Overcoming Forgetting Using Adaptive Federated Learning for IIoT Devices with Non-IID Data

Benteng Zhang, *Student Member, IEEE*, Yingchi Mao, *Member, IEEE*, Haowen Xu, *Student Member, IEEE*, Yihan Chen, Tasiu Muazu, Xiaoming He, *Member, IEEE*, and Jie Wu, *Fellow, IEEE*

***Abstract*—In real-world Industrial Internet of Things (IIoT) scenarios, due to the limited storage capacity of IIoT devices, fresh data continuously received by diverse devices will overwrite the outdated data and change the local data distribution. However, state-of-the-art studies have demonstrated that Federated Learning (FL) tends to focus on training with fresh data, and the latest global model may forget the historical update directions (i.e., catastrophic forgetting). This issue can significantly degrade the global model accuracy. Existing methods primarily focus on integrating outdated data characteristics into fresh data but overlook the large parameter update gap between global and local models during global aggregation. This gap can cause the global model updates to deviate from the optimal direction. To this end, we propose a federated adaptive weighted aggregation method based on model consistency (FedAWAC). Specifically, FedAWAC measures the model consistency on devices and dynamically adjusts the aggregation weights of each local model, thereby guiding the global model toward optimal updates. Furthermore, FedAWAC integrates $\mathcal{M}$ historical global models most correlated to the latest global model on the cloud server to overcome catastrophic forgetting. Experiments on 4 different datasets (Non-IID settings) indicate that compared to 5 baselines, FedAWAC can improve global model accuracy by an average of 1.86%, reduce the forgetting rate by an average of 3.93%, and save average memory usage by up to 2.57GB.**

***Index Terms*—Federated learning, industrial internet of things, catastrophic forgetting, global aggregation**

I. Introduction

To provide Artificial Intelligence (AI) services and applications in the Industrial Internet of Things (IIoT) [1], Machine Learning (ML) [2, 3] and Deep Neural Networks (DNNs) [4, 5] are widely used to train deep learning models. As depicted in ***Challenge 1*** in Fig. 1, in real-world IIoT scenarios, IIoT devices come in many types and have limited storage capacity. Moreover, IIoT data (e.g., images and text) is typically Non-IID (non-identically and independently distributed [8]) and decentralized (i.e., distributed across numerous devices [6]). This creates high communication burdens and privacy concerns for centralized ML. Federated Learning (FL) can coordinate numerous IIoT devices to cooperatively train high-quality AI models without sharing the raw data (FL-IIoT) [7], which becomes a promising way to train AI models in IIoT.

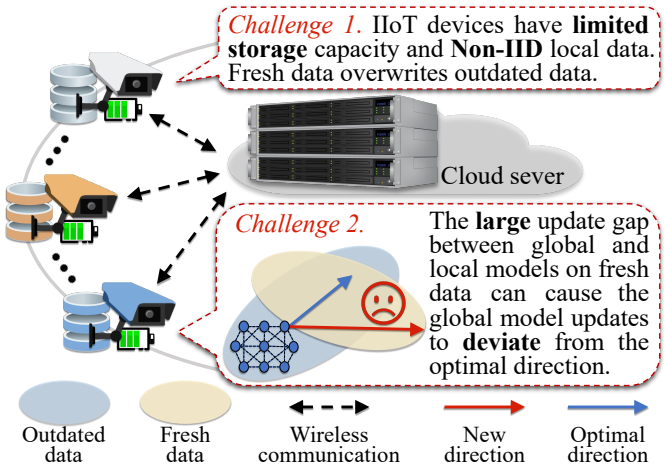

Fig. 1. Two challenges in handling the catastrophic forgetting for FL-IIoT.

However, as FL training goes on, IIoT devices continuously receive fresh data for a new given task, which potentially changes the local data distributions on devices. State-of-the-art studies have demonstrated that FL tends to use fresh data for training and the latest global model may forget the historical update directions (i.e., catastrophic forgetting) [9]. As depicted in ***Challenge 2*** in Fig. 1, there are significant differences in data distribution across IIoT devices in each training round. Consequently, the update directions of the selected local models in each round may differ significantly from its historical update directions (i.e., local forgetting) [10]. However, the global model only aggregates the local model updates from the current training round, which may cause the global model to forget the update directions from historical rounds (i.e., global forgetting) [11]. As the global model updates, catastrophic forgetting will significantly degrade the global model accuracy. Therefore, to improve model accuracy in FL-IIoT with heterogeneous data distributions, IIoT devices should try to minimize the deviation of local model updates from the optimal direction caused by catastrophic forgetting, while continuously receiving fresh data and new given tasks.

Existing methods focus on integrating outdated data char-

This work was supported in part by the Key Research and Development Program of China under Grant 2022YFC3005401; in part by the Key Research and Development Program of China, Yunnan Province under Grant 202203AA080009; and in part by the National Natural Science Foundation of China under Grant 62402246. *(Benteng Zhang and Haowen Xu contributed equally to this work.) (Corresponding author: Yingchi Mao.)*

Yingchi Mao, Benteng Zhang, Haowen Xu, Yihan Chen, and Tasiu Muazu are with the College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China (e-mail: yingchimao@hhu.edu.cn; 230407040003@hhu.edu.cn; 231607010096@hhu.edu.cn; 241307010028@hhu.edu.cn; tmuazu@yahoo.com).

Xiaoming He is with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China (e-mail: hexiaoming@njupt.edu.cn).

Jie Wu is with the Center for Networked Computing, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

TABLE I
COMPARISON BETWEEN RELATED WORKS

Focus	Methods	Optimization ideas	IIoT scenarios
Data Heterogeneity	Wang [18]	Re-weighting strategy	✗
	Zhang [19]	Fine-tuning parameters	✓
	FedNova [21]	Normalized average	✓
Local update	FedDC [22]	Corrected local update	✗
	Scaffold [23]		✗
	GEM [27]	Gradient memory	✗
	GradMA [28]		✗
	FedCurv [13]	Penalty term	✓
	FedCL [14]		✓
	LwF [15]	Regularization term	✓
	FedProx [20]		✓
	Kirkpatrick [26]		✗
	FedGA [42]	Gradient alignment	✓
Global aggregation	FedSeIT [31]	Knowledge distillation	✗
	Wu [32]		✗
	Wang [34]		✓
	iCaRL [17]	Incremental learning	✗
	FCIL [33]		✓
	Generative-Replay [16]	Memory-replay strategy	✓
	Rebufi [17]		✓
	FedCM [25]		✓

acteristics into fresh data during local training to overcome catastrophic forgetting [12]. Replay-based methods, such as FedCM [25], construct storage areas to retain outdated data or historical gradients to retrain the network, but they require significant storage overhead. Additionally, these works (e.g., FedCurv [13], FedCL [14], LwF [15], etc.) avoid a large update gap to parameters that are significantly associated with the global model by adding penalty or regularization terms. For instance, to reduce model inconsistency, FedProx [20] introduces a regularization term by utilizing local information aggregation to constrain local objectives. Inspired by short-term memory generation in the brain, Generative-Replay employs a dual-model structure based on deep generative models and task-solving models [16]. Rebufi *et al.* use short-term and long-term memory to handle recent and all old data [17], respectively. However, the above methods can solve local forgetting, but they overlook global forgetting during global aggregation. If the update direction of the aggregated global model significantly deviates from the historical update direction, this update direction may deviate from the optimal direction and degrade model accuracy. As the final step of FL, global aggregation determines the accuracy of the global model. To our knowledge, we found that overcoming catastrophic forgetting during global aggregation to improve the global model accuracy is still a gap that needs to be filled.

Given the state-of-the-art studies and motivated by these issues above, we aim to effectively suppress global model update deviation caused by catastrophic forgetting during the global aggregation in an adaptive manner, thereby ensuring faster convergence speed and higher model accuracy on both old and new tasks. To this end, we propose a **Fed**erated **A**daptive **W**eighted **A**ggregation method based on model **C**onsistency (FedAWAC). Briefly, FedAWAC can dynamically adjust the aggregation weights of each device and effectively overcome catastrophic forgetting during global aggregation.

The **contributions** of our paper are depicted as follows.

- **IIoT device side.** A model consistency evaluation mechanism is designed to measure model consistency and dynamically adjust the aggregation weight of each device, thereby guiding the global model to update in the optimal direction. When the classifier dimensions of devices are the same, this mechanism is compatible with various FL methods, including heterogeneous models.
- **Cloud server side.** To incorporate historical update directions into global aggregation, we construct a historical model-assisted global aggregation mechanism by utilizing a sliding window to integrate $\mathcal{M}$ historical global models that are most correlated to the latest global model.
- **Effectiveness.** Experiments on 4 different datasets demonstrate that compared to 5 baselines, FedAWAC can improve global model accuracy by an average of 1.86%, reduce the forgetting rate by an average of 3.93%, and save average memory usage by up to 2.57GB.

The remainder of this paper is organized as follows. Section II presents the related work. The proposed system model is shown in Section III. The design details of FedAWAC are discussed in Section IV. The experiments and analysis are given in Section V. Finally, we conclude with Section VI.

II. RELATED WORK

A. Data Heterogeneity in FL-IIoT

Data heterogeneity is one of the objective realities that FL must face in IIoT. When training models using data with highly skewed label distributions, the model's prediction accuracy across different categories varies significantly, potentially leading to overfitting and poor accuracy. Wang *et al.* design a ratio loss based on a re-weighting strategy [18], redistributing variable weights for each category to mitigate the adverse effects of data heterogeneity by focusing on minority classes. For label distribution imbalance based on quantity, Zhang *et al.* fine-tuned hyperparameters [19], sacrificing time costs for considerable performance gains. Moreover, data heterogeneity can lead to accuracy discrepancies between different local models, a phenomenon known as model inconsistency. Additionally, FedNova (normalized averaging) proposes adaptive adjustments to the frequency of local updates to eliminate inconsistencies [21], but it overlooks the impact of local imbalanced data on accuracy across different categories.

B. Catastrophic Forgetting During Local Update

When there are significant differences in the update directions among local models, the update direction of the global model may deviate from the optimal direction after global aggregation, thereby reducing the accuracy and convergence speed of the global model. Inspired by the EWC (elastic weight consolidation) algorithm, FedCurv avoids large parameter update differences associated with the global model through a penalty term [13]. LwF (learning without forgetting) mitigates catastrophic forgetting by feeding the training data of the new task into the outdated network and using its output as synthetic data labels while optimizing both synthetic and real data training [15]. To accelerate the convergence of the global model, FedDC (local drift decoupling and correction) [22]

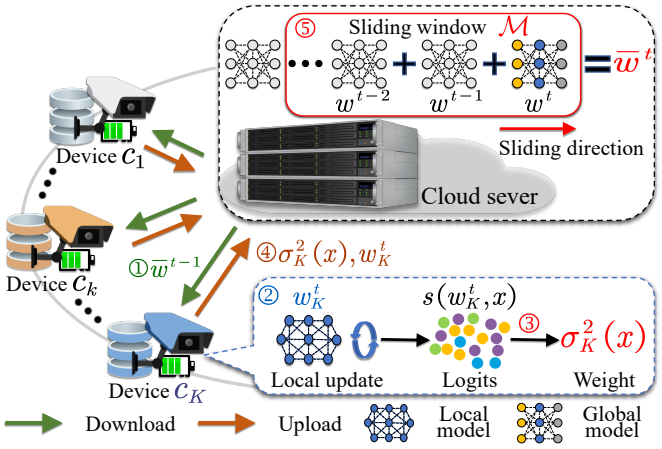

Fig. 2. The overview of FedAWAC. ① Global model delivery. ② Local update. ③ Model consistency evaluation. ④ Upload parameters. ⑤ Global aggregation and global model integration.

and Scaffold (stochastic controlled averaging) [23] correct the update direction of the local models using control variables. Additionally, Kirkpatrick adds a regularization term to the local optimization objective to constrain drastic changes in important model parameters [26]. Inspired by GEM (gradient episodic memory) [27], GradMA (gradient-memory-based accelerated federated learning) restricts the direction of local gradient updates using historical local models and global gradients [28]. To address catastrophic forgetting caused by class imbalance and class absence, FedGA uses a GA (gradient alignment)-based approach to implement label calibration during the model backpropagation [42]. Although the methods above can limit the differences in updates between local models to a certain extent, they overlook the fact that catastrophic forgetting can also occur during global aggregation.

C. Catastrophic Forgetting During Global Aggregation

As the local data distribution on IIoT devices continuously changes over time, in edge scenarios with varying tasks, it becomes challenging for the global model to quickly adapt to the received fresh data distribution while maintaining high accuracy on outdated data [24]. Knowledge distillation methods [29, 30] can train a teacher model using outdated data and use the teacher model as a guideline for training the student model. For example, FedSeIT selectively combines local model parameters and task information [31], which can also be applied to natural language processing. Wu *et al.* used proxy datasets for training [32], and although this method no longer requires a teacher model, it requires access to private datasets, without considering data privacy protection issues. In federated incremental learning, FCIL (federated class-incremental learning) allows edge devices to add new category data continuously [33], and during the local training, FCIL effectively balances the learning weights of fresh and outdated categories. However, FCIL strictly requires task data to be orthogonal in each round, so it only applies to certain extreme scenarios and has poor generalization. To measure the impact of fresh and outdated categories on model accuracy, Wang *et al.* reduce the imbalance between both categories by aligning the model's Logits outputs on both [34]. Moreover, Rebufi *et al.* propose iCaRL (incremental classifier and representation learning) to select exemplars for each class based on feature space [17], but iCaRL has a high computational cost for network parameters and relies on stored samples, making it less feasible. However, the methods above require significant storage and computational overhead, which IIoT devices (with limited storage and computing capacity) are unable to provide. Therefore, effectively mitigating catastrophic forgetting during global aggregation while improving global model accuracy still remains a major challenge in FL-IIoT.

III. System Model

As shown in Fig. 2, our system model consists of N IIoT devices and a cloud server for training image classification models. We assume that all IIoT devices are in the same network with good channel quality (both uplink and downlink) and that all devices remain online throughout the FL training without dropping out. To protect data privacy, only the cloud server has access to an unlabeled public dataset P, and the categories in dataset P are the same as in the local datasets. At the beginning of each training round, the cloud server randomly selects a subset of IIoT devices S^t, containing K devices. Each IIoT device $c_k \in S^t$ has a private local dataset $\mathcal{D}_k$, which contains C types of samples, and each sample is represented as $x \in \mathbb{R}^d$ with a corresponding label y. All IIoT devices are initialized with a global model w^0. In the t-th training round, device c_k receives the latest global model from the cloud server, the updated local model parameters of device c_k are denoted as w_k^t, and the aggregated global model parameters are denoted as w^t. Each IIoT device optimizes its local model based on the global model parameters, and the optimization objective is given by

$$w_k^t = \arg\min \mathbb{E}_{(x,y)\sim\mathcal{D}_k}[\mathcal{L}(w; w^{t-1}, x, y)], \tag{1}$$

where $\mathcal{L}$ is the composite loss function. The IIoT device c_k outputs the Logits vector $s(w_k^t, x)$ for the sample x under the model parameters w_k^t, which is denoted as

$$s(w_k^t, x) = z_k = [z_{1,k}, z_{2,k}, ..., z_{i,k}, ..., z_{C,k}] \in \mathbb{R}^{1\times C}, \tag{2}$$

where $z_{i,k}$ is the Logits output of class i for device c_k. Then, the classification probability vector p is obtained through the *Softmax* function, which is denoted as

$$p = [p_{1,k}, p_{2,k}, ..., p_{i,k}, ..., p_{C,k}]. \tag{3}$$

At the beginning of the t-th training round, the FL system randomly selects $|S^t| = K$ IIoT devices to participate in FL. The global model needs to satisfy the optimization objective:

$$\min_{w\in\mathbb{R}^d} f(w) := \sum_{k=1}^{K} \alpha_k F_k(w), \tag{4}$$

where w is the global model parameters, α_k is the aggregation weight of the local model on IIoT device c_k, and $F_k(w)$ is the objective for local updates on device c_k. $F_k(w)$ is defined as

$$F_k(w) = \frac{1}{|\mathcal{D}_k|} \sum_{k=1}^{K} \mathcal{L}(h_k(w; x), y), \tag{5}$$

TABLE II
LIST OF MAIN SYMBOLIC PARAMETERS

Symbols	Descriptions
d	Dimensions of target models
x	Samples in the dataset $\mathcal{D}$
y	True label corresponding to sample x
r	Local update epoch
η	Learning rate of target models
K	Total number of IIoT devices in subset S^t
N	Total number of IIoT devices in FL training
C	Total number of sample categories
T	Total training rounds in FL training
B	Mini-batch size in local training
P	Public dataset
c_k	The k-th IIoT device
w^t	Aggregated global model in t-th training round
w_k^t	Local model of device c_k in t-th training round
z_i	Logits output of sample x for the i-th class
z_y	Logits output of sample x for the target class
$\bar{w}^t$	Integrated Global model in t-th training round
σ_k^t	Model consistency of device c_k in t-th training round
α_k^t	Aggregate weights of device c_k in t-th training round
$s(\cdot)$	Logits output vector of the model
$\mathcal{M}$	Sliding window size
S^t	A randomly selected subset of N in t-th training round
$\mathcal{D}_k$	Local dataset of device c_k

where h_k is the local model representation. The optimization of the IIoT device c_k is conducted by minimizing (5). Then, the cloud server receives and aggregates the local model parameters w_k^t to update the global model w^t.

However, when the local data is highly heterogeneous, there is an inconsistency between global and local models on IIoT devices. The traditional average aggregation loses important knowledge from the local models and fails to preserve historical models. This leads to a significant degradation in the accuracy of the updated global model on certain data distributions. To address this issue, we design an adaptive weighted model aggregation mechanism based on model consistency.

IV. ADAPTIVE WEIGHTED AGGREGATION AND HISTORICAL MODEL INTEGRATION

A. Model Consistency Evaluation Mechanism

In FL-IIoT, models are trained on heterogeneous data distributions, where some local models may exhibit higher prediction accuracy compared to other devices. To improve the accuracy of IIoT devices, we need to amplify the consensus (i.e., the consistency of model predictions on the same data, which is called model consistency in this section) reached by the local models. On the other hand, even if some IIoT devices do not have high prediction accuracy for certain data samples, their uploaded gradients may still carry useful model update parameters. Therefore, we set different aggregation weights for IIoT devices by measuring the model consistency.

First, the local models of IIoT devices are trained based on the local datasets. To guide the gradients of IIoT devices to update in the optimal direction, inspired by FedProx [20], we add a regularization term to the local loss, which is given by

$$\min F_k(w) + \frac{\delta}{2}\left\|w^{t-1} - w_k^t\right\|_2. \tag{6}$$

Then, we measure the model consistency by evaluating the variance of the model output Logits vector $s(w_k^t, x)$ and $x \in P$, which is denoted as

$$\sigma_k^t(w_k^t, x) = Var(s(w_k^t, x)), \tag{7}$$

where $Var(\cdot)$ is the variance of $s(w_k^t, x)$. A higher $\sigma_k^t(w_k^t, x)$ indicates that IIoT device c_k has higher confidence in predicting sample x. Compared to devices with lower variance Logits, if an IIoT device has a higher $\sigma_k^t(w_k^t, x)$, it should be assigned a higher aggregation weight. Therefore, we set a confidence-based weighted average according to the distribution of Logits for sample $x \in P$, and the global model is denoted as

$$w^t = \frac{1}{|K|}\sum_{k\in S^t} \alpha_k^t(x) w_k^t, \tag{8}$$

where the aggregation weight coefficient $\alpha_k^t(x)$ is given by

$$\alpha_k^t(x) = \sigma_k^t(w_k^t, x) / \sum_{k\in S^t} \sigma_k^t(w_k^t, x). \tag{9}$$

By measuring the consistency of the model Logits, we can effectively filter out anomalous models, which can help the global model aggregation reach a consensus faster.

B. Historical Model-Assisted Global Aggregation Mechanism

In each training round, the global model only aggregates the gradients uploaded by the selected IIoT devices, which ignores the parameters of the historical global model. To overcome the catastrophic forgetting of the global model, we integrate $\mathcal{M}$ historical global models to provide more comprehensive global aggregation information while maintaining the global model's performance on outdated data. To determine the correlation between the number of historical global models $\mathcal{M}$ and catastrophic forgetting, relevant experimental analyses will be discussed in Section V-B. We give **Definition 1**.

Definition 1 (Global Distribution Generalization Bound)**.** *For N IIoT devices in federated learning and one central server, let $\mathcal{D}$ define the global model test data distribution, and $\mathcal{D}_{trg}$ and $\hat{\mathcal{D}}_{trg}$ represent the true data distribution and empirical data distribution, respectively. For device c_k, let $h_k = \arg\min \mathcal{L}_{\mathcal{D}_{trg}}(h)$ and $\hat{h}_k = \arg\min \mathcal{L}_{\hat{\mathcal{D}}_{trg}}(h)$, with aggregation weights $\alpha_k^t, k \in S^t$. The global model aggregation is $\sum_{k=1}^K \alpha_k^t h_{\hat{\mathcal{D}}_{trg}}, \sum_{k=1}^K \alpha_k^t = 1$. With a sampling probability of at least $1-\varepsilon$, the classification boundary is given by*

$$\begin{aligned}\mathcal{L}_D(\sum_{k=1}^K \alpha_k^t h_{\hat{\mathcal{D}}_{trg}}) \le & \sum_{k=1}^K \alpha_k^t \mathcal{L}_{\hat{\mathcal{D}}_{trg}}(h_{\hat{\mathcal{D}}_{trg}}) \\ & + \frac{1}{2}\sum_{k=1}^K \alpha_k^t d(\mathcal{D}_k, \mathcal{D}) \\ & + \sqrt{\log^{\varepsilon-1}} \sum_{k=1}^K \frac{\alpha_k^t}{\sqrt{|[\mathcal{D}_k]_b|}} \\ & + \sum_{k=1}^K \alpha_k^t \nu_k,\end{aligned} \tag{10}$$

where $v_k = \arg\min \mathcal{L}_{\mathcal{D}_k}(h) + \mathcal{L}_{\mathcal{D}}(h)$, $[\mathcal{D}_k]_b$ is the mini-batch samples, and $d(\mathcal{D}_k, \mathcal{D})$ calculates the difference between the data distributions.

According to (9), the global model aggregation in the t-th training round is given by

$$w^t = \frac{1}{|K|} \sum_{k=1}^{K} \frac{\sigma_k^t(w_k^t, x) w_k^t}{\sum_{k \in S^t} \sigma_k^t(w_k^t, x)}. \tag{11}$$

According to **Definition 1**, the classification accuracy of the aggregated global model depends on *1)* the size of the training dataset, *2)* the accuracy of the local model, and *3)* the differences between local and global data distribution. Through model consistency evaluation, we assign lower aggregation weights to IIoT devices with poor local model accuracy or extreme local data distributions that are significantly different from the global data distribution.

In $(t+1)$-th training round, device c_k receives the latest global model and performs mini-batch Stochastic Gradient Descent (SGD) on the local dataset $\mathcal{D}_k$. The local model update for device c_k is given by

$$w_k^{t,r} = w_k^{t,0} - \frac{\eta}{|[\mathcal{D}_k]_b|} \sum_{e=0}^{r-1} \sum_{b=1}^{B} \nabla f(w_k^{t,e}, [\mathcal{D}_k]_b), \tag{12}$$

where $[\mathcal{D}_k]_b$ is the b-th mini-batch of data randomly sampled from $\mathcal{D}_k$, and r is the number of local updates epochs.

To retain important historical update parameters that impact catastrophic forgetting, we construct a sliding window to integrate $\mathcal{M}$ historical global models that are strongly correlated with w^t into $\bar{w}^t$, which is given by

$$\overline{w}^t = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} w^{t-m+1}, \mathcal{M} \in \mathbb{Z}^+. \tag{13}$$

Using the integrated $\bar{w}^t$ as the latest global model $w^t \leftarrow \bar{w}^t$ to guide the local model w_k^{t+1} for the next training round can prevent the local gradient update from deviating from the optimal direction. Although a random sample of all IIoT devices is synthesized each round to form a subset, the information on historical global model parameters is still contained in $\bar{w}^t$. The integrated model helps different devices reach a consensus on data representation while retaining important historical global update parameters. We will prove the effectiveness and range of the positive integer $\mathcal{M}$ in Section IV-C (**Theorem 4**).

C. Convergence and Effectiveness Analysis

To prove the convergence of the FedAWAC, we need to demonstrate that during FL training, after T training rounds, the loss function of the global model will gradually decrease and eventually converge to the optimal solution. For the convergence analysis, we must rely on some common assumptions, we give **Assumptions 1, 2, 3** and **Theorems 1, 2**. To prove that FedAWAC can minimize the variance of global model aggregation updates, we give **Theorem 3**. For the effectiveness analysis of $\mathcal{M}$, we give **Theorem 4**.

Assumption 1 (L-Smoothness)**.** *Assume that the local loss function $F_k(w)$ for each device is L-smooth, which means the gradient changes of the loss function are bounded.*

$$\|\nabla F_k(w_1) - \nabla F_k(w_2)\| \leq L\|w_1 - w_2\|, \quad \forall w_1, w_2. \tag{14}$$

Assumption 2 (μ-Strong Convexity)**.** *Assume that the local loss function $F_k(w)$ is μ-strongly convex.*

$$\begin{aligned} F_k(w_2) \geq F_k(w_1) + \langle \nabla F_k(w_1), w_2 - w_1 \rangle \\ + \frac{\mu}{2}\|w_2 - w_1\|^2, \quad \forall w_1, w_2. \end{aligned} \tag{15}$$

Assumption 3 (Randomness)**.** *In each training round, the selected devices are chosen randomly, so the probability of each device being selected is the same. This randomness ensures that after multiple rounds, all devices' updates contribute to the global model.*

Theorem 1 (Convergence Guarantee of Local Updates)**.** *If (12) is feasible and **Assumptions 1, 2, 3** hold, then after e local update epochs, the loss function of each device will gradually converge to the local optimal value $F_k(w_k^*)$.*

Proof. As shown in (12), in FedAWAC, each device performs multiple steps of local SGD. According to the convergence theory of gradient descent, with **Assumptions 1, 2**, the local updates gradually converge to the local optimal solution. Specifically, for device c_k, the loss function value $F_k(w_k^t)$ of the local model w_k^t satisfies:

$$F_k(w_k^{t+1}) \leq F_k(w_k^t) - \left(\eta - \frac{L\eta^2}{2}\right) \|\nabla F_k(w_k^t)\|^2, \tag{16}$$

where η is the learning rate. Based on **Assumption 2**, we have

$$\|\nabla F_k(w_k^t)\|^2 \geq 2\mu \left(F_k(w_k^t) - F_k(w_k^*)\right), \tag{17}$$

where w_k^* is the optimal model of device c_k. Therefore, the local update satisfies the following recursive relationship:

$$\begin{aligned} F_k(w_k^{t+1}) \leq F_k(w_k^t) \\ - 2\mu \left(\eta - \frac{L\eta^2}{2}\right) \left(F_k(w_k^t) - F_k(w_k^*)\right). \end{aligned} \tag{18}$$

This means that after e local update epochs, the loss function of each device gradually converges to the local optimal value $F_k(w_k^*)$. Therefore, **Theorem 1** concludes. □

Theorem 2 (Convergence Guarantee of the Global Model)**.** *If (11) is feasible and **Assumptions 1, 2,** and **3** hold, then after T training rounds, the loss function of the global model will converge to the optimal value $F(w^*)$.*

Proof. The update of the global model can be denoted as

$$w^t = w^{t-1} - \eta \nabla F(w^{t-1}), \tag{19}$$

where $\nabla F(w^{t-1})$ is the gradient of the global loss function. Based on (11) and **Theorem 1**, we can prove that the value of the global model's loss function $F(w^t)$ decreases monotonically in each round, which is given by

$$F(w^t) \leq F(w^{t-1}) - \eta \|\nabla F(w^{t-1})\|^2. \tag{20}$$

Algorithm 1: FedAWAC

Input: N, T, B, $\mathcal{M}$, r, η
Output: Global model w^t

Initial global model parameters w^0
Procedure Sever Execution
for *each training round* $t \in T$ **do**
  Random sample a set of devices $S^t \in N$
  if $t < \mathcal{M}$ **then**
    Send the global model w^{t-1} to the selected devices
  else
    Send integrated history global model $\bar{w}^{t-1}$ to the selected devices
  end
  for *each device* c_k *in parallel* **do**
    $w_k^t \leftarrow ClientUpdate(w^{t-1}, \mathcal{D}_k)$
    $\sigma_k^t(w_k^t, x) = Var(s(w_k^t, x))$
    $\alpha_k^t(x) = \sigma_k^t(w_k^t, x) / \sum_k \sigma_k^t(w_k^t, x)$
  end
  $w^t = \frac{1}{K} \sum_{k=1}^{K} \alpha_k^t(x) w_k^t$
  $\bar{w}^t = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} w^{t-m+1}$
end
function $ClientUpdate(w^{t-1}, \mathcal{D}_k)$
begin
  for *each local epoch* $e = 1, ..., r$ **do**
    for *each batch* $b \in B$ **do**
      $w_k^t \leftarrow w_k^{t,0} - \frac{\eta}{|[\mathcal{D}_k]_b|} \sum_{e=0}^{r-1} \sum_{b=1}^{B} \nabla f(w_k^{t,e}, [\mathcal{D}_k]_b)$
      $f = \min F_k(w) + \frac{\delta}{2} \left\| w^{t-1} - w_k^t \right\|_2$
    end
  end
  return w_k^t back to sever
end

Furthermore, based on **Assumption 2**, we can get

$$\|\nabla F(w^{t-1})\|^2 \geq 2\mu(F(w^{t-1}) - F(w^*)). \tag{21}$$

Thus, the change in $F(w^t)$ satisfies:

$$F(w^t) - F(w^*) \leq (1 - 2\mu\eta)(F(w^{t-1}) - F(w^*)). \tag{22}$$

This means that after T rounds of global training, the global model's loss function will converge to the optimal value $F(w^*)$ at an exponential rate, which is given by

$$F(w^T) - F(w^*) \leq (1 - 2\mu\eta)^T (F(w^0) - F(w^*)). \tag{23}$$

Therefore, **Theorem 2** concludes. □

Theorem 3 (Convergence Bound Improvement through Adaptive Weighting Based on Consistency)**.** *If (9) is feasible and* ***Assumptions 1, 2*** *hold, then FedAWAC can minimize the variance of aggregated updates, yielding a tighter convergence bound. The convergence of the global model w^T after T training rounds satisfies the bound:*

$$\begin{aligned} &\mathbb{E}[f(w^T) - f(w^*)] \\ &\leq \frac{1}{T} \sum_{t=1}^{T} \left(\frac{L}{2} \sum_{k=1}^{K} \alpha_k^t \left|\left| w^t - w^* \right|\right|^2 + \frac{\eta^2 \sigma_k^t}{2\alpha_k^t} \right), \end{aligned} \tag{24}$$

where $f(w^*)$ is the optimal value of the global objective function, L is the Lipschitz constant for smoothness, and η is the learning rate.

Proof. Based on **Assumption 1**, in the t-th training round, $F_k(w^t)$ satisfies:

$$\begin{aligned} F_k(w^t) \leq\ & F_k(w^*) \\ & + \nabla F_k(w^*)^T (w^t - w^*) \\ & + \frac{L}{2} \parallel w^t - w^* \parallel^2 . \end{aligned} \tag{25}$$

According to (4), in the t-th training round, the global objective function $f(w^t)$ with weight α_k^t is given by

$$f(w^t) = \sum_{k=1}^{K} \alpha_k^t F_k(w^t). \tag{26}$$

Substituting (25) into (26), we can get

$$\begin{aligned} f(w^t) \leq\ & \sum_{k=1}^{K} \alpha_k^t (F_k(w^*) \\ & + \nabla F_k(w^*)^T (w^t - w^*) \\ & + \frac{L}{2} \|w^t - w^*\|^2). \end{aligned} \tag{27}$$

Based on **Assumption 2**, in the t-th training round, we have

$$f(w^t) - f(w^*) \leq \frac{1}{2\mu} \|\nabla f(w^t)\|^2. \tag{28}$$

According to (9), to minimize the variance of aggregated updates, we ensure that devices with lower variance $\sigma_k^t(w_k^t, x)$ contribute more to the global model. Then, we can infer

$$\begin{aligned} \|w^{t+1} - w^*\|^2 \leq\ & \|w^t - w^*\|^2 \\ & - 2\eta \sum_{k=1}^{K} \alpha_k^t \nabla F_k(w^*)^T (w^t - w^*) \\ & + \eta^2 \sum_{k=1}^{K} \frac{\sigma_k^t}{\alpha_k^t}. \end{aligned} \tag{29}$$

Summing (29) over T rounds, we can get

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}[f(w^t) - f(w^*)] \\ &\leq \frac{1}{T} \sum_{t=1}^{T} \left(\frac{L}{2} \sum_{k=1}^{K} \alpha_k^t \parallel w^t - w^* \parallel^2 + \frac{\eta^2 \sigma_k^t}{2\alpha_k^t} \right). \end{aligned} \tag{30}$$

Therefore, **Theorem 3** concludes. According to **Theorems 1, 2, 3**, we have proven the convergence of FedAWAC. □

Theorem 4 (Effectiveness of Sliding Window $\mathcal{M}$)**.** *If (13) is feasible and* ***Assumption 1*** *holds, then moderate $2 \leq \mathcal{M} \leq \sqrt{T}$ can make the integrated model $\bar{w}^t$ suppress the fluctuations in global model updates.*

Proof. In the t-th training round, based on **Assumption 1** and SGD, we have

$$F(w^t) - F(w^*) \leq \frac{1}{2\eta t} \|w^0 - w^*\|^2. \tag{31}$$

TABLE III
DATASETS DETAILS AND HYPERPARAMETER SETTINGS

Datasets	CIFAR-10	CIFAR-100	Tiny-ImageNet	AG News
Type	Image	Image	Image	Text
Model	CNN	CNN	ResNet-18	FastText
Device	10/100	10/100	10/100	10/100
Category	10	100	200	4
Train Size	50,000	50,000	100,000	84000
Test Size	10,000	10,000	10,000	36000
Batch Size	50	50	50	10
Training Round	200	200	200	200
Learning Rate	0.1	0.1	0.1	0.1

According to (13) , we can get

$$F(\bar{w}^t) - F(w^*) \leq \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} (F(w^{t-m+1}) - F(w^*)). \quad (32)$$

Since $F(w^{t-m+1})$ is decreasing (i.e., the error of the global model gradually decreases as training goes on), we have

$$F(\bar{w}^t) \leq \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} F(w^{t-m+1}) \leq F(w^{t-\mathcal{M}+1}). \quad (33)$$

Thus, the error of the integrated model $\bar{w}^t$ will not be larger than the worst error within the most recent $\mathcal{M}$ rounds. As training goes on, if too many historical global models are integrated, it can slow the convergence of the global model while imposing substantial computational costs. This indicates that when $2 \leq \mathcal{M} \leq \sqrt{T}$ is moderate, the integrated model $\bar{w}^t$ can effectively suppress the fluctuations in global model updates. **Theorem 4** concludes. □

D. Complexity Analysis of FedAWAC

The computational complexity of FedAWAC mainly comes from the local training on the device and the global aggregation on the server. *1) Gradient computation.* Each client computes the gradient on its local dataset $\mathcal{D}_k$. Assuming the dimension of the model parameters is d, the complexity of calculating the gradient for each batch is $O(dB)$. *2) Local update epoch.* In each communication round, clients perform r local epochs of updates, and each epoch requires gradient computation for all batches. Assuming each batch contains B samples, the computational complexity of the entire local training is $O(r \cdot B \cdot d)$. Therefore, the computational complexity for each client is $O(r \cdot B \cdot d)$. *3) Server-side.* The cloud server's computation mainly involves the weighted aggregation of the global model. Specifically, after receiving the updated models from the clients, the server computes the weighted average. Assuming K clients are selected, and each client model has d parameters, the complexity of the server-side aggregation operation is $O(K \cdot d)$. Additionally, the FedAWAC introduces a sliding window ensemble operation, where the server averages the models from the past $\mathcal{M}$ rounds. The complexity of this operation is $O(\mathcal{M} \cdot d)$. Therefore, the total computational complexity on the server side is $O(K \cdot d + \mathcal{M} \cdot d) = O((K + \mathcal{M}) \cdot d)$.

E. Algorithm Design

FedAWAC can be divided into the following two key stages.

1) Model consistency evaluation. During the local training, as shown in step ② in Fig. 2, IIoT devices perform multiple rounds of SGD on their local heterogeneous datasets to obtain local updates. Subsequently, the devices output the Logits vector for unlabeled samples $x \in P$ and compute the variance of the vector $\sigma_k^t(w_k^t, x)$ as the model consistency (step ③). A higher value of $\sigma_k^t(w_k^t, x)$ indicates that the device c_k has a higher confidence in its prediction for sample x. Based on $\sigma_k^t(w_k^t, x)$, the influence of the local models' consensus on the data prediction is assessed, and the aggregation weight $\alpha_k^t(x)$ is assigned to each device.

2) Historical model-assisted global aggregation. During the global model aggregation, the cloud server aggregates the local models participating in the current training round based on the aggregation weights $\alpha_k^t(x)$ (step ④), effectively reducing the negative impact of more anomalous local models. By utilizing a sliding window, $\mathcal{M}$ historical global models are integrated to obtain $\bar{w}^t$ (step ⑤). This $\bar{w}^t$ can replace the updated global model and is sent to the devices participating in the next training round for local training (step ①).

The design details of FedAWAC are presented in ***Algorithm 1***. Judge the training rounds (lines 4-8). Calculate the variance of the Logits vector output from the local model and measure the model consistency (lines 9-11). Assign aggregate weights to IIoT devices (line 12). Global model aggregation (line 13). Historical global model integration (line 14). Update the local model with the local dataset (lines 15-20).

V. PERFORMANCE EVALUATION

A. Experimental Settings

Experimental Environment. All experiments in this section are conducted on a server equipped with 2 NVIDIA $A100$ GPUs with 80GB memory and 256GB RAM, running the Ubuntu 20.04 operating system, powered by a 64-core Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz, and utilizing a CUDA 11.8 computing platform with the PyTorch 1.8 framework.

Non-IID Datasets and Target Models. We choose three image datasets (CIFAR-$10/100$ and Tiny-ImageNet) and one text dataset (AG News) and train three models (CNN and RestNet-18 for image classification tasks, FastText for text classification tasks). To simulate the real heterogeneous environment in IIoT, we utilize the *Dirichlet* function $Dir(\alpha)$ to partition the datasets (i.e., reallocating the proportion of class C samples assigned to each device) to generate Non-IID datasets [36, 37]. The smaller the parameter α, the more different the distribution of training datasets allocated to devices. The details of the datasets and models are as follows.

- *CIFAR-10* and *CIFAR-100* datasets comprise images of 10 and 100 categories [38], respectively, with a fixed size of 32×32 pixels in color. We train a Convolutional Neural Network (CNN) model to classify images, which consists of 2 convolutional layers (5×5, each activated by ReLU and followed by 2×2 max pooling), 2 fully connected layers, and Softmax normalizes the final output.

TABLE IV
GLOBAL MODEL ACCURACY Acc_g OF FEDAWAC WITH DIFFERENT SLIDING WINDOWS $\mathcal{M}$ (%)

Sliding	CIFAR-10			CIFAR-100		
windows	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$
$\mathcal{M}=3$	41.89	68.75	**75.89**	30.27	31.90	39.45
$\mathcal{M}=$ **5**	**43.65**	**69.82**	74.49	**31.26**	**32.73**	**40.10**
$\mathcal{M}=7$	42.97	68.02	74.51	31.08	32.12	39.82
$\mathcal{M}=9$	41.64	66.79	73.10	29.65	30.89	38.97

TABLE V
FORGETTING RATE $\mathcal{F}$ OF FEDAWAC WITH DIFFERENT SLIDING WINDOWS $\mathcal{M}$ (%)

Sliding	CIFAR-10			CIFAR-100		
windows	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$
$\mathcal{M}=3$	54.10	20.62	19.10	42.68	28.46	**24.17**
$\mathcal{M}=$ **5**	**47.30**	**19.15**	**17.20**	**38.24**	**27.29**	24.29
$\mathcal{M}=7$	50.71	21.36	18.49	40.29	30.71	28.13
$\mathcal{M}=9$	51.79	23.78	18.20	42.10	34.16	30.56

- *Tiny-ImageNet* dataset consists of 200 categories with approximately $120,000$ samples [39], where each class contains 500 training images, 50 validation images, and 50 test images, with each image sized at 64×64. Compared to the CIFAR-$10/100$ datasets, the Tiny-ImageNet dataset has greater complexity in terms of image categories and RGB channels. Therefore, we train a ResNet-18 model to classify images, which incorporates residual structures and consists of 18 layers (including convolutional layers, normalization layers, and fully connected layers).
- *AG News* dataset consists of article titles and descriptions, comprising 4 categories with $127,600$ samples [40]. It is one of the commonly used datasets for text classification tasks. Given that the FastText structure is simple and parallelization-friendly, making it suitable for resource-constrained IIoT devices, we train a FastText model to classify text (including an input layer, a fully connected hidden layer, and a fully connected output layer).

Baselines. Five comparative methods are as follows.

- *FedAvg* [41], the most classic and popular baseline, uses the empirical loss of randomly selected devices' training data as the optimization objective and updates the global model by averaging the selected local model parameters.
- *FedCurv* adds penalty terms to the local model based on the EWC algorithm to mitigate catastrophic forgetting under heterogeneous data [13]. The weight of the penalty term is determined by the *Fisher* information matrix.
- *FedProx* adds a regularization term to the local loss function to control gradient drift by limiting the *Euclidean* distance between the local and global models [20].
- *FedCM* aggregates the global gradient from the previous training round and uses a momentum term to adjust the local gradient toward the global gradient [25].
- *FedGA* [42], the state-of-the-art method to mitigate local forgetting, reduces forgetting of minority and unseen classes during local updates through label calibration.

Hyperparameter Settings. The total number of IIoT devices $N = 100$ and $K = 10$ devices are randomly selected in each training round. To ensure the fairness of our experiments, we adopt the same settings as FedCM [25]. The momentum coefficient is 0.9, the learning rate scheduler has a decay factor of 0.99, and the number of local training epochs $r = 5$. Other parameters follow common settings used in image and text classification tasks, with details provided in Table III.

Metrics. Three evaluation criteria are depicted as follows.

- *Global model accuracy* Acc_g. We use the global model accuracy Acc_g as a metric for evaluating the global model's performance. The higher the Acc_g, the better the model training performance of the method.
- *Forgetting rate* $\mathcal{F}$. Similar to [43], we use $\mathcal{F}$ as the catastrophic forgetting evaluation metric for heterogeneous data and $\mathcal{F}$ represents the average gap between the maximum accuracy and the final accuracy for each class at the end of FL, which is given by

$$\mathcal{F} = \frac{1}{C}\sum_{c=1}^{C} \max_{t\in\{1,2,\cdots,T-1\}} (Acc_c^t - Acc_c^T), \tag{34}$$

  where Acc_c^t is the accuracy of class c at t-th training round. A smaller $\mathcal{F}$ indicates that the model forgets less about tasks on different data and the method is better at overcoming catastrophic forgetting.
- *Memory usage* m_u. The memory usage of the model during training is employed as an evaluation metric. The smaller the memory usage m_u, the fewer memory resources are consumed during the model training.

B. Selection of Sliding Window $\mathcal{M}$

In FedAWAC, a larger $\mathcal{M}$ may include unstable parameters, potentially reducing the accuracy of the global model while a smaller $\mathcal{M}$ cannot overcome catastrophic forgetting. To select a reasonable $\mathcal{M}$ for contrast experiments, this section discusses the hyperparameter performance of FedAWAC on the CIFAR-$10/100$ datasets under different levels of heterogeneity $\alpha = \{0.05, 0.5, 1\}$ from the perspectives of Acc_g and $\mathcal{F}$.

To increase training instability and better amplify the impact of $\mathcal{M}$ on FedAWAC training performance, we set $T = 100$ and $K = 5$ for the experiments (with other hyperparameters unchanged) in this section. By comparing different settings of $\mathcal{M}$ (e.g., $3, 5, 7, 9$), the model's performance on Acc_g during oscillations can be directly observed, enabling the selection of an appropriate $\mathcal{M}$. Additionally, the training instability will magnify the catastrophic forgetting phenomenon under different $\mathcal{M}$ settings, allowing for a more intuitive observation of how the metric $\mathcal{F}$ changes with $\mathcal{M}$. This can help us quickly determine the optimal value of $\mathcal{M}$.

1) Global model accuracy Acc_g with different $\mathcal{M}$. The Acc_g of FedAWAC with different $\mathcal{M}$ on the CIFAR-10 dataset are reported in columns 2 to 4 of Table IV. When the level of data heterogeneity $\alpha = 1$ and $\mathcal{M} = 3$, the global model achieves the highest $Acc_g = 75.89\%$. As $\mathcal{M}$ increases, the Acc_g of the global model gradually declines. Since the CIFAR-10 dataset is relatively simple and the level of data heterogeneity $\alpha = 1$ is low, the impact of different $\mathcal{M}$ on the forgetting rate is minimal. As the level of data heterogeneity increases, when

TABLE VI
GLOBAL MODEL ACCURACY Acc_g OF DIFFERENT TRAINING METHODS (%)

Methods	CIFAR-10			CIFAR-100			Tiny-ImageNet			AG News			Average
	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	
FedAvg	35.13	66.97	73.25	31.31	37.32	38.62	13.62	16.93	17.70	79.15	83.13	86.97	48.34
FedCurv	34.51	67.34	72.82	30.85	35.99	39.16	14.26	17.97	19.62	78.67	82.95	87.01	48.43
FedProx	38.16	63.98	62.65	32.13	32.10	35.70	16.73	18.17	21.57	79.08	83.25	87.10	47.55
FedCM	42.19	**71.64**	74.65	30.43	38.75	**42.35**	15.82	24.75	26.25	81.12	84.68	87.55	51.68
FedGA	42.30	67.22	**74.77**	33.74	37.64	39.07	14.65	20.82	23.50	81.84	**84.77**	**87.97**	50.69
FedAWAC (Ours)	**45.24**	69.82	74.49	**35.50**	**39.33**	40.10	**18.38**	**26.42**	**26.93**	**81.90**	84.75	87.74	**52.55**

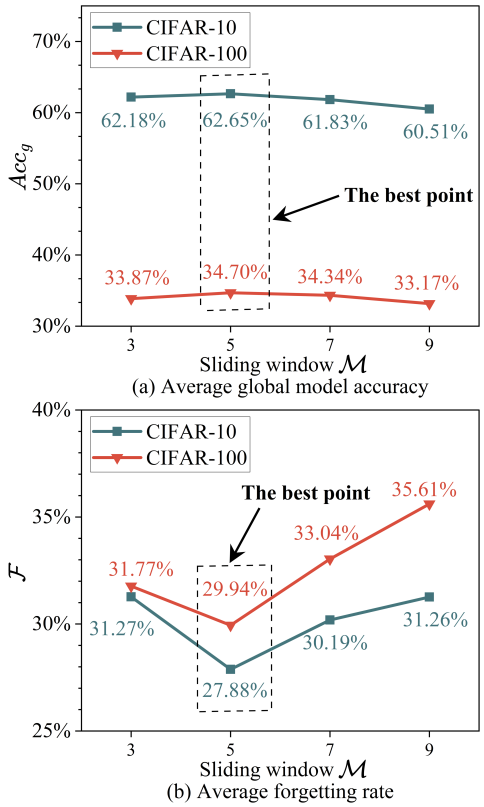

Fig. 3. The average Acc_g and $\mathcal{F}$ of FedAWAC with variable $\mathcal{M}$ (%).

$\alpha = \{0.05, 0.5\}$, FedAWAC achieves the highest $Acc_g = 43.65\%$ and 69.82% with $\mathcal{M} = 5$, respectively.

Furthermore, columns 5 and 7 of Table IV report the Acc_g of FedAWAC with different $\mathcal{M}$ on the CIFAR-100 dataset. The experimental results report that when $\alpha = \{0.05, 0.5, 1\}$ and $\mathcal{M} = 5$, the Acc_g of FedAWAC is consistently the highest. When the level of data heterogeneity is low ($\alpha = 1$), FedAWAC achieves the highest accuracy across different α, demonstrating robustness in selecting $\mathcal{M}$, with the Acc_g only 0.65% lower than that of $\mathcal{M} = 3$. Notably, when $\mathcal{M} = 9$, FedAWAC shows the lowest global model accuracy across all α on both datasets, where the Acc_g on the CIFAR-10 dataset ($\alpha = 0.5$) is 3.03% lower than that of $\mathcal{M} = 5$. This is because if $\mathcal{M}$ is large, there are more historical global models being integrated, and the differences in the global models across different rounds gradually become large. This can slow the global model convergence and degrade the model accuracy.

2) Forgetting rate $\mathcal{F}$ with different $\mathcal{M}$. To balance global model accuracy Acc_g and forgetting rate $\mathcal{F}$, Table V reports the forgetting rates $\mathcal{F}$ of FedAWAC with different $\mathcal{M}$. Columns 2 to 4 of Table V show the Acc_g of CIFAR-10 with different levels of data heterogeneity α. When $\alpha = \{0.05, 0.5, 1\}$ and $\mathcal{M} = 5$, FedAWAC achieves the lowest $\mathcal{F} = 47.30\%$, 19.15%, and 17.20%, respectively. Therefore, $\mathcal{M} = 5$ enables FedAWAC to forget more appropriate historical global models to trade for higher model accuracy.

Columns 5 to 7 of Table V present the forgetting rates on the CIFAR-100 dataset with different $\mathcal{M}$. When the level of heterogeneity is low ($\alpha = 1$) and $\mathcal{M} = 3$, FedAWAC achieves the lowest forgetting rate of 24.17%, although the global model accuracy at this point (39.45%) is lower than the accuracy at $\mathcal{M} = 3$. As the data heterogeneity increases, when $\alpha = \{0.05, 0.5\}$ and $\mathcal{M} = 5$, FedAWAC achieves the lowest forgetting rates of 38.24% and 27.29%, respectively, reaching the optimal sliding window setting.

Therefore, considering the objective factors of data heterogeneity, by balancing the global model accuracy and the performance of mitigating catastrophic forgetting as reported in this section, we set $\mathcal{M} = 5$ for subsequent experiments.

C. Analysis of Global Model Accuracy Acc_g

1) CIFAR-10 dataset. As reported in columns 2 to 4 of Table VI, when $\alpha = 0.5$, by utilizing the set of historical model gradients, FedCM achieves the highest $Acc_g = 71.64\%$, which is 1.82% higher than FedAWAC, respectively. However, FedCM needs to record the gradient changes of the local model in each round, aggregate the average gradient changes of all devices, and transmit additional momentum terms to the server, leading to significantly higher computational cost and communication overhead compared to FedAWAC. When $\alpha = 1$, FedGA achieves the highest $Acc_g = 74.77\%$. However, when $\alpha = 0.05$, FedAWAC achieves the highest $Acc_g = 45.24\%$, which is about 10.11% higher than FedAvg.

2) CIFAR-100 dataset. As reported in columns 5 to 7 of Table VI, the increased complexity of the dataset degrades model accuracy for all methods. Specifically, when $\alpha = 1$, FedCM achieves the highest Acc_g, which is 2.25% higher than FedAWAC. This indicates that when data heterogeneity is low, the differences in data distribution between devices are small, and FedAWAC utilizes well-integrated consistency model representations but demonstrates slightly weaker performance compared to FedCM in adjusting local gradient updates. However, the average Acc_g of FedAWAC is significantly higher than that of FedAvg, FedCurv, FedProx, and FedGA.

TABLE VII
FORGETTING RATE $\mathcal{F}$ OF DIFFERENT TRAINING METHODS (%)

Methods	CIFAR-10			CIFAR-100			Tiny-ImageNet			AG News			Average
	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	$Dir(0.05)$	$Dir(0.5)$	$Dir(1)$	
FedAvg	67.10	27.02	20.17	59.15	32.90	28.11	76.17	59.72	58.13	58.75	26.11	15.51	44.07
FedCurv	62.36	28.09	18.34	57.30	29.75	22.90	75.89	55.31	53.95	56.20	26.10	17.16	41.95
FedProx	64.97	26.60	19.20	54.18	31.67	21.42	74.22	58.75	51.50	53.61	23.47	15.32	41.24
FedCM	61.92	21.40	16.55	52.17	**25.14**	18.93	73.60	51.24	49.61	51.44	22.66	15.25	38.33
FedGA	62.05	20.32	**16.51**	51.60	30.55	20.15	73.10	53.13	51.53	50.69	22.29	**15.21**	38.93
FedAWAC (Ours)	**60.03**	**19.25**	17.38	**50.35**	27.32	**18.57**	**71.41**	**48.79**	**48.04**	**50.51**	**21.80**	15.40	**37.40**

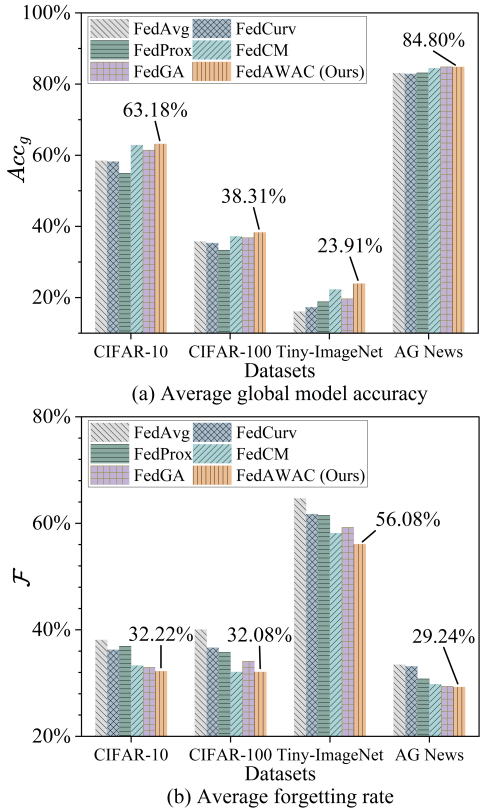

Fig. 4. The average Acc_g and $\mathcal{F}$ of six methods (%).

3) Tiny-ImageNet dataset. As reported in columns 8 to 10 of Table VI, FedAWAC achieves the highest $Acc_g = 18.38\%$, 26.42%, and 26.93% when $\alpha = \{0.05, 0.5, 1\}$, respectively, significantly outperforming FedAvg, FedCurv, FedProx, FedCM, and FedGA. FedCM only achieves the scend-highest $Acc_g = 15.82\%, 24.75\%$, and 26.25%, respectively.

4) AG News dataset. As reported in columns 11 to 13 of Table VI, FedGA achieved the highest accuracy of $Acc_g = 84.77\%$ and 87.97% when $\alpha = \{0.5, 1\}$, respectively, which is only 0.02% and 0.23% higher than FedAWAC. This is because the AG News dataset contains fewer categories, allowing FedGA to better align sample labels. However, when the data is highly heterogeneous ($\alpha = 0.05$), FedAWAC achieved the highest accuracy of $Acc_g = 81.90\%$, surpassing FedGA's 81.84%. This indicates that FedAWAC can adapt to datasets with a higher level of heterogeneity. Therefore, FedAWAC can maintain good model accuracy on the text dataset (AG News), while also being applicable to other models.

D. Analysis of Forgetting Rate $\mathcal{F}$

1) CIFAR-10 dataset. As reported in columns 2 to 4 in Table VII, when the level of data heterogeneity is low ($\alpha = 1$), FedCM has the lowest $\mathcal{F}$, which is 0.83% lower than FedAWAC. When the data heterogeneity increases to $\alpha = \{0.05, 0.5\}$, FedAWAC achieves the lowest $\mathcal{F} = 60.03\%$ and 19.25%, respectively, significantly lower than other baseline methods. When $\alpha = 0.05$, FedAWAC outperforms FedAvg, FedCurv, FedProx, FedCM, and FedGA by approximately 7.07%, 2.33%, 4.94%, 1.89%, and 2.02%, respectively.

2) CIFAR-100 dataset. As reported in columns 5 to 7 in Table VII, when $\alpha = 0.5$, FedCM achieves the lowest $\mathcal{F}$ among all methods. This is because FedCM retains the information on model gradient changes during each round of local training, which can help mitigate the negative impact of data heterogeneity on the global model by improving the correction of local model update directions in cases of low data heterogeneity. When the data heterogeneity $\alpha = 0.05$, the $\mathcal{F}$ of FedAWAC is significantly lower than FedCM and FedGA by approximately 1.82% and 1.25%, respectively.

3) Tiny-ImageNet dataset. As reported in columns 8 to 10 in Table VII, with the increase in the complexity of sample categories in the Tiny-ImageNet dataset, the forgetting rate $\mathcal{F}$ of FedAWAC with $\alpha = 0.5$ is also about 2.45% lower than that of FedCM, which is a greater reduction than the 1.82% observed on the CIFAR-100 dataset. This indicates that FedAWAC can better overcome forgetting caused by the global model on complex datasets. Furthermore, FedAWAC achieve the lowest $\mathcal{F} = 71.41\%, 48.79\%$, and 48.04% when $\alpha = \{0.05, 0.5, 1\}$, which is significantly lower than FedGA by approximately $1.69\%, 4.34\%$, and 3.49%, respectively.

4) AG News dataset. As reported in columns 11 to 13 in Table VII, FedGA achieved the lowest $\mathcal{F} = 15.21\%$ when $\alpha = 1$, which is 0.19% lower than FedAWAC. When $\alpha = \{0.05, 0.5\}$, FedAWAC obtained the lowest $\mathcal{F} = 55.51\%$ and 21.80%, respectively. On the AG News dataset, where the number of sample categories is relatively small, the gradient alignment in FedGA helps it retain more features under low data heterogeneity. However, as data heterogeneity increases, gradient alignment becomes more challenging, leading to a higher forgetting rate in FedGA compared to FedAWAC. FedAWAC effectively reduces the forgetting rate under high data heterogeneity by integrating $\mathcal{M}$ historical global models.

E. Analysis of Memory Usage m_u

Table VIII reports the memory usage m_u of six methods on 3 datasets (CIFAR-10/100 and Tiny-ImageNet). The m_u

TABLE VIII
MEMORY USAGE m_u OF DIFFERENT TRAINING METHODS (GB).

Methods	CIFAR-10 $Dir(1)$	CIFAR-100 $Dir(1)$	Tiny-ImageNet $Dir(1)$	Average
FedAvg	0.75	0.75	8.49	3.33
FedCurv	0.95	2.30	14.69	5.98
FedProx	1.10	1.10	12.70	4.97
FedCM	1.24	1.24	13.55	5.34
FedGA	0.89	0.89	9.24	3.67
FedAWAC (Ours)	**0.85**	**0.85**	**8.54**	**3.41**

of the trained model is primarily determined by the model's structure and the number of parameters. Since FastText is the most lightweight model among the three models and has the fewest parameters, the m_u required to train FastText with the six methods differs only slightly. Therefore, in this section, we choose the more complex CNN and ResNet-18 for comparison. The model structures for all methods on the same dataset are identical, and during the training process, the parameter matrices or tensors of the models remain fixed, independent of the feature distribution or label distribution of the input data (i.e., independent of α). Therefore, in this section, we set the same data heterogeneity $\alpha = 1$ for six methods on three datasets.

1) CIFAR-$10/100$ *datasets.* As reported in columns 2 to 3 in Table VIII, FedAvg requires the least amount of memory, while the proposed FedAWAC is the scend-least. This is because FedAvg exchanges model parameters with the server without involving additional weight calculations. FedAWAC integrates and stores five historical global models, which occupy 0.1GB more memory than FedAvg. Since the computational complexity of FedAvg, FedProx, FedCM, FedGA, and FedAWAC is related to the model parameters, and the models trained on the CIFAR-$10/100$ datasets have the same CNN structure, the memory usage for the same method on both datasets is the same. Since each device in FedCurv needs to send the diagonal elements of the *Fisher* matrix to other devices, FedCurv occupies more memory on the CIFAR-100 dataset. FedGA dynamically aggregates the global model based on gradient similarity, which needs to store all devices' gradients in each round. Therefore, the memory usage of FedGA is slightly higher than that of FedAWAC.

2) Tiny-ImageNet dataset. As the number of sample categories in the Tiny-ImageNet dataset increases, the global model is a more complex ResNet-18, leading to a significant increase in memory usage across all five methods. Due to its simplicity, FedAvg has the lowest memory usage, while the proposed FedAWAC only requires 50MB more than FedAvg. However, according to Sections V-C and V-D, FedAWAC significantly outperforms FedAvg in terms of model accuracy Acc_g and forgetting rate $\mathcal{F}$. Due to the increased gradients of the ResNet-18, the amount of gradient information that needs to be stored by FedGA significantly increases, leading to a substantial rise in memory usage. Although FedCurv and FedGA can mitigate forgetting to some extent, their memory usage is approximately 1.72 and 1.08 times that of FedAWAC, respectively. Experiments demonstrate that FedAWAC can occupy significantly less memory than other baselines and better accommodate environments with limited storage capacity.

VI. CONCLUSION

Due to the limited storage capacity of IIoT devices, fresh data continuously received by diverse devices will overwrite the outdated data and change the local data distribution in FL-IIoT. As training goes on, FL tends to train with fresh data and the latest global model may forget the historical update direction. Catastrophic forgetting can significantly degrade the accuracy of the global model. To overcome catastrophic forgetting during global aggregation, we propose a Federated Adaptive Weight Aggregation method based on model Consistency (FedAWAC). Specifically, we extract reliable consensus from the Logits output of unlabeled data on the local model to measure model consistency and dynamically adjust global aggregation weights for each device. Meanwhile, a sliding window integrates $\mathcal{M}$ historical global models on the cloud server side to overcome catastrophic forgetting and ensure higher global model accuracy on Non-IID data. Experiments on 4 different datasets show that, compared to 5 baselines, FedAWAC can improve global model accuracy by an average of 1.86%, reduce the forgetting rate by an average of 3.93%, and save average memory usage by up to 2.57GB. Furthermore, when the dimensions of the classifiers for IIoT devices are the same, FedAWAC is compatible with various FL training methods and heterogeneous models. These improvements indicate that FedAWAC is a promising approach to overcome catastrophic forgetting in FL-IIoT. In the future, we will consider how to overcome catastrophic forgetting through collaboration between local updates and global aggregation.

REFERENCES

[1] S. Bhatnagar *et al.*, "Efficient Logistics Solutions for E-Commerce Using Wireless Sensor Networks," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024.

[2] J. Bian *et al.*, "Machine Learning in Real-Time Internet of Things (IoT) Systems: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8364-8386, 2022.

[3] S. El khediri *et al.*, "Integration of Artificial Intelligence (AI) with Sensor Networks: Trends, Challenges, and Future Directions," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 1, p. 101892, 2024.

[4] A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim and M. Alrashoud, "An Efficient Spam Detection Technique for IoT Devices Using Machine Learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 903-912, 2021.

[5] M. Revanesh, S. S. Gundal, J. R. Arunkumar, P. J. Josephson, S. Suhasini, and T. K. Devi, "Artificial Neural Networks-based Improved Levenberg–Marquardt Neural Network for Energy Efficiency and Anomaly Detection in WSN," *Wireless Netw*, vol. 30, no. 6, pp. 5613–5628, 2024.

[6] N. Kumar, P. Rani, V. Kumar, P. K. Verma, and D. Koundal, "TEEECH: Three-Tier Extended Energy Efficient Clustering Hierarchy Protocol for Heterogeneous

Wireless Sensor Network," *Expert Systems with Applications*, vol. 216, p. 119448, 2023.

[7] A. Abu-Khadrah, A. M. Ali, and M. Jarrah, "An Amendable Multi-Function Control Method using Federated Learning for Smart Sensors in Agricultural Production Improvements," *ACM Trans. Sen. Netw.*, 2023,.

[8] M. Le, D. T. Hoang, D. N. Nguyen, W.-J. Hwang, and Q.-V. Pham, "Wirelessly Powered Federated Learning Networks: Joint Power Transfer, Data Sensing, Model Training, and Resource Allocation," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[9] K. Luo, X. Li, Y. Lan, and M. Gao, "GradMA: A Gradient-Memory-Based Accelerated Federated Learning With Alleviated Catastrophic Forgetting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3708–3717, 2023.

[10] C. Xu, Z. Hong, M. Huang, and T. Jiang, "Acceleration of Federated Learning with Alleviated Forgetting in Local Training," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[11] G. Wei and X. Li, "Knowledge Lock: Overcoming Catastrophic Forgetting in Federated Learning," in *Advances in Knowledge Discovery and Data Mining*, Cham: Springer International Publishing, pp. 601–612, 2022.

[12] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual Lifelong Learning with Neural Networks: A Review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[13] N. Shoham *et al.*, "Overcoming Forgetting in Federated Learning on Non-IID Data," in *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2019.

[14] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "FedCL: Federated Contrastive Learning for Privacy-Preserving Recommendation," *arXiv preprint* arXiv:2204.09850, 2022.

[15] Z. Li and D. Hoiem, "Learning without Forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935-2947, 2018.

[16] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual Learning with Deep Generative Replay," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017.

[17] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

[18] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[19] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021.

[20] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing Class Imbalance in Federated Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, Art. no. 11, 2021.

[21] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., pp. 7611–7623, 2020.

[22] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10121, 2022.

[23] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning, PMLR*, pp. 5132–5143, 2020.

[24] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, Vol. 10, No. 2, 2019.

[25] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, "FedCM: Federated Learning with Client-level Momentum," *arXiv preprint* arXiv:2106.10874, 2021.

[26] J. Kirkpatrick *et al.*, "Overcoming Catastrophic Forgetting in Neural Networks," in *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[27] D. Lopez-Paz and M. A. Ranzato, "Gradient Episodic Memory for Continual Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017.

[28] K. Luo, X. Li, Y. Lan, and M. Gao, "GradMA: A Gradient-Memory-Based Accelerated Federated Learning with Alleviated Catastrophic Forgetting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3708–3717, 2023.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in A Neural Network," *arXiv preprint* arXiv:1503.02531, 2015.

[30] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-Efficient Federated Learning via Knowledge Distillation," *Nat Commun*, vol. 13, no. 1, p. 2032, 2022.

[31] Y. Chaudhary, P. Rai, M. Schubert, H. Schütze, and P. Gupta, "Federated Continual Learning for Text Classification via Selective Inter-client Transfer," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4789–4799, 2022.

[32] Z. Wu *et al.*, "Exploring the Distributed Knowledge Congruence in Proxy-data-free Federated Distillation," *ACM Trans. Intell. Syst. Technol.*, p. 3639369, 2023.

[33] J. Dong *et al.*, "Federated Class-Incremental Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10164–10173, 2022.

[34] L. Wang and K.-J. Yoon, "Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp.

3048–3068, 2022.

[35] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *arXiv preprint* arXiv:1610.05492, 2017.

[36] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[37] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification," *arXiv preprint* arXiv:1909.06335, 2019.

[38] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," MIT and NYU, Tech. Rep., 2009.

[39] L. Yao and J. Miller, "Tiny ImageNet Classification with Convolutional Neural Networks", *CS 231N*, vol. 2, no. 5, p. 8, 2015.

[40] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2015.

[41] M. Abadi *et al.*, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria: ACM*, 2016.

[42] C. Xiao, Z. Zuo, and S. Wang, "FedGA: Federated Learning with Gradient Alignment for Error Asymmetry Mitigation," Dec. 21, 2024, *arXiv preprint* arXiv:2412.16582.

[43] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Art. no. 1, 2018.

Benteng Zhang (Student Member, IEEE) received the B.S. degree in software engineering from the College of Computer Science and Technology, Qingdao University, Qingdao, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Computer Science and Software Engineering, Hohai University, Nanjing.

His research interests include distributed machine learning, edge computing, and federated learning.

Yingchi Mao (Member, IEEE) received the Ph.D. degree in computer software and theory from the Department of Computer Science and Technology, Nanjing University, Nanjing, China in 2007. She serves with the Key Laboratory of Water Big Data Technology, Ministry of Water Resources, Nanjing, and she is also currently a Professor with the College of Computer Science and Software Engineering, Hohai University, Nanjing. Her main research interests include edge intelligent computing, Internet of Things data analysis, and mobile sensing systems.

Prof. Mao is a Senior Member of the China Computer Federation and the Chinese Association of Automation.

Haowen Xu (Student Member, IEEE) received his B.E. degree in computer and technology from Hohai University, Nanjing, in 2023. He is currently a Master student in the College of Computer and Information at Hohai University, Nanjing. His research interests include distributed learning, edge computing, and federated learning.

Yihan Chen received his B.E. degree in computer science and technology from Hohai University, Nanjing in 2024. He is currently a Master's student in the College of Computer Science and Software Engineering at Hohai University, Nanjing. His research interests include distributed machine learning and edge computing.

Tasiu Muazu received the B.Sc. degree in mathematics from Ahmadu Bello University, Zaria, Nigeria, in 2015, the M.Sc. degree in mathemaics and Ph.D. degree in computer science and technology from Hohai University, Nanjing, China, in 2021 and 2024 respectively. His research interests include federated learning, blockchain, edge computing, and privacy optimization.

Xiaoming He (Member, IEEE) received the Ph.D. degree in Computer Science and Software Engineering from Hohai University, Nanjing, China, in 2023. He is currently a Lecturer with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. Prior to work, he was a Visiting Research Fellow in Singapore University of Technology and Design.

His current research interests include edge intelligence and FPGA-based AI accelerators.

Jie Wu (Fellow, IEEE) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989. He is the Director of the Center for Networked Computing and a Laura H. Carnell Professor with Temple University, Philadelphia, PA, USA, and also serves as the Director of International Affairs, College of Science and Technology. Dr. Wu is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He was an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair for the IEEE Technical Committee on Distributed Processing. He is a Fellow of the AAAS.