

Understanding and Improving Model Averaging in Federated Learning on Heterogeneous Data

Tailin Zhou, *Graduate Student Member, IEEE*, Zehong Lin, *Member, IEEE*, Jun Zhang, *Fellow, IEEE*,
and Danny H.K. Tsang, *Life Fellow, IEEE*

Abstract—Model averaging is a widely adopted technique in federated learning (FL) that aggregates multiple client models to obtain a global model. Remarkably, model averaging in FL yields a superior global model, even when client models are trained with non-convex objective functions and on heterogeneous local datasets. However, the rationale behind its success remains poorly understood. To shed light on this issue, we first visualize the loss landscape of FL over client and global models to illustrate their geometric properties. The visualization shows that the client models encompass the global model within a common basin, and interestingly, the global model may deviate from the basin's center while still outperforming the client models. To gain further insights into model averaging in FL, we decompose the expected loss of the global model into five factors related to the client models. Specifically, our analysis reveals that the global model loss after early training mainly arises from *i*) the client model's loss on non-overlapping data between client datasets and the global dataset and *ii*) the maximum distance between the global and client models. Based on the findings from our loss landscape visualization and loss decomposition, we propose utilizing iterative moving averaging (IMA) on the global model at the late training phase to reduce its deviation from the expected minimum, while constraining client exploration to limit the maximum distance between the global and client models. Our experiments demonstrate that incorporating IMA into existing FL methods significantly improves their accuracy and training speed on various heterogeneous data setups of benchmark datasets. Code is available at <https://github.com/TailinZhou/FedIMA>.

Index Terms—Federated learning, model averaging, heterogeneous data, loss landscape visualization, loss decomposition.

1 INTRODUCTION

FEDERATED learning (FL) [1] enables clients to collaboratively train a machine learning model while keeping their data decentralized to protect privacy. One of the primary challenges in FL is the heterogeneous data across clients, which diverges client models and deteriorates the performance of FL [2]. Despite this challenge, numerous works have effectively integrated FL into the artificial intelligence (AI) services of large-scale networks with enormous data to ensure the smooth operation of these networks, including the Internet of Things (IoT) [3], [4], wireless networks [5], [6], mobile networks [7], [8] and vehicular networks [9]. The FL empirical success suggests that FL may surpass its theoretical expectations [10].

A common view of the empirical success is that federated model averaging (FMA) mitigates the effect of heterogeneous data in FL, as per [11]. Model averaging, first introduced in

[12], is a widely used technique to reduce communication overhead [13] and the variance of gradients [14] in distributed/decentralized learning [15] by periodically averaging models trained over parallel workers with homogeneous data. In this work, we refer to model averaging in FL on heterogeneous data as FMA to distinguish it from model averaging in other communities (e.g., distributed learning on homogeneous data). Specifically, at each round, FMA aggregates K client models updated locally on heterogeneous data to obtain a global model \mathbf{w} as $\mathbf{w} \leftarrow \sum_{k=1}^K (n_k / \sum_k n_k) \mathbf{w}_k$, where \mathbf{w}_k is the k -th client model and n_k is the size of the k -th client dataset. Surprisingly, FMA can work with divergent client models and alleviate their impact on FL [10].

However, it remains unclear how FMA mitigates the effect of divergent client models and enables the global model to converge throughout the training process. Existing works, e.g., [16], [17], analyze the convergence rate of FMA-based FL under the assumption of bounded gradient dissimilarity. Specifically, these analyses use an assumed upper bound on the distance between the global and client gradients to ensure a theoretical convergence rate of FL. Nevertheless, the assumed bound omits the correlations (i.e., covariance) across clients. A recent work [18] demonstrates that the gradient dissimilarity can be arbitrarily large, and the data heterogeneity has no negative impact when ensuring convergence by making this assumption. Meanwhile, the actual drift of the global gradient (i.e., average client gradients) is significantly smaller than expected based on this bound. This indicates that the bounded gradient dissimilarity cannot accurately characterize the effect of heterogeneous data on FL. Since the bound neglects the overall relationship among all client

This work was supported in part by the Hong Kong Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R, in part by NSFC/RGC Collaborative Research Scheme grant CRS_HKUST603/22, in part by Guangzhou Municipal Science and Technology Project under Grant 2023A03J0011, in part by Guangdong Provincial Key Laboratory of Integrated Communications, Sensing and Computation for Ubiquitous Internet of Things, and in part by National Foreign Expert Project under Project Number G2022030026.

T. Zhou is with IPO, Academy of Interdisciplinary Studies, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR, China (Email: tzhouaq@connect.ust.hk). Z. Lin and J. Zhang are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR, China (E-mail: {eezhlin, eejzhang}@ust.hk). D. H.K. Tsang is with the Internet of Things Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China, and also with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR, China (Email: eetsang@ust.hk). (Corresponding author: Zehong Lin.)

gradients, the effect of FMA on FL is ignored, while FMA plays a practical role in alleviating the drift of the global gradient [11]. Therefore, a conclusive explanation of how FMA assists FL is still lacking.

To fill this gap, we first investigate the geometric properties of FMA by visualizing the loss/error landscape based on the global and client models. Our investigation reveals that the global model is closely surrounded by client models within a common basin and consistently achieves lower test loss and error. Then, we decompose the expected global model loss to establish a connection between the global model's loss and the client models' losses. Based on this connection, we analyze how the global model loss is affected by five factors throughout the training process: training bias, heterogeneous bias, model-prediction variance, covariance between client models, and locality. Our visualization and decomposition demonstrate that FMA may deviate the global model from the expected center of the loss basin when facing heterogeneous data. To mitigate the deviation, we employ iterative moving averaging (IMA) on global models along their optimization trajectory. By integrating IMA into various FL methods, we can effectively reduce the global model's deviation and keep the model in a low-loss region, thereby improving the training performance.

This work aims to unravel the underlying mechanism of FMA and improve it based on the properties of loss landscape and loss decomposition in FL on heterogeneous data. Unlike previous works on loss landscapes with homogeneous data, this study focuses on unraveling the landscape on FMA with heterogeneous data. Meanwhile, compared to previous loss decomposition, our analysis decomposes the FMA model loss on heterogeneous data into five factors instead of two. This comprehensive decomposition quantifies the performance of the FMA model by analyzing the performance of client models on their respective local datasets. It is worth noting that our work aims to decompose the global model loss to evaluate the impact of FMA on data heterogeneity in FL, in contrast to [18], which focuses on computing the average client gradient drift at the optimum to ensure a tighter FL convergence rate. Our main contributions are summarized as follows:

- We investigate the dynamics of test loss and classification error landscapes over the global and client models. Through these landscape visualizations, we observe that while achieving lower loss/error than client models, the global model is closely surrounded by client models in a common basin but may deviate from its lowest point.
- We decompose the global model loss by analyzing the bias and variance of client models on the global dataset. We demonstrate that after early training, the global model loss is dominated by the losses of client models on non-overlapping data between their datasets and the global dataset, as well as their maximum distance from the global model.
- Our loss visualization and decomposition indicate that FMA may shift the global model away from the expected point. We introduce IMA on global models and decay client exploration in late training stages to mitigate this deviation.

- Our experiments show that IMA improves the performance of existing FL methods on various benchmark datasets, enhancing model accuracy and reducing communication costs.

The remainder of this paper is organized as follows. Section 2 reviews related works to ours. Section 3 introduces preliminaries on FL and loss landscape visualization. The loss landscape of FMA is visualized in Section 4, and we present our theoretical and empirical analysis of FMA in Section 5. Section 6 outlines our proposed method for improving FMA, while simulation results are presented in Section 7. Finally, the concluding remarks are presented in Section 8.

2 RELATED WORKS

2.1 Model Averaging

Model averaging in machine learning (ML) is a technique developed to reduce the variance of model updating by periodically averaging models trained over multiple rounds. It was first introduced to average models along the training trajectory in centralized training [12], and then widely adopted to average models over parallel workers in distributed learning [13], [14], [19]. Izmailov et al. [20] discover that a converged ML model tends to end up at the boundary rather than the center of its loss basin while maintaining low loss. To encourage convergence to the basin center, they propose stochastic weight averaging (SWA) to average model weights along the optimization path in the final stage. Notably, SWA does not reinitialize training with the averaged model, thus preserving the optimization trajectory. This approach has been extended to distributed learning [21] and FL [22]. Furthermore, maintaining models with mild diversity in the model ensembling and model average [23], [24] has improved the model generalization.

A comprehensive survey [10] indicates that although FMA has achieved empirical success in FL, its underlying mechanisms remain unclear. Unlike traditional model averaging on homogeneous data, FMA needs to accommodate the challenges posed by heterogeneous data in FL [1]. Notably, despite clients optimizing non-convex ML objectives on heterogeneous local datasets, FMA consistently achieves a converged global model by aggregating divergent client models. Therefore, to understand the mechanism of FMA, we begin by analyzing its geometric properties on heterogeneous data through loss landscape visualization, followed by the decomposition of the expected loss.

2.2 FL on Heterogeneous Data

Heterogeneous data across clients is one of the primary challenges in FL [2]. Common solutions involve improving the local training on clients or modifying model aggregation on the server. Client-side methods typically introduce regularization to local loss functions to prevent local models from converging to their local minima instead of the FL minima. For example, regularization can be designed as the distance between client and global models in FedProx [25], the distance between feature anchors and features in FedFA [26], or the distance among client-invariant features in FedCiR [27]. These approaches aim to handle heterogeneous data on the client side, but they do not improve FMA. On

the other hand, server-side methods develop alternative aggregation schemes building upon FMA. For instance, before performing FMA at the server, FedNova [28] normalizes local updates to mitigate the impact of varying numbers of local updates, FedAdam and FedYogi [29] introduce adaptive momentum to mitigate updating oscillation of the global model, and FedGMA [30] applies the AND-Masked gradient update to sparsify the global model and improve the loss flatness. Moreover, some methods allow clients to tackle heterogeneous data by sharing data with privacy guarantees, such as sharing synthesized [31] or coded data [32], [33]. In addition to addressing heterogeneous data, several studies have explored techniques, including specification [34], quantization [35], and low-rank decomposition [36], to reduce the communication overhead from a global model perspective.

While improving FL performance on heterogeneous data, analyses of existing works like [17], [25], [28], [37] mainly focus on the overall convergence of their proposed methods, rather than elucidating the success of FMA. A common assumption of these analyses is the bounded dissimilarity of client gradients [16], [28], but this assumption is overly pessimistic, as indicated in [18]. It fails to characterize the practical drift of the global model, which is much smaller after FMA on client updates than the theoretical expectation. As suggested in [11], FMA may maintain the drift close to zero on heterogeneous data, though the underlying rationale remains unclear. To fill this gap, we investigate how FMA achieves success and how to improve it during the training.

2.3 Loss Landscape Analysis

Loss landscape [38] analysis refers to the visualization and understanding of the optimization landscape of a model's loss function. It is a common approach to provide insights into models' convergence, generalization, and geometric properties. The loss landscape is typically visualized by plotting the loss function through low-dimensional projections along random or meaningful directions in the parameter space [20], [39], [40]. In [39], Goodfellow et al. take the first step to visualize the optimization trajectory of ML models using low-dimensional projections, enabling comparison of different optimization algorithms. In [40], sharp and flat minima concepts are introduced, where flat minima generally yield better generalization. Subsequently, Izmailov et al. [20] leverage landscape visualization to show that a converged ML model tends to end up at the boundary of the wide flat region (i.e., a loss basin) instead of its flatter center. In addition, sharpness aware minimization (SAM) is introduced to seek flat minima [41] and extended to domain generalization [42] and FL [22]. Regarding geometric structure, Garipov et al. [43] discover that different minima in ML models have a connected structure called mode connectivity despite facing non-convex challenges. This implies that local minima are not isolated but interconnected within a manifold [44].

While loss landscape visualization has been extensively studied, previous research has primarily focused on centralized training with homogeneous data. For instance, the researchers of [20], [38] only consider data shuffling, where the data distribution remains consistent across all workers. In distributed learning, which is more similar to FL, some studies like [14] have explored when model averaging helps

distributed training, suggesting that model averaging brings models of different workers to a common basin of attraction. However, these studies do not provide further visualization analysis and only consider the workers' data independently drawn from the same data pool, i.e., homogeneous data.

In contrast, our work specifically addresses FL scenarios, which have received comparatively less attention in terms of loss landscape visualization. A few preliminary studies [22], [45], [46], [47] have attempted to visualize the loss landscape and improve FL performance by enhancing loss flatness. Nonetheless, these studies have not directly analyzed how FMA handles data heterogeneity for FL, nor have they explored the bias introduced by the global model in the landscape. In contrast, our work stands out by delving into the geometric properties of FMA to understand its underlying mechanism and provide a clear visualization of its loss landscape. Specifically, our work aims to fill this gap by investigating how FMA enables the convergence of the global model aggregated by client models trained on heterogeneous data, despite using a non-convex objective function. More importantly, we demonstrate a novel finding that the global model may deviate from the expected point when using FMA. To the best of our knowledge, our study is the first to identify the deviation of global models from the basin's center on the landscape.

2.4 Bias-variance Loss Decomposition

Bias-variance loss decomposition is a helpful concept for understanding the performance of ML models [48], [49], [50], [51]. Specifically, the expected model loss is decomposed into bias, variance, and irreducible error components. Bias quantifies the fitting capability of models on the training data, while variance reflects the models' sensitivity to small fluctuations in training data [48]. In [50], a unified bias-variance decomposition framework for regression and classification tasks is proposed to guide model selection in the model ensembling. Belkin et al. [52] study how deep ML models achieve low bias and variance via this decomposition.

Compared to previous bias-variance decomposition works that focus on homogeneous data, our analysis delves into heterogeneous data and further decomposes the bias and variance into five factors: training bias, heterogeneous bias, variance, covariance, and locality. This novel decomposition aims to elucidate the underlying mechanism of the global model on heterogeneous data. Specifically, we decompose the bias factor into training bias and heterogeneous bias, and re-derive the variance as the variance and covariance factors among client models that are not independent and identically distributed. In addition, we introduce a locality factor to control the effectiveness of our decomposition by measuring the maximal distance between the client and global models. This comprehensive decomposition enables us to identify the main factors that affect the global model when FL performs FMA on heterogeneous data.

Moreover, it is important to note that our decomposition differs from bias/variance-reduction optimization techniques, such as FedADAM/FedYogi [29] and Scaffold [37], which aim to control the bias/variance of gradient updates to accelerate convergence. Our primary focus is to characterize the performance of the global model on the

global dataset by decomposing the performance of client models on their respective local datasets, which supports our geometric observation of the loss landscape on FMA. The different motivation behind the decomposition allows our proposed IMA approach to complement and enhance these variance-reducing methods rather than conflict with them.

3 PRELIMINARIES

3.1 Federated Learning (FL)

3.1.1 FL Problem Formulation

We consider an FL framework with K clients, each possessing its dataset $\mathcal{D}_k = \{(x, y)\} \sim \mathcal{P}_k$ consisting of n_k data samples, where x and y denote a labeled data sample and its corresponding label, respectively. The global dataset of FL is the union of all client datasets and denoted by $\mathcal{D} = \bigcup_{k=1}^K \mathcal{D}_k \sim \mathcal{P}$, comprising $n = \sum_{k=1}^K n_k$ data samples. Here, \mathcal{P}_k and \mathcal{P} represent the client and global data distribution, respectively. When dealing with a ML task on the global dataset \mathcal{D} , FL uses a finite-sum objective to minimize the expected global loss $\mathcal{L}(\mathbf{w}) := \mathbb{E}_{(x,y) \in \mathcal{D}} [l(\mathbf{w}; (x, y))]$, where $l(\mathbf{w})$ denotes the global loss function for model parameters \mathbf{w} . As shown in [1], this objective can be reformulated as:

$$\min_{\mathbf{w} \in \mathbb{R}^n} \mathcal{L}(\mathbf{w}) = \sum_{k=1}^K \frac{n_k}{n} \mathcal{L}_k(\mathbf{w}) = \sum_{k=1}^K \frac{n_k}{n} \sum_{i=1}^{n_k} l_k(\mathbf{w}; (x_i, y_i) \in \mathcal{D}_k), \quad (1)$$

where $\mathcal{L}_k(\cdot)$ is the expected local loss of the k -th client on its local dataset \mathcal{D}_k , and $l_k(\mathbf{w})$ is the local loss function on \mathbf{w} .

An FL method called FedAvg [1] optimizes the objective (1) by averaging client models at the server in a periodic manner. In each round, the method has the following steps:

- 1) Clients update their local models $\{\mathbf{w}_k\}_{k=1}^K$ independently by minimizing their local losses $\{\mathcal{L}_k(\mathbf{w}_k)\}_{k=1}^K$ on the local datasets $\{\mathcal{D}_k\}_{k=1}^K$;
- 2) Clients upload their updated models to the server;
- 3) The server performs FMA on the local models to calculate the new global model, i.e., $\mathbf{w} = \sum_{k=1}^K \frac{n_k}{n} \mathbf{w}_k$;
- 4) The new global model \mathbf{w} is sent back to clients to initialize the next round of local training.

This process repeats until the global model converges.

3.1.2 Heterogeneous Data Problem and FMA in FL

The objective (1) assumes that the client data distribution \mathcal{P}_k is formed by uniformly and randomly distributing the training examples from the global data distribution \mathcal{P} . However, the assumption does not generally hold in FL due to heterogeneous data among clients, where $\mathcal{P}_k \neq \mathcal{P}_{k'} \neq \mathcal{P}$ when $k \neq k'$. As per [2], [25], [28], FL performance can be negatively impacted by heterogeneous data, leading to a slower convergence speed and worse model generalization.

There are two common types of heterogeneous data [10]: feature distribution skew and label distribution skew, referred to as *feature skew* and *label skew*, respectively, in this work for brevity. Our work delves into the effect of these two types of heterogeneous data on FMA in FL. Suppose that the k -th client data distribution follows $\mathcal{P}_k(x, y) = \mathcal{P}_k(x|y)\mathcal{P}_k(y) = \mathcal{P}_k(y|x)\mathcal{P}_k(x)$, where $\mathcal{P}_k(x)$ and $\mathcal{P}_k(y)$ denote the input feature marginal distribution

and label marginal distribution of the k -th client, respectively. Specifically, label skew means that $\mathcal{P}_k(y)$ varies from $\mathcal{P}_{k'}(y)$ while $\mathcal{P}_k(x|y) = \mathcal{P}_{k'}(x|y)$ for clients $k \neq k'$; feature skew means that $\mathcal{P}_k(x)$ varies from $\mathcal{P}_{k'}(x)$ while $\mathcal{P}_k(y|x) = \mathcal{P}_{k'}(y|x)$ for clients $k \neq k'$.

In FL, when optimizing the objective (1) on heterogeneous data, \mathcal{L}_k can be an arbitrarily poor approximation to \mathcal{L} [1], e.g., an inconsistent local objective with the FL objective, potentially hindering the FL convergence. Nonetheless, FL typically outperforms its theoretical convergence expectation despite data heterogeneity [18]. For example, FedAvg shows empirical success as per [10], with FMA keeping the global model converging throughout the training process. A recent survey [53] discovers that FMA effectively balances sharing information among clients while preserving privacy. This highlights the crucial role of FMA in FedAvg, while it remains unclear how FMA deals with heterogeneous data on FL.

3.2 Loss Landscape Visualization

The loss landscape depicts the distribution of loss values throughout the model's weight space. As per [38], exploring the loss landscape can enhance our understanding of ML problems. While it is generally difficult to visualize the landscape in high-dimensional spaces, there have been many attempts to achieve it by dimensionality reduction. This helps reveal the geometric properties of neural networks, such as flatness [43] and optimization trajectory [39]. In this work, we employ two common approaches to visualize the loss landscape of FL: 1D and 2D visualizations.

For 1D visualization, we follow [39] to draw the loss landscape in a line segment (1D) by using linear interpolation between two models \mathbf{w}_1 and \mathbf{w}_2 . Specifically, given a target dataset, we evaluate the loss of different model weights along the line segment between \mathbf{w}_1 and \mathbf{w}_2 , i.e., $\mathcal{L}_{[\mathbf{w}_1, \mathbf{w}_2]}(\beta) = \mathcal{L}(\beta\mathbf{w}_1 + (1 - \beta)\mathbf{w}_2)$, where β is the interpolation coefficient of line model interpolation between \mathbf{w}_1 and \mathbf{w}_2 .

For 2D visualization, we explore the loss landscape in a plane (2D) by drawing the contour based on three models according to [20]. Specifically, we take $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ to form a plane by constructing two base vectors $\bar{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|^2$ and $\bar{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|^2$, where $\mathbf{u} = \mathbf{w}_2 - \mathbf{w}_1$ and $\mathbf{v} = (\mathbf{w}_3 - \mathbf{w}_1) - \langle \mathbf{w}_3 - \mathbf{w}_1, \mathbf{w}_2 - \mathbf{w}_1 \rangle / \|\mathbf{w}_2 - \mathbf{w}_1\|^2 \cdot (\mathbf{w}_2 - \mathbf{w}_1)$. Next, each plane point P , with coordinates (a, b) , represents a model $\mathbf{w}_P = \mathbf{w}_1 + a \cdot \bar{\mathbf{u}} + b \cdot \bar{\mathbf{v}}$. Finally, given a dataset, we evaluate the loss $\mathcal{L}(\mathbf{w}_P)$ of all the points in this plane and draw the loss contour by $\{\mathcal{L}(\mathbf{w}_P) = c\}$ with a contour value c .

4 LOSS LANDSCAPE VISUALIZATION IN FL

In this section, we explore the geometric properties of FMA through 2D loss landscape visualization. As depicted in Figure 1, to construct the plane of 2D visualization, we use three client models from the same training round in the first three columns and three global models from different training rounds in the fourth column. The FL setup related to Figure 1 involves training a global model on the CIFAR-10 dataset [54] across 100 clients over 400 rounds. To introduce data heterogeneity, each client dataset contains two class shards of CIFAR-10, following [1] (see specific setups in Table 6). In this setup, the ideal accuracy of client models on the

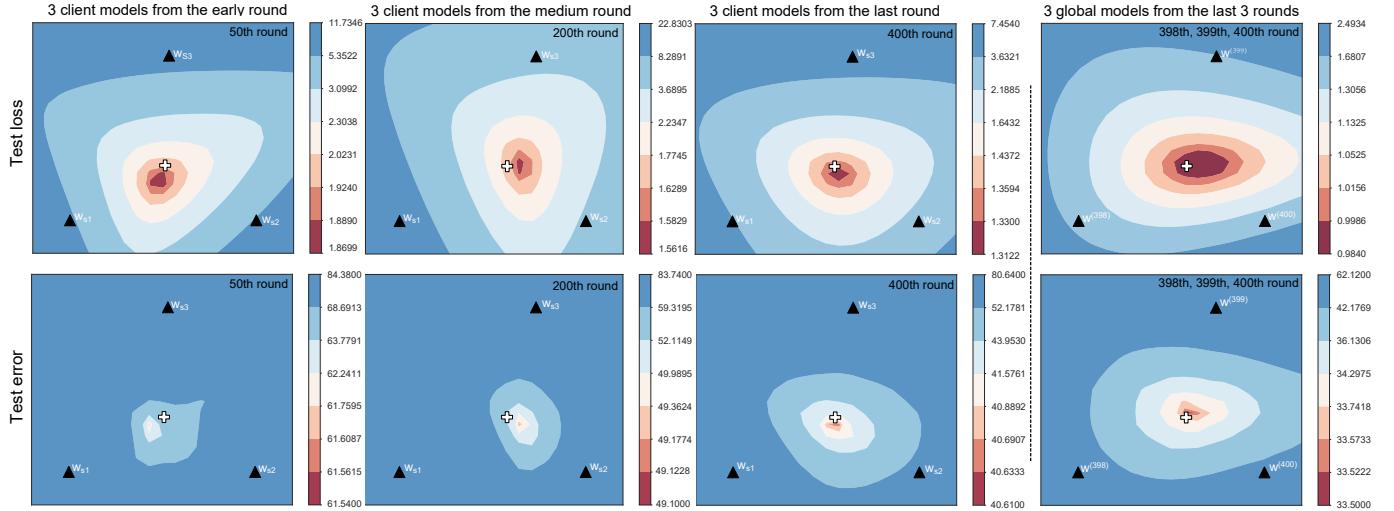


Figure 1: Visualization of the loss (**top row**) and classification error (**bottom row**) landscapes on the CIFAR-10 test dataset, along with three client models from the early stage (**first column**), middle stage (**second column**), and final stage (**third column**), as well as the visualization of three global models from the final three rounds (**fourth column**). The *black triangles* represent the location of three models in the plane, while the *white cross* represents their average model's location. The loss/error landscape can be viewed as a basin, where client models reach the basin's wall and the global model approaches the basin's bottom as FL training proceeds. FMA helps move the global model towards the basin's bottom by averaging client models on the basin's wall, while heterogeneous data deviates the global model from the basin's center.

CIFAR-10 test set is 20% if clients train their models by local datasets. More results for various setups, including different datasets, data heterogeneity, models, and FL settings, are provided in Figures 8, 9, and 10 of the Appendix.

4.1 Lower Test Loss with FMA

In Figure 1, we observe that the averaged model (i.e., the *white cross*) of the three client models (i.e., the *black triangles*) is consistently located at the regions with lower test loss and classification error than individual client models. This implies that FMA can effectively aggregate local client information into the global model. Furthermore, the first three columns of Figure 1 correspond to three different training stages. As training progresses, since the newly-aggregated global model re-initiates client models with lower loss, FMA prevents them from over-fitting to their respective datasets. Meanwhile, FMA leverages client models with lower losses to anchor the global model more precisely in a lower-loss landscape area. In other words, FMA prevents over-fitting information from aggregating into the global model.

Moreover, we observe a bias between the *white cross* and the lowest loss/error point in the first three columns of Figure 1. This bias can be caused by the deviation between the averaged models (i.e., the *white cross*) and the global model or between the global model and its optimal model. To further investigate this bias, we visualize the loss/error landscape of global models obtained from the final three rounds (i.e., the 398th, 399th, and 400th rounds) in the fourth column of Figure 1. There is a bias between the global models (i.e., the *black triangle*) and the lowest loss point in the loss landscape, similar to the bias observed over client models, and their averaged model (i.e., the *white cross*) is closer to the lowest point. This reveals that the only performing FMA on client models may fail to achieve the optimal global model.

In summary, FMA helps move the global model towards the center of the loss basin during the FL training process. However, while the global model is converging, the presence of heterogeneous data causes the global model's movement to deviate from the basin's center. In Section 6, we will address the deviation of the global model aggregated by FMA from the lowest loss point.

4.2 Global Model and Client Models in a Common Basin

In Figure 1, the second row demonstrates that the test classification errors of client models are around 80%. These errors almost reach the lowest classification error obtained by client models through local training, indicating their proximity to local optima. Moreover, Figure 1 illustrates that the averaged model is surrounded by client models and located near a local optimum of the global model throughout the entire training process. Meanwhile, the distance between global and client models remains limited, as presented in Figure 3. These observations suggest that client models within a common basin closely surround the global model.

Geometrically, the test loss/error landscape in FL can be viewed as a basin, with client models reaching the basin's wall and the global model near the basin's bottom, as shown in Figure 1. This geometric property provides a novel insight into the mechanism behind FMA in FL. For example, Wang et al. [18] have empirically found that the client-update drifts' practical impact on the global model's convergence speed is less than predicted by theoretical analysis. This observation can be explained by the geometric property as follows: In the earlier stages of training, clients update their models towards the basin, resulting in client updates that are roughly in the same direction but not identical. Consequently, the client-update drifts are small in the earlier stages. However, as client models approach their optimal points in the later

stages, they encounter the basin walls in the loss landscape. That is, clients' optimal points are scattered around the basin's wall. These client models are then initialized by the global model, which is near the basin's bottom, and the direction of client updates may radiate in all directions, from the basin's bottom to the wall. Fortunately, when FL performs FMA on the drifts of client updates, these drifts tend to cancel each other out, resulting in a limited impact on the global model and preventing it from drifting away from the basin's bottom. Therefore, the client-update drifts remain small even in the later stages of training, although their updated directions may be more dissimilar compared to earlier stages.

5 EXPECTED LOSS DECOMPOSITION

In this section, we will analyze the relationship of the losses between the global and client models when FL performs FMA. To decompose the expected global model loss, we first examine the connection between FMA and the weighted-model ensembling (WENS). Next, we decompose the global model's expected loss using this connection based on the client models' losses. Finally, we empirically validate our decomposition analysis to show which factors dominate the global model's loss throughout the training process.

We represent the forward function of \mathbf{w} as $f_{\mathbf{w}} : \mathcal{X} \rightarrow \mathcal{Y}$, where \mathcal{X} and \mathcal{Y} are input and output spaces, respectively. For simplicity, we focus on the mean-square error (MSE) loss in the theoretical analysis, i.e., $l(\mathbf{w}; (x, y)) = (y - f_{\mathbf{w}}(x))^2$. It is worth noting that this framework can be extended to other loss functions [50]. Due to mode connectivity of neural networks [20], [44], [47], given a model architecture $\mathcal{W} \subset \mathbb{R}^d$, a loss function \mathcal{L} , and a training dataset \mathcal{D}_{tr} , there exists a single connected low-loss manifold that contains all the minima trained on \mathcal{D}_{tr} . In other words, there exists a model solution subspace $\mathcal{W}_{\mathcal{D}_{\text{tr}}} = \{\mathbf{w}_{\text{tr}}\} \subset \mathcal{W}$, where \mathbf{w}_{tr} denotes a model optimized on \mathcal{D}_{tr} . When the model \mathbf{w} is uniformly distributed in $\mathcal{W}_{\mathcal{D}_{\text{tr}}}$, the bias-variance decomposition of the expected loss of \mathbf{w} evaluated on a test dataset \mathcal{D}_{te} can be expressed as [49], [51]:

$$\begin{aligned} \mathbb{E}_{\mathbf{w} \in \mathcal{W}_{\mathcal{D}_{\text{tr}}}} \mathcal{L}_{\mathcal{D}_{\text{te}}}(\mathbf{w}) &= \mathbb{E}_{(x, y) \in \mathcal{D}_{\text{te}}} [\underbrace{(y - \bar{f}_{\mathcal{D}_{\text{tr}}}(x))^2}_{\text{Bias}\{f|(x, y)\}} \\ &\quad + \underbrace{\mathbb{E}_{\mathbf{w} \in \mathcal{W}_{\mathcal{D}_{\text{tr}}}} [(f_{\mathbf{w}}(x) - \bar{f}_{\mathcal{D}_{\text{tr}}}(x))^2]}_{\text{Var}\{f|x\}}], \end{aligned} \quad (2)$$

where $f_{\mathbf{w}}(\cdot)$ and $\bar{f}_{\mathcal{D}_{\text{tr}}}(\cdot) = \mathbb{E}_{\mathbf{w} \in \mathcal{W}_{\mathcal{D}_{\text{tr}}}} [f_{\mathbf{w}}(\cdot)]$ are the model output of \mathbf{w} and the expected output on $\mathcal{W}_{\mathcal{D}_{\text{tr}}}$, respectively.

Since $\bar{f}_{\mathcal{D}_{\text{tr}}}(\cdot)$ represents the ensemble output of all models in $\mathcal{W}_{\mathcal{D}_{\text{tr}}}$, we rewrite it as a finite-sum formulation, $\bar{f}_{\mathcal{D}_{\text{tr}}}(\cdot) = \frac{1}{N} \sum_{i \in [N]} f_{\mathbf{w}_i \in \mathcal{W}_{\mathcal{D}_{\text{tr}}}}(\cdot)$. Specifically, given $\mathcal{W}_{\mathcal{D}_{\text{tr}}}$ and a sample $(x, y) \in \mathcal{D}_{\text{te}}$, $\text{Bias}\{f|(x, y)\}$ denotes the bias between the ground truth y and the ensemble output $\bar{f}_{\mathcal{D}_{\text{tr}}}(x)$ and $\text{Var}\{f|x\}$ denotes the expected MSE between $f_{\mathbf{w}}(x)$ and $\bar{f}_{\mathcal{D}_{\text{tr}}}(x)$, which depends on the discrepancy between \mathcal{D}_{tr} and \mathcal{D}_{te} according to [55]. Note that the bias captures the capability of the models to fit the training data distribution \mathcal{D}_{tr} , while the variance measures the models' sensitivity to small fluctuations in \mathcal{D}_{tr} .

5.1 Connection between FMA and WENS

At each round, FMA performs weighted averaging, defined as $\mathbf{w}_{\text{FMA}} \leftarrow \sum_{k=1}^K \frac{n_k}{n} \mathbf{w}_k$, where the averaging weight depends on n_k [1] and $n = \sum_k n_k$. According to [20], the model average is a first-order approximation of the model ensembling when the averaged models are closely located in the weight space, where the model ensembling represents the averaging of outputs from multiple diverse models given the same input. Based on this approximation, we establish the connection between FMA and WENS as follows:

Lemma 1. (*FMA and WENS. See proof in Appendix*) Given K models $\{\mathbf{w}_k\}_{k=1}^K$ and $n_i/n_j \neq \infty$ when $i \neq j$, we denote $\Delta_k = \|\mathbf{w}_k - \mathbf{w}_{\text{FMA}}\|$ and $\Delta = \max_{k=1}^K \Delta_k$. Then, we have:

$$f_{\text{WENS}}(x) - f_{\mathbf{w}_{\text{FMA}}}(x) = \langle \Delta f_{\mathbf{w}_{\text{FMA}}}(x), \sum_{k=1}^K \frac{n_k}{n} \Delta_k \rangle + O(\Delta^2),$$

where the WENS on the K models is to conduct weighted averaging on the outputs of these models when given the same input, represented as $f_{\text{WENS}}(x) = \sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}(x)$.

Lemma 1 shows that the output of the FMA model $f_{\text{FMA}}(\cdot)$, i.e., the global model in FL, is a first-order approximation of weighted averaging on the outputs of client models, i.e., the WENS $f_{\text{WENS}}(\cdot)$. The term $O(\Delta^2)$ measures the quadratic of the maximum distance between the client and global models and controls the approximation error. With a limited maximum distance, the approximation error is expected to be small, which will be verified in Figure 3. Note that WENS involves averaging model outputs, while FMA involves averaging model parameters. The connection between FMA and WEN enables us to conduct a bias-variance decomposition on FMA using equation (2), which relates to model outputs.

5.2 Expected Loss Decomposition of Global Model

With Lemma 1, we can incorporate FMA and adapt the bias-variance decomposition (2) to the FL version. Specifically, the model \mathbf{w} in (2) is substituted by K client models $\{\mathbf{w}_k\}_{k=1}^K$, \mathcal{D}_{tr} and \mathcal{D}_{te} are modified to client datasets $\{\mathcal{D}_k\}_{k=1}^K$ and the global dataset \mathcal{D} , respectively. Meanwhile, $\bar{f}_{\mathcal{D}_{\text{tr}}}(\cdot) = \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K \in \prod_k^K \mathcal{W}_{\mathcal{D}_k}} [f_{\{\mathbf{w}_k\}_{k=1}^K}(\cdot)] = \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} [f_{\text{WENS}}(\cdot)]$ denotes the ensemble output of the combination subspace on K client models, where $\prod_k^K \mathcal{W}_{\mathcal{D}_k} = \mathcal{W}_{\mathcal{D}_1} \times \cdots \times \mathcal{W}_{\mathcal{D}_K}$. Then, we decompose the expected loss of $f_{\mathbf{w}_{\text{FMA}}}(\cdot)$ on \mathcal{D} in the following theorem:

Theorem 1. (*Loss decomposition of the global model. See proof in Appendix*) Given K client models $\{\mathbf{w}_k\}_{k=1}^K \in \prod_k^K \mathcal{W}_{\mathcal{D}_k}$, the expected loss of the global model \mathbf{w}_{FMA} on \mathcal{D} is decomposed as:

$$\begin{aligned} \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} \mathcal{L}(\mathbf{w}_{\text{FMA}}) &= \frac{1}{n} \sum_{(x, y) \in \mathcal{D}} [\sum_{k=1}^K \frac{n_k}{n} \text{TrainBias}\{f_{\mathbf{w}_k}|(x, y)\} \\ &\quad + \frac{n_k}{n} \text{HeterBias}\{f_{\mathbf{w}_k}|(x, y)\}]^2 + \sum_{k=1}^K \frac{n_k^2}{n^2} \text{Var}\{f_{\mathbf{w}_k}|x\} \\ &\quad + \sum_k \sum_{k' \neq k} \frac{n_k n_{k'}}{n^2} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\} + O(\Delta^2), \end{aligned}$$

where $\text{TrainBias}\{f_{\mathbf{w}_k}|(x, y)\} = \mathbb{I}[(x, y) \in \mathcal{D}_k] (y - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)])$; $\text{HeterBias}\{f_{\mathbf{w}_k}|(x, y)\} = \mathbb{I}[(x, y) \in \mathcal{D} \setminus \mathcal{D}_k]$

$$(y - \mathbb{E}_{\mathbf{w}_k}[f_{\mathbf{w}_k}(x)]); \text{Var}\{f_{\mathbf{w}_k}|x\} = \mathbb{E}_{\mathbf{w}_k}[(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k}[f_{\mathbf{w}_k}(x)])^2]; \text{Cov}\{f_{\mathbf{w}_k}, f'_{\mathbf{w}_k}|x\} = \mathbb{E}_{\mathbf{w}_k, \mathbf{w}'_k}[(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k}[f_{\mathbf{w}_k}(x)])(f'_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}'_k}[f'_{\mathbf{w}_k}(x)])].$$

In Theorem 1, the underlying meanings of the five factors are elaborated as follows:

- $\text{TrainBias}\{f_{\mathbf{w}_k}|(x, y)\}$ measures the fitting capability of a client model \mathbf{w}_k on the samples of client dataset (i.e., $(x, y) \in \mathcal{D}_k$), where \mathbf{w}_k is trained on \mathcal{D}_k ;
- $\text{HeterBias}\{f_{\mathbf{w}_k}|(x, y)\}$ measures the degree of catastrophic forgetting of a client model \mathbf{w}_k on the non-overlapping samples between the global dataset and client dataset (i.e., $(x, y) \in \mathcal{D} \setminus \mathcal{D}_k$), where \mathbf{w}_k is trained on \mathcal{D}_k ;
- $\text{Var}\{f_{\mathbf{w}_k}|x\}$ measures the sensitivity of a client model \mathbf{w}_k to small fluctuations in the given sample input $x \in \mathcal{D}_k$, which does not depend on y ;
- $\text{Cov}\{f_{\mathbf{w}_k}, f'_{\mathbf{w}_k}|x\}$ denotes the output correlation between client models \mathbf{w}_k and \mathbf{w}'_k given the same input x , which does not depend on y ;
- $O(\Delta^2)$ represents the locality in [20], [24], i.e., the maximum distance between client and global models.

Based on these five factors, the capability of the global model on the global dataset can be quantified by the capabilities of client models on their local datasets. Due to the presence of unseen samples for the client model \mathbf{w}_k in its local dataset, $\text{HeterBias}\{f_{\mathbf{w}_k}|(x, y)\}$ is expected to have a more significant impact on the global model compared to $\text{TrainBias}\{f_{\mathbf{w}_k}|(x, y)\}$. This implies that client models that are more robust to catastrophic forgetting contribute to a lower loss for their corresponding global model. In other words, HeterBias can effectively measure the performance of the global model on the client side. Moreover, unlike the decomposition in (2), Theorem 1 incorporates a covariance term to account for the fact that client models are not independent and identically distributed within a model solution subspace due to the heterogeneity of the data. In the following, we empirically validate the effect of these factors on global model capability throughout the training.

5.3 Empirical Validation of Decomposition Analysis

The FL setup involves training a global model on CIFAR-10 across ten clients for 400 rounds, where clients hold two class shards of CIFAR-10 for heterogeneous data setup (see specific setups in Table 6).

5.3.1 The effect of bias factor: heterogeneous bias dominates the loss of the global model after the early training

Figure 2 shows that the $\text{TrainBias}\{f_{\mathbf{w}_k}|(x, y)\}$ is effectively reduced to almost zero, which is because the number of local updates is sufficient for client models to fit their datasets. However, heterogeneous data introduce a non-zero $\text{HeterBias}\{f_{\mathbf{w}_k}|(x, y)\}$, which individual client struggles to address through local training due to missing samples from the global dataset. Nonetheless, due to the geometric property observed in Section 4, FMA provides an initialization point with enriched global information for client models to mitigate this bias.

Moreover, the larger the local update step, the more global information the FMA provides is forgotten, and the greater

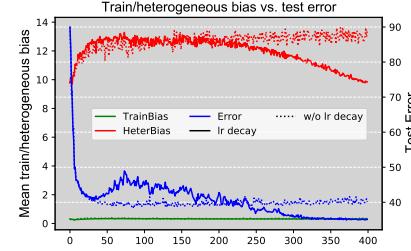


Figure 2: Train and heterogeneous biases w.r.t rounds (x-axis).

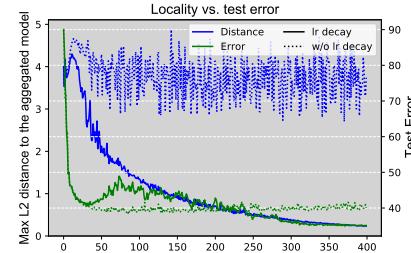


Figure 3: Locality (L2 distance) w.r.t rounds (x-axis).

the heterogeneous bias becomes due to the catastrophic forgetting phenomenon in neural networks [56], [57]. This is validated by the cases with and without learning rate (lr) decay shown in Figure 2. We use a round-exponential decay lr to control the update size, a straightforward approach to preventing catastrophic forgetting in FL [29]. In the early phase, heterogeneous bias does not significantly impact the test classification error because the error continues to decrease even if the bias increases. However, both the error and the bias show a positive correlation in both cases. For example, after approximately 40 rounds, they both increase and decrease in the case of lr decay, and the error grows slightly with the bias in the case without lr decay.

5.3.2 The effect of locality factor: controlling the locality helps reduce the global model loss at the late training

In Figure 3, we employ the L2 distance to quantify the locality term $O(\Delta^2)$. Theorem 1 demonstrates that the test loss decreases as the maximum distance between client models and the global model, i.e., Δ , reduces.

Figure 3 shows that the locality is larger in the case without lr decay, which results in a more significant test error. Before the 40th round, the test classification errors of both cases continue to decline despite an increase in the locality occurring within this period. Then, in the case of lr decay, the locality reduces while the error increases from 40 to 75 rounds. This indicates that the locality does not correlate strongly with the error during the early training. The locality stabilizes after the early training phase (i.e., the locality is upper-bounded in both cases). This further validates the proximity of client models to the global model, as discussed in Section 4.

5.3.3 The effect of variance factor: reducing global model loss by aggregating more client models in FMA is limited

When the client dataset \mathcal{D}_k does not change during the training, the variance factor $\mathbb{E}_{(x,y) \in \mathcal{D}}[\text{Var}\{f_{\mathbf{w}_k}|x\}] = 1/n_k \sum_{(x,y) \in \mathcal{D}_k} \text{Var}\{f_{\mathbf{w}_k}|x\}$ in Theorem 1 can be viewed as

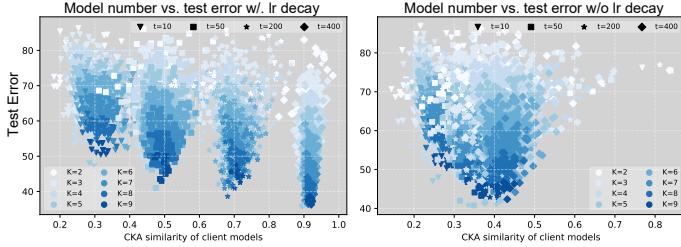


Figure 4: Test error w.r.t model number and model similarity.

a constant V_k since $\text{Var}\{f_{\mathbf{w}_k}|x\}$ depends on the discrepancy between \mathcal{D} and \mathcal{D}_k . Specifically, from Theorem 1 in [55] and Proposition 2 in [24], we have the following property:

Theorem 2. (Bounded variance.) Given a kernel regime $f_{\mathbf{w}_k}$ trained on client dataset \mathcal{D}_k (of size n_k) with neural tangent kernel $K_{f_{\mathbf{w}_k}}$, when $\exists(\lambda_{\mathcal{D}_k}, \epsilon)$ with $0 \leq \epsilon \ll \lambda_{\mathcal{D}_k}$ such that $\forall x_i \in \mathcal{D}_k, K_{f_{\mathbf{w}_k}}(x_i, x_i) = \lambda_{\mathcal{D}_k}$ and $\forall x_i, x_j \in \mathcal{D}_k$ and $i \neq j, |K_f(x_i, x_j)| \leq \epsilon$, the variance on the global dataset \mathcal{D} is:

$$\begin{aligned} \mathbb{E}_{x \in \mathcal{D}}[\text{Var}\{f_{\mathbf{w}_k}|x\}] &= \frac{n_k}{2\lambda_{\mathcal{D}_k}} MMD^2(\mathcal{D}_k, \mathcal{D}) \\ &\quad + \lambda_{\mathcal{D}} - \frac{n_k}{2\lambda_{\mathcal{D}_k}} \beta_{\mathcal{D}} + O(\epsilon), \end{aligned} \quad (3)$$

where $MMD(\cdot)$ is the empirical maximum mean discrepancy in the reproducing kernel Hilbert space (RKHS) of $K_{f_{\mathbf{w}_k}}(x_i, x_j)$; $\lambda_{\mathcal{D}} = \mathbb{E}_{x \in \mathcal{D}} K_{f_{\mathbf{w}_k}}(x, x)$ and $\beta_{\mathcal{D}} = \mathbb{E}_{x_i, x_j \in \mathcal{D}, i \neq j} K_{f_{\mathbf{w}_k}}^2(x_i, x_j)$ denote the empirical mean similarities of identical and different samples averaged over \mathcal{D} , respectively.

In Theorem 2, both $\lambda_{\mathcal{D}}$ and $\beta_{\mathcal{D}}$ depend exclusively on the global dataset \mathcal{D} for a $f_{\mathbf{w}_k}$. The global dataset \mathcal{D} represents the combination of all client datasets and can be viewed as a fixed dataset in FL. Consequently, $\lambda_{\mathcal{D}}$ and $\beta_{\mathcal{D}}$ can be regarded as constants that depend on \mathcal{D} in Theorem 2. Therefore, Theorem 2 demonstrates that the variance term in Theorem 1 is solely associated with $MMD^2(\mathcal{D}_k, \mathcal{D})$, which quantifies the distance between the client dataset \mathcal{D}_k and the global dataset \mathcal{D} in FL setups.

From Theorem 2, we have $\max_k\{V_k\} \leq \epsilon$. The whole variance term in Theorem 1 is $\sum_k n_k^2/n^2 V_k$ and it is upper bounded by $\epsilon \sum_{k=1}^K n_k^2/n^2$. Then, FMA can keep this upper bound diminishing by averaging more client models (i.e., larger K induces smaller $\sum_{k=1}^K n_k^2/n^2$) to reduce the global model loss when ϵ is tight for $\max_k\{V_k\}$. Figure 4 verifies this effect on FMA throughout the training process. However, it is important to note that the impact of the client number becomes negligible when all client models are identically distributed (i.e., client models are trained on homogeneous datasets with the same training configurations). This is because the sum of the third and fourth terms on the right-hand side of Theorem 1 (i.e., $\text{Var}\{f_{\mathbf{w}_k}|x\} = \text{Cov}\{f_{\mathbf{w}_k}, \mathbf{w}_{k'}|x\}$ when client models are identically distributed) is equal to the variance of a single client model.

In summary, the variance term decreases as the number of client models being averaged in FMA increases. Nonetheless, this effect weakens as more models are incorporated.

5.3.4 The effect of covariance factor: heterogeneous data inherently results in a small but lower bounded covariance

To measure the covariance factor $\text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$, we employ the CKA similarity [54] to compute the output correlation among client models given the same input. As shown in Figure 4, heterogeneous data inherently lead to a small covariance term, especially for the case without lr decay. That is, maintaining high diversity among client models (e.g., [23], [24]) may not significantly reduce the loss of the global model. Indeed, it can negatively impact the performance in the late training stage, as illustrated by the comparison between both cases at the 400th round in Figure 4.

Furthermore, we show that the covariance term has a non-zero lower bound that depends on the maximum discrepancy across client datasets. Let $n_i = n_j, \forall i, j \in [K]$ (i.e., the number of client samples is the same). By ablating the impact of weighted averaging, we can further decompose the covariance term in Theorem 1 as follows:

Corollary 1. (Lower bound of the covariance term.) For $n_i = n_j$ when $i \neq j$, the covariance term in Theorem 1 is bounded by:

$$\begin{aligned} \mathbb{E}_{(x,y) \in \mathcal{D}} (\frac{1}{K^2} \sum_k \sum_{k'} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}) \\ = \frac{1}{nK^2} \sum_{(x,y) \in \mathcal{D}} \sum_k \sum_{k'} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\} \\ \geq \frac{K-1}{nK} \sum_{(x,y) \in \mathcal{D}} \min_{(k,k')} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}, \end{aligned} \quad (4)$$

where $\min_{(k,k')} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$ measures the maximum discrepancy among all client models.

The physical meaning of $\min_{(k,k')} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$, when given a sample $(x, y) \in \mathcal{D}$, can be understood as follows: Firstly, $\text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$ calculates the covariance between client models \mathbf{w}_k and $\mathbf{w}_{k'}$. Then, $\min_{(k,k')} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$ finds the minimal value of $\text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$ across all client pairs (k, k') , where $\forall k, k' \in [K], k \neq k'$. This minimal value measures the largest diversity among client models on the given sample (x, y) . The maximum discrepancy across client datasets determines the diversity and remains constant since client datasets do not change in the generic FL setups.

Therefore, Corollary 1 demonstrates that the covariance term has a lower bound that depends on the maximum discrepancy across client datasets. Consequently, the effect of FMA on reducing the loss of the global model by controlling the diversity of client models is limited.

5.3.5 Summary

From the above discussion, we summarize the impact of the five factors in Theorem 1 on the loss of the global model during training as follows:

- TrainBias $\{f_{\mathbf{w}_k}|(x, y)\}$ keeps almost zero throughout the training process;
- HeterBias $\{f_{\mathbf{w}_k}|(x, y)\}$ and $O(\Delta^2)$ dominate the loss of the global model after the early training;
- The weighted sum of $\text{Var}\{f_{\mathbf{w}_k}|x\}$ can be reduced to some extent with a large number of client models;
- $\text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}$ is too small to affect the loss of the global model.

Therefore, FMA can reduce the loss of the global model by controlling HeterBias $\{f_{\mathbf{w}_k} | (x, y)\}$ and locality $O(\Delta^2)$, in addition to aggregating more client models.

6 PROPOSED METHOD

In this section, we will begin by discussing our motivation, i.e., to alleviate the deviation of global models in FMA. After that, we will introduce IMA and mild client exploration to address this deviation. Lastly, we will discuss the advantages of using IMA for FL.

6.1 Motivation: Deviation of Global Models from the Basin's Center

As shown in the fourth column of Figure 1, when performing FMA on heterogeneous data, global models deviate from the basin center of the loss landscape. Specifically, FMA tends to move the global model towards the center of the loss basin. However, due to heterogeneous data, the movement of the global model deviates from the basin's center. This deviation causes global models obtained in different rounds to be scattered around the basin's center, as illustrated in Figure 5(a). This geometric property can be leveraged to improve the aggregation process and bring the global model closer to the basin's center.

The loss decomposition analysis of the global model presented in Theorem 1 provides a new perspective on this geometric property. In FL, a small number of clients participating in each round makes the one-cohort dataset of participating clients $\mathcal{D}_C^{(t)} = \cup_{i=1}^C \mathcal{D}_i$ inconsistent from the global dataset \mathcal{D} . Moreover, the weighted averaging tends to assign higher weights to clients with datasets that are large but imbalanced compared to \mathcal{D} , further exacerbating the inconsistency between $\mathcal{D}_C^{(t)}$ and \mathcal{D} . Consequently, heterogeneous bias cannot be completely reduced since the one-cohort dataset misses data samples $(x, y) \in \mathcal{D}^{(t)} \setminus \mathcal{D}_C$. This leads to the observed deviation of global models from the basin's center.

In contrast, a combination of one-cohort datasets from different rounds, denoted by $\mathcal{D}_{\text{IMA}} = \cup_{i=0}^{P-1} \mathcal{D}_C^{(t-i)}$, contains fewer missing data samples than $\mathcal{D}^{(t)}$ alone. As summarized in Section 5.3.5, reducing heterogeneous bias can decrease the global model's loss. This implies that aggregating historical global models can reduce the heterogeneous bias on missing data samples since the global model $\mathbf{w}^{(t-i)}$ carries the information of $\mathcal{D}_C^{(t-i)}$. To verify this, we linearly interpolate two global models from different rounds and evaluate the performance of the interpolated models on the CIFAR-10 dataset, as depicted in Figure 5(b). The figure demonstrates that lower loss/error points consistently exist within the global models' interpolation. In other words, interpolated models retain more global information than a solo global model while remaining within a common basin, thus reducing the heterogeneous bias. Therefore, we apply Theorem 1 to leverage the geometric properties of FMA to alleviate the deviation of global models.

6.2 Iterative Moving Averaging (IMA)

The missing information of one-cohort client models on $(x, y) \in \mathcal{D} \setminus \mathcal{D}_C$ can be compensated by utilizing historical

global models. This compensation can be achieved by aggregating historical global models into the latest one, as supported by the observation in Figure 5(b). Therefore, we propose applying IMA to historical global models after sufficient training rounds instead of ignoring them in conventional FMA. Specifically, as illustrated in Figure 5(a), after t_s rounds, the server performs FMA with IMA to obtain an averaged model from a time window of previous rounds as:

$$\mathbf{w}_{\text{IMA}}^{(t)} \leftarrow \frac{1}{P} \sum_{i=0}^{P-1} \mathbf{w}^{(t-i)}, \quad t \geq t_s, \quad (5)$$

where $\mathbf{w}_{\text{IMA}}^{(t)}$ is the IMA model for the t -th round, P is the size of the time window, and t_s is the starting round of IMA. The complete process of IMA is illustrated in Algorithm 1.

To mitigate the impact of information noise introduced by historical global models, we initiate IMA in the later training phase, such as $0.75R$ with R denoting the total number of training rounds. Note that IMA provides a better initialization for client models to perform local training, thus resulting in faster convergence and higher accuracy. Importantly, IMA only requires storing P global models $\{\mathbf{w}^{(\tau)}\}_{\tau=t-P}^t$ obtained by FMA and initializing client models with $\mathbf{w}_{\text{IMA}}^{(t)}$ for the next round, without modifying client participation or weighted aggregation. Consequently, IMA can be readily integrated into various FL methods to maintain the global model within the low-loss landscape region, as demonstrated in Figures 5(c) and 7(c).

6.3 Mild Client Exploration in IMA

Theorem 1 indicates that controlling the locality can reduce the loss of the global model in the late training stage. Based on this insight and the geometric properties discussed in Section 4, we highlight the importance of regulating the magnitude of client updates once the global model enters the low-loss area after sufficient training rounds, as illustrated in Figure 5(a). Otherwise, clients may converge to their local optimal models such that the global model deviates from the low-loss area. This is because these local models reach the wall of the loss basin of the global model instead of the bottom, as suggested by the geometric properties of FMA, even when they are close to the global model.

To address this issue, we adopt a more aggressive learning rate decay, called mild client exploration, to control updates during late training. This involves a significant exponential lr decay, such as 0.03 lr decay per round. Table 3 demonstrates that some methods, using a small and constant lr in IMA, yield similar results to ours when they sufficiently constrain client updates. In contrast, as shown in Table 3, when the locality is not adequately controlled during late training (i.e., non-additional lr decay), the deviation of the global model in FMA may impact its performance.

6.4 Advantages of IMA for FL

In contrast to FMA without considering previous global models, IMA uses a sliding window to average the global models over successive training rounds. As discussed in Section 6.1, FMA may deviate the global model from the expected loss-basin center when facing heterogeneous data. Therefore, IMA is built upon FMA and leverages the geometric property of

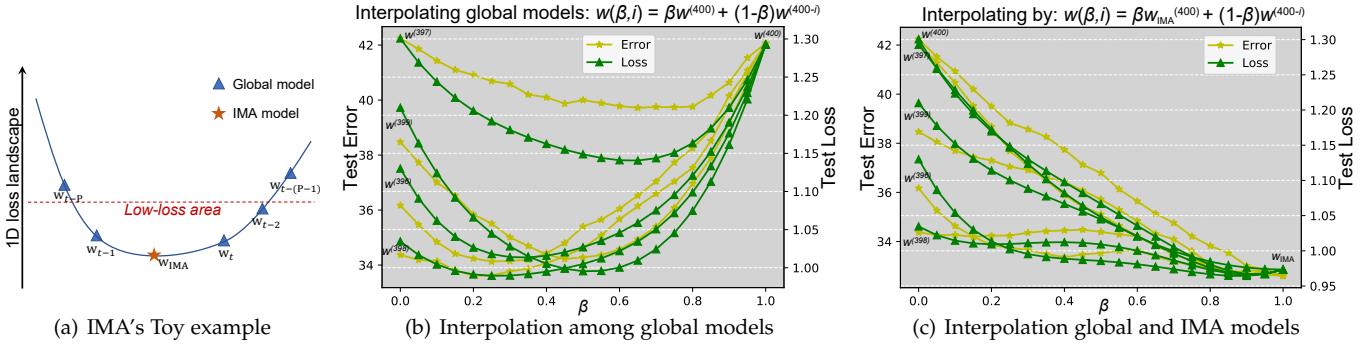


Figure 5: (a) A toy example of 1D loss landscape visualization to show the motivation of IMA; (b) interpolation among global models to validate the effect of interpolation on alleviating global models’ deviation from the basin’s center; (c) interpolation between the IMA and global models to indicate the flatness of the IMA model near the basin’s center.

global models in the loss landscape to mitigate the deviation introduced by FMA, as validated in Figures 5(b) and 5(c).

SWA [20] has been widely used to enhance the model performance by aggregating the training checkpoints when convergence is near. Compared with SWA, IMA collects P previous global models when the FL training has not converged. In addition, IMA and SWA apply to different scenarios: SWA involves centralized training on homogeneous data, while IMA is tailored for FL on heterogeneous data. Specifically, due to heterogeneous data, FMA causes the global model to deviate towards the wall of the loss basin, instead of the expected basin center, as illustrated in Figure 5(a). Notably, the deviation causes the global models to surround the center from different rounds. IMA leverages this geometric property to average global models over a time window, bringing the IMA model closer to the loss-basin center while avoiding the injection of outdated information. Then, IMA re-initiates the client models with the IMA model, whereas SWA does not. The re-initiation with the IMA model corrects the training trajectory and speeds up FL training.

Moreover, it is worth noting that model compression techniques, such as sparsification [30], [34], quantization [35], and low-rank decomposition [36], can be seamlessly integrated with IMA to further reduce the communication overhead and accelerate FL training on heterogeneous data. These techniques operate orthogonally to IMA, which focuses on correcting the trajectory of the global model in the loss landscape. By combining IMA with model compression, we can achieve a two-fold benefit: mitigating the impact of heterogeneous data on model convergence and reducing the communication burden. This synergistic effect is exemplified in Table 2, which demonstrates that IMA is compatible with FedGMA, a method that employs AND-Masked sparsification to accelerate the FL training.

7 EXPERIMENTS

In this section, we present experimental results to verify the effectiveness of IMA by comparing it with existing methods. We will first describe the experimental setups. Next, we will present results on different heterogeneous data setups, datasets, models, FL setups, and baselines. Finally, we conduct a comprehensive ablation study on IMA, including different starting rounds, window sizes, and lr decays.

Algorithm 1 FL with IMA

```

Input: model  $w$ , total client number  $K$ , IMA’s start round  $t_s$ , IMA’s window size  $P$ 
for each round  $t = 1, \dots, R$  do
    Server samples clients  $\mathcal{S} \subseteq [K]$ 
    if  $t \geq t_s$  do
        Server sends  $w_{\text{IMA}}^{(t-1)}$  to all clients  $i \in \mathcal{S}$ 
    else:
        Server sends  $w^{(t-1)}$  to all clients  $i \in \mathcal{S}$ 
    on client  $i \in \mathcal{S}$  in parallel do
        if  $t \geq t_s$  do
            Initialize the local model  $w_i \leftarrow w_{\text{IMA}}^{(t-1)}$ 
            Local training with mild exploration and get  $w_i^{(t)}$ 
        else:
            Initialize the local model  $w_i \leftarrow w^{(t-1)}$ 
            Local training and get  $w_i^{(t)}$ 
        end for
        Send  $w_i^{(t)}$  back to the server
    end on client
    Server performs FMA  $w^{(t)} \leftarrow \sum_{i \in \mathcal{S}} (n_i / \sum_{i \in \mathcal{S}} n_i) w_i^{(t)}$ 
    if  $t \geq t_s$  do
        Server performs IMA  $w_{\text{IMA}}^{(t)} \leftarrow \frac{1}{P} \sum_{\tau=0}^{P-1} w^{(t-\tau)}$ 
    end for

```

7.1 Experimental Setups

7.1.1 Heterogeneous Data Setups

We examine label/feature distribution skew in heterogeneous data [10] and refer to them as label/feature skew. To simulate label skew, we divide FMNIST [58] and CIFAR-10/100 into *data shards with the same sample number* for clients (e.g., $\#C = 2$ indicates that each client holds two classes as in [1]). We use the Dirichlet distribution $Dir(\alpha)$ to create client datasets with *different sample numbers* according to [59]. Moreover, we combine label skew and feature skew on Digit Fives [60] and PACS [61]. Specifically, we divide each domain dataset (i.e., feature skew) into 20 subsets, each for one client, with diverse label distributions (i.e., label skew). The combined skew on Digit Fives and PACS is a more heterogeneous case than their inherent feature domain shift.

Table 1: Mean top-1 last-10-round accuracy comparison of all methods with and without IMA under label skew (including $\#C = 2$ and $\alpha = 0.1$) and feature skew (FS). We follow [29] to use Pachinko Allocation (PA) [62] to create a federated CIFAR-100. Bold text indicates the best results between IMA and IMA-free methods, while underlined text denotes the best results with or without IMA.

Dataset (Model)	Heter Data	FedAvg (+IMA)	FedProx (+IMA)	FedASAM (+IMA)	FedFA (+IMA)	FedNova (+IMA)	FedAdam (+IMA)	FedYogi (+IMA)	FedGMA (+IMA)
FMNIST (CNN)	# $C = 2$	81.17(84.68)	79.78(83.77)	84.69(85.01)	85.60(88.06)	81.23(84.65)	83.53(86.99)	82.42(86.86)	80.89(84.56)
	$\alpha = 0.1$	80.13(83.06)	78.76(81.64)	80.81(82.88)	<u>82.97</u> (86.45)	79.98(83.15)	78.85(83.54)	79.66(83.95)	80.18(83.14)
CIFAR-10 (CNN)	# $C = 2$	62.34(67.37)	61.71(67.03)	62.60(63.64)	67.49(69.19)	62.34(67.46)	64.49(69.59)	66.68(68.74)	62.25(67.47)
	$\alpha = 0.1$	61.00(64.57)	61.31(64.80)	56.92(59.10)	<u>64.99</u> (67.03)	55.11(60.09)	61.61(66.25)	64.12(65.86)	61.18(64.36)
CIFAR-10 (ResNet)	# $C = 2$	50.10(59.64)	53.98(61.65)	49.01(56.78)	46.56(56.15)	49.65(59.30)	54.04(59.05)	54.45(59.73)	49.42(58.79)
	$\alpha = 0.1$	49.96(56.37)	<u>52.13</u> (55.07)	48.96(54.41)	42.84(48.88)	33.72(40.52)	47.47(47.60)	50.92(51.26)	49.89(55.93)
CIFAR-100 (VGG)	$\alpha = 0.1$ (+PA)	38.99(39.89)	38.88(39.93)	37.51(38.25)	<u>43.47</u> (44.68)	39.21(39.96)	38.96(39.83)	38.89(39.29)	39.30(40.02)
CIFAR-100 (ResNet)	$\alpha = 0.1$ (+PA)	31.60(32.97)	32.06(33.27)	28.35(29.34)	31.24(34.03)	32.01(33.50)	<u>37.87</u> (40.93)	37.55(40.27)	31.65(32.90)
Digit Five (CNN)	# $C = 2$ (+FS)	87.90(90.15)	88.14(90.04)	88.68(89.97)	90.26(91.16)	87.77(89.53)	85.63(91.50)	86.31(91.25)	87.91(90.33)
	$\alpha = 0.1$ (+FS)	90.45(91.38)	90.52(91.48)	90.53(91.41)	90.57(91.58)	90.10(90.76)	90.55(92.20)	91.06(92.30)	90.50(91.49)
PACS (AlexNet)	# $C = 2$ (+FS)	57.47(58.01)	60.88(61.51)	<u>61.15</u> (61.46)	56.57(57.36)	60.24(63.53)	54.63(60.09)	55.54(57.03)	57.33(62.17)
	$\alpha = 0.1$ (+FS)	40.36(47.36)	<u>42.15</u> (49.13)	39.57(43.29)	41.95(47.12)	13.96(16.10)	33.76(43.23)	39.97(40.56)	41.73(47.46)

7.1.2 Datasets and Models

We evaluate the performance of baselines with and without IMA on different models and datasets, considering both label and feature skews. Table 1 presents the mean accuracy of the global model for the last ten rounds (mean top-1 accuracy of all domains in Digit Five and PACS). For label skew, we train CNN models [1] on FMNIST and CIFAR-10, and train ResNet18 [63] and VGG11 [64] on CIFAR-10/100. For label-feature skew, we train CNN on Digit Fives and AlexNet [65] on PACS. We replace BN layers with GN layers following [66]. Detailed settings are presented in Table 4 in the Appendix. We aim to demonstrate the effectiveness of IMA on FL by considering different model architectures and datasets.

7.1.3 FL Setup and Baselines

In the FL setup, unless otherwise specified, we use a batch size of 50 and 5 local epochs, with 100 clients participating in FL for 400 rounds, and one-tenth of the clients participate in each round. For the client optimizer, we follow the standard configuration from the FL benchmark [67] and use the SGD optimizer with a learning rate (lr) of 0.01 and momentum of 0.9 (see Tables 5 and 7 for more details in the Appendix).

For baselines, in addition to FedAvg [1], we include other methods that improve FedAvg on the client side, such as parameter-regularization: FedProx [25], flatness-improvement: FedASAM [22], and feature-classifier-alignment: FedFA [26]), and on the server side, such as update-normalization: FedNova [28], gradient-masking: FedGMA [30], and server-momentum: FedADAM/FedYogi [29]. Here, we choose FedAdam and FedYogi as our momentum-based baselines because our method aligns with their approach of using global model updates for momentum, instead of SCAFFOLD [37], which relies on receiving updates from a sufficient number of clients in each round and is ineffective when clients have unpredictable availability and may drop out during the training process [67], [68]. Meanwhile, we implement IMA on these baselines with a window size $P = 5$ and the starting round $t_s = 0.75R$ with $R = 400$, unless otherwise specified. It is worth noting that

IMA provides a better initialization for client models and is thus compatible with these baselines.

7.2 Experimental Results

7.2.1 Performance with Label Skew

Table 1 illustrates that, for label skew (i.e., $\#C = 2$ and $\alpha = 0.1$), IMA enhances the performance of all methods on different datasets and models. Adding IMA consistently improves performance across all datasets (FMNIST, CIFAR-10, and CIFAR-100). For instance, when training a CNN model on FMNIST, FedFA with IMA achieves the highest accuracy of 88.06% among baselines, compared with 79.7% for FedProx without IMA. The most significant improvement is achieved by training ResNet on CIFAR-10, where the performance rises from 49.65% to 59.30%, i.e., with a gain of 9.65%. Moreover, for the same setup of label skew, e.g., $\alpha = 0.1$, the performance gain for CIFAR-10 is 6.42% (FedAvg with ResNet), which is twice of the case of CIFAR-100 (3.06%), as shown in Table 1. Note that CIFAR-100 employs Pachinko Allocation [62] to make date heterogeneity milder. Thus, the benefits of IMA depend on the heterogeneity level of label skew, with greater heterogeneity resulting in more significant performance gains.

Meanwhile, the performance of various models on the same dataset varies in FL with the same heterogeneous data setup. For example, in the case of $\alpha = 0.1$ on CIFAR-10, the CNN model achieves approximately 10% accuracy improvement over the ResNet model across all methods. This is partly because the CNN model, with fewer parameters, is faster to train within a given total training round, thereby achieving higher accuracy. In addition, compared to CNN, the better fitting ability of the ResNet model causes it to overfit heterogeneous local data more significantly after multiple local epochs. This leads to larger divergence among local models and reduced accuracy of the global model.

7.2.2 Performance on Various Heterogeneous Degrees

To further investigate the effect of heterogeneous data on IMA, we conduct tests on Digit Five and PACS datasets

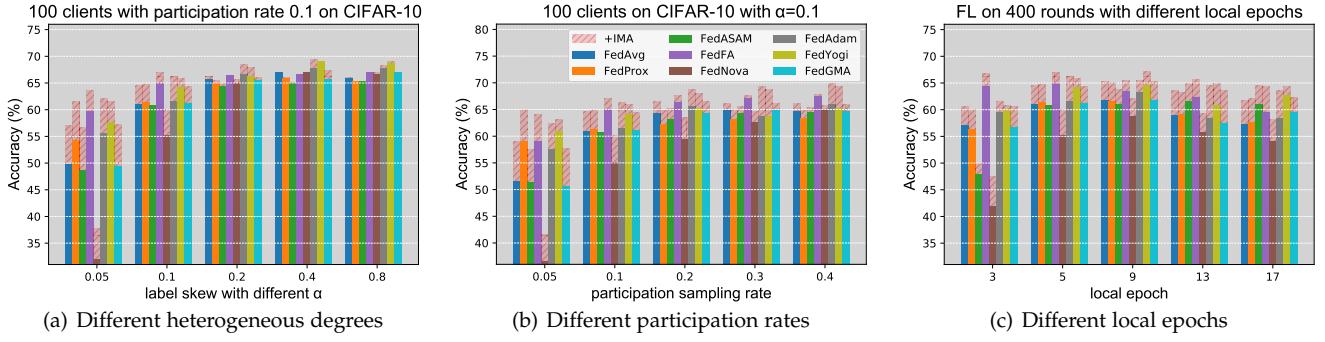


Figure 6: Performance of all methods with and without IMA on different federated setups.

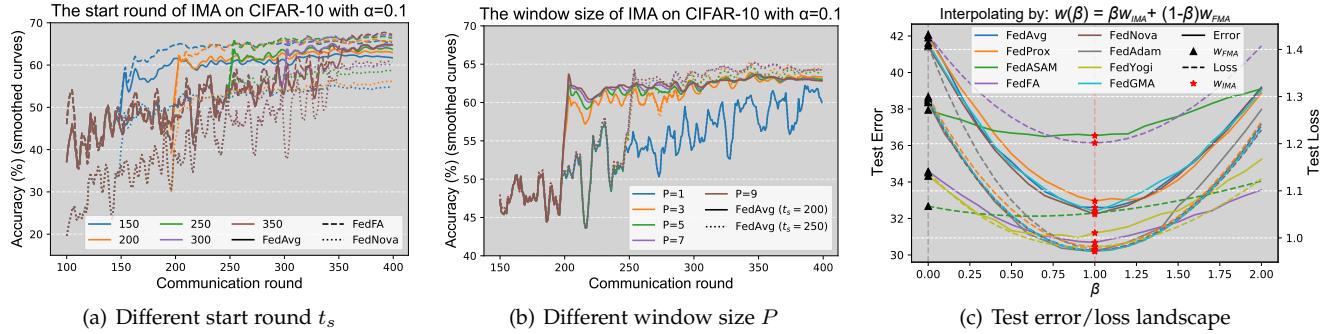


Figure 7: Ablation studies on IMA and landscape visualization between FMA and IMA models. Note that all curves in Figures 7(a) and 7(b) are smoothed by a Savitzky-Golay filter [69] with a window length of 10 and a polynomial order of 2 to mitigate the noise in the visualization of the results while preserving the essential trends.

under both label and feature skew. Our findings on feature skew, as shown in Table 1, are similar to those observed in the cases of label skew. For instance, we observe a greater performance gain with IMA on PACS than Digit Five due to the more severe heterogeneity of feature skew in PACS with $\alpha = 0.1$. To validate these findings, we test different levels of label skew on CIFAR-10 and present the results in Figure 6(a). The figure indicates that IMA substantially improves performance on more heterogeneous data, represented by smaller α . Specifically, the IMA gain for $\alpha = 0.05$ and $\alpha = 0.1$ is approximately 5% for all baselines, whereas the gain becomes insignificant for $\alpha = 0.8$. Moreover, the performance gain of IMA diminishes as α increases. This implies that IMA is superior to FMA, except for homogeneous data. Therefore, Table 1 and Figure 6(a) demonstrate the effectiveness of IMA in mitigating the negative effect of heterogeneous data, especially in scenarios with extreme heterogeneity.

7.2.3 Reduction in Communication Overhead

Table 2 presents the communication efficiency of IMA with different starting rounds to achieve a target accuracy on CIFAR-10 with $\alpha = 0.1$, where FedASAM and FedNova are not reported because their performance is worse than the targeted accuracy. The results illustrate that initiating IMA at earlier rounds significantly reduces the communication overhead, compared with three-quarters of the total rounds in Table 1. For instance, starting IMA at the 150th round saves communication by nearly half for FedAdam and FedProx.

Table 2: Required rounds by IMA with different start rounds t_s when the accuracy reaches 61.61% from Table 1.

(+IMA) t_s	150	200	250	300	FMA
FedAvg	318($\times 1.24$)	257($\times 1.53$)	260($\times 1.51$)	309($\times 1.28$)	394($\times 1$)
FedProx	201($\times 1.96$)	210($\times 1.88$)	260($\times 1.52$)	309($\times 1.28$)	394($\times 1$)
FedFA	183($\times 1.87$)	212($\times 1.61$)	258($\times 1.32$)	308($\times 1.11$)	341($\times 1$)
FedAdam	193($\times 2.03$)	210($\times 1.87$)	259($\times 1.51$)	306($\times 1.28$)	392($\times 1$)
FedYogi	195($\times 1.85$)	210($\times 1.71$)	257($\times 1.40$)	306($\times 1.18$)	360($\times 1$)
FedGMA	316($\times 1.25$)	242($\times 1.63$)	260($\times 1.52$)	309($\times 1.28$)	394($\times 1$)

7.2.4 Performance on Different Client Participation Rates

We evaluate the performance of IMA under varying participation rates from 0.05 to 0.4 in Figure 6(b), in addition to the results obtained with a 0.1 participation rate in Table 1. The figure indicates that the gain achieved by IMA generally increases as the client participation rate decreases. For example, the gain with a 0.05 participation rate is approximately twice that observed with a 0.2 participation rate. Furthermore, Figure 6(b) verifies the global model deviation induced by low participation rates, as highlighted in Section 6. It illustrates that lower participation rates lead to larger deviations between the cohort and the global datasets, amplifying the negative effect of heterogeneous data.

7.2.5 Performance on Different Local Epochs

To assess the robustness of IMA, we evaluate its performance on different local epoch settings ranging from 3 to 17, as shown in Figure 6(c). The results show that IMA consistently improves all baseline methods across different epochs. We also observe that the performance gain remains stable even

Table 3: Accuracy v.s. decay schemes.

IMA w/ decay	Exp Decay	Const LR	Cyclic Decay	Epoch Decay	NA Decay
FedAvg	64.57	64.50	64.27	62.96	63.96
FedProx	64.80	64.73	64.59	63.14	64.12
FedASAM	59.10	58.14	59.33	57.48	58.95
FedFA	67.03	66.62	66.94	66.63	66.40
FedNova	60.09	59.86	59.38	59.04	58.90
FedAdam	66.25	65.89	66.06	64.00	65.62
FedYogi	65.86	65.53	65.51	63.13	65.12
FedGMA	64.36	64.41	64.12	62.97	63.75

when the number of local epochs increases. This is because client models are closely located around the global model within the same basin due to FMA, as observed in Section 4. Consequently, the advantages of IMA persist even when client models are close to their local optima, as IMA may bring global models closer to the global optimum.

7.3 Ablation Study on IMA

7.3.1 Ablation on Starting Rounds and Window Size of IMA
The results presented in Table 2 indicate that initiating IMA at a later round leads to increased communication overhead when considering a target accuracy. For example, setting $t_s = 300$ on the FedFA baseline results in an additional 96 rounds compared to the case of $t_s = 200$. In contrast, Figure 7(a) demonstrates that starting IMA at a later round leads to better accuracy performance. For instance, the case of $t_s = 300$ on FedFA shows an approximately 3% increase in accuracy compared to the case of $t_s = 200$. These findings highlight the existence of a trade-off between communication efficiency and performance in IMA. Moreover, increasing the window size improves the training stability, but it impairs the final accuracy if IMA starts early. This can be observed in the case of FedAvg with $t_s = 200$ and $P = 9$, where a lower accuracy is achieved compared to other cases, as shown in Figure 7(b). Note that due to the oscillation of the original results, all curves in Figures 7(a) and 7(b) have been smoothed for better visualization clarity.

7.3.2 Ablation on Mild Client Exploration in IMA

As mentioned in Section 6, we adopt a more aggressive exponential lr decay per round in IMA than in FMA to restrict client exploration. To evaluate this design choice, we conduct experiments on CIFAR-10 with $\alpha = 0.1$ to ablate IMA with different decay schemes, including a small constant lr (i.e., lr is 5×10^{-5} in IMA), cyclic lr decay [20] (i.e., decaying lr from 1×10^{-2} to 5×10^{-5} every 20 rounds), epoch decay [70] (i.e., decaying one local epoch per 20 rounds), and non-additional decay (NA). As shown in Table 3, more aggressive decay schemes that sufficiently constrain client updates (e.g., exponential lr decay or small constant lr) outperform milder schemes. For instance, exponential decay achieves 64.57% on FedAvg, compared with 62.96% of epoch decay.

7.3.3 Test Loss Landscape between FMA and IMA Models

Figure 7(c) depicts the interpolation model between FMA and IMA models (both from the final round) to visualize the landscape of test error and test loss. The figure shows that the IMA models reach almost the center (i.e., the lowest

point) of the test error and loss basins for all baselines, effectively alleviating the deviation mentioned in Section 6. In contrast, the FMA models only reach the basin's wall, which verifies the deviation observed in Section 4. Moreover, Figure 7(c) shows that these methods reach various basins with different curvature. However, it does not necessarily hold that a flatter basin corresponds to lower error. For example, while FedASAM reaches the basin with the flattest curvature, it achieves the highest test error.

8 DISCUSSIONS AND FUTURE WORKS

This work advanced the understanding of how FMA operates in the presence of heterogeneous data and proposed employing IMA to enhance its performance. Firstly, we investigated the dynamics of the loss landscape of FMA during training and observed that client models closely surround the global model within the same basin. By employing test loss decomposition, we illustrated the relationship between the global model and client models, demonstrating that the client models' heterogeneous bias and locality dominate the global model's error after the early training stage. These findings motivated us to adopt IMA on global models in the late training stage rather than disregarding them in FMA. Our experiments showed that IMA significantly improves existing FL methods' accuracy and communication efficiency under both label and feature skews.

Although we demonstrate the error relationship between the global model and client models based on expected loss decomposition in Section 5, it remains necessary to explicitly quantify this relationship in general cases. Future works should analyze how each factor dominates the error throughout the training process. In addition, an IMA variant with an adaptive starting round demonstrates promising results in Table 2 and deserves investigation to reduce communication overhead without compromising generalization. Moreover, employing more flexible regularization between the global model and client models (e.g., elastic weight consolidation [57]) can further reduce the bias and locality in Theorem 1. We hope our study will serve as a valuable reference for further analysis and improvement of FL methods.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Ft. Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.
- [2] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," [Online]. Available: <https://arxiv.org/pdf/1806.00582.pdf>.
- [3] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2020.
- [4] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial iot: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Oct. 2021.
- [5] S. Wang, M. Chen, C. G. Brinton, C. Yin, W. Saad, and S. Cui, "Performance optimization for variable bitwidth federated learning in wireless networks," *IEEE Trans. Wireless Commun.*, Mar. 2023.
- [6] Z. Lin, H. Liu, and Y.-J. A. Zhang, "CFLIT: Coexisting federated learning and information transfer," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8436–8453, Sept. 2023.

- [7] M. N. Nguyen, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, "Toward multiple federated learning services resource sharing in mobile edge networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 541–555, Jun. 2021.
- [8] Z. Feng, X. Chen, Q. Wu, W. Wu, X. Zhang, and Q. Huang, "Feddd: Toward communication-efficient federated learning with differential parameter dropout," *IEEE Trans. Mobile Comput.*, no. 01, pp. 1–18, Aug. 2023.
- [9] X. Zhang, Z. Chang, T. Hu, W. Chen, X. Zhang, and G. Min, "Vehicle selection and resource allocation for federated learning-assisted vehicular network," *IEEE Trans. Mobile Comput.*, pp. 1–12, Jun. 2023.
- [10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [11] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," [Online]. Available <https://arxiv.org/pdf/2107.06917.pdf>.
- [12] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, 1992.
- [13] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 5330–5340.
- [14] J. Zhang, C. De Sa, I. Mitliagkas, and C. Ré, "Parallel sgd: When does averaging help?" [Online]. Available <https://arxiv.org/pdf/1606.07365.pdf>.
- [15] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Virtual Event, Jul. 2020, pp. 5381–5393.
- [16] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.
- [17] H. Yu, S. Yang, and S. Zhu, "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, Honolulu, Hawaii, USA, Jan. 2019, pp. 5693–5700.
- [18] J. Wang, R. Das, G. Joshi, S. Kale, Z. Xu, and T. Zhang, "On the unreasonable effectiveness of federated averaging with heterogeneous data," *Trans. Mach. Learn. Res. (TMLR)*, May 2024.
- [19] P. Jain, S. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, "Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification," *J. Mach. Learn. Res.*, vol. 18, pp. 223:1–223:42, Jul. 2017.
- [20] P. Izmailov, D. Podoprikin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Int. Conf. on Uncert. in Artif. Intell. (UAI)*, Monterey, California, USA, Aug. 2018, pp. 876–885.
- [21] V. Gupta, S. A. Serrano, and D. DeCoste, "Stochastic weight averaging in parallel: Large-batch training that generalizes well," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.
- [22] D. Calderola, B. Caputo, and M. Ciccone, "Improving generalization in federated learning by seeking flat minima," in *Proc. Eur. Conf. Comp. Vision (ECCV)*, Tel Aviv, Israel, Oct. 2022, pp. 654–672.
- [23] S. Lee, S. Purushwalkam Shiva Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra, "Stochastic multiple choice learning for training diverse deep ensembles," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Barcelona, Spain, Dec. 2016, pp. 2119–2127.
- [24] A. Rame, M. Kirchmeyer, T. Rahier, A. Rakotomamonjy, P. Gallinari, and M. Cord, "Diverse weight averaging for out-of-distribution generalization," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, LA, CA, USA, May 2022.
- [25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, Austin, TX, USA, Mar. 2020.
- [26] T. Zhou, J. Zhang, and D. H. K. Tsang, "FedFA: Federated learning with feature anchors to align feature and classifier for heterogeneous data," *IEEE Trans. Mobile Comput.*, pp. 1–17, Oct. 2023.
- [27] Z. Li, Z. Lin, J. Shao, Y. Mao, and J. Zhang, "FedCiR: Client-invariant representation learning for federated non-iid features," *IEEE Trans. Mobile Comput.*, pp. 1–17, Mar. 2024.
- [28] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Virtual Event, Dec. 2020, pp. 7611–7623.
- [29] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, Virtual Event, May 2021.
- [30] I. Tenison, S. A. Sreeramadas, V. Mugunthan, E. Oyallon, I. Rish, and E. Belilovsky, "Gradient masked averaging for federated learning," *Trans. Mach. Learn. Res. (TMLR)*, Sep. 2023.
- [31] Z. Li, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Federated learning with gan-based data synthesis for non-iid clients," in *FL Workshop in Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 13448, Vienna, Austria, Jul. 2022, pp. 17–32.
- [32] Y. Sun, J. Shao, S. Li, Y. Mao, and J. Zhang, "Stochastic coded federated learning with convergence and privacy guarantees," in *IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Aug. 2022, pp. 2028–2033.
- [33] J. Shao, Y. Sun, S. Li, and J. Zhang, "Dres-fl: Dropout-resilient secure federated learning for non-iid clients via secret data sharing," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, LA, CA, USA, May 2022.
- [34] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, and P. Mittal, "SparseFed: Mitigating model poisoning attacks in federated learning with sparsification," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 151, Virtual Event, Sep 2022, pp. 7587–7624.
- [35] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, Mar. 2021.
- [36] G. Lan, X.-Y. Liu, Y. Zhang, and X. Wang, "Communication-efficient federated learning for resource-constrained edge devices," *IEEE Trans. Mach. Learn. Commun. Netw.*, Aug 2023.
- [37] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, Virtual Event, 2020, pp. 5132–5143.
- [38] R. Sun, D. Li, S. Liang, T. Ding, and R. Srikanth, "The global landscape of neural networks: An overview," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 95–108, Oct. 2020.
- [39] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, San Diego, CA, USA, May 2014.
- [40] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, Montreal, Canada, Dec 2018, pp. 6391–6401.
- [41] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, Virtual Event, May 2021.
- [42] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park, "Swad: Domain generalization by seeking flat minima," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Virtual Event, Dec. 2021, pp. 22405–22418.
- [43] T. Garipov, P. Izmailov, D. Podoprikin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, Montreal, Canada, Dec. 2018, pp. 8803–8812.
- [44] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht, "Essentially no barriers in neural network energy landscape," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 80, 10–15 Jul 2018, pp. 1309–1318.
- [45] Z. Li, H.-Y. Chen, H. W. Shen, and W.-L. Chao, "Understanding federated learning through loss landscape visualizations: A pilot study," in *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.
- [46] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Honolulu, Hawaii, USA, Jul. 2023, pp. 19767–19788.
- [47] T. Zhou, J. Zhang, and D. H. K. Tsang, "Mode connectivity and data heterogeneity of federated learning," in *Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, Dec. 2023.
- [48] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, Sep. 1992.
- [49] R. Kohavi, D. H. Wolpert *et al.*, "Bias plus variance decomposition for zero-one loss functions," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 96, Bari, Italy, Jul. 1996, pp. 275–83.

- [50] P. Domingos, "A unified bias-variance decomposition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Austin, Texas, USA, Jul. 2000, pp. 231–238.
- [51] G. Brown, J. Wyatt, and P. Sun, "Between two extremes: Examining decompositions of the ensemble objective function," in *Multip. Classif. Syst.: 6th Int. Workshop, Seaside, CA, USA*, Jun. 2005, pp. 296–305.
- [52] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias-variance trade-off," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 116, no. 32, pp. 15 849–15 854, Jul. 2019.
- [53] J. Shao, Z. Li, W. Sun, T. Zhou, Y. Sun, L. Liu, Z. Lin, and J. Zhang, "A survey of what to share in federated learning: Perspectives on model utility, privacy leakage, and communication efficiency," [Online]. Available: <https://arxiv.org/pdf/2307.10655.pdf>.
- [54] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," [Online]. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- [55] N. Ye, K. Li, H. Bai, R. Yu, L. Hong, F. Zhou, Z. Li, and J. Zhu, "Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 7947–7958.
- [56] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," [Online]. Available: <https://arxiv.org/pdf/1312.6211.pdf>.
- [57] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
- [58] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," [Online]. Available: <https://arxiv.org/pdf/1708.07747.pdf>.
- [59] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Long Beach, California, USA, Jun 2019, pp. 7252–7261.
- [60] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, Virtual Event, May 2021.
- [61] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 5542–5550.
- [62] W. Li and A. McCallum, "Pachinko allocation: Dag-structured mixture models of topic correlations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 148, Pittsburgh, Pennsylvania, USA, Jun. 2006, pp. 577–584.
- [63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Repr. (ICLR)*, San Diego, CA, USA, May 2015.
- [65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Mar 2017.
- [66] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Virtual Event, Jul. 2020, pp. 4387–4398.
- [67] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. Int. Conf. Data Eng. (ICDE)*, Kuala Lumpur, Malaysia, May 2022, pp. 965–978.
- [68] Y. Sun, Y. Mao, and J. Zhang, "Mimic: Combating client dropouts in federated learning by mimicking central updates," *IEEE Trans. Mobile Comput.*, Nov. 2023.
- [69] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, May 2011.
- [70] G. Pu, Y. Zhou, D. Wu, and X. Li, "Server averaging for federated learning," [Online]. Available: <https://arxiv.org/pdf/2103.11619.pdf>.



Tailin Zhou (Graduate student member, IEEE) received his B.Eng. degree in Electrical Engineering and Automation Engineering from Sichuan University in 2018, and his Master's degree in Electrical Engineering from South China University of Technology in 2021. He is pursuing a Ph.D. degree at the Hong Kong University of Science and Technology under the supervision of Professor Jun Zhang and Professor Danny H.K. Tsang. His research interests include federated learning and cooperative AI agents.



Zehong Lin (Member, IEEE) received the B.Eng. degree in information engineering from South China University of Technology in 2017, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2022. Since 2022, he has been with the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology, where he is currently a Research Assistant Professor. His research interests include federated learning and edge AI.



Jun Zhang (Fellow, IEEE) received the B.Eng. degree in Electronic Engineering from the University of Science and Technology of China in 2004, the M.Phil. degree in Information Engineering from the Chinese University of Hong Kong in 2006, and the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 2009. He is an Associate Professor in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology. His research

interests include wireless communications and networking, mobile edge computing and edge AI, and cooperative AI.

Dr. Zhang co-authored the book *Fundamentals of LTE* (Prentice-Hall, 2010). He is an Editor of *IEEE Transactions on Communications*, *IEEE Transactions on Machine Learning in Communications and Networking*, and was an editor of *IEEE Transactions on Wireless Communications* (2015–2020). He was a MAC track co-chair for *IEEE Wireless Communications and Networking Conference (WCNC)* 2011 and a co-chair for the *Wireless Communications Symposium of IEEE International Conference on Communications (ICC)* 2021. He is an IEEE Fellow and an IEEE ComSoc Distinguished Lecturer.



Danny H.K. Tsang (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA, USA, in 1989. After graduation, he joined the Department of Computer Science, Dalhousie University, Halifax, NS, Canada. He later joined the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 1992, where he is currently a Professor. He has also been serving

as the Thrust Head of the Internet of Things Thrust, HKUST (Guangzhou), Guangzhou, China, since 2020. His current research interests include next-generation networking, mobile edge computing, online algorithm design, and smart grids.

Dr. Tsang is a member of the Special Editorial Cases Team of *IEEE Communications Magazine*. He was a Guest Editor of the *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*' special issue on Advances in P2P Streaming Systems, an Associate Editor of *Journal of Optical Networking* published by the Optical Society of America, and a Guest Editor of *IEEE SYSTEMS JOURNAL*. He invented the 64B/65B encoding (U.S. Patent No.: U.S. 6 952 405 B2) and contributed it to the proposal for Transparent GFP in the T1X1.5 standard that was advanced to become the ITU G.GFP standard. The coding scheme has now been adopted by International Telecommunication Union (ITU)'s Generic Framing Procedure Recommendation GFP-T (ITUT G.7041/Y.1303) and Interfaces of the Optical Transport Network (ITU-T G.709). He has been elevated to an IEEE Fellow in 2012 and an HKIE Fellow in 2013.

APPENDIX

APPENDIX A: PROOF

Proof of Lemma 1

Suppose clients do not have an extremely imbalanced dataset (i.e., $n_i/n_j \neq \infty$ when $i \neq j$). For client k and the FMA's model, we have:

$$f_{\mathbf{w}_k}(x) = f_{\mathbf{w}_{\text{FMA}}}(x) + \langle \Delta f_{\mathbf{w}_{\text{FMA}}}(x), \Delta_k \rangle + O(\|\Delta_k\|^2), \quad (6)$$

where $\langle \cdot, \cdot \rangle$ is the dot product, and $\Delta_k = \mathbf{w}_k - \mathbf{w}_{\text{FMA}}$. Thus, we establish the relationship between the FMA-model function and the WENS function as:

$$\begin{aligned} f_{\text{WENS}}(x) &= \sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}(x) \\ &= f_{\mathbf{w}_{\text{FMA}}}(x) + \sum_{k=1}^K \frac{n_k}{n} \langle \Delta f_{\mathbf{w}_{\text{FMA}}}(x), \Delta_k \rangle + \sum_{k=1}^K \frac{n_k}{n} O(\|\Delta_k\|^2) \\ &= f_{\mathbf{w}_{\text{FMA}}}(x) + \langle \Delta f_{\mathbf{w}_{\text{FMA}}}(x), \sum_{k=1}^K \frac{n_k}{n} \Delta_k \rangle + O(\|\Delta\|^2) \\ &= f_{\mathbf{w}_{\text{FMA}}}(x) + O(\|\Delta\|^2), \end{aligned} \quad (7)$$

where $\|\Delta\| = \max_k^K \|\Delta_k\|$, $n = \sum_k n_k$ is the total sample number.

Proof of Theorem 1

Substituting $f_{\text{WENS}}(x) = \sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}$ into (2), we have:

$$\begin{aligned} &\mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K \in \prod_k^K \mathcal{W}_{D_k}} \mathcal{L}(\{\mathbf{w}_k\}_{k=1}^K) \\ &= \mathbb{E}_{(x,y) \in \mathcal{D}} [(\text{Bias}\{f_{\text{WENS}}|(x,y)\})^2 + \text{Var}\{f_{\text{WENS}}|x\}]. \end{aligned} \quad (8)$$

For the bias term, we have:

$$\begin{aligned} \text{Bias}\{f_{\text{WENS}}|(x,y)\} &= y - \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} f_{\text{WENS}}(x) \\ &= y - \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} \left[\sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}(x) \right] \\ &= y - \sum_{k=1}^K \frac{n_k}{n} \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)] \\ &= \sum_{k=1}^K \frac{n_k}{n} (y - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)]). \end{aligned} \quad (9)$$

Taking the expectation of the bias term for the global dataset, we have:

$$\begin{aligned} &\mathbb{E}_{(x,y) \in \mathcal{D}} (\text{Bias}\{f_{\text{WENS}}|(x,y)\})^2 \\ &= \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} \left[\sum_{k=1}^K \frac{n_k}{n} \underbrace{\mathbb{I}[(x,y) \in \mathcal{D}_k] (y - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)])}_{\text{TrainBias}\{f_{\mathbf{w}_k}|(x,y)\}} \right]^2 \\ &\quad + \sum_{k=1}^K \frac{n_k}{n} \underbrace{\mathbb{I}[(x,y) \in \mathcal{D} \setminus \mathcal{D}_k] (y - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)])^2}_{\text{HeterBias}\{f_{\mathbf{w}_k}|(x,y)\}} \end{aligned} \quad (10)$$

For the variance term, we have:

$$\begin{aligned} &\text{Var}\{f_{\text{WENS}}|(x,y)\} \\ &= \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} \left[\left(\sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}(x) - \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} \left[\sum_{k=1}^K \frac{n_k}{n} f_{\mathbf{w}_k}(x) \right] \right)^2 \right] \\ &= \sum_{k=1}^K \frac{n_k^2}{n^2} \underbrace{\mathbb{E}_{\mathbf{w}_k} [(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)])^2]}_{\text{Var}\{f_{\mathbf{w}_k}|x\}} \\ &\quad + \sum_k \sum_{k' \neq k} \frac{n_k n_{k'}}{n^2} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}, \end{aligned} \quad (11)$$

where $\text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\} = \mathbb{E}_{\mathbf{w}_k, \mathbf{w}_{k'}} [(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k} [f_{\mathbf{w}_k}(x)]) (f_{\mathbf{w}_{k'}}(x) - \mathbb{E}_{\mathbf{w}_{k'}} [f_{\mathbf{w}_{k'}}(x)])]$. Taking the expectation of the variance term for the global dataset, we have:

$$\begin{aligned} &\mathbb{E}_{(x,y) \in \mathcal{D}} (\text{Var}\{f_{\text{WENS}}|(x,y)\}) \\ &= \mathbb{E}_{(x,y) \in \mathcal{D}} \left(\sum_{k=1}^K \frac{n_k^2}{n^2} \text{Var}\{f_{\mathbf{w}_k}|x\} \right. \\ &\quad \left. + \sum_k \sum_{k'} \frac{n_k n_{k'}}{n^2} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\} \right) \\ &= \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} \sum_{k=1}^K \frac{n_k^2}{n^2} \text{Var}\{f_{\mathbf{w}_k}|x\} \\ &\quad + \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} \sum_k \sum_{k'} \frac{n_k n_{k'}}{n^2} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\}. \end{aligned} \quad (12)$$

Using the Taylor expansion at the zeroth order of the loss, we extend Lemma 1 and obtain:

$$\begin{aligned} \mathcal{L}(\mathbf{w}_{\text{FMA}}) &= \mathbb{E}_{(x,y) \in \mathcal{D}} [l(f_{\mathbf{w}_{\text{FMA}}}(x); y)] \\ &= \mathbb{E}_{(x,y) \in \mathcal{D}} [l(f_{\text{WENS}}(x); y)] \\ &\quad + O(\|f_{\mathbf{w}_{\text{FMA}}}(x) - f_{\text{WENS}}(x)\|_2) \\ &= \mathcal{L}(\{\mathbf{w}_k\}_{k=1}^K) + O(\Delta^2). \end{aligned} \quad (13)$$

Finally, combining (10) and (12) with (8), we have:

$$\begin{aligned} &\mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K \in \prod_k^K \mathcal{W}_{D_k}} \mathcal{L}(\mathbf{w}_{\text{FMA}}) \\ &= \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K \in \prod_k^K \mathcal{W}_{D_k}} \mathcal{L}(\{\mathbf{w}_k\}_{k=1}^K) + O(\Delta^2) \\ &= \mathbb{E}_{(x,y) \in \mathcal{D}} [(\text{Bias}\{f_{\text{WENS}}|(x,y)\})^2 + \text{Var}\{f_{\text{WENS}}|x\}] + O(\Delta^2) \\ &= \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} \left[\sum_{k=1}^K \frac{n_k}{n} \text{TrainBias}\{f_{\mathbf{w}_k}|(x,y)\} \right. \\ &\quad \left. + \frac{n_k}{n} \text{HeterBias}\{f_{\mathbf{w}_k}|(x,y)\} \right]^2 \\ &\quad + \sum_{k=1}^K \frac{n_k^2}{n^2} \text{Var}\{f_{\mathbf{w}_k}|x\} + \sum_k \sum_{k'} \frac{n_k n_{k'}}{n^2} \text{Cov}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}}|x\} \\ &\quad + O(\Delta^2). \end{aligned} \quad (14)$$

APPENDIX B: LOSS LANDSCAPE VISUALIZATION

Loss Landscape Visualization of Cross-device and Cross-silo FL

We examine two common FL frameworks to demonstrate the similarity of loss landscapes across different FL frameworks:

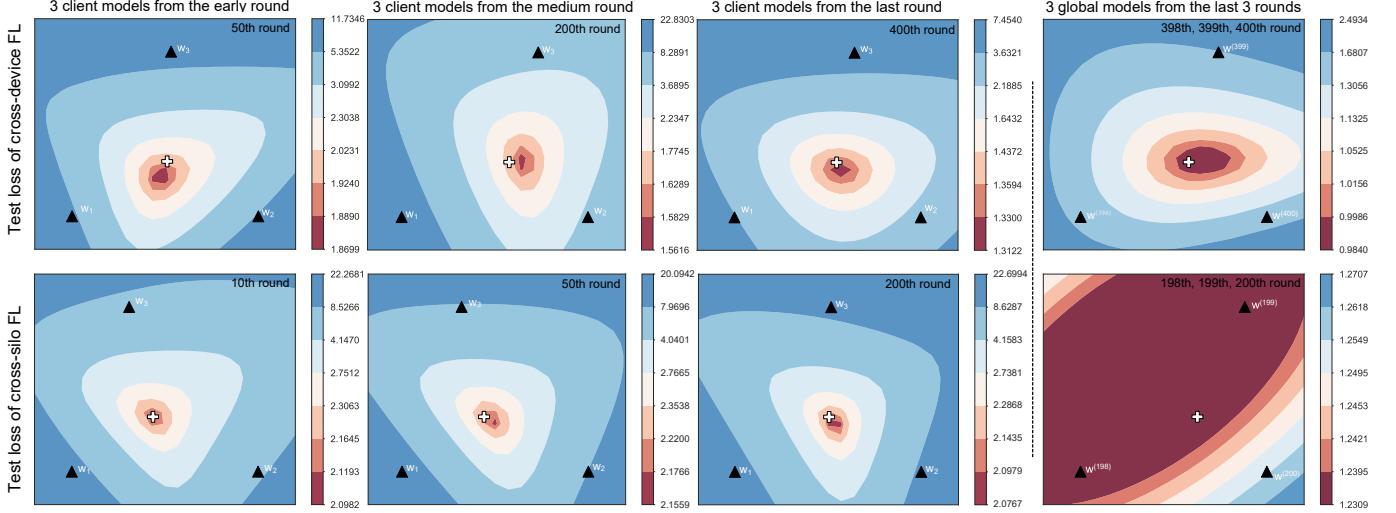


Figure 8: Loss landscape visualization with three models on global test set in cross-device and cross-silo FL.

cross-device FL and cross-silo FL [10]. The number of clients involved in cross-silo FL is small (e.g., the cross-silo FL shown in Figure 8 includes ten clients), and all clients participate fully in each communication round. On the other hand, cross-device FL requires a large number of clients, and only a subset of them participate in each round (e.g., the cross-device FL in Figure 8 involves 100 clients with 0.1 participation rate for each round). Figure 8 depicts the loss landscape visualization with three models on the global dataset for both cross-device and cross-silo FL settings.

Similar to the FMA's geometric properties observed from Figure 1, the FMA model (i.e., the *white cross*) achieves lower test loss and error than the individual client models (i.e., the *black triangles*) throughout the training process in both settings. Furthermore, both FL settings illustrate that FMA maintains the client and global models closely within a shared basin. Notably, the deviation between the *white cross* and the lowest point of the basin in terms of loss/error is smaller in cross-device FL than cross-silo FL, as shown in Figure 8. This finding supports the analysis presented in Section 6, which suggests that low participation rates exacerbate the deviation.

In addition to the loss landscape, we visualize the classification error landscape on the global dataset for both settings in Figure 9. The observed geometric properties of FMA in Figure 9 are similar to those in Figure 8. Therefore, we omit the detailed descriptions here to avoid repetition.

Loss Landscape Visualization under Different Models, Datasets, and Heterogeneous Data Settings

To further explore the geometric properties of FMA, we visualize the loss landscape of FL under various models (including the CNN model and the ResNet model), datasets (including FMNIST and CIFAR-10), and data heterogeneity (including label skews with $\#C = 2$ and $\alpha = 0.1$). The visualization results are presented in Figure 10. The geometric properties of FMA discussed in Section 4 are consistent with those observed in Figure 10. Regardless of the specific FL setup, FMA ensures that client and global models reside within a common basin. This geometric insight sheds light on

how FMA effectively prevents client models from over-fitting to their respective datasets (i.e., FMA mitigates the over-fitting information of client models being aggregated into the global model) and improves the generalization performance of the global model.

APPENDIX C: FURTHER ANALYSIS

To analyze the variance term, we set $n_i = n_j$ (i.e., the number of client samples is the same) to isolate the impact of weighted averaging on the loss decomposition in Theorem 1. Consequently, we have the following corollary:

Corollary 2. (*Loss decomposition of FMA with the same client sample sizes. Extended from Theorem 1.*) Given K client models $\{\mathbf{w}_k\}_{k=1}^K \in \prod_k \mathcal{W}_{D_k}$ and $n_i/n = n_j/n = 1/K$, we can decompose the expected loss of the FMA's model \mathbf{w}_{FMA} on \mathcal{D} as:

$$\begin{aligned} \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K} \mathcal{L}(\mathbf{w}_{\text{FMA}}) &= \frac{1}{nK^2} \sum_{(x,y) \in \mathcal{D}} [\sum_{k=1}^K \text{TrainBias}\{f_{\mathbf{w}_k}|(x,y)\} \\ &\quad + \text{HeterBias}\{f_{\mathbf{w}_k}|(x,y)\}]^2 + \frac{1}{K} \sum_{k=1}^K \underbrace{\frac{1}{K} \text{Var}\{f_{\mathbf{w}_k}|x\}}_{\overline{\text{Var}}\{f_{\{\mathbf{w}_k\}_{k=1}^K}|x\}} \\ &\quad + \sum_k \frac{K-1}{K^2} \sum_{k' \neq k} \underbrace{\frac{1}{K-1} \text{Cov}\{f_{\mathbf{w}_k}, f_{\mathbf{w}_{k'}}|x\}}_{\overline{\text{Cov}}\{f_{\mathbf{w}_k}, f_{\mathbf{w}_{k'}}|x\}} + O(\Delta^2), \end{aligned} \quad (15)$$

where $\text{Var}\{f_{\mathbf{w}_k}|x\} = \mathbb{E}_{\mathbf{w}_k}[(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k}[f_{\mathbf{w}_k}(x)])^2]$; $\text{Cov}\{f_{\mathbf{w}_k}, f_{\mathbf{w}_{k'}}|x\} = \mathbb{E}_{\mathbf{w}_k, \mathbf{w}_{k'}}[(f_{\mathbf{w}_k}(x) - \mathbb{E}_{\mathbf{w}_k}[f_{\mathbf{w}_k}(x)])(f_{\mathbf{w}_{k'}}(x) - \mathbb{E}_{\mathbf{w}_{k'}}[f_{\mathbf{w}_{k'}}(x)])]$; $\text{Var}\{f_{\{\mathbf{w}_k\}_{k=1}^K}|x\}$ denotes the mean variance of client models when given a sample $(x,y) \in \mathcal{D}$; $\text{Cov}\{f_{\mathbf{w}_k}, f_{\mathbf{w}_{k'}}|x\}$ denotes the mean covariance between two client models when given a sample $(x,y) \in \mathcal{D}$.

With Corollary 2, the mean covariance $\overline{\text{Cov}}\{f_{\mathbf{w}_k}, f_{\mathbf{w}_{k'}}|x\}$ and the mean variance $\overline{\text{Var}}\{f_{\{\mathbf{w}_k\}_{k=1}^K}|x\}$ become equivalent when client models are identically distributed (i.e., client models are trained on homogeneous datasets with the same

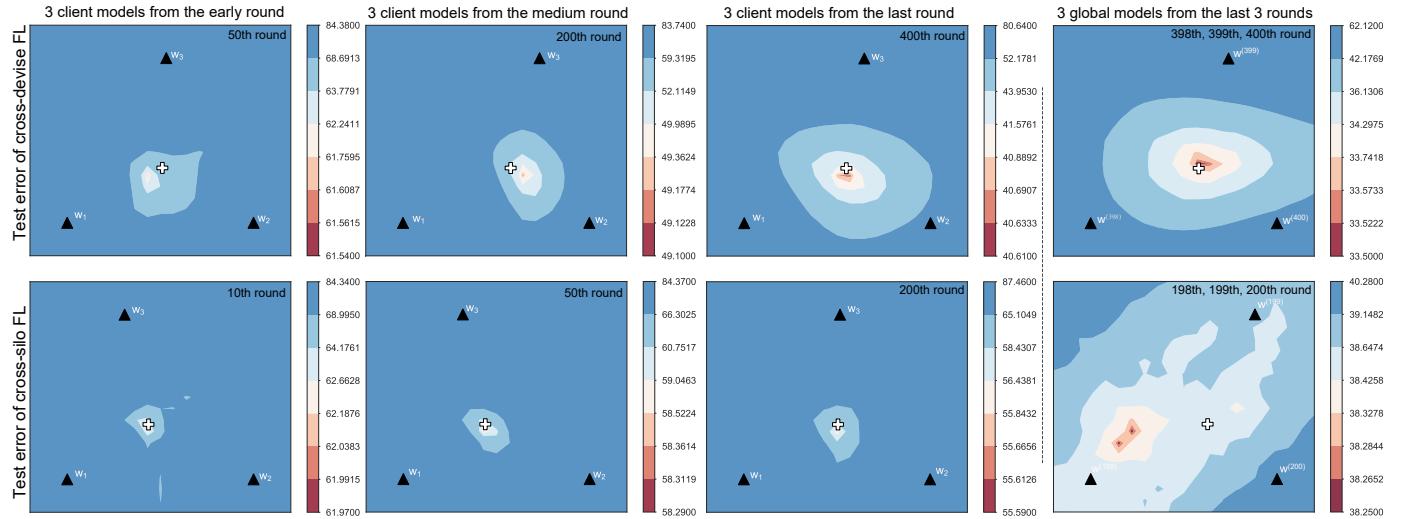


Figure 9: Error landscape visualization with three models on global test set in cross-device and cross-silo FL.

training configurations). In this case, $\overline{\text{Var}}\{f_{\{\mathbf{w}_k\}_{k=1}^K} | x\} = \overline{\text{Cov}}\{f_{\mathbf{w}_k, \mathbf{w}_{k'}} | x\} = \text{Var}\{f_{\mathbf{w}_k} | x\}$, and thus the effect of the variance and covariance factors in Theorem 1 is the same as that of a single client model, making the aggregation of more client models in FMA useless.

APPENDIX D: EXPERIMENT SETTINGS

Models

Table 4 outlines the models used in all the experiments, including validation, test, and ablation experiments. To isolate the controversial effect of BN layers on FL, we follow [66] to replace the BN layer with the GroupNorm layer in all experiments. Our models adhere to the architectures reported in the respective baseline works for a fair comparison. Specifically, for the experiments conducted on the FMNIST and CIFAR-10 datasets, we employ a CNN model consisting of two 5x5 convolutional layers followed by 2x2 max pooling and two fully connected layers with ReLU activation. This architecture aligns with the model used in [1], [26]. For the CIFAR-10/100 experiments, we adopt the ResNet-18 architecture [63] with a linear projector. This choice is consistent with the models employed in [29] and [60]. Lastly, for the Digit Fives dataset, we employ a CNN model with three 5x5 convolutional layers followed by five GroupNorm layers.

Baseline Settings

Table 5 provides the additional hyper-parameters specific to different baselines. These hyper-parameters are chosen based on the setups reported in the respective baseline works. Here is a brief description of the role of hyper-parameters in each baseline:

- FedProx and FedFA: These baselines modify the loss function by adding a proximal term at the client side. The best coefficient of the proximal term is selected from the given range [0.1, 0.01, 0.001]. This coefficient controls the trade-off between the proximal and main loss functions.

- FedASAM: This baseline utilizes the SAM technique as the client loss function. The hyper-parameters η_{SAM} and ρ_{SAM} control the noise introduced in SAM, affecting the exploration-exploitation trade-off during optimization.
- FedAdam and FedYogi: These baselines apply adaptive momentum to the global update on the server side. The hyper-parameters include the server learning rate (lr) η , decay parameters β_1, β_2 , and the degree of adaptivity τ_1 . These hyper-parameters control the adaptation of the server-side optimizer's momentum over time.
- FedGMA: This baseline employs the AND-Masked gradient update based on the masking threshold ϵ . The masking threshold determines the sparsity level in the gradient updates and improves the flatness of the global model.

It is noteworthy that FedAvg and FedNova do not require additional hyper-parameters beyond the standard optimization parameters.

Settings of Visualization and Validation experiments

Table 6 provides the specific setups for all visualization and validation experiments. The experiments are performed using PyTorch on a single node of the High-Performance Computing platform. The node has 4 NVIDIA A30 Tensor Core GPUs, each with 24GB of memory. The setups include hyper-parameters and configurations specific to each experiment, such as the model architecture, dataset, number of clients, batch size, learning rate, optimizer, and other relevant details.

Settings of Test Experiments

Table 7 provides the setup details for all test experiments conducted in this work. The table includes information such as the client number, participation rate, local epoch number, lr, decay scheme, total communication rounds, model, and specific setups of IMA. Please refer to Table 7 for a comprehensive overview of the experimental configurations.

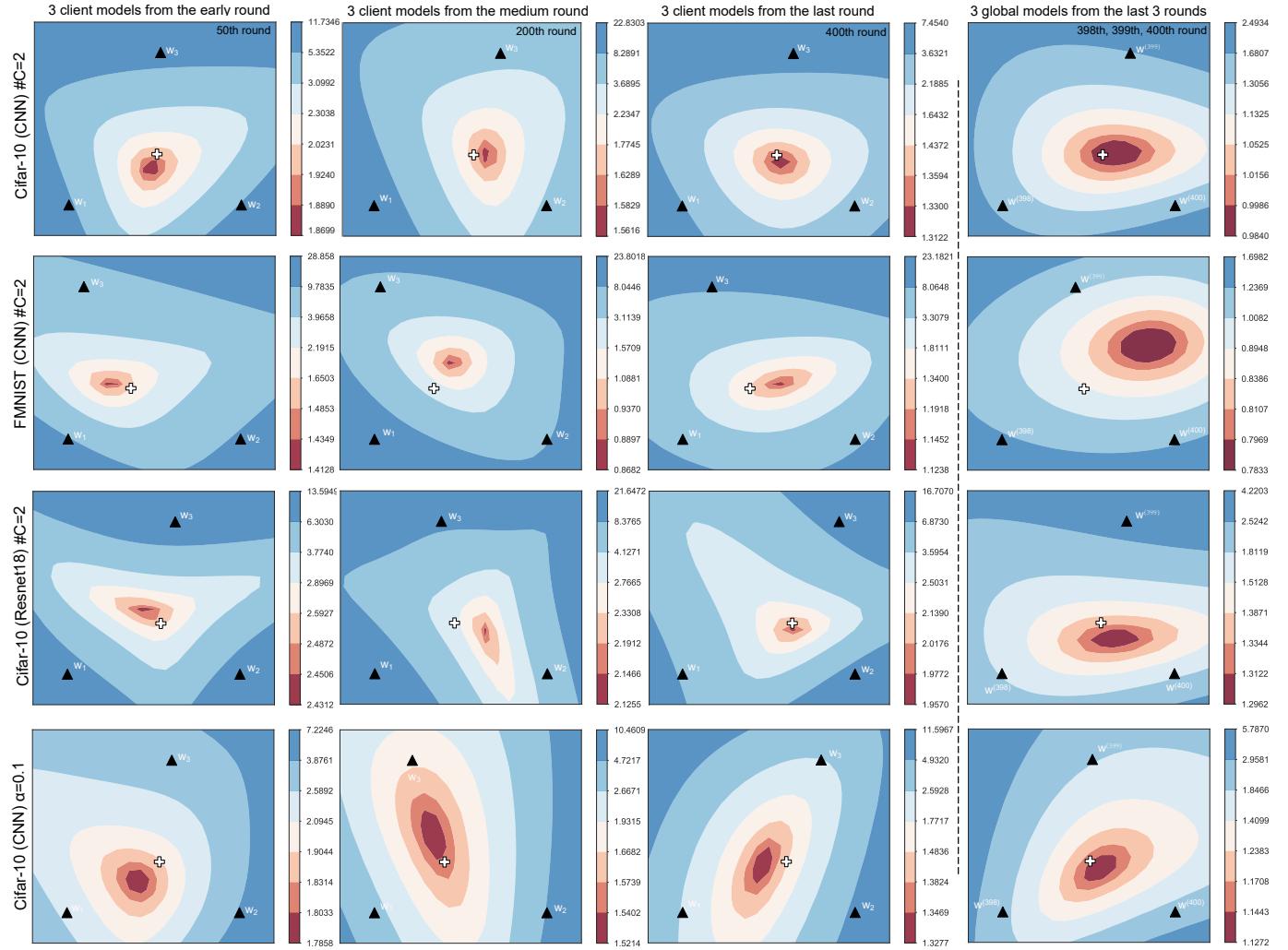


Figure 10: Loss landscape visualization of client and global models in the early, medium, and final stages of FL under different setups: **The first row** illustrates the visualization of CNN models trained under data heterogeneity $\#C = 2$ on CIFAR-10; **The second row** illustrates the visualization of CNN models trained under $\#C = 2$ on FMNIST; **The third row** illustrates the visualization of ResNet18 trained under $\#C = 2$ on CIFAR-10; **The fourth row** illustrates the visualization of CNN models trained with a Dirichlet parameter $\alpha = 0.1$ on CIFAR-10. Each row shows the loss landscape visualization for local and global models in the early, medium, and final stages.

Table 4: Parameter settings for all the models used in our experiments. Group normalization (GN) layers split the input channels into two groups in all models. Conv2d(a, b, c) represents a convolutional layer with a input channels, b output channels, and $c \times c$ kernel sizes. FC(a, b) denotes a fully connected (FC) layer with a input channels and b output channels. MaxPool2D(a, b) is a max pooling layer with dimensions $a \times b$, and ReLU refers to the ReLU activation function. The backbone refers to the framework excluding the last layer. For example, in ResNet18, the backbone corresponds to its feature extractor. ResNet18.FC(a, b) represents ResNet18 with the classifier replaced by an FC(a, b) layer.

Block	Dataset (Used Model)					
	FMNIST (CNN)	(CNN)	CIFAR-10/100 (VGG11)	(ResNet18)	Digit Five (CNN)	PACS (AlexNet)
1	Conv2d(1,32,5) ReLU,MaxPool2D(2,2)	Conv2d(3,64,5) ReLU,MaxPool2D(2,2)	Backbone of VGG11 with GN	Backbone of ResNet18 with GN	Conv2d(3,64,5,1,2) GN(2,64) ReLU,MaxPool2D(2,2) Conv2d(64,64,5,1,2) GN(2,64) ReLU,MaxPool2D(2,2) Conv2d(64,128,5,1,2) GN(2,128) ReLU,MaxPool2D(2,2) FC(6272, 2048) ReLU FC(2048,128) FC(128,10)	Backbone of AlexNet with GN
	Conv2d(1,32,5) ReLU,MaxPool2D(2,2)	Conv2d(64,64,5) ReLU,MaxPool2D(2,2)	VGG11.FC(512,128)	ResNet18.FC(512,128)	ResNet18.FC(4096,512)	
3	FC(512,384) ReLU	FC(1600,384) ReLU	FC(128,10)	FC(128,10)		FC(512,10)
4	FC(384,128)	FC(384,128)				
5	FC(128,10)	FC(128,10)				
6						

Table 5: Hyper-parameter setups for all the baselines. FedAvg and FedNova are excluded as they do not require additional hyper-parameters. In FedASAM, η_{SAM} and ρ_{SAM} control the noise to affect the exploration-exploitation trade-off during optimization. For FedAdam and FedYogi, the server learning rate η , momentum decay parameters β_1, β_2 , and the adaptivity degree τ_1 control the adaptation of the server-side optimizer's momentum over time. In FedGMA, the masking threshold ϵ determines the sparsity level in the gradient updates to improve the flatness of the global model loss. Our code for these baselines follows the hyperlinks provided below.

Hyper-parameter	FedProx	FedASAM	FedFA	FedAdam	FedYogi	FedGMA
coefficient of proximal term: Best from [0.1, 0.01, 0.001]	CNN models: $\rho_{SAM}=0.7$, $\eta_{SAM}=0.2$ Other models: $\rho_{SAM}=0.2$, $\eta_{SAM}=0.05$	coefficient of proximal term: Best from [0.1, 0.01, 0.001]	coefficient of anchor updates: 0.9	$\eta = 0.01$	$\epsilon = 0.8$	
Refer	-	Authors' codes	Authors' codes	Benchmark:Flower	Reproduces codes	

Table 6: Setup of all visualization and validation experiments in this work. Each row details the specific experiment setup corresponding to the figures. Cross-device FL and cross-silo FL indicate that some and all clients participate in each training round, respectively. $\#C = 2$ implies that each client holds two class shards of the training dataset, with each shard containing 250 samples. Moreover, $\alpha = 0.1$ represents the splitting of the training dataset using a Dirichlet distribution $Dir(\alpha) = 0.1$ as in [59].

	FL	client number	client participation	local epoch	local batch	lr (momentum)	lr decay per round	round	dataset	heterogeneous data(C:class)	model
Figure 1, 9	Cross device	100	0.1	5	50	0.01(0.9)	0	400	CIFAR-10	$\#C = 2$	CNN
Figure 8, 9	Cross silo	10	1	5	50	0.01(0.9)	0	200	CIFAR-10	$\#C = 2$	CNN
Figure 10	Cross device	100	0.1	5	50	0.01(0.9)	0	400	CIFAR-10, FMNIST	$\#C = 2, \alpha = 0.1$	CNN, ResNet18
Figure 2, 3,4	Cross silo	10	1	5	50	0.01(0.9)	0/0.01	400	CIFAR-10	$\#C = 2$	CNN
Figure 5	Cross device	100	0.1	5	50	0.01(0.9)	0	400	CIFAR-10	$\#C = 2$	CNN

Table 7: Setup of all test experiments in Table 1. Each row shows the specific experiment setup for the corresponding datasets, including the FL, IMA, and tested models. For the IMA setup, the columns "IMA windows" and "starting IMA" denote P and $t_s = 0.75R$ in (5), respectively.

Dataset	client number	client participation	local epoch	local batch	lr (momentum)	lr decay per round	round (R)	IMA windows	IMA lr decay	starting IMA	model
FMNIST	100	0.1	5	50	0.01(0.9)	0.01	300	5	0.03	225	CNN
CIFAR-10	100	0.1	5	50	0.01(0.9)	0.01	400,400	5	0.03	300,300	CNN,ResNet18
CIFAR-100	100	0.1	5	50	0.01(0.9)	0.01	300,400	5	0.03	225,300	VGG11,ResNet18
Digit Five	100	0.1	5	50	0.01(0.9)	0.01	200	5	0.03	150	CNN w/ GN
PACS	80	0.2	5	50	0.01(0.9)	0.01	400	5	0.03	300	AlexNet w/ GN