorch 1.12.0. We use 4 Quadro RTX 8000 GPUs for computation.

D.2. Data

Data partition. To generate NonIID data partition amongst clients, we use Dirichlet distribution sampling in the trainset of each dataset. In our implementation, apart from clients having different class distributions, clients also have different dataset sizes; we think this partition is more realistic in practical scenarios. For the data partition in Figure 5 and Figure 12, we use a hybrid Dirichlet sampling to generate an FL system with both class-balanced clients and class-imbalanced clients. Specifically, we first generate all-client distribution with α_1, and we only keep half of these clients. Then we use the remaining data to generate the distribution of remaining clients with α_2. For the data in Figure 5, we first generate a 20-client distribution with $\alpha_1 = 10$ and keep the first 10 clients as the balanced clients; then we use the remaining data to generate distributions of the last 10 imbalanced clients with $\alpha_2 = 0.1$. The distribution is shown in Figure 13.

Data augmentation. We adopt no data augmentation in the experiments.

Proxy dataset. We use a small and class-balanced proxy dataset on the server. In Table 3, we use proxy datasets with 10 samples per class, which means, for FashionMNIST and CIFAR-10, there are 100 samples in the proxy datasets, and for CIFAR-100, there are 1000 samples in the proxy datasets. The proxy datasets are randomly selected from the testset of each dataset. Then we use the remaining data in the testset to test the global models' performance for all compared methods. For Table 8 and the right of Figure 8, we use CIFAR-10 and a 100-sample proxy dataset, while in Table 5, we use CIFAR-10 and a 1000-sample proxy dataset.

D.3. Model

SimpleCNN and MLP. The SimpleCNN for CIFAR-10 and CIFAR-100 is a convolution neural network model with ReLU activations which consists of 3 convolutional layers followed by 2 fully connected layers. The first convolutional layer is of size (3, 32, 3) followed by a max pooling layer of size (2, 2). The second and third convolutional layers are of sizes (32, 64, 3) and (64, 64, 3), respectively. The last two connected layers are of sizes (64*4*4, 64) and (64, num_classes, respectively. The MLP model for FasionMNIST is a three-layer MLP model with ReLU activations. The first layer is of size (28*28, 200), the second is of size (200, 200), and the last is (200, 10).

ResNet and DenseNet. We followed the model architectures used in (Li et al., 2018). The numbers of the model names mean the number of layers of the models. Naturally, the larger number indicates a deeper network. For WRN56_4 in Table 5,

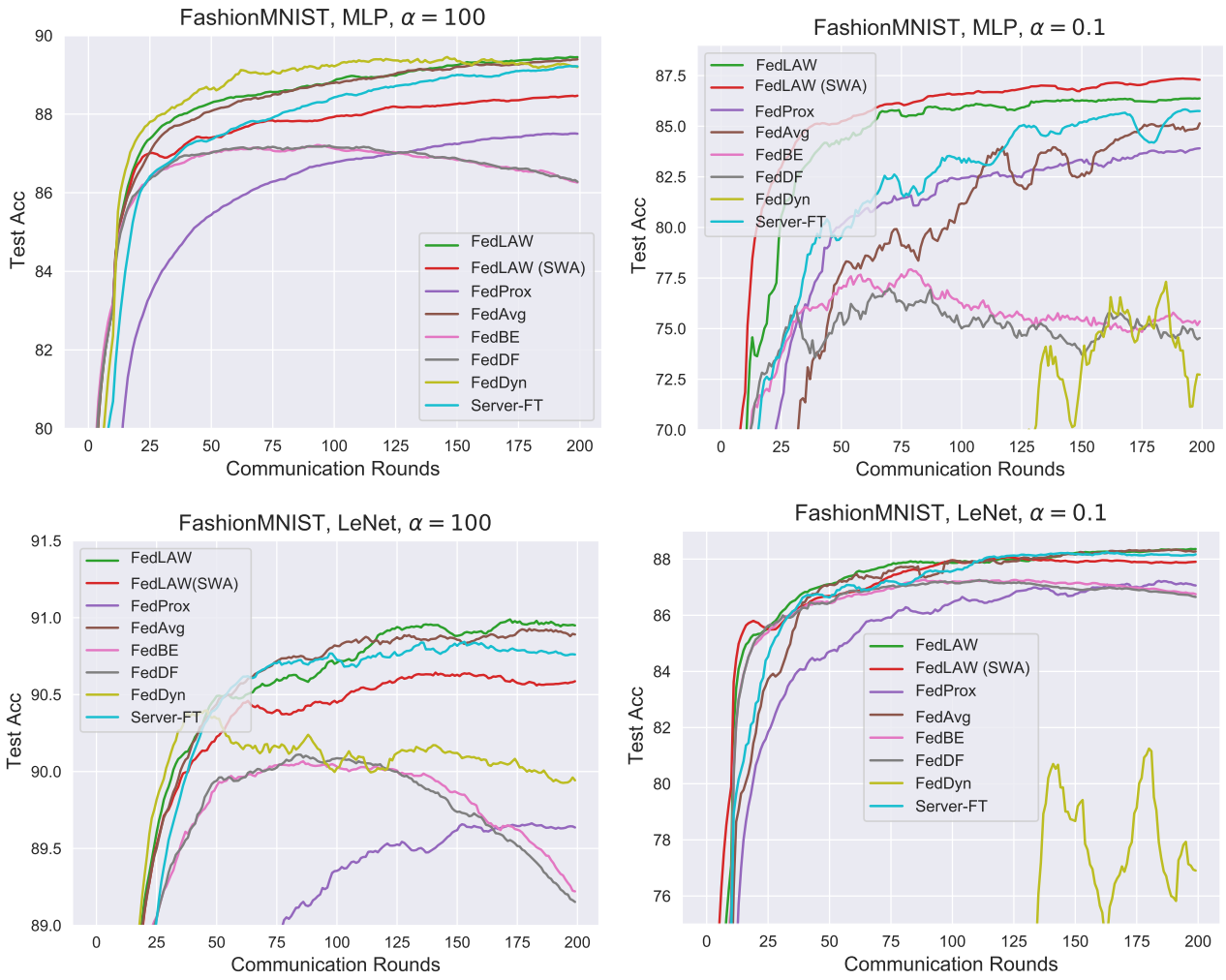

Figure 15. **Test accuracy curves of algorithms under FashionMNIST.** According to the results in Table 3.

it is an abbreviation of Wide-ResNet56-4, where "4" refers to four times as many filters per layer.

D.4. Randomness

Randomness is important for fair comparisons. In all experiments, we implement the experiments three times with different random seeds and report the averaged results. We use random seeds 8, 9, and 10 in all experiments. Given a random seed, we set torch, numpy, and random functions as the same random seed to make the data partitions and other settings identical. To make sure all algorithms have the same initial model, we save an initial model for each architecture and load the saved initial model at the beginning of one experiment. Also, for the experiments with partial participation, the participating clients in each round are vital in determining the model performance, and to guarantee fairness, we save the sequences of participating clients in each round and load the sequences in all experiments. This will make sure that, given a random seed and participation ratio, every algorithm will have the same sampled clients in each round.

D.5. Evaluation

We evaluate the global model performance on the testset of each dataset. The testset is mostly class-balanced and can reflect the global learning objective of an FL system. Therefore, we reckon the performance of the model on the testset can indicate the generalization performance of global models. In all experiments, we run 200 rounds and take the average test accuracy of the last 10 rounds as the final test accuracy for each experiment. For the indicators during training in section 4, like γ, r, the norm of global gradient, and the norm of GWS pseudo gradient, we take the averaged values in the middle stage of training, that is the average of 90-110 rounds.

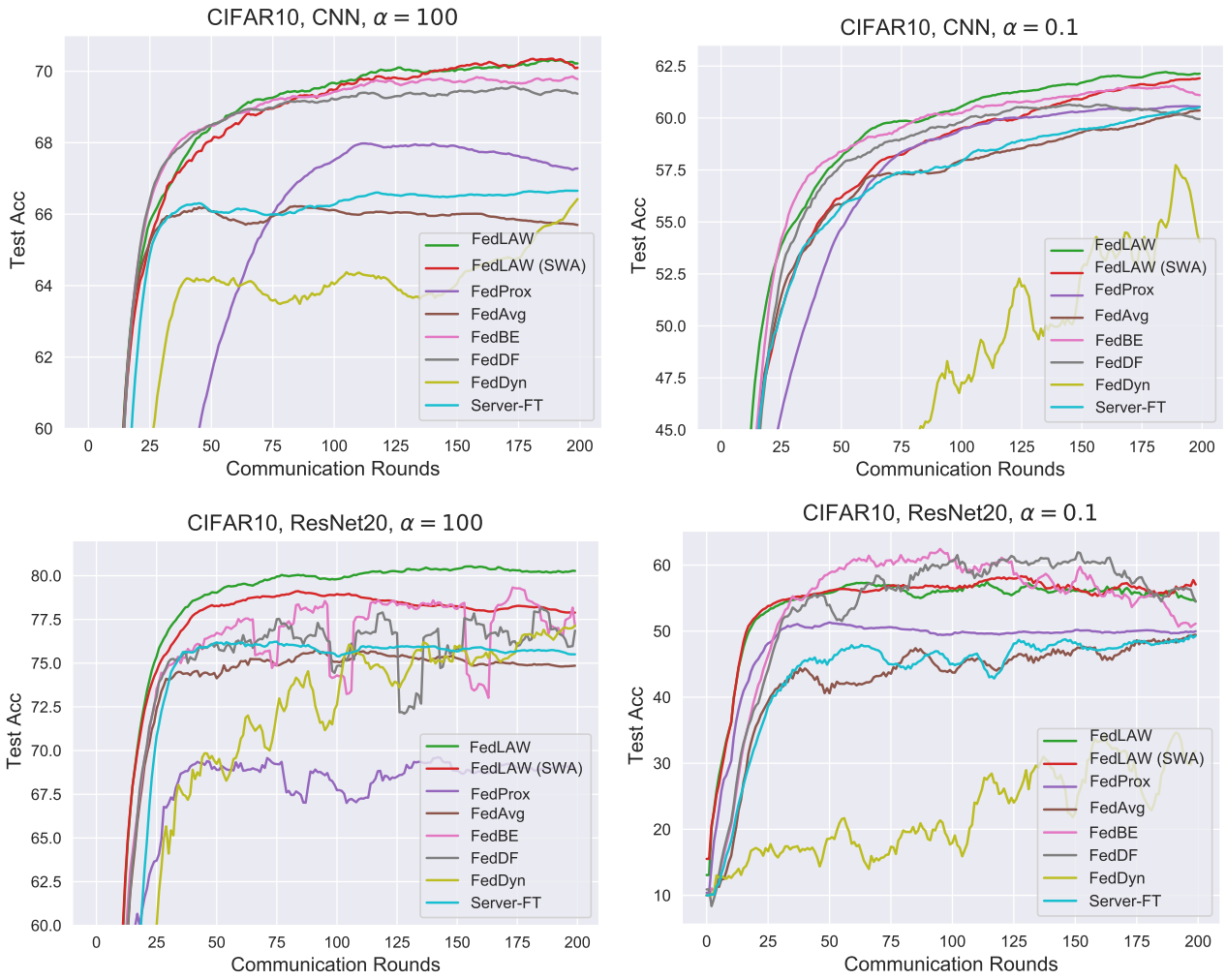

Figure 16. **Test accuracy curves of algorithms under CIFAR-10.** According to the results in Table 3.

D.6. Hyperparameter

Learning rate and the scheduler. We set the initial learning rates (LR) as 0.08 in CIFAR-10 and FashionMNIST and set LR as 0.01 in CIFAR-100. We set a decaying LR scheduler in all experiments; that is, in each round, the local LR is 0.99*(LR of the last round).

Local weight decay. We adopt local weight decay in all experiments. For CIFAR-10 and FashionMNIST, we set the weight decay factor as 5e-4, and for CIFAR-100, we set it as 5e-5.

Optimizer. We set SGD optimizer as the clients' local solver and set momentum as 0.9. For the server-side optimizer (FEDDF, FEDBE, SERVER-FT, and FEDLAW), we use Adam optimizer and betas=(0.5, 0.999).

Hyperparameter for FL algorithms. For FEDDF, FEDBE and FEDLAW, we set the server epoch as 100. We observe for SERVER-FT, this epoch is too large that it will cause negative effects, so we set the epoch as 2 for SERVER-FT. We set $\mu_{FedProx} = 0.001$ in FEDPROX and $\alpha_{FedDyn} = 0.01$ in FEDDYN as suggested in their official implementations or papers. For FEDBE, we use the Gaussian mode in SWAG server. We did not use temperature smoothing in the ensemble distillation methods FEDDF and FEDBE.

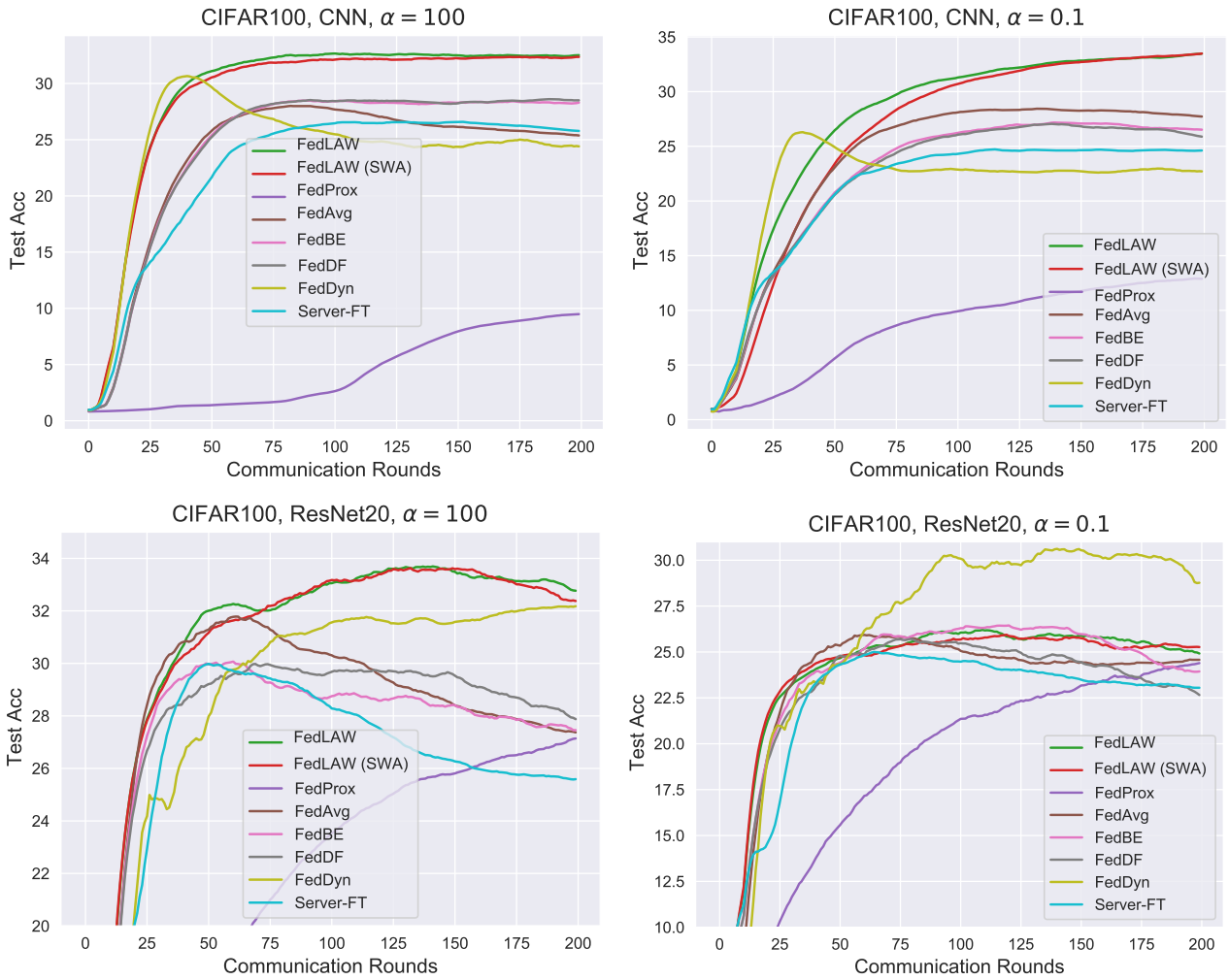

Figure 17. **Test accuracy curves of algorithms under CIFAR-100.** According to the results in Table 3.