Window-based Model Averaging Improves Generalization in Heterogeneous Federated Learning

Window-based Model Averaging Improves Generalization in Heterogeneous Federated Learning

Debora Caldarola[1] Barbara Caputo[1] Marco Ciccone[1]
[1]Politecnico di Torino
name.surname@polito.it

Abstract

Federated Learning (FL) aims to learn a global model from distributed users while protecting their privacy. However, when data are distributed heterogeneously the learning process becomes noisy, unstable, and biased towards the last seen clients' data, slowing down convergence. To address these issues and improve the robustness and generalization capabilities of the global model, we propose WIMA *(**Window-based Model Averaging**). WIMA aggregates global models from different rounds using a window-based approach, effectively capturing knowledge from multiple users and reducing the bias from the last ones. By adopting a windowed view on the rounds,* WIMA *can be applied from the initial stages of training. Importantly, our method introduces no additional communication or client-side computation overhead. Our experiments demonstrate the robustness of* WIMA *against distribution shifts and bad client sampling, resulting in smoother and more stable learning trends. Additionally,* WIMA *can be easily integrated with state-of-the-art algorithms. We extensively evaluate our approach on standard FL benchmarks, demonstrating its effectiveness.*

1. Introduction

Federated Learning (FL) [34] is a distributed machine learning framework aiming at learning a shared global model from edge users' data (the *clients*) while ensuring their privacy. Instead of centrally collecting their data, federated training is based on the exchange of model parameters between clients and the server. The actual training is performed on the client side, and the updates are later aggregated on the server side. In real-world scenarios, the number of clients typically reaches billions [18], and their data collection depends on numerous factors such as geographical location [15, 8, 43, 35], or personal habits [7, 52]. For instance, autonomous vehicles may collect images and videos of largely different cities with varying weather and light conditions [8, 43]. This results in highly diverse local data distributions, creating inherent *statistical heterogeneity* within the context of FL [27, 18]. As a consequence, training a global model capable of addressing the overall underlying distribution becomes particularly challenging: as only a fraction of clients participates in each round, the convergence speed is drastically reduced [30, 19], the learning trend becomes noisy and unstable [20, 2], the clients' biased updates drift the model from its convergence points [20, 28, 1], and the global model suffers from *catastrophic forgetting* [23, 2], resulting in the loss of knowledge acquired from previous users as training progresses. Most of the approaches addressing these issues focus on client-side training: several methods [28, 20, 1, 36, 48] regularize the local objective to reduce the *client drift*, while others leverage momentum to incorporate knowledge from previous updates and lead the local optimization onto the path defined by the global models across rounds [19, 49, 51, 21, 32]. More theoretical studies reveal that learning rate decay is fundamental in local training to reach global convergence in heterogeneous settings [30, 53, 4, 31]. Building upon [9, 25], another promising research direction focuses on the sharpness of reached minima as an indicator of the model's generalization ability, and explicitly guides the local updates towards flatter minima [2, 37, 46, 45]. Less attention has been given to server-side aggregation. The de-facto standard approach for merging models is FedAvg [34], where the updated parameters are averaged based on the number of samples seen by each client. Recent studies [14, 38] reveal that FedAvg aligns with a step in the optimization path defined by SGD (Stochastic Gradient Descent) [41] with a unitary learning rate, and suggest that using server-side momentum or adaptive optimizers could be beneficial when dealing with heterogeneous scenarios. Differently, [2] introduces Stochastic Weight Averaging (SWA) [16] to ensemble global models across rounds in the later stages of training with the goal of improving stability and generalization. A significant limitation of this approach lies in its impracticality during the early stages of training, rendering it challenging to deploy in real-world contexts.

In addition, less attention has been given to research in FL related to vision domains [2].

In this paper, we aim at building a robust and stable global model without incurring in additional communication or client-side computational burden, with a specific focus on vision tasks. Building upon the insights of [2], we propose **Window-based Model Averaging** (WIMA), a method for aggregating global models from the initial stages of training. In particular, WIMA leverages a server-side window-based approach that averages the last W global models. This strategy helps to mitigate the drift introduced by the last seen clients, preserving information from previous users with reduced forgetting. The model built with WIMA is robust towards both distribution shifts and bad client sampling and can be easily applied on top of any existing state-of-the-art FL algorithm.

Our main contributions are summarized as follows:

- We propose WIMA which averages the last W global models on the server side, building a model more robust towards distribution shifts and bad client sampling from the earliest stages of training.
- We show that averaging these models is equivalent to using learning rate decay in the server-side aggregation process.
- We evaluate WIMA's performances on multiple FL datasets and observe smoother learning trends. Furthermore, we show that the use of WIMA helps narrow the gap with runs that involve higher client participation rates.

2. Related works

Federated settings FL [34] enables the training of a shared model among edge devices or institutions while ensuring the privacy of their sensitive data. Real-world scenarios comprise cross-silo and cross-device FL [18]. The former involves silos like companies or hospitals in the training process, with access to extensive data from multiple clients (*e.g.*, patients). In contrast, the cross-device scenario utilizes billions of edge devices, such as smartphones, which possess limited data and computational resources. Moreover, their data is often *biased* towards various distributions, influenced by factors such as capturing devices, personal habits, and geographical locations [15, 18, 8, 43, 7]. Lastly, the devices are not always online and reachable, resulting in only a fraction of them available for training. Thus, it is essential to account for constraints related to resource limitations, communication capabilities, and small skewed datasets when designing federated algorithms [27, 18]. In this work, we focus on the cross-device setting, aiming to avoid adding complexity for the resource-constrained clients while improving the robustness of the global model.

Heterogeneity in Federated Learning Federated training is based on communication rounds, during which clients and server exchange the global model updated parameters, with the server never accessing the local data. On the server side, the updates are aggregated, usually using a weighted average as introduced by the de-facto standard FedAvg [34]. While being effective in homogeneous scenarios, FedAvg fails at achieving comparable performances in heterogeneous cross-device ones [18, 27]. In particular, local different distributions lead to the so-called *client drifts* [20], *i.e.* the local models converging towards different solutions in the loss landscape, making server-side aggregation more challenging. As a consequence, convergence is slowed down [20, 30], the learning trend becomes noisy and unstable [2] and the global model suffers from catastrophic forgetting of the knowledge acquired by previously involved clients [23, 44]. As a first step to overcome these issues, [38] explains that applying FedAvg on the global updates is equivalent to globally using SGD with learning rate 1, and shows that adaptive optimizers can help address heterogeneous scenarios more effectively. To reduce the client drift, FedProx [28] introduces a regularization term in the local objectives, while SCAFFOLD [20] leverages stochastic variance reduction [39] and FedDyn [1] aligns local and global stationary points at convergence. However, [48] shows that FedDyn is often prone to parameter explosion in particularly skewed and cross-device settings, introducing AdaBest as a solution. Other approaches use a momentum term [47] to preserve the history of the previous updates and reduce the bias towards the last fraction of selected clients [49, 19, 32]. In particular, FedAvgM [14] uses momentum on the server-side aggregation, while FedACG [21] and FedCM [51] leverage a momentum term to guide local updates in the direction followed by global models. MIME [19] combines both stochastic variance reduction and momentum so that local updates mimic the behavior of training on *i.i.d.* data. Since our approach only looks at the server-side aggregation of global models, it can be easily combined with any of these methods. Furthermore, [30, 53, 4, 31] highlight that employing learning rate decay in the client-side training is essential to achieve convergence in heterogeneous scenarios. In this study, we illustrate that averaging the model parameters at different rounds is equivalent to applying global decay in the SGD steps of FedAvg, resulting in notable improvements in stability during the training process. Lastly, the authors in [33] reveal that the classifier is the network component most affected by local distribution shifts. In this context, our work demonstrates how WIMA improves the backbone's ability to extract better features, consequently enhancing the stability of the classifier.

Building upon [9, 25, 16], other works look at the generalization of the global model through the lens of the loss

landscape, linking it with convergence to flat minima. FedSAM [2, 37] uses Sharpness-Aware Minimization (SAM) [9] optimizing both loss value and sharpness, and FedSpeed [46] follows a similar approach, while FedSMOO [45] introduces the concept of global sharpness. Following this line of research, [2] shows that using Stochastic Weight Averaging (SWA) [16] on the server-side to ensemble global models leads to more robust and stable results with significant gains in performances and generalization. SWA achieves this by averaging the weights obtained by SGD during its optimization path, utilizing a cyclic learning schedule to explore broader regions in the weight space. However, it's important to note that SWA can only be effectively applied near convergence; otherwise, it may hinder the training process. In this work, we address this last issue emerging with SWA by leveraging a window-based approach. Instead of collecting *all* global models from the beginning, WIMA only averages the last k ones, resulting in significant improvements and overcoming the issues faced with SWA.

Model ensembling for robustness Our work is inspired by research conducted outside the federated scenario, which demonstrates the effectiveness of model ensembling in enhancing accuracy and robustness. We leverage these valuable insights to face challenges proper of the heterogeneous federated scenarios, aiming at improving the performance of the learned models. In their work, [26] demonstrate that ensembling predictions in the output space can lead to performance boosts due to the diversity of the networks. Perhaps surprisingly, [10] reveals that randomly initialized networks independently trained on the same task are connected by simple curves of low-loss, and proposes FGE (Fast Geometric Ensembling) to ensemble predictions at the end of weight space exploration. Finally, [16] shows that solutions found by FGE are found on the edge of the most desirable ones, and presents SWA to ensemble models in the weight space and move towards the center of the minimum. However, SWA is most effective near convergence, *e.g.* after 75% of training was performed. To speed up convergence and reduce training hours for large models, [17] proposes LAWA (Latest Weight Averaging), focusing on the middle stages of training. LAWA averages the last checkpoints found at the end of each epoch. Our approach draws inspiration from both [16, 17], incorporating their intuitions into the federated scenario. By doing so, we aim to improve the performance of federated learning in the presence of heterogeneous data distributions.

3. Window-based Model Averaging

In this Section, we provide details regarding the objectives of the federated training in cross-device settings (Sec. 3.1) and introduce the specifics of WIMA (Sec. 3.2).

3.1. Problem formulation

The goal of training in FL is to learn a global model $f(w) : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the input space (*e.g.*, images), $\mathcal{Y}$ the output space (*e.g.*, labels), and $w \in \mathbb{R}^d$ the model parameters. Training proceeds over T communication rounds and is distributed among a set of devices $\mathcal{S}$ (*i.e.*, *clients*), having access to local private datasets $\mathcal{D}_i = \{(x_j, y_j) | \, x_j \in \mathcal{X}, \, y_j \in \mathcal{Y}, \, j \in [N_i], \, i \in \mathcal{S}\}$, where $N_i = |\mathcal{D}_i|$. We define the overall number of clients $|\mathcal{S}| =: K$ to ease the notation. The global objective is

$$\min_{w \in \mathbb{R}^d} F(f_1(w), f_2(w), \dots, f_K(w)), \tag{1}$$

where $F(\cdot)$ is the aggregating function and $f_i \, \forall i \in \mathcal{S}$ is the local objective (*e.g.*, cross-entropy loss). In this work, $F(\cdot)$ is defined by FedAvg as

$$\min_{w \in \mathbb{R}^d} \sum_{i \in \mathcal{S}} \frac{N_i}{\sum_{j \in \mathcal{S}} N_j} f_i(w_i), \tag{2}$$

where w_i are the locally updated parameters. At each round $t \in [T]$, this minimization problem translates into performing a weighted average of the parameters updated by the subset of selected clients $\mathcal{S}^t$. Additionally, [38] shows that the FedAvg global update can be generally seen as one step of SGD with unitary learning rate (FedOpt), *i.e.*

$$w^{t+1}_{\text{FedAvg}} = \sum_{i \in \mathcal{S}^t} \frac{N_i}{N} w_i^t = w^t - \eta_s \sum_{i \in \mathcal{S}^t} \frac{N_i}{N} (w^t - w_i^t), \tag{3}$$

where $N = \sum_{i \in \mathcal{S}^t} N_i$ and η_s is the server-side learning rate, equal to 1 in FedAvg. The difference $w^t - w_i^t =: \Delta w_i^t$ defines the i-th client's pseudo-gradient, and their average the global pseudo-gradient Δw^t at round t. The local updates w_i are usually computed using SGD. The server-side update can be also generalized for a generic optimizer as $w^{t+1} = w^t - \text{SERVEROPT}(w^t, \Delta w^t, \eta_s, t)$, where SERVEROPT indicates any optimizer, *e.g.* SGD, Adam [22], AdaGrad [6].

In realistic settings, local datasets likely follow different distributions, *i.e.* $\mathcal{P}_i \neq \mathcal{P}_j \, \forall i, j \in \mathcal{S}$, resulting in local updates directing towards distinct minima in the typically non-convex loss landscape. This leads to unfavorable behavior, *e.g.* noisy and unstable learning trends, slowed down convergence (Fig. 1). In addition, as only a fraction of client $|\mathcal{S}^t| \ll K$ is selected at each round t, the resulting model is extremely biased towards the just seen distributions $\mathcal{P}_i \, \forall i \in \mathcal{S}^t$ [2, 33], leading to catastrophic forgetting.

3.2. WIMA for Federated Learning

To overcome the instability and bias proper of training in heterogeneous cross-device federated scenarios, in this work, we introduce ***W**indow-based **M**odel **A**veraging*

(WIMA). Defined a window size of W rounds, at the end of round $t \geq W$, WIMA averages the last W global models built using FedAvg as:

$$w_{\text{WIMA}}^{t+1} = w_{\text{WIMA}}^{t'+W} := \frac{1}{W} \sum_{\tau=t'}^{t'+W-1} w_{\text{FEDAVG}}^{\tau+1}, \quad (4)$$

where $t' = t + 1 - W$ is the first round comprised in the window frame. The rationale behind this approach is to enhance robustness of the global model towards distribution shifts across rounds and diminish bias towards the last-seen clients by averaging models that are still experiencing significant changes. By considering the last W rounds, we retain sufficient history to stabilize the model without hindering the training process, as observed with SWA.

3.2.1 Unveiling the window contents

We now try to answer the question *"What information is stored inside the window?"* To do so, we reformulate Eq. 4 using the updates provided in Eq. 3:

$$\begin{aligned} w_{\text{WIMA}}^{t'+W} &= \frac{1}{W} \sum_{\tau=t'}^{t'+W-1} w_{\text{FEDAVG}}^{\tau+1} && \text{(Eq. 4)} \\ &= \frac{1}{W} \sum_{\tau=t'}^{t'+W-1} \sum_{i \in \mathcal{S}^\tau} \frac{N_i}{N} w_i^\tau && \text{(FedAvg in Eq. 3)} \\ &= \frac{1}{W} \sum_{\tau=t'}^{t'+W-1} \left(w^\tau - \eta_s \sum_{i \in \mathcal{S}^\tau} \frac{N_i}{N} (w^\tau - w_i^\tau)\right). && (5) \end{aligned}$$

By unraveling the summation over the last W rounds and writing each update using Eq. 3, we find out that the WIMA model's update is equivalent to

$$w^{t'} - \eta_s \sum_{\tau=t'}^{t'+W-1} \frac{t'+W-\tau}{W} \sum_{i \in \mathcal{S}^\tau} \frac{N_i}{N} (w^\tau - w_i^\tau), \quad (6)$$

or more in general

$$w^{t'} - \sum_{\tau=t'}^{t'+W-1} \frac{t'+W-\tau}{W} \text{SERVEROPT}(w^\tau, \Delta w^\tau, \eta_s, \tau). \quad (7)$$

The term $^{t'+W-\tau}/_{W}$ tends to 1 when $\tau = t'$, *i.e.* at the beginning of the queue, and to $^{1}/_{W}$ when $\tau = t' + W - 1$, *i.e.* in the last round. Thus, Eq. 7 can be interpreted as $W - 1$ SGD steps starting from the initial model $w^{t'}$ with a ***learning rate decay*** that depends on the position in the queue, given by $^{t'+W-\tau}/_{W}$. Indeed, WIMA assigns higher significance to previous updates, as they are perceived as more stable, while also integrating new knowledge at a rate proportional to the window size W. This sets it apart from methods like momentum, which prioritize more recent updates. Additional details can be found in Appendix A.

Table 1: Datasets statistics. Clf-X indicates the classification task over X classes, while NCP stands for Next Character Prediction.

Dataset	Distribution	Task	Clients	Imbalance
CIFAR10	$\alpha = 0, 0.05$	Clf-10	100	✗
CIFAR100	$\alpha = 0, 0.5$	Clf-100	100	✗
	PAM	Clf-100	500	✗
FEMNIST	NIID	Clf-62	3,400	✓
GLDV2	NIID	Clf-2,028	1,262	✓
SHAKESPEARE	NIID	NCP	715	✓

4. Experiments

In this Section, we provide numerical results on the application of WIMA to different heterogeneous federated scenarios. Sec. 4.1 informs on datasets used, model architectures, and training details. Final results and comparison with state-of-the-art approaches can be found in Sec. 4.2, while Sec. 4.3 studies WIMA more in depth.

4.1. Implementation details

Here we provide a detailed description of the experimental settings. Large-scale experiments were performed using an NVIDIA DGX A100, while the others run on one NVIDIA GeForce GTX 1070. The code was built starting from the FedJAX framework [40]. All runs are averaged over 3 seeds.

4.1.1 Datasets

Table 1 summarizes the information on the used datasets, chosen among common FL benchmarks. As for vision tasks, we focus on classification and use the federated CIFAR10, CIFAR100 [24] and FEMNIST [3]. We introduce large-scale experiments on LANDMARKS-USER-160K [15], the federated version of GOOGLE LANDMARKS V2 [50], which we will refer to as GLDV2 for short. To further prove the wide applicability of our method, we additionally test it on SHAKESPEARE [3] for the next character prediction task. The α value in Table 1 refers to the parameter of the latent Dirichlet's distribution applied to the labels, as proposed by [14]. A smaller value of α identifies a more skewed setting, with $\alpha = 0$ being its extreme scenario in which each client only sees one class. CIFAR100/PAM leverages the Pachinko Allocation Method [29] instead. More details can be found in [38]. FEMNIST is split according to the writer's information, while clients in SHAKESPEARE correspond to characters in Shakespearean plays and each user is the author of the picture in GLDV2. Images are pre-processed using standard data augmentation techniques, *e.g.* random crop, horizontal flip.

4.1.2 Models

We use a ResNet20 [11] on all the distributions of the CIFAR datasets, substituting Batch Normalization with Group Normalization layers, as suggested by [13]. For FEMNIST and SHAKESPEARE we use the architectures proposed in FedJAX, a 2-layer Convolutional Neural Network and an LSTM [12] network respectively, following [34, 3]. As done in [15, 2], we train MobileNetV2 [42] pre-trained on ImageNet [5] for GLDV2, replacing Batch Normalization layers with Group Normalization ones.

4.1.3 Training details

In all cases, on the server side, we use the standard FedAvg with $\eta_s = 1$ unless otherwise specified and momentum 0, and clients locally train with SGD. Experiments on the Dirichlet's CIFAR datasets are run for $10k$ rounds, selecting 10 clients at each round, *i.e.* with 10% participation rate. On the client side, we select learning rate 0.1 from $\{0.1, 0.01\}$, momentum 0 from $\{0, 0.9\}$, weight decay 0 unless otherwise specified, batch size 100 among $\{32, 64, 100, 128\}$, and train for 1 local epoch chosen from $\{1, 2, 4\}$. For CIFAR100/PAM we train for $10k$ rounds with 20% participation rate, using learning rate 0.05, weight decay $4e$-4, batch size 20, server-side momentum 0.9 from [2]. For FEMNIST we use client learning rate 0.1 from $\{0.1, 0.01, 0.001\}$, momentum 0 from $\{0, 0.9\}$, weight decay 0, batch size 10 from $\{10, 20, 32\}$. We train for $1,500$ rounds with 10 clients per round ($\approx 0.3\%$ participation rate), performing 1 local epoch each. For GLDV2, we follow the setup of [2] except for the batch size equal to 50 and train the model for $3k$ rounds with 10 clients selected at the time. In SHAKESPEARE, local learning rate is 1, momentum 0, weight decay 0, batch size 4, 1 epoch from [19]. Training is spanned over $1,500$ rounds with 10 clients per round ($\approx 1.4\%$ participation rate). The WIMA parameter W is set to 100 for all settings except for GLDV2, where $W = 370$ (see Sec. 4.3 for additional analyses). For all datasets, the reported final results are averaged over the last 100 rounds for increased robustness [2].

4.1.4 SOTA algorithms details

We provide here details on the tuning intervals for the state-of-the-art (SOTA) algorithms used for comparison. We apply WIMA on top of methods proposed for addressing statistical heterogeneity in FL. Looking at momentum-based approaches, we select FedAvgM [14] (server-side momentum $\beta = 0.9$, $\eta_s \in \{0.1, 1\}$), MIME SGD ($\eta_s \in \{0.1, 1\}$) and SGDm [19], *i.e.* with momentum 0.9, MIMELite SGDm [19] ($\eta_s \in \{0.1, 1\}$, momentum 0.9), FedCM [51] ($\alpha_{\text{CM}} \in \{0.05, 0.1, 0.5\}$) and FedACG [21] ($\beta_{\text{ACG}} \in \{0.01, 0.001\}$, $\lambda_{\text{ACG}} \in \{0.8, 0.85, 0.9\}$). We additionally test SCAFFOLD [20], FedProx [28] ($\mu_{\text{PROX}} \in \{0.1, 0.01, 0.001\}$), FedDyn [1] ($\alpha_{\text{DYN}} \in \{0.01, 0.001\}$) and AdaBest [48] ($\mu_{\text{ADABEST}} \in \{0.01, 0.02\}$, $\beta_{\text{ADABEST}} \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$) to reduce the client drift. Lastly, we compare WIMA with SWA applied from 75% of training onwards, for which we use $c \in \{10, 20\}$ and second learning rate equal to $\eta \cdot 10^{-2}$, following [2].

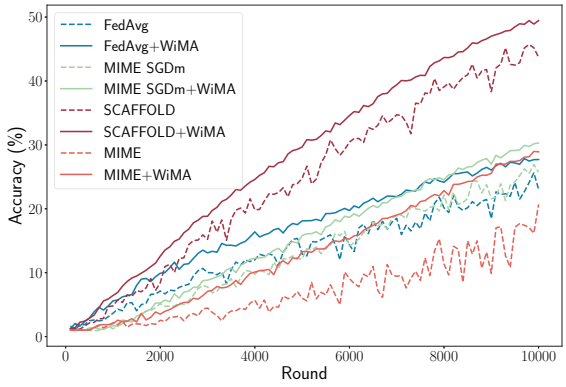


Figure 1: Accuracy trends of different SOTA algorithms on CIFAR100 $\alpha = 0$ across rounds, with and without WIMA (dashed lines). The application of WIMA results in smoother and more stable trends, leading to enhanced robustness and improved performance. Best seen in colors.

4.2. Results

Reducing noise and increasing stability with WIMA. Thanks to the window-based average of global models, WIMA mitigates the negative impact of statistical heterogeneity inherent in cross-device federated settings. As shown in Fig. 1, WIMA effectively smooths the learning trends, resulting in enhanced robustness and reduced instability. Notably, these benefits are observed across all performance levels, with improvements evident in low-performing approaches (*e.g.*, MIME in Fig. 1) as well as the best-performing ones (*e.g.*, SCAFFOLD).

The effective combination of WIMA with SOTA. Table 2 presents the results achieved by combining WIMA with SOTA federated algorithms designed to handle statistical heterogeneity. Looking at standalone algorithms (*i.e.*, w/o WIMA), SCAFFOLD achieves the best performances overall. FedDyn is not able to converge in the most heterogeneous settings, as already shown by [48, 2]. Notably, WIMA enables each method to achieve better final accuracy, showcasing substantial improvements compared to the algorithm without WIMA. The most significant gains are observed on the more challenging CIFAR datasets. In particular, WIMA proves especially beneficial for the worst-performing methods, increasing the final ac-

Table 2: WIMA combined with state-of-the-art FL algorithms. For each configuration, the first column reports the accuracy (%) reached by each standalone method; in the second column, the performance achieved when adding WIMA. Between brackets the improvements introduced by WIMA, underlined the best ones in each dataset. For simplicity, we only report gains in improvements ≥ 1.5. Best overall accuracy in bold.

Algorithm	CIFAR10				CIFAR100						FEMNIST		SHAKESPEARE	
	$\alpha = 0$		$\alpha = 0.05$		$\alpha = 0$		$\alpha = 0.5$		PAM					
		w/ WIMA		w/ WIMA		w/ WIMA		w/ WIMA		w/ WIMA		w/ WIMA		w/ WIMA
FedAvg	64.37	69.95 (↑ 5.3)	68.50	72.69 (↑ 4.2)	23.00	27.91 (↑ 4.9)	31.21	34.45(↑ 3.2)	47.41	48.53	83.59	85.06 (↑ 1.5)	**56.86**	57.74
FedAvgM	73.32	75.72 (↑ 2.4)	73.10	75.30 (↑ 2.2)	24.27	28.77 (↑ 4.5)	31.78	33.97 (↑ 2.2)	55.96	61.63 (↑ 5.7)	85.00	85.26	56.91	57.57
MIME SGD	74.92	80.65 (↑ 5.7)	78.82	82.81 (↑ 4.0)	17.55	29.05 (↑ **11.5**)	27.30	40.37 (↑ **13.1**)	54.33	57.44 (↑ 3.1)	85.37	86.40	56.06	57.43
MIME SGDm	74.58	76.20 (↑ 1.6)	78.39	80.38 (↑ 2.0)	25.78	30.11 (↑ 4.3)	38.42	43.08 (↑ 4.7)	54.62	57.28 (↑ 2.7)	86.67	87.40	54.00	54.68
MIMELite	64.42	67.78 (↑ 3.4)	68.27	71.21 (↑ 2.9)	20.00	24.69 (↑ 4.7)	35.56	39.15 (↑ 3.6)	53.97	60.34 (↑ **6.4**)	**86.82**	**87.51**	52.45	53.01
FedCM	78.83	81.73 (↑ 2.9)	73.94	80.28 (↑ **6.3**)	19.62	25.29 (↑ 5.7)	36.12	40.10 (↑ 4.0)	53.16	54.12	83.88	84.90	38.90	39.29
FedACG	55.27	60.09 (↑ 4.8)	63.20	66.35 (↑ 3.2)	20.09	23.55 (↑ 3.5)	29.74	32.46 (↑ 2.7)	58.88	61.38 (↑ 2.5)	85.73	86.14	56.79	58.03
FedProx	64.25	69.90 (↑ 6.7)	67.82	71.90 (↑ 4.1)	22.59	27.58 (↑ 5.0)	30.70	33.68 (↑ 3.0)	55.91	62.25 (↑ 6.3)	84.50	85.21	55.92	56.71
SCAFFOLD	**81.45**	**83.96** (↑ 2.5)	**83.24**	**85.17** (↑ 1.9)	**45.65**	**49.77** (↑ 4.1)	**50.93**	**53.75** (↑ 2.8)	56.09	57.64 (↑ 1.6)	85.87	86.61	56.68	57.48
FedDyn	N/A	N/A	N/A	N/A	5.88	8.48 (↑ 2.6)	20.88	24.54 (↑ 3.7)	**57.42**	**63.00** (↑ 5.6)	N/A	N/A	54.54	55.09
AdaBest	66.05	73.95 (↑ **7.9**)	71.54	77.42 (↑ 5.9)	24.92	31.41 (↑ 6.5)	37.45	43.81 (↑ 6.4)	54.98	57.57 (↑ 2.6)	84.95	86.02	56.60	**58.12** (↑ 1.5)

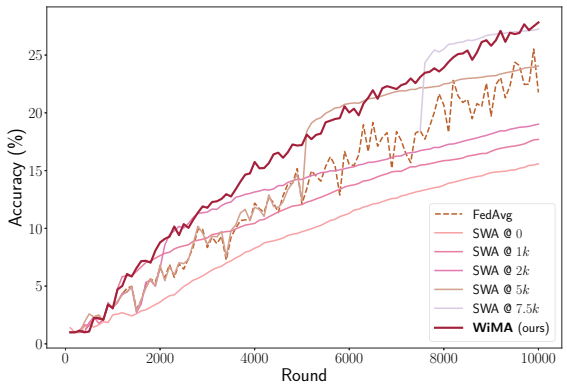


Figure 2: Accuracy trends of WIMA and SWA starting at different rounds on CIFAR100 with $\alpha = 0$, using FedAvg as reference. WIMA has a stable trend from the beginning, leading to final better performances, while SWA suffers from early initialization. Best seen in colors.

curacy by over 11 points for MIME SGD on both α values in CIFAR100. On the other hand, using the aggregation proposed by WIMA is effective even on the overall best-performing SCAFFOLD, or on the less challenging FEMNIST and SHAKESPEARE datasets. Thus, all methods and settings are positively affected by the increased robustness and stability introduced by WIMA.

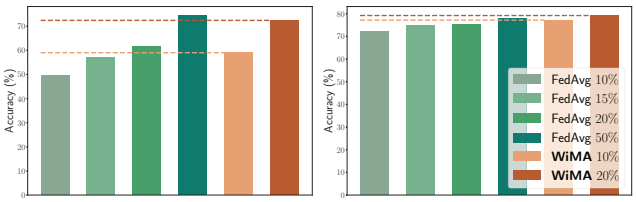
WIMA in large-scale classification. In Table 3, we introduce the results obtained when using WIMA for large-scale classification on GLDV2. Without redundancy and loss of generality, we present the performance of WIMA when integrated with both the standard FedAvg and the best-performing SCAFFOLD. Even in this more complex vision scenario, WIMA achieves large gains in accuracy.

Table 3: Large-scale experiments. Results in test accuracy (%) on GLDV2. Best result in bold.

Algorithm	w/o WIMA	w/ WIMA
FedAvg	58.17	63.05
SCAFFOLD	62.32	**68.30**

WIMA vs SWA. Fig. 2 compares the accuracy trends of WIMA and SWA starting at different rounds. We note that SWA suffers from early initialization, leading to saturation and worse performances than FedAvg, reaching a final accuracy comparable to our method only if close to the last rounds. Differently, thanks to the windowed view of the global models across rounds, WIMA can be applied from the beginning of training, and presents a constantly stable and better trend than SWA.

WIMA allows less client participation. In Fig. 3, we observe that the enhanced generalization capability achieved with WIMA allows narrowing the gap with runs involving higher client participation rates. Specifically, we compare FedAvg training with and without our method, using varying numbers of clients selected at each round on CIFAR10. Experiments are run using batch size 20 to account for more local iterations, highlighting the client drift. WIMA enables the model with a 10% participation rate to attain a final accuracy that is *at least* comparable to the run involving 1.5 times the number of devices with $\alpha = 0$ and twice that number with $\alpha = 0.05$. When 20% of clients are involved instead, WIMA reaches performances comparable ($\alpha = 0$) or better ($\alpha = 0.05$) than FedAvg involving half the devices (50% rate). This result holds significant impor-



(a) $\alpha = 0$ (b) $\alpha = 0.05$

Figure 3: WIMA performances compared with varying client participation rates at each round on CIFAR10 using FedAvg. **a)** WIMA achieves higher accuracy with 10% participation compared to FedAvg with 1.5 times the number of devices per round. WIMA with 20 clients perform similarly to FedAvg with half the clients. **b)** WIMA with 10% rate performs almost on par with FedAvg w/o WIMA selecting 50% of the devices.

tance in cross-device settings, where devices are often unavailable due to factors such as limited battery life, network connectivity issues, and communication overload [18]. The ability to achieve improved results with fewer clients involved aligns favorably with real-life requirements, making it a valuable contribution.

4.3. Ablation study

Studying the window size. The dimension W of the window used by WIMA plays a crucial role in achieving a trade-off between retaining useful historical information and avoiding excessively old data. In Table 4 we compare the accuracy reached with varying values of W on the heterogeneous CIFARs. Smaller W values lead to lower performance as the WIMA model fails to capture sufficient information from the underlying distribution, while excessively large W values slow down training by relying on outdated updates. The optimal results are obtained with $W = 100$ in both cases.

Table 4: WIMA accuracy (%) with varying W on the CIFAR datasets with $\alpha = 0$. Best results in bold.

Window size W	**WIMA Accuracy** (%)	
	CIFAR10	CIFAR100
5	67.25	22.71
10	69.12	25.70
50	69.79	22.75
100	**69.94**	**27.91**
200	68.74	27.72

WIMA outputs better features. We now wonder where WIMA helps the model the most. In particular, with the goal of understanding which part of the architecture our method affects the most, we evaluate its performances when acting only on the feature extractor, or the classifier, *i.e*. the last linear layer of the model. To allow for more client-side finetuning, we use a batch size of 20. The analyses reported in Table 5 demonstrate that WIMA is mainly acting on the feature extractor. Thanks to the more robust and less biased output features, the classifier is consequently able to give more accurate predictions.

Table 5: Accuracy (%) reached when applying WIMA only on the classifier (*clf*), the feature extractor (*feat. extr.*) or all the model parameters (*all*) as reference.

Dataset	α	**WIMA** *clf*	**WIMA** *feat. extr.*	**WIMA** *all*
CIFAR10	0	47.76	59.03	59.53
	0.05	71.01	76.72	78.87
CIFAR100	0	25.13	27.10	27.91
	0.5	36.12	36.29	37.88

5. Conclusions

In this work, we proposed *Window-based Model Averaging* (WIMA) to address the negative impacts of statistical heterogeneity in federated learning scenarios. In particular, our goal is to reduce the noise and instability proper of learning trends of models trained in non-*i.i.d*. federated settings. To addresses these issues, WIMA averages the last W global models built using any server-side optimizer at each round. Thanks to the windowed view of the rounds, we keep sufficient history to stabilize the model without hindering the training process. WIMA can be easily combined with most of the existing state-of-the-art algorithms, significantly improving the performance of each method and leading to smoother and more stable trends. We showed that WIMA mainly affects the backbone of the network, producing better output features and consequently enabling the classifier in giving more accurate predictions. Lastly, WIMA helps narrowing the gap with runs using higher client participation rates, a favorable result for realistic federated settings.

Acknowledgments This project was partially funded by CINI (Consorzio Interuniversitario Nazionale per l'Informatica). Large scale experiments were run using the CINECA infrastructure. We thank the anonymous reviewers for their valuable feedback.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *International Conference on Learning Representations*, 2021. 1, 2, 5

[2] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022. 1, 2, 3, 5

[3] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *Workshop on Data Privacy and Confidentiality*, 2019. 4, 5

[4] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR, 2022. 1, 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 3

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *Advances in Neural Information Processing Systems*, 2020. 1, 2

[8] Lidia Fantauzzo, Eros Fanì, Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11504–11511. IEEE, 2022. 1, 2

[9] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *International Conference on Learning Representations*, 2021. 1, 2, 3

[10] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, 2016. 5

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[13] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020. 5

[14] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *Neurips Workshop on Federated Learning*, 2019. 1, 2, 4, 5

[15] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020. 1, 2, 4, 5

[16] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *Conference on Uncertainty in Artificial Intelligence*, 2018. 1, 2, 3

[17] Jean Kaddour. Stop wasting my time! saving days of imagenet and bert training with latest weight averaging. *arXiv preprint arXiv:2209.14981*, 2022. 3

[18] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1, 2, 7

[19] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *Advances in Neural Information Processing Systems*, 2020. 1, 2, 5

[20] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 1, 2, 5

[21] Geeho Kim, Jinkyu Kim, and Bohyung Han. Communication-efficient federated learning with acceleration of global momentum. *arXiv preprint arXiv:2201.03172*, 2022. 1, 2, 5

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 3

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 2

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 4

[25] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021. 1, 2

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 3

[27] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future

directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1, 2

[28] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 1, 2, 5

[29] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006. 4

[30] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *International Conference on Learning Representations*, 2020. 1, 2

[31] Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. *International Conference on Machine Learning*, 2023. 1, 2

[32] Yixing Liu, Yan Sun, Zhengtao Ding, Li Shen, Bo Liu, and Dacheng Tao. Enhance local consistency in federated learning: A multi-step inertial momentum approach. *arXiv preprint arXiv:2302.05726*, 2023. 1, 2

[33] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021. 2, 3

[34] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 5

[35] Jiaxu Miao, Zongxin Yang, Leilei Fan, and Yi Yang. Fedseg: Class-heterogeneous federated learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8042–8052, 2023. 1

[36] Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Fedadc: Accelerated federated learning with drift control. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 467–472. IEEE, 2021. 1

[37] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, pages 18250–18280. PMLR, 2022. 1, 3

[38] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *International Conference on Learning Representations*, 2021. 1, 2, 3, 4

[39] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016. 2

[40] Jae Hun Ro, Ananda Theertha Suresh, and Ke Wu. FedJAX: Federated learning simulation with JAX. *arXiv preprint arXiv:2108.02117*, 2021. 4

[41] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 1

[42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[43] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 444–454, 2023. 1, 2

[44] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *NeurIPS Workshop*, 2019. 2

[45] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. *International Conference on Machine Learning*, 2023. 1, 3

[46] Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. *International Conference on Learning Representations*, 2023. 1, 3

[47] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 2

[48] Farshid Varno, Marzie Saghayi, Laya Rafiee Sevyeri, Sharut Gupta, Stan Matwin, and Mohammad Havaei. Adabest: Minimizing client drift in federated learning via adaptive bias estimation. In *European Conference on Computer Vision*, pages 710–726. Springer, 2022. 1, 2, 5

[49] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *International Conference on Learning Representations*, 2019. 1, 2

[50] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 4

[51] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021. 1, 2, 5

[52] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018. 1

[53] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019. 1, 2