IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 26 June 2023; revised 16 October 2023; accepted 11 December 2023.
Date of publication 18 December 2023; date of current version 15 January 2024.

The associate editor coordinating the review of this article and approving it for publication was M. Chen.

Digital Object Identifier 10.1109/TMLCN.2023.3344074

Privacy-Preserving Federated Class-Incremental Learning

JUE XIAO [1,2], XUEMING TANG [1], AND SONGFENG LU [1,2]

[1]School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2]Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518063, China

Corresponding authors: X. TANG (xmtang@hust.edu.cn) AND S. LU (lusongfeng@hust.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grand 2021YFB2012202; in part by the Key Research and Development Plan of Hubei Province of China under Grant 2021BAA038; in part by the Major Program (JD) of Hubei Province under Grant 2023BAA027; in part by the Project of Science, Technology and Innovation Commission of Shenzhen Municipality of China, under Grant JCYJ20210324120002006 and Grant JSGG20210802153009028.

ABSTRACT Federated Learning (FL) offers a collaborative training framework, aggregating model parameters from decentralized clients. Many existing models, however, assume static, predetermined data classes within FLa frequently unrealistic assumption. Real-time data additions from clients can degrade global model recognition of established classes due to catastrophic forgetting. This is exacerbated when new clients, unfamiliar to previous participants, join sporadically. Additionally, there's an imperative for client data privacy. Addressing these, we propose the Privacy-Preserving Federated Class-Incremental Learning (PP-FCIL) approach. This methodology ensures content-level privacy and significantly alleviates the risk of catastrophic forgetting in FCIL. To our knowledge, this is the first research seeks to embed differential privacy into the FCIL settings. Specifically, we introduce a dual-model structure that uses adaptive fusion of new and old knowledge to obtain a new global model. We also propose a multi-factor dynamic weighted aggregation strategy that considers several factors, such as data imbalance timeliness of the model, to speed up global model aggregation and accuracy. For privacy protection, we use Bayesian differential privacy to provide more balanced privacy protection for different datasets. Finally, we conducted experiments on CIFAR-100 and ImageNet to compare our method with other methods and verify its superiority.

INDEX TERMS Federated learning, class-incremental learning, catastrophic forgetting, local differential privacy, dynamic aggregation.

I. INTRODUCTION

FEDERATED learning (FL) is a recent advancement in distributed machine learning (ML). It permits data collection and processing at the client level, subsequently transmitting updated ML parameters to a central server for amalgamation. Its main goal is safeguarding edge and personal data, while facilitating effective ML across diverse parties or computational nodes [1], [2]. To date, FL has seen effective applications in healthcare [3], [4], intelligent robotics [5], [6], autonomous driving [7], [8], among other domains.

FL offers a notable advantage in enabling local training without exchanging personal data between the server and clients, thereby protecting clients' data from being eavesdropped by hidden adversaries [9], [10], [11], [12], [13]. However, with the growing emphasis on data security and individual privacy, privacy preservation has gained prominence, especially in big data applications and distributed learning setups [14], [15], [16]. It has been shown in several studies that even if only gradients are uploaded, they are still subject to model inference attacks, membership inference attacks, and so on, leading to private information leakage [17], [18], [19], [20]. To prevent information leakage, a natural approach is to add artificial noise, with differential privacy (DP) being one prominent example [21], [22]. Existing works on DP-based learning algorithms include local DP (LDP) [23], [24], DP-based distributed SGD [25], and DP meta-learning [26].

We observed that nearly all existing federated learning (FL) methods [27], [28], [29], [30] assume a static learning

process, where the local data of clients remains fixed and unaltered over time. However, in the real world, this assumption is unrealistic as clients' local data often grows over time. Moreover, with continuous data accumulation, catastrophic forgetting (CF) can occur, leading to a significant decline in model performance [31], [32]. CF is when a neural network trained on the original task crashes and decreases its performance on the original task after training on a new task. CF occurs when the neural network's weights are influenced by the new task's data and objectives, causing the original task's knowledge to be corrupted or overwritten. This is more likely to happen when the neural network has fewer parameters and the complexity and diversity of the tasks are higher. The effect of CF is to limit the ability of neural networks to continual learning and the ability to adapt to new tasks in changing environments while retaining knowledge of old tasks. These abilities is essential for developing AI, which needs to deal with a wide variety of problems rather than focusing on a single task. Additionally, in practical applications, the timing when some clients join the FL training is flexible, and they might introduce new categories of data unseen by other clients. Employing existing FL methods in such scenarios can also lead to CF [33].

To solve these practical problems, we study a challenging problem called Federated Class-Incremental Learning (FCIL). Unlike traditional FL, in the FCIL setting, each local client continuously collects training data according to its preference, while new clients with unseen new classes may join the FL training at any time. Another pivotal concern of our study is privacy security within FCIL. To our knowledge, this is the first research seeks to embed differential privacy into the FCIL settings. In addition, the performance of the FCIL model can be significantly affected by issues such as wireless signal distortion and global aggregation errors, which arise from resource-constrained IoT devices. Therefore, it is crucial to design an innovative and communication-efficient model aggregation solution tailored for FCIL.

In this paper, we introduce a novel Privacy-Preserving Federated Class-Incremental Learning (PP-FCIL) approach. This approach is a groundbreaking effort to address the real-world FCIL challenges while ensuring data privacy. Specifically, our approach employs a dual-model structure designed to retain previous knowledge while adeptly adapting to new information, significantly reducing the issue of catastrophic forgetting. We also use model compression to maintain the stability of the dual-model structure and reduce memory pressure on clients. For privacy preservation, considering that in FCIL, clients need to upload gradients frequently, which can lead to greater vulnerability to different ways of attack, we introduce a Bayesian differential mechanism that corrects noise intensity according to data distribution and provides more balanced privacy preservation for different datasets. Additionally, based on data balancing optimization, we propose a multi-factor dynamic weighted aggregation strategy that considers timeliness, accuracy, and frequency of participation in model aggregation of local models to improve convergence speed and accuracy of the global model.

Our main contributions are as follows:

- **Pioneering Privacy Security in FCIL:** While tackling the challenging FCIL issues, we also focus on protecting client privacy. To our knowledge, this is the first research that seeks to embed differential privacy into the FCIL settings.
- **Dual-Model Structure:** We introduce a dual-model structure to avoid catastrophic forgetting problems using a dual-model adaptive feature fusion strategy to dynamically balance old and new knowledge while maintaining stability through model compression.
- **Novel Approach in Bayesian differential privacy:** We introduce a novel approach to measure privacy loss in Bayesian differential privacy that enables users to customize privacy budgets based on data distribution across datasets. Compared with traditional differential privacy models, Bayesian differential privacy not only upholds privacy protection but also elevates the quality of service.
- **Multi-Factor Dynamic Weighted Aggregation Strategy:** We design a multi-factor dynamic weighted aggregation strategy considering several factors, such as data imbalance and model timeliness. Additionally, we utilize Mahalanobis distance to select clients participating in the aggregation process. Compared with traditional FedAvg algorithms, our method significantly improves the convergence speed and accuracy of the global model.

The remainder of the paper is organized as follows. In Section II, we overview the related work, including federated learning, privacy protection and federated class-incremental Learning. Sections III and IV introduce the preliminaries and problem description. In Section V, we describe our approach in detail. We present the experiments and evaluations in Section VI. Section VII concludes the paper.

II. RELATED WORK

A. FEDERATED LEARNING

Federated Learning (FL) [33], [34], [35], [36] primarily trains a decentralized global model by aggregating the network parameters from various local models. McMahan et al. [37] proposed an average-weighted strategy to aggregate multiple local models during the collaborative learning of a global model. Subsequently, they developed the FedProx framework [38] to tackle the data heterogeneity challenges inherent in federated local models. Additionally, [39] introduced a penalty regularizer in the objective function, promoting local models to learn shared optimal results. To reduce communication overhead in federated learning, Chen et al. [40] edsigned a time-aggregation mechanism that employs synchronous learning. This mechanism aggregates deeper layers during the final iterations and shallower layers in each iteration. References [33] and [35] designed a Bayesian

non-parametric strategy to aggregate model parameters of local clients. Peng et al. [41] relied on domain adaptation technology [42], [43] to improve the generalization performance on unsupervised target domains at the local side under federated learning settings. Qu et al. [44] addressed the performance degradation of heterogeneous data across local clients when deployed on different devices with significant shifts. Ding et al. [45] presented the concept of feature hotness metrics to address disparities in optimization speeds across different features, leading to a notable enhancement in federated learning performance. Liu et al. [46] employed a two-layer optimization strategy to derive ''gradient of gradients'' insights from local data. They further introduced two mechanismsdynamic weight assignment and meta-knowledge sharingto address the heterogeneity issue. Yang et al. [47] achieves comparable recognition accuracy on edge devices, broadening the scope of federated learning's potential applications. However, none of the above methods can learn new classes consecutively in a streaming manner and suffer from catastrophic forgetting of old classes when local memory is limited to store old classes.

B. PRIVACY PROTECTION

Due to its strictly provable property, differential privacy is one of the most representative methods used in federated learning to preserve privacy. One widely adopted approach is the differential privacy stochastic gradient Descent (DP-SGD) algorithm proposed by Song et al. [48]. This method segments the gradient and incorporates noise during training, ensuring that the resulting deep model adheres to (ε, δ)-differential privacy. Furthermore, Abadi et al. [25] proposed Moment Account based on DP-SGD. The method obtains a smaller privacy loss by choosing an appropriate scale of noise and fragmentation thresholds. Dong et al. [49] proposed Gaussian differential privacy (GDP) and defined f-DP to accurately describe the optimizer's consumed privacy in each epoch. In recent years, Wei et al. [50] proposed a novel framework called NbAFL for preventing information leakage in federated learning. The method works by adjusting the variance of artificial noise to meet the DP requirements under different protection levels. Hu et al. [51] developed a sparse amplification privacy technique, merging stochastic sparsification and gradient perturbation, to bolster privacy assurances while minimally impacting model accuracy. Wang et al. [52] tackle the limitations of conventional federated learning algorithms by incorporating three differential privacy mechanisms. These enhancements aim to boost model efficacy while ensuring robust user data protection. In this paper, we employ the Bayesian differential mechanism [53] to tailor noise based on data distribution, offering enhanced privacy assurance for similarly distributed data. Concurrently, we also propose a general method for accounting for privacy based on Bayesian differential privacy algorithm, which improves the practicality of differential privacy.

C. FEDERATED CLASS-INCREMENTAL LEARNING

Class-incremental learning (CIL) [31], [54], [55], [56] aims to continuously identify new classes in the real world, which is widely used in image classification tasks [57]. Existing CIL methods [58] can be categorized into three groups: learning without access to old classes, generative replay of old classes, and exemplar memory construction of old classes. When training data for old classes is unavailable, Kirkpatrick et al. [59] propose compensating for biased optimization caused by new classes. In contrast, [60], [61] employ knowledge distillation to address performance degradation (i.e., catastrophic forgetting) on old classes. For generative replay of old classes, [62] relies on adversarial learning to design a memory replay generator for old classes and overcomes catastrophic forgetting by performing replay alignment. Additionally, [32] proposes a dual cooperative model that includes a memory generator to synthesize old class and a task solver to address forgetting. Reference [63] considers distilling causal relations of class-imbalanced training samples. Simon et al. [64] propose an improved knowledge distillation technology and utilize geodesic path to measure the similarity between old and new predictions. Meanwhile, [65], [66], [67] design an adaptive network to balance stability and plasticity. Furthermore, [68] introduces the transformer framework to address forgetting on old classes by designing expandable task tokens. However, addressing the FCIL problem with existing CIL methods [31], [58], [62], [69] requires strong prior knowledge about where and when to collect new classes, which is impractical and violates the requirement of privacy preservation in the real world.

To date, less work has been done on FCIL studies. Dong et al. [70] address the issue of incremental class learning in federated learning with the Global-Local Forgetting Compensation model (GLFC). This method counteracts the forgetting of old classes at local client levels by implementing gradient compensation loss for class perception and distillation loss for class semantic relationships. And in LGA [71], Dong further improves the model performance. However, both GLFC and LGA need a proxy server to achieve their best performance, leading to high communication costs and privacy issues. Based on Enhancer-Transformer, Liu et al. [72] proposed a framework called FedET for communication-efficient federated class incremental learning. This method uses Enhancer and Transformer modules to absorb and transfer new knowledge, respectively, and proposes an enhancer distillation method to solve the problems of local forgetting caused by new classes of new tasks and global forgetting that is not independently and identically distributed. However, neither LGA nor FedET considers data privacy security and requires more computational resources and storage space when dealing with local forgetting.

III. PRELIMINARIES

In this section, we detail the critical premises involved in our scheme, including federated learning, differential privacy and incremental learning.

A. FEDERATED LEARNING

We consider a federated learning architecture optimized with FedAvg, the backbone of commonly-adopted federated learning algorithms. Assuming that there are K clients participating in federated learning, each client has a local private data set $D_i = \left\{x_i^k, y_i^k\right\} (k = 1, 2, \ldots, K)$, where x_i^k and y_i^k are the feature vector and label respectively. Let n_k and n be the data set sizes of the k-th client and all clients respectively, $\rho\left(x_i, y_i; \omega\right)$ is the loss of using the model parameter ω to predict the sample (x_i, y_i). Before participating in the federation training, all clients will uniformly determine the local model training objectives, namely:

$$\min_{\omega \in R^d} \rho(x, y; \omega) = \min_{\omega \in R^d} \frac{1}{n} \sum_{i=1}^{n} \rho\left(x_i, y_i; \omega\right). \tag{1}$$

In the federation training process, the objective equation can be transformed into:

$$\rho(x, y; \omega) = \sum_{k=1}^{K} \frac{n_k}{n} L_k\left(x_k, y_k; \omega\right), \tag{2}$$

where $L_k\left(\boldsymbol{x}_k, \boldsymbol{y}_k; \boldsymbol{\omega}\right) = n/n_k \sum_{i \in D_i} \rho\left(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\omega}\right)$ is the objective equation of client k, $n = \sum_{i=1}^{K} |D_i| = \sum_{i=1}^{K} n_k$ is the size of the data set constructed by all the clients participating in the federation learning, $n_k = |D_i|$ represents the local dataset of client k.

The server first initializes the client's local model parameters. In each round of communication t, clients with proportion F will be randomly selected to communicate directly with the server. Then, each client participating in the federated learning downloads the current global model parameters from the central server and updates their local models synchronously. The server will aggregate the uploaded model parameters to optimize the global shared model further:

$$\omega_{t+1} \leftarrow \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} f_k, \tag{3}$$

where $f_k = \nabla L_k\left(x_k, y_k; \omega_t\right)$ is the average loss on the local dataset of client k. For each client k, $\omega_{t+1}^k \leftarrow \omega_t - \eta f_k$, combined with the above formula, then:

$$\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k. \tag{4}$$

B. DIFFERENTIAL PRIVACY

1) CLASSIC DIFFERENTIAL PRIVACY

(ε, δ)-DP provides a strong criterion for privacy preservation of distributed data processing systems. It provides an information theory security guarantee that the output result of the function is not sensitive to any specific record in the data set. We will now formally define DP as follows.

Definition 1 ((ε, δ)-DP [21]): A random function (algorithm) $\mathcal{M}$:$\mathcal{X} \rightarrow \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ε, δ)-DP, if for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and for any two adjacent databases $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$,

$$\Pr\left[\mathcal{M}\left(\mathcal{D}_i\right) \in \mathcal{S}\right] \leq e^{\varepsilon} \Pr\left[\mathcal{M}\left(\mathcal{D}_i'\right) \in \mathcal{S}\right] + \delta, \tag{5}$$

where $\Pr\left[\mathcal{M}\left(\mathcal{D}_i\right) \in \mathcal{S}\right] / \Pr\left[\mathcal{M}\left(\mathcal{D}_i'\right) \in \mathcal{S}\right]$ is the privacy loss. $\varepsilon > 0$ is privacy budget, which can denote the degree of privacy preservation, and the smaller it is, the better the privacy preservation is, but the more noise is added, and the data usability is decreased. δ represents the probability that the privacy loss breaks the privacy budget.

Definition 2: For numerical data, a Gaussian mechanism defined in [21] can be used to guarantee (ε, δ)-DP. According to [21], we present the following DP mechanism by adding artificial Gaussian noise.

In order to ensure that the given noise distribution $n \sim \mathcal{N}\left(0, \sigma^2\right)$ preserves (ε, δ)-D where $\mathcal{N}$ represents the Gaussian distribution, we choose noise scale $\sigma \geq c\Delta s/\varepsilon$ and the constant $c \geq \sqrt{2\ln(1.25/\sigma)}$ for $\varepsilon \in (0, 1)$. In this result, n is the value of an additive noise sample for a data in the dateset, Δs is the sensitivity of the function s given by $\Delta s = \max_{\mathcal{D}_i, \mathcal{D}_i'} \left\| s\left(\mathcal{D}_i\right) - s\left(\mathcal{D}_i'\right) \right\|$, and s is a real-valued function.

2) BAYESIAN DIFFERENTIAL PRIVACY

Our privacy-preservation scheme includes Bayesian differential privacy [53]. This method offers a privacy loss accounting technique that is more efficient than the traditional moments' accountant.

Definition 3: Bayesian differential privacy. A random function (algorithm) $\mathcal{A}$: $\mathcal{X} \rightarrow \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy. There are any two brother datasets $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$, differing in a single data record $x' \sim \mu(x)$, and for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ it holds that:

$$\Pr\left[\mathcal{M}\left(\mathcal{A}_i\right) \in \mathcal{S}\right] \leq e^{\varepsilon_\mu} \Pr\left[\mathcal{M}\left(\mathcal{A}_i'\right) \in \mathcal{S}\right] + \delta_\mu. \tag{6}$$

The basic properties of Bayesian differential privacy and related proofs can be found in the literature [73]. To derive tighter sequential composition, an alternative definition that implies the above can be used:

$$\Pr\left[L_{\mathcal{M}}\left(\omega, \mathcal{A}_i, \mathcal{A}_i'\right) \geq \varepsilon_\mu\right] \leq \delta_\mu, \tag{7}$$

where $L_{\mathcal{M}}\left(\omega, \mathcal{A}_i, \mathcal{A}_i'\right) = \log \frac{\Pr[\mathcal{M}(\mathcal{A}_i)=\omega]}{\Pr[\mathcal{M}(\mathcal{A}_i')=\omega]}$ is used to represents the privacy loss random variable. Besides, $\Pr\left[L_{\mathcal{M}}\left(\omega, \mathcal{D}_i, \mathcal{D}_i'\right) \geq \varepsilon_\mu\right]$ depends on ω and x', where ω is the randomness of the outcome and x' is the additional example.

The definition of Bayesian differential privacy is very similar to the definition of classical DP. The difference is that it considers the randomness of x' and satisfies two assumptions [53]:

- All data records are not limited to a particular dataset but rather a specific type of data (e.g. emails, MRI images, etc.) or a mixture of such types.
- All data records are exchangeable.

This means that the Bayesian DP can provide worse-case guarantees than the classical DP.

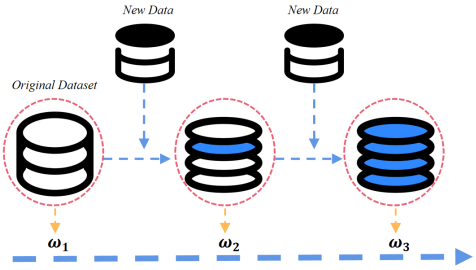

FIGURE 1. Incremental learning.

Definition 4: Rényi divergence of λ between distributions P and Q, denoted as $D_\lambda(P\|Q)$ as:

$$D_\lambda(P\|Q) = \frac{1}{\lambda-1}\log\int p(x)\left[\frac{p(x)}{q(x)}\right]^{\lambda-1}dx = \frac{1}{\lambda-1}\log\int q(x)\left[\frac{p(x)}{q(x)}\right]^{\lambda}dx, \quad (8)$$

where $p(x)$ and $q(x)$ are corresponding density functions of P and Q.

Definition 5: Privacy Cost. Define as the privacy cost of the iteration t:

$$c_t(\lambda) = \log \mathbb{E}_x\left[e^{\lambda D_{\lambda+1}(p_t\|q_t)}\right], \quad (9)$$

where $p_t = p(\omega^{(t)}\omega^{(t-1)}, \mathcal{D})$, $q_t = p(\omega^{(t)}\omega^{(t-1)}, \mathcal{D}').p_t$ and q_t denotes the distribution of private outcomes.

C. INCREMENTAL LEARNING

The core principle of incremental learning is that deep learning models can continuously acquire new data knowledge, maintaining a balance between new and old knowledge. The classification model can perform multi-class tasks on these previously learned classes as new classes are continually added.

In image classification, each incremental step introduces n old classes and m new classes. The goal is to train a model to test the classification performance on $n+m$ classes. The representation of old and new class samples is as follows:

$$X^n = \{(x_i, y_i), 1 \le i \le N, y_i \in [1, \ldots, n]\}, \quad (10)$$

$$X^m = \{(x_i, y_i), 1 \le i \le M, y_i \in [n+1, \ldots, n+m]\}, \quad (11)$$

where N and M represent the number of old and new class samples, respectively, and x_i and y_i denote the image and its corresponding class label.

Incremental learning, mimicking human learning patterns, is devised to handle continuous data streams. It learns from new samples without forgetting old knowledge, as depicted in Fig. 1.

Incremental learning algorithms possess several key features:

- **Continuous learning of new knowledge:** Unlike traditional machine learning algorithms, incremental learning can learn new knowledge continuously and update the model, addressing the issue of continuous data streams and reducing training time.
- **Avoids reprocessing old samples:** Incremental learning applies only new data without requiring the training data of old knowledge.
- **Retains old knowledge:** In practical applications, a proficient incremental learning system retains old knowledge while assimilating new, capable of recognizing previously learned old samples.

IV. PROBLEM DESCRIPTION

In this section, we outline federated incremental tasks, privacy threat model and the efficiency issue that must be addressed.

A. FEDERATED INCREMENTAL TASKS

Most existing models make an unreasonable assumption that the data classes in the FL framework are predetermined and fixed beforehand. The issue arises when local clients receive new classes successively, leading to limited memory to store old classes. This assumption significantly reduces the recognition performance of the global model on old classes, also known as catastrophic forgetting. Furthermore, introducing new local clients that collect novel and unseen classes adds to the irregularity in FL training, further exacerbating catastrophic forgetting of old classes. Therefore, one of the issues this article seeks to address is handling incremental tasks in FL and mitigating catastrophic forgetting.

B. PRIVACY THREAT MODEL

Similar to traditional FL, there are primarily two roles in FCIL: a server responsible for aggregating the global model and K clients, each possessing their own data. The objective for the clients is to collaboratively train a machine learning model on the server using their individual data while ensuring privacy. Throughout this process, the original data from the clients is not uploaded to the server. However, studies [17], [18], [19], [20] have shown that merely transmitting gradients can still be susceptible to various attacks. To address these issues, researchers typically modify the original data to prevent privacy leaks, such as using differential privacy, generalization, and perturbation. However, while protecting privacy, these methods also alter sample features that affect the accuracy of the analytical service model. Therefore, how to guarantee privacy-preservation while improving the quality of the service model remains another pressing issue in this paper.

C. EFFICIENCY ISSUE

In traditional federated learning, the weighted aggregation of the global model is determined based on the data size contributed by participating users. For example, in the FedAvg algorithm, participating users with larger local training datasets significantly impact the global model. However, this approach can lead to model skewness, which can affect the convergence speed and accuracy of the global model,

especially when faced with severely imbalanced data or users with varied data quality. In FCIL, this issue is further amplified, necessitating consideration of other factors. Therefore, how to design a dynamic aggregation algorithm that aligns with FCIL is a critical issue that this paper aims to explore.

V. OUR PROPOSED SCHEME

In this section, we will discuss in detail the solutions to the three problems in Section IV. Specifically, they include framework design, Federated incremental learning based on dual-model, Local Bayesian differential privacy, and Multi-factor dynamic weighted aggregation strategy.

A. FRAMEWORK DESIGN

In our framework, we use $C_1,C_2,\ldots,C_k$ to represent K clients, each of which has its own corresponding data set $\mathcal{D}^1,\mathcal{D}^2,\ldots,\mathcal{D}^k$, let $\mathcal{D}^k = \{\mathcal{D}_t^k\}_{t=1}^T$ denote a sequence of training data of client k, where $\mathcal{D}_t^k = \{\mathcal{X}_t^k, \mathcal{Y}_t^k\}$ is the coming dataset at round t. $\mathcal{X}_t^k$ and $\mathcal{Y}_t^k$ denote data and corresponding labels. For different step i and j, $\mathcal{X}_i^k \cap \mathcal{X}_j^k = \emptyset$ and $\mathcal{Y}_i^k \cap \mathcal{Y}_j^k = \emptyset$. The server represents the cloud server that aggregates the local models. In the t round, each client C_i uses its local dataset to train the local model with the model parameters represented by ω_t^i.

We have used local Bayesian differential privacy mechanism, Multi-facto dynamaic weighted aggregation algorithm and Federated incremental learning algorithm based on the fusion of dual models in the framework. Below we will detail these techniques and how to utilize them.

B. FEDERATED INCREMENTAL LEARNING BASED ON DUAL-MODEL

The current federated learning approach typically involves training a federated model on a static dataset, which is not updated dynamically. However, in real-world scenarios, client data is often subject to change and requires dynamic updates. We are aware that incremental learning often leads to the catastrophic forgetting problem. Similarly, if we train the previous federated model directly with newly added data, it will also result in catastrophic forgetting. To address this issue, we have proposed a federated incremental learning algorithm that utilizes dual-model fusion. The specific approach is as follows:

1) CONSTRUCTING DUAL-MODEL

In order to construct a dual model, the training dataset must first be processed. To ensure the sensitivity of the federated model to old samples, we take into account memory limitations. Therefore, after completing model training at each stage, we select a small number of representative paradigm samples from the old samples. By using both the new samples and the paradigm samples, an incremental local model is obtained through joint training during the incremental learning stage.

We use the Herding algorithm [74] to select paradigm samples from the old samples. First, calculate the class mean μ of the round t training data for client k:

$$\mu = \frac{1}{|D_t^k|}\sum_{x\in n_t^k}\Omega_s(x), \tag{12}$$

where $|D_t^k|$ is the size of the training data volume for the t-th model aggregation client k and Ω_s is the feature extractor. If p paradigm samples are selected, then the paradigm samples S_t^k are:

$$s_p \leftarrow \underset{x\in D_t^k}{\operatorname{argmin}} \left\| \mu - \frac{1}{p}\left[\Omega_s(x) + \sum_{i=1}^{p-1}\Omega_s(s_i)\right]\right\|, \tag{13}$$

$$S_t^k \leftarrow \{s_1, s_2, \ldots, s_p\}. \tag{14}$$

In the initial step, we train with $\mathcal{D}_1^k$, resulting in the model ω_1^k. Due to memory limitations, we can't store the entirety of $\mathcal{D}_1^k$. Instead, we select a subset S_1^k and store it in the memory bank. At round t ($t > 1$), we train the model ω_t^k using $\mathcal{D}_t^k \cup S_{t-1}^k$. Upon completion of the model training, the memory bank is updated with the subset S_t^k. Throughout the incremental learning process, we consistently iterate through training the model and updating the memory bank.

2) DESIGNING THE LOSS FUNCTION

In order to achieve a balance between model stability and plasticity during the training of the dual-model, we have introduced a loss function that considers three aspects: classification loss to address class imbalance, distillation loss to preserve old knowledge, and SupCon loss to learn new knowledge [75].

a: CLASSIFICATION LOSS

In FCIL scenario, as new data classes are added, the model tends to increasingly classify data into these new classes. To address this issue, we adopted the Balanced Softmax approach proposed in [76] as a replacement for the softmax activation function. Consequently, the classification loss is defined as follows:

$$\mathcal{L}_{\mathcal{C}} = -\sum_{i=1}^{K_t}\sigma_{y=i}\log h_i(x), \tag{15}$$

where x and y represent the input image and its corresponding label, respectively. $\sigma_{y=i}$ is an indicator function. K_t denotes the total number of classes in the t round. $h_i(x)$ is the output probability for the i-th class, and is computed as follows:

$$h_i(x) = \frac{n_i e^{Z_i(x)}}{\sum_{j=1}^{K_t} Z_j(x)}, \tag{16}$$

where n_i is the number of samples in i-th class and $Z_i(x)$ is the i-th output logit of $\mathcal{Z}_{mix}$. The Balanced Softmax significantly alleviates the adverse effects of data imbalance without introducing additional computational overhead during training.

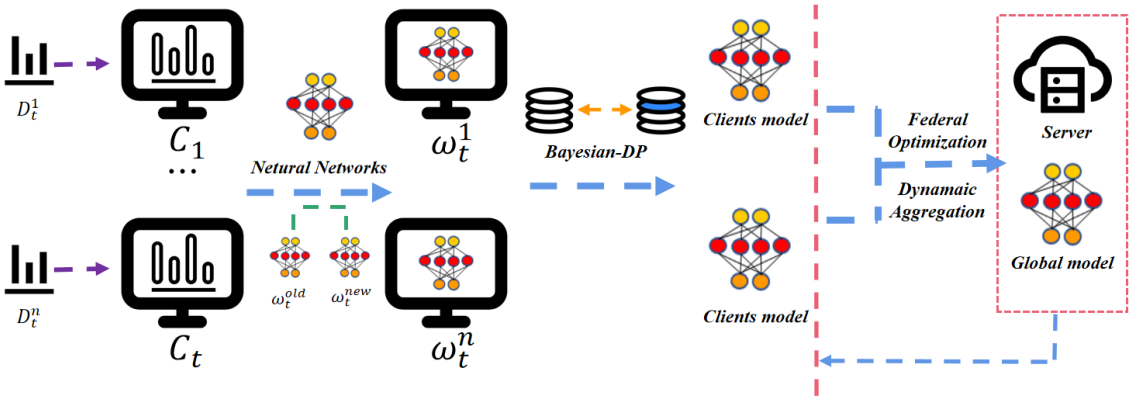

FIGURE 2. **Framework design. (Note that the figure only describes one iteration process.)**

b: DISTILLATION LOSS

While the dual-model reutilizes the old model, it remains imperative to conduct knowledge distillation to control the retention of old knowledge within the mixed features $\mathcal{F}_{mix}$. We denote Ω_{old} and the output probabilities of the current dual-model as $\hat{h}_i(x)$ and $h_i(x)$, respectively. The distillation loss is then adopted as follows:

$$\mathcal{L}_{\mathcal{D}} = -\sum_{i=1}^{K_{t-1}} \hat{h}_i(x) log(h_i(x)), \tag{17}$$

$$\hat{h}_i(x) = \frac{e^{\hat{Z}_i(x)/\tau}}{\sum_{j=1}^{K_{t-1}} e^{\hat{Z}_j(x)/\tau}}, \; h_i(x) = \frac{e^{Z_i(x)/\tau}}{\sum_{j=1}^{K_{t-1}} e^{Z_j(x)/\tau}}, \tag{18}$$

where τ represents the temperature, and K_{t-1} denotes the number of old classes in the round t. $\hat{Z}_i(x)$ and $Z_i(x)$ are the output logits for the i-th class from $\mathcal{Z}_{mix}$ and $\mathcal{Z}_{old}$, respectively.

In the knowledge distillation scenario, there exist two models, an old model and a current dual model. Specifically, we calculate the output probability of the old model and use it as the training target for the student model. By minimizing the difference in output probabilities between the old model and the new model (knowledge distillation loss), the student model can learn the knowledge of the old model and perform well on the new tasks.

c: SUPCON LOSS

The second branch of the dual-model is tasked with learning new knowledge and extracting new features, denoted as $\mathcal{F}_{new}$. We aspire for $\mathcal{F}_{new}$ to be distinguishable and expressive rather than merely passively acquired intermediate features during the end-to-end training process. To achieve this, we employ the Supervised Contrastive Learning (SCL) approach, inspired by the work of Khosla et al. [77], which leverages both labels and data augmentation to constrain Ω_{new}. Specifically, to compute the SupCon loss, we use the projection head G_t to map $\mathcal{F}_{new}$ to a new space:

$$G_t(\mathcal{F}_{new}) = \varphi(M_2 \cdot \varphi(M_1 \cdot \mathcal{F}_{new} + b_1) + b_2), \tag{19}$$

where M_1 and M_2 are weight matrices, b_1 and b_2 are bias terms, and φ is the **ReLU** activation function.

Let $\mathcal{D}_t^{k:all} = \left\{\mathcal{X}_t^{k:all}, \dagger_t^{k:all}\right\}$ denotes the union of training data $\hat{\mathcal{D}}_t^k$ in t round for client k and its augmentation. For a given input image x and its corresponding label y, we define $p(x)$ as the positive set. This set comprises images from the same class within $\mathcal{X}_t^{k:all}$ but excludes x. Specifically:

$$p(x) = \left\{x_i \in \mathcal{X}_t^{k:all} | x_i \neq x, y_i \neq y\right\}. \tag{20}$$

Thus, the SupCon loss is formulated as:

$$\mathcal{L}_{\mathcal{S}} = -\frac{1}{|p(x)|} \sum_{x_i \in p(x)} \log \frac{e^{z \cdot z_i/\tau}}{\sum_{x_j \in \mathcal{X}_t^{k:all}} e^{z \cdot z_j/\tau}}, \tag{21}$$

where z, z_i, and z_j are the corresponding projection features of x, x_i, and x_j, respectively. τ is the temperature scalar.

d: FINAL OBJECTIVE

The loss function of the dual-model is:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\mathcal{C}} + \lambda\mathcal{L}_{\mathcal{D}} + \varrho\mathcal{L}_{\mathcal{S}}, \tag{22}$$

where $\lambda = \frac{K_{t-1}}{K_t}$ is used to balance classification loss and distillation loss, and ϱ is a hyper-parameter adjusting the weight of the SupCon loss.

3) DUAL-MODEL ADAPTIVE FEATURE FUSION

For $\mathcal{F}_{old}$ and $\mathcal{F}_{new}$, we posit that a straightforward addition or concatenation operation might compromise the intrinsic knowledge embedded within them. Consequently, we propose a method of adaptive feature fusion to get a more expressive and separable set of mixed features, denoted as $\mathcal{F}_{mix}$, which is beneficial for imbalanced classification.

$\mathcal{F}_{new}$ offers superior classification performance for images of new classes. This is because Ω_{new} is trained based on all data and the SupCon loss. Similarly, $\mathcal{F}_{old}$ demonstrates

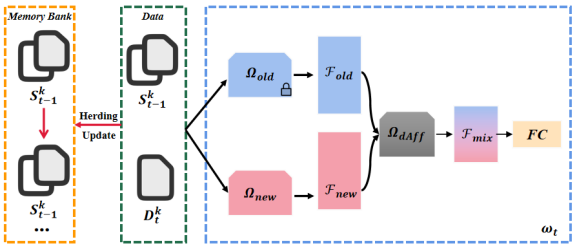

FIGURE 3. Dual-model Strategy. The dual-model strategy comprises two feature extractors, namely Ω_{old} and Ω_{new}, along with a dual-model adaptive feature fusion module, Ω_{dAff}. At round t for client k, we freeze the model trained previously at round $t-1$ to serve as Ω_{old} to preserve existing information. Subsequently, we introduce a trainable network, Ω_{new}, with a consistent architecture to learn new knowledge.

enhanced classification capabilities for images of old classes, given that Ω_{old} is inherently designed to classify the old classes. To optimally harness the information from both $\mathcal{F}_{old}$ and $\mathcal{F}_{new}$, we introduce the following dual-model adaptive feature fusion module [75]:

$$\begin{aligned}\mathcal{F}_{mix} &= \Omega_{dAff}(\mathcal{F}_{old}, \mathcal{F}_{new}) \\ &= \mathcal{F}_{old} \otimes \Delta + \mathcal{F}_{new} \otimes (1-\Delta).\end{aligned} \tag{23}$$

The $\otimes$ and Δ represent the multiplications between matrices and the weight of each channel, respectively. We use the channel attention mechanism proposed in [78] to compute the fusion weights Δ:

$$\Delta = \sigma_m\left(W_2\varphi\left(W_1(\mathcal{F}_{old})\right)\right), \tag{24}$$

where σ_m is the **Sigmoid** function and φ is the **ReLU** activation function. Δ is a simple gating mechanism with a **Sigmoid** function. Ω_{dAff} consist of two fully connected layers $W_1 \in \mathbb{R}^{\frac{C}{r}\times r}$ and $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$, r is the reduction ratio.

The specific implementation process of Dual-model Adaptive Feature Fusion is as follows:

- **Initial Processing:** The input channel descriptor, denoted as $\mathcal{F}_{old}$, is first processed by the initial fully-connected layer. This layer assimilates the weights W_1 to produce an intermediate representation.
- **Nonlinear Transformation:** This intermediate representation is then subjected to a nonlinear transformation using the **ReLU** activation function, resulting in a new intermediate state.
- **Deducing Modulation Weight:** The second fully connected layer takes this newly derived representation and deduces the channel's modulation weight Δ by assimilating the weights W_2.
- **Final Fusion:** The final step involves calculating $\mathcal{F}_{mix}$ to obtain the fused model.

Furthermore, focusing on the channels proves sufficient in this approach [78], yielding significant improvements with only a minimal increase in computational demand and parameter count.

4) MODEL COMPRESSION

Since we use Ω_{new} in every round of training, this leads to an infinite increase in the size of the dual-model. In order to preserve the dual-model's two-branch structure and alleviate the pressure on client storage caused by the model's increase in size, we compress the dual-branch feature extraction network Ω_t into a single-branch network Ω_t':

$$\mathcal{L}_{\mathcal{M}} = (1-\lambda)\mathcal{L}_{\mathcal{C}} + \lambda\Gamma\left\|\mathcal{F}_{mix} - \mathcal{F}'_{mix}\right\|^2, \tag{25}$$

where $\lambda = \frac{K_{t-1}}{K_t}$, $\mathcal{F}'_{mix}$ is the mixed feature of $t-1$ round, and Γ is a hyper-parameter controlling the weight of feature distillation loss. Compressing the model necessitates additional knowledge integration time, thereby incurring extra computational costs. However, experimental results indicate that the increased computational costs are justified, given the improvements in model performance.

C. LOCAL BAYESIAN DIFFERENTIAL PRIVACY

Compared with classical DP, BDP can add the noise based on the client's local data distribution, which makes the noise addition more reasonable. Besides, we propose a Local Bayesian Account to monitor the privacy loss of clients in the FCIL by utilizing the principle and advanced composition theorem of BDP. The implementation steps are as follows:

(1) During training, computing the privacy cost $c_t(\lambda)$ for each iteration of the local model requires knowledge of the data distribution $\mu(x)$, which is difficult to do in FCIL. According to Definition 5, we need to estimate $\mathbb{E}_x\left[e^{\lambda D_{\lambda+1}(p_t\|q_t)}\right]$.

(2) We use a subsampled Gaussian noise mechanism for privacy-preserving FCIL. Thus, the output distribution $p(\omega^{(t)}\omega^{(t-1)}, \mathcal{D}')$ is calculated as follows [53]:

$$p\left(w^{(t)}w^{(t-1)}, \mathcal{D}'\right) = (1-q)\mathcal{N}\left(g_t, \sigma^2\right) + q\mathcal{N}\left(g'_t, \sigma^2\right), \tag{26}$$

where g_t and g'_t are the non-private outputs of t iterations. They correspond to the cases without x' and with x', respectively. σ is the noise parameter of the mechanism and q represents the probability of data sampling.

(3) According to $p(\omega^{(t)}\omega^{(t-1)}, \mathcal{D}')$ and Definition 4, the cost of privacy is given by:

$$\begin{aligned}c_t(\lambda) &= log\mathbb{E}_x\left[\mathbb{E}_{k\sim B(\lambda+1,q)}[e^{\frac{k^2-k}{2\sigma^2}\|g_t-g'_t\|^2}]\right] \\ &= og\mathbb{E}_x\left[\sum_{k=0}^{\lambda+1}\binom{\lambda+1}{q}q^k(1-q)^{(\lambda+1-k)}e^{\frac{k^2-k}{2\sigma^2}\|g_t-g'_t\|^2}\right],\end{aligned} \tag{27}$$

where $B(\lambda+1, q)$ represents the binomial distribution, and $\lambda+1$ and q are the number of experiments and the probability of success, respectively.

(4) Therefore, the privacy cost of each iteration is summed up to obtain the privacy cost of the whole learning process. To ensure that the privacy cost does not exceed the privacy

budget, for a fixed ε, $c_t(\lambda)$ must satisfy the following condition [53]:

$$log\delta \leq \sum_{t=1}^{T} c_t(\lambda) - \lambda\varepsilon. \tag{28}$$

In conclusion, we have advanced the Bayesian differential privacy framework to embrace local Bayesian differential privacy. Specifically, Steps 1 and 2 outline the methodology for determining the privacy cost in each iteration of federated learning. Step 3 introduces a precise formula to compute this per-iteration privacy cost. Lastly, Step 4 presents an inequality to ensure that the cumulative privacy cost throughout the learning procedure remains within a predefined privacy budget. By introducing noise to the model parameters, attackers are unable to obtain precise output results through querying the model, thus preventing them from recovering the training data or inferring whether a specific sample belongs to the model's training data or not based on the output results.

D. MULTI-FACTOR DYNAMIC WEIGHTED AGGREGATION STRATEGY

In the context of federation incremental learning, the communication pressure between clients and central servers will gradually increase and the speed of model aggregation will slow down due to increasing data. To address this problem, we propose a multi-factor dynamic weighted aggregation strategy and Federal Preferred client Strategy by the following methods:

1) DATA BALANCING WEIGHTED AGGREGATION

FedAVG only considers the amount of data in the training set of participating clients to influence the model aggregation(the larger the amount of data the more impact the client has on the global model), which may lead to the global model being heavily skewed towards the participating clients with large amounts of data and does not handle incremental data. To address this issue, we propose incremental weight, which influence the aggregation strategy by modifying the weight values of the clients model.

The incremental weight of the client k can be calculated using the following formula:

$$I_t^k = \frac{|d_t^k|}{|D_{t-1}^k| + |d_t^k|}, \tag{29}$$

where $|D_{t-1}^k|$ denotes the sample size of data for client k at round $t-1$ and $|d_t^k|$ denotes the incremental sample size of data for that client at round t. The incremental weight I_t^k indicates the proportion of the number of newly added samples from the client k to the total number of samples in the client k.

We define a parameter depth value $\theta_t^k = \theta_{t-1}^k - (I_t^k \cdot \theta_{t-1}^k)$, which represents new data added to the local dataset after the client has completed its iterative learning, and to a certain extent, reflects the degree of update at the client. The initial value of θ_t^k can be set to $|D_0^k|/|D_0|$, $|D_0^k|$ denotes the original sample size of data for client k, and $|D_0|$ denotes the original sample size of data for all client. During the time interval between downloading and uploading parameters by the client, new training data is generated and the depth values are updated to some extent.

In order to ensure that clients with larger depth values have smaller parameter weights and a relatively smooth decay process, the inverse tangent function is chosen as the incremental weighted decay function in this paper:

$$\gamma_t^k = \tanh(\theta_t^k + \alpha) = \frac{e^{\theta_t^k+\alpha} - e^{-(\theta_t^k+\alpha)}}{e^{\theta_t^k+\alpha} + e^{-(\theta_t^k+\alpha)}}, \tag{30}$$

where a is α hyperparameter, set according to the size of the data. Then the aggregation strategy changes to:

$$\omega_t = \sum_{k=1}^{K} \gamma_t^k \cdot \omega_t^k. \tag{31}$$

2) TEMPORALLY WEIGHTED AGGREGATION

With the increase in the number of clients to be involved and the uncertainty of the random participant selection method, some clients may not be involved in the model aggregation process of the federated learning system for a long period of time. As a result the model parameters of these clients remain unchanged for a long period of time and their local model parameters may differ significantly from the federated model parameters. In order to avoid the excessive impact on the federated model when the clients that have not joined the federated learning for a long time participate in the aggregation, some scholars have proposed a Temporally Weighted Aggregation algorithm [40], [79] that is conducive to improving the convergence speed of the federated model,the basic formula of which is as follows:

$$T_t^k = (\frac{e}{2})^{-(t-t_k)}. \tag{32}$$

Temporally Weighted Aggregation algorithm affixes timestamps to all clients involved in federated learning. Where t represents the number of federated model training rounds, t_k represents the training round (timestamp) of the client k whose model was last updated, and e is the base of the natural logarithm. The equation can be interpreted to mean that the parameter information uploaded by participants with more frequent model updates will be given more weight in the model aggregation process, given that the participants' local datasets are of equal size.

Then the aggregation strategy changes to:

$$\omega_t = \sum_{k=1}^{K} \frac{|D_t^k|}{|D_t|} \cdot T_t^k \cdot \omega_t^k. \tag{33}$$

3) MULTI-FACTOR DYNAMIC WEIGHTED AGGREGATION

The weighted aggregation algorithm optimized based on data balance considers the influence of the size of each participating user's local dataset on model aggregation and the weighted aggregation algorithm based on model timeliness

considers the time span between the last communication round that a client participated in model aggregation and the current communication round of federated model aggregation. Additionally, because the accuracy of each client's trained model may vary, aggregating models with the same weight for poorly and highly accurate models will slow down the convergence speed of the model. To address this issue, we consider increasing the weight ratio of highly accurate client models in model aggregation, with the specific formula as follows:

$$A_t^k = \frac{\beta_t^k}{\beta_t}, \tag{34}$$

where β_t represents the sum of the accuracy of local models of all participating clients in the current round of model aggregation on their respective local training sets at round t, while β_t^k represents the size of the accuracy of the local model of participating client k on the local test set at round t.

Generally speaking, the client models that participate in model aggregation more frequently are closer to the final trained model. Therefore, we also consider the weight proportion of the total number of times that a client model participates in model aggregation during the model aggregation process, with the specific formula as follows:

$$R_t^k = (\frac{r_t^k}{r_t} + 1)/2, \tag{35}$$

where r_t represents the total number of times all client models participate in model aggregation in round t, while r_t^k represents the total number of times client k participates in model aggregation at present in round t. The calculation of R_t^k is shown in Algorithm 1. Note that in the sixth line of Algorithm 1, we compress R_t^k to between 0.5 and 1 to prevent extreme values from having an impact on the model.

It is worth noting that Eq. (32) focuses on the last round in which the client participated in the aggregation, while Eq. (35) focuses on the total number of times the client participated in the aggregation.

Algorithm 1 Rounds_ratio (at Round t)

Input The set of selected clients SC_t, the frequency of client k participation in aggregation r_t^k
Output Weight proportion R_t^k

1: Initialize $r_t \leftarrow 0$
2: **for** each client $k \in SC_t$ in parallel **do**
3: $\quad r_t \leftarrow r_t + r_t^k$
4: **end for**
5: **for** each client $k \in SC_t$ in parallel **do**
6: $\quad R_t^k \leftarrow (r_t^k / r_t + 1)/2$
7: **end for**
8: $F_t \leftarrow \{R_t^k\}$
9: **return** F_t

Combining all of the above factors, we obtain the multi-factor dynamic weighted aggregation strategy, which is:

$$\omega_t = \sum_{k=1}^{K} (a \cdot \gamma_t^k + b \cdot \frac{|D_t^k|}{|D_t|} \cdot T_t^k + c \cdot A_t^k + d \cdot R_t^k) \cdot \omega_t^k, \tag{36}$$

where, a, b, c, and d are hyper-parameters, and $a + b + c + d = 1$. Multi-facto Dynamaic Weighted Aggregation Algorithm is an improvement upon the FedAvg algorithm that aims to mitigate the influence of other factors on the model's accuracy.

4) FEDERAL PREFERRED CLIENT STRATEGY

In traditional federated learning, the clients participating in federated learning are mostly determined by setting a fixed proportion or based on a set fixed threshold. However, in FCIL, this approach has many shortcomings.

- In FCIL, the client's local dataset is constantly growing, simply randomly selecting a subset of participants according to the communication ratio can easily lead to a skewed training process for global models.
- Depending on a fixed threshold can sometimes be time-consuming, and in the later stages of training, the global model may fluctuate and struggle to converge.

Our system assigns a weight value to each client. Before each round of communication, the server calculates and sorts the weight values of all clients, and then selects a subset of clients based on a pre-set proportion F. If a client is not selected, they accumulate local parameter information and proceed to the next round of learning iterations until they are selected again.

It is worth noting that the weight value calculation is carried out throughout the learning process. Whether selected or not, the weight value must be computed after the upcoming round of learning.

To obtain a comprehensive and accurate one-dimensional performance indicator that accurately describes the participation level of each customer, we use Mahalanobis distance [80] to calculate the accuracy, loss, and kappa feature vectors of customers. We use the sum of Mahalanobis distances between each client's performance indicators and those of other customers as the client's rank value SD. The larger the weight value, the less similarity in performance indicators; vice versa.

Assuming that the sub-ends of two factories are represented by $x = (x_{acc}, x_{loss}, x_{kappa})$ and $y = (y_{acc}, y_{loss}, y_{kappa})$, respectively, the formula for calculating the covariance of x and y is: formula for calculating the covariance of x and y is:

$$p = \sum_{ij} Cov(x_i, y_i) = E[(x_i - \mu_x)(y_i - \mu_y)], \tag{37}$$

where $\mu_x = E[x_i]$, $\mu_y = E[y_i]$.

The calculation formula of Mahalanobis distance $MD(x, y)$ of two non-independent and identically distributed client x and y is:

$$MD(x, y) = \sqrt{(x - y)^T p^{-1} (x - y)}. \tag{38}$$

Algorithm 2 SeverExecution

Input: Total number of clients K, participation proportion F
Output: Global model ω_t

Initialize a, b, c, d
for each client $k \in 1, 2, \ldots, K$ **do**
  $t_k \leftarrow 0$
  $r_t^k \leftarrow 0$
end for
for each rount $t = 1, 2, \ldots$ **do**
  $m \leftarrow \max(K \cdot F, 1)$
  $M_t \leftarrow order(SD_1, SD_2, \ldots, SD_k)$
  The set of selected clients $SC_t \leftarrow$ (top m set of M_t)
  $F_t \leftarrow$ **Rounds_ratio**(SC_t, r_t^k)
  for each client $k \in SC_t$ in parallel **do**
    $\omega_t^k, \gamma_t^k \leftarrow$ Execution of Algorithm **ClientUpdate**
    $\beta_t = \beta_t + \beta_t^k$
    $T_t^k \leftarrow (\frac{e}{2})^{-(t-t_k)}, A_t^k \leftarrow \frac{\beta_t^k}{\beta_t}, R_t^k \leftarrow F_t[k]$
  end for
  $\omega_t \leftarrow \sum_{k=1}^{K}(a \cdot \gamma_t^k + b \cdot \frac{|D_t^k|}{|D_t|} \cdot T_t^k + c \cdot A_t^k + d \cdot R_t^k) \cdot \omega_t^k$
  for each client $k \in SC_t$ **do**
    $r_t^k \leftarrow r_t^k + 1$
    $t_k \leftarrow t$
  end for
  return ω_t (Send it to all clients)
end for

TABLE 1. The effect of scale factor F on model performance on CIFAR-100 ($\mathcal{T} = 10$ and $K = 50$).

F	**Ours1**	None-OP
0.3	**69.28**	64.08
0.5	**74.44**	68.44
0.7	**75.21**	70.56
1	**75.90**	73.35

Then it is concluded that the weight value of the clients is:

$$SD_i = \sum_{j=0}^{K} MD(x_i, y_j), \tag{39}$$

where, $i \neq j$, K represents the number of all clients.

VI. EXPERIMENT AND EVALUATION

To validate the effectiveness of our method, we build a federated learning model and simulate our experiments. And we also evaluated the performance of our scheme.

A. EXPERIMENT SETTING

1) BENCHMARK PROTOCOL

We follow the benchmark protocol proposed in [31]: for a given multi-class classification dataset, the classes are arranged in a fixed random order. Each method is then trained in a class-incremental way on the available training data. After each batch of classes, the resulting classifier is evaluated on the test part data of the dataset, considering only those classes that have already been trained. Note that, even though the test data is used more than once, no overfitting can occur, as the testing results are not revealed to the algorithms. The result of the evaluation are curves of the classification accuracies after each batch of classes. If a single number is preferable, we report the average of these accuracies, called average incremental accuracy (Avg). Therefore, average incremental accuracy (Avg) is used throughout the experiments in this paper.

We divided all classes into $\mathcal{T}$ steps in our experiments and fixed the memory bank's size. We use the CIFAR-100 (ImageNet) dataset, where client local data is incremented in batches of 5, 10, and 20 classes at a time (20, 10, and 5 steps), each containing a random amount of data. A total of 100 classes of data are trained for each client. All experiments were repeated 10 times, and the average value of the classification performance of the corresponding parameters was compared and analyzed.

2) DATA AUGMENTATION

To obtain better experimental results, we utilized several data augmentation methods to create a more extensive training

Algorithm 3 ClientUpdate (at round t)

Input: Paradigm samples S_{t-1}^k, compressed module Ω_{old}, previous FC layer $\mathcal{FC}_{t-1}$
Output: Local model ω_t^k, data balance weighting factor γ_t^k, new FC layer $\mathcal{FC}_t$

Get the global model ω_{t-1} from the Server
if $|D_t^k|$ **then**
  $I_t^k \leftarrow |d_t^k|/(|D_{t-1}^k| + |d_t^k|)$
  $\theta_t^k \leftarrow \theta_{t-1}^k - (I_t^k \cdot \theta_{t-1}^k)$
else
  $\theta_t \leftarrow t$
end if
$\gamma_t^k \leftarrow [e^{\theta_t^k + \alpha} - e^{-(\theta_t^k + \alpha)}]/[e^{\theta_t^k + \alpha} + e^{-(\theta_t^k + \alpha)}]$
$V_t^k \leftarrow D_t^k \cup S_{t-1}^k$
for epochs **do**
  for mini-barches x in V_t^k **do**
    $x' =$ augmentation data on x
    $\mathcal{F}_{old} = \Omega_{old}(x), \mathcal{F}_{new} = \Omega_{new}(x)$
    $\mathcal{F}_{mix} = \Omega_{dAff}(\mathcal{F}_{old}, \mathcal{F}_{new}) = \mathcal{F}_{old} \otimes \Delta + \mathcal{F}_{new} \otimes (1 - \Delta)$
    $\mathcal{Z}_{mix} = \mathcal{FC}_t^k(\mathcal{F}_{mix}), \mathcal{Z}_{old} = \mathcal{FC}_{t-1}^k(\mathcal{F}_{old})$
    Train ω_t^k by loss:
    $\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{C}}(\mathcal{Z}_{mix}) + \lambda\mathcal{L}_{\mathcal{D}}(\mathcal{Z}_{mix}, \mathcal{Z}_{old}) + \varrho\mathcal{L}_{\mathcal{S}}(G_t(\mathcal{F}_{new}), G_t(\Omega_{new}(x')))$
  end for
end for
Compress model by $\mathcal{L}_{\mathcal{M}}$
Obtain $\omega_t^{k*} \leftarrow$ local Bayesian differential privacy
Obtain $s_t^k \leftarrow \{s_1, s_2, \ldots, s_a\}$ of D_t^k
UPdate $S_t^k \leftarrow s_t^k \cup S_{t-1}^k$ in memory
return $\omega_t^{k*} \rightarrow \omega_t^k, \gamma_t^k, \mathcal{FC}_{t-1}$

TABLE 2. Comparisons in terms of accuracy on CIFAR-100 dataset when the number of Tasks $\mathcal{T} = 5$.

Methods	20	40	60	80	100	Avg (%)	Imp (%)
iCaRL+FL	80.6	64.1	52.8	48.0	43.5	57.80	16.94
BiC+FL	81.1	65.3	58.0	50.6	44.2	59.84	14.90
PODNet+FL	82.3	65.9	60.8	52.6	47.0	61.72	13.02
DyTox+FL	82.3	71.2	66.7	63.4	58.8	68.48	6.26
GLFC	82.4	76.9	66.8	64.1	55.4	69.12	5.62
LGA	82.8	76.4	72.8	66.7	62.3	72.20	2.54
FedET	84.8	78.0	79.1	70.2	68.0	76.02	-1.28
Ours1	**82.7**	**79.2**	**75.4**	**71.0**	**65.4**	**74.74**	**-1.28 ∼ 16.94**

TABLE 3. Comparisons in terms of accuracy on ImageNe dataset when the number of tasks $\mathcal{T} = 5$.

Methods	20	40	60	80	100	Avg (%)	Imp (%)
iCaRL+FL	76.7	61.4	47.3	43.7	38.8	53.58	17.28
BiC+FL	76.8	62.3	53.0	45.8	41.2	55.82	15.04
PODNet+FL	77.9	62.3	57.1	49.6	43.0	57.98	12.88
DyTox+FL	71.2	67.5	63.2	57.7	48.8	61.68	9.18
GLFC	78.4	72.9	64.3	56.4	50.4	64.48	6.38
LGA	81.6	76.3	65.2	58.5	55.1	67.34	3.52
FedET	79.1	70.3	66.9	62.2	54.8	66.66	4.20
Ours1	**79.8**	**75.6**	**71.1**	**66.7**	**61.1**	**70.86**	**3.52 ∼ 17.28**

dataset, including cropping, flipping, and colour distorting. Firstly, we randomly cropped the CIFAR-100 (ImageNet) image with the scale of [0.2, 1] and resized it to 32 × 32 (224 × 224). Then, we applied horizontal flipping with a 50% probability and color distortion with a 50% probability for image augmentation. Finally, we converted the images to grayscale with a 30% possibility. We assigned each client 100 classes.

3) IMPLEMENTATION DETAILS

We evaluated the performance of our scheme through simulations on the CIFAR-100 and ImageNet datasets. The implementation of our algorithms is based on PyTorch-implemented federated learning and DP-SGD. We utilized a 32-layer ResNet as the backbone network for CIFAR-100 and an 18-layer ResNet for ImageNet. Throughout all experiments, the models were trained using an SGD optimizer with an initial learning rate of 0.1 and momentum of 0.8. The batch size was established at 128, with weight decay parameters set at 0.001 for CIFAR-100 and 0.0005 for ImageNet, respectively. The dual-model underwent training for 160 epochs on CIFAR-100 and 90 epochs on ImageNet, with learning rate reductions by a factor of 10 following the 80th and 120th epochs for CIFAR-100, and the 30th and 60th epochs for ImageNet. Furthermore, we used a noise parameter sigma of 0.5 and a privacy budget epsilon of 8 for differential privacy.

B. PERFORMANCE EVALUATION OF FCIL

1) PERFORMANCE COMPARISON

In this section, wIn detail, we compare our method with the following baselines in the FL scenario: iCaRL [31], BiC [69], PODNet [58], DyTox [68], GLFC [70], LGA [71], FedET [72]. It is worth noting that for comparison purposes, we did not apply differential privacy in this experiment. Therefore, we did not use our local Bayesian differential privacy strategy (**Ours1**).

As shown in Tables 2 - 7, we executed an extensive set of comparative experiments to assess the accuracy of our model against other methods on both the CIFAR-100 and ImageNet datasets. For these experiments, the number of consecutive learning tasks was designated as $\mathcal{T} = \{5, 10, 20\}$. The Avg (%) column denotes the average incremental accuracy across the various tasks, and the Imp (%) column indicates the improvement in Avg (%) of our method compared to other methods. It is worth stating that among these methods, LGA and FedET are the current state-of-the-art methods, but our scheme only slightly lags behind FedET on the CIFAR-100 dataset at $\mathcal{T} = \{5, 10\}$. However, our method outperformed all baseline methods when $\mathcal{T}$ increased to 20.

Based on the presented results in Tables 2-7, we have the following conclusions:

- Our dual-model facilitates collaborative training of a global class-incremental model by local clients and consistently outperforms other methods in incremental tasks. This underscores our model's efficacy in addressing the challenge of forgetting within the FCIL framework.
- The dual-branch structure demonstrates a more effective approach in identifying optimal solutions for all classes during training compared to the single-branch structure, underscoring the superiority of our method.
- Our dual-model achieved state-of-the-art performance against other comparison methods when tested on different types of consecutive learning tasks (i.e., $\mathcal{T} = \{5, 10, 20\}$). This demonstrates the resilience and effectiveness of our dual-Model in addressing cataclysmic forgetting under various experimental settings in FCIL.

2) ABLTION STUDIES

This subsection provides qualitative ablation studies to highlight the efficacy of each component within our dual-model across various experimental scenarios.

a: EFFECT OF SCALE FACTOR F

We tested and validated the scale factor F of clients participating in federated training to ensure the high efficiency and performance of our scheme. We set $\mathcal{T} = 10$ and the total number of clients $K = 50$. Additionally, we conducted comparison experiments (randomly selected) with the case where our preferred client strategy was not used (None-OP). The results presented here are all average incremental accuracy and do not use differential privacy (**Ours1**).

TABLE 4. Comparisons in terms of accuracy on CIFAR-100 dataset when the number of tasks $\mathcal{T} = 10$.

Methods	10	20	30	40	50	60	70	80	90	100	Avg (%)	Imp (%)
iCaRL+FL	88.5	60.6	55.3	53.5	50.8	49.0	46.5	42.8	41.6	38.2	52.68	21.76
BiC+FL	88.3	70.4	67.5	63.0	58.7	50.5	47.3	44.5	42.3	40.8	57.33	17.11
PODNet+FL	89.0	75.6	73.5	66.7	58.4	56.0	53.2	47.6	46.8	45.3	61.21	13.23
DyTox+FL	85.0	81.3	76.5	71.2	66.4	63.8	60.0	57.6	54.9	52.2	66.89	7.55
GLFC	88.8	81.2	78.2	71.4	66.2	62.0	58.5	53.7	51.6	49.8	66.14	8.30
LGA	88.5	80.4	76.9	73.7	71.7	69.7	65.4	64.0	61.6	59.5	71.14	3.30
FedET	92.2	87.5	82.2	79.3	77.1	74.7	67.4	68.6	66.4	65.0	76.04	-1.60
Ours1	**91.8**	**89.1**	**85.2**	**80.0**	**74.9**	**69.6**	**67.7**	**65.1**	**61.9**	**59.1**	**74.44**	**-1.60 $\sim$ 21.76**

TABLE 5. Comparisons in terms of accuracy on ImageNet dataset when the number of tasks $\mathcal{T} = 10$.

Methods	10	20	30	40	50	60	70	80	90	100	Avg (%)	Imp (%)
iCaRL+FL	73.2	65.6	58.4	49.2	44.7	43.1	40.5	37.8	35.6	33.2	48.13	20.48
BiC+FL	75.0	67.1	63.1	53.5	49.7	45.7	43.4	39.2	36.8	34.1	50.76	17.85
PODNet+FL	74.4	71.0	65.3	56.8	51.4	48.1	46.5	43.9	39.8	38.3	53.55	15.06
DyTox+FL	76.3	73.2	59.5	54.5	49.6	46.1	42.5	38.1	37.0	35.8	51.26	17.35
GLFC	73.8	69.0	69.8	60.8	57.5	54.4	53.2	49.5	47.2	44.3	57.95	10.66
LGA	81.5	72.7	70.8	70.7	66.6	64.3	62.5	58.1	56.9	56.0	66.01	2.60
FedET	83.8	81.2	74.0	73.3	68.8	68.6	59.9	58.2	54.8	52.9	67.55	1.06
Ours1	**85.7**	**83.3**	**79.4**	**74.2**	**69.1**	**63.8**	**61.9**	**59.3**	**56.1**	**53.3**	**68.61**	**1.06 $\sim$ 20.48**

TABLE 6. Comparisons in terms of accuracy on CIFAR-100 dataset when the number of tasks $\mathcal{T} = 20$.

Methods	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	Avg (%)	Imp (%)
iCaRL+FL	83.5	82.4	70.3	65.0	63.2	61.7	59.1	56.7	55.3	52.3	51.4	50.6	50.0	49.3	47.2	46.2	46.9	46.0	45.5	42.9	56.28	15.84
BiC+FL	82.8	78.3	70.0	67.5	66.6	64.2	62.8	60.5	57.3	54.5	53.2	52.1	51.8	50.2	49.4	48.3	47.2	47.0	46.9	44.5	57.76	14.36
PODNet+FL	84.0	77.2	75.9	72.4	69.5	68.1	67.7	63.4	61.3	59	58.5	57.3	55.8	53.8	52.7	51.8	49.6	48.6	48.0	47.1	61.09	11.03
DyTox+FL	81.5	80.3	75.5	73.6	71.4	68.1	65.0	65.4	63.1	61.3	59.5	58.2	56.8	57.3	56.6	54.0	53.0	51.6	52.0	51.1	62.77	9.35
GLFC	85.2	83.5	76.3	77.6	73.1	70.9	69.4	67.0	64.5	63.3	62.4	60.2	59.6	57.8	58.1	57.0	56.5	54.8	52.6	52.9	65.14	6.98
LGA	86.9	85.8	81.4	81.7	80.1	77.6	75.0	68.0	65.8	69.9	64.1	63.3	61.1	59.3	56.9	56.6	54.8	66.4	64.3	62.2	69.06	3.06
FedET	88.2	85.0	80.2	76.2	75.4	72.8	69.3	68.9	70.1	66.5	63.7	63.6	61.4	59.1	57.2	55.2	52.8	51.6	51.6	50.9	65.99	6.13
Ours1	**87.5**	**86.9**	**84**	**81.5**	**80.6**	**78.2**	**76.3**	**74.5**	**72.8**	**70.7**	**69.4**	**68.3**	**66.8**	**64.9**	**66.2**	**65.5**	**64.2**	**62.7**	**61.1**	**60.2**	**72.12**	**3.06 $\sim$ 15.84**

TABLE 7. Comparisons in terms of accuracy on ImageNet dataset when the number of tasks $\mathcal{T} = 20$.

Methods	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	Avg (%)	Imp (%)
iCaRL+FL	81.8	75.8	66.3	61.3	58.4	55.2	52.9	50.3	47.8	45.3	44.5	42.2	40.2	39.0	37.6	38.0	34.4	33.1	31.5	29.9	48.28	17.37
BiC+FL	82.6	72.3	65.4	63.4	62.7	60.2	55.8	54.0	52.5	48.2	47.1	44.9	43.2	42.0	41.2	40.6	36.6	35.4	34.2	33.6	50.78	14.87
PODNet+FL	83.0	72.3	67.2	65.7	64.7	63.3	58.9	57.4	55.9	53.6	51.1	48.6	46.8	45.2	44.4	43.2	39.3	38.8	37.9	36.5	53.69	11.96
DyTox+FL	73.5	68.7	62.3	58.4	60.2	57.4	56.5	55.2	52.7	50.0	47.7	45.2	43.1	41.2	38.6	36.2	34.5	32.7	30.8	28.7	48.68	16.97
GLFC	83.8	74.4	70.6	68.2	67.2	68.1	64.2	62.9	62.8	59.0	56.5	54.6	52.7	51.5	50.4	48.3	45.2	44.3	42.5	41.4	58.43	7.22
LGA	77.0	77.6	76.8	72.3	70.8	69.4	68.2	66.6	63.0	61.9	60.8	60.2	57.7	56.6	52.9	51.9	51.2	49.7	49.0	49.2	62.14	3.51
FedET	79.5	73.6	74.3	73.4	69.2	65.0	64.4	59.8	58.8	58.0	55.6	54.3	54.3	50.2	48.6	47.3	46.9	46.2	45.6	45.0	58.50	7.15
Ours1	**83.5**	**79.7**	**75.9**	**74.3**	**73.5**	**71.3**	**70.8**	**69.5**	**68.0**	**66.5**	**64.9**	**62.6**	**59.7**	**57.9**	**58.3**	**57.2**	**56.3**	**55.4**	**54.1**	**53.6**	**65.65**	**3.51 $\sim$ 17.37**

Table 1 shows the effect of different scale factor F values (i.e., the number of clients participating in federated training in each round) on the performance of the model in various aspects. As shown in the table, as the number of clients participating in federated training increases, the performance of the model also improves to a certain extent. To obtain

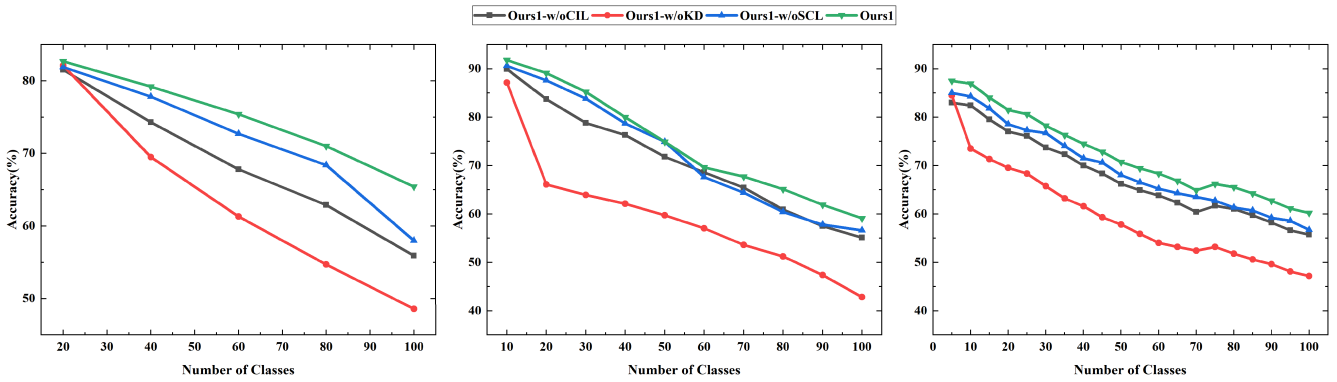

FIGURE 4. Ablation studies on CIFAR-100 when $\mathcal{T} = 5$ (left), $\mathcal{T} = 10$ (middle) and $\mathcal{T} = 20$ (right).

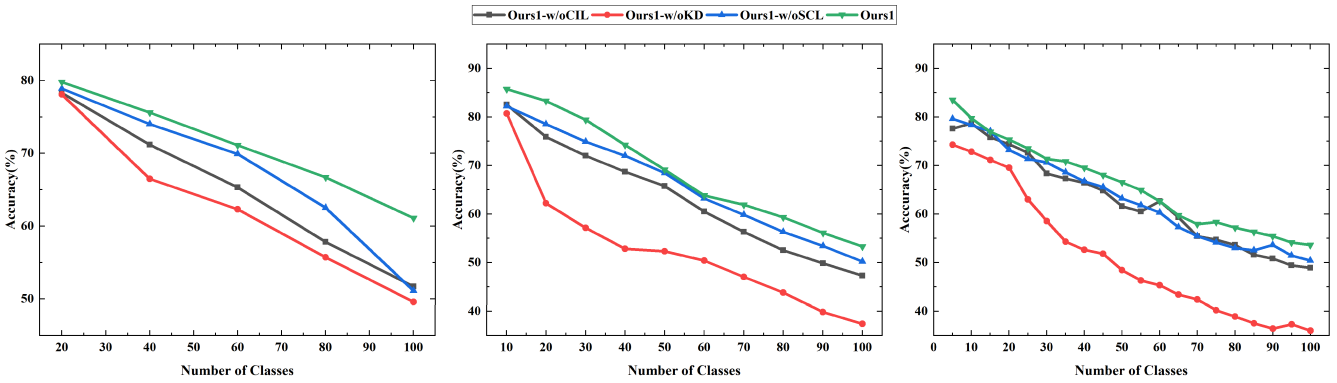

FIGURE 5. Ablation studies on ImageNet when $\mathcal{T} = 5$ (left), $\mathcal{T} = 10$ (middle) and $\mathcal{T} = 20$ (right).

assurance of the experiment's fairness and verify the validity of our scheme, we set $F = 0.5$ and $K = 50$ for most experiments.

b: EFFECTS OF DIFFERENT COMPONENTS

As shown in Fig. 4 and Fig. 5. Ours1-w/oCIL, Ours1-w/oKD, Ours1-w/oSCL represent the performance of training proposed dual-model without utilizing the classification loss $\mathcal{L}_{\mathcal{C}}$, distillation loss $\mathcal{L}_{\mathcal{D}}$ and SupCon loss $\mathcal{L}_{\mathcal{S}}$.

Compared with Ours1, Ours1-w/oCIL exhibits an average accuracy decrease ranging from 3.05% to 6.26%. This result underscores the significant improvement in accuracy brought about by the Balanced Softmax across various experimental settings. Moreover, it indicates that the Balanced Softmax can effectively address the class imbalance challenge inherent in CIL. In addition, Ours1 surpasses Ours1-w/oKD by a substantial margin, showing an increase of 8.42% to 16.26% in average accuracy on benchmark datasets. This result corroborates that $\mathcal{L}_{\mathcal{D}}$ is adept at addressing the distillation losses arising from the retention of old knowledge. Furthermore, the performance of Ours1-w/oSCL witnesses a decrease in average accuracy by 2.2% to 3.58%. It confirms that $\mathcal{L}_{\mathcal{S}}$ is effective in alleviating the losses incurred due to the acquisition of new knowledge in CIL.

Additionally, we conducted ablation experiments focusing on the dual-branch structure using the CIFAR-100 dataset. Specifically, we utilized a single-branch structure (BiC) as the baseline model and refined it employing our loss function. As shown in Table 8, the results reveal that the Balanced Softmax can enhance accuracy across various experimental settings. This provides evidence for its efficacy in addressing the class imbalance challenge in FCIL. Notably, the most significant factor impacting the performance of both models is the distillation loss, underscoring its pivotal role in determining model efficacy in FCIL. We deduced that while the single-branch approach makes the new data classes distinguishable after incorporating the SupCon loss, it inadvertently neglects the retention of previously acquired knowledge. It can be seen that under the same experimental conditions, the accuracy of the two-branch structure significantly surpasses that of the single-branch structure, suggesting that the strategy of reutilizing old knowledge in the two-branch structure is viable.

c: SENSITIVE STUDY OF HYPER-PARAMETERS

In our approach, the parameter ϱ modulates the weight of the SupCon loss within the loss function, while Γ is utilized to adjust the weight of the feature distillation loss during the model compression phase. We performed a sensitivity

TABLE 8. Comparison of the effects of each component.

Methods	Ours1-w/oCIL	Ours1-w/oKD	Ours1-w/oSCL	Normal
Base (BiC)	62.24	56.32	67.61	66.38
Ours1	70.80	59.09	72.24	74.44

TABLE 9. Sensitive study of hyper-parameters in aggregation ($\mathcal{T} = 10$, $K = 50$).

Ours1	[a, b, c, d]	Avg(%)
①	[0.7, 0.1, 0.1, 0.1]	70.59
②	[0.5, 0.2, 0.2, 0.1]	71.86
③	[0.3, 0.3, 0.2, 0.2]	72.19
④	[0.3, 0.5, 0.1, 0.1]	74.44
⑤	[0.3, 0.1, 0.3, 0.3]	68.43
⑥	[0.1, 0.5, 0.2, 0.2]	59.88
⑦	[0.1, 0.1, 0.4, 0.4]	48.58

analysis to explore the implications of different values of ϱ and Γ on both CIFAR-100 and ImageNet datasets when $\mathcal{T} = 10$.

The effect of the hyper-para ϱ is shown in Fig. 6. We observe that the model exhibits the lowest accuracy when $\varrho = 0$, signalling that SupCon loss significantly enhances the performance of our model. This enhancement can be attributed to the model's proficiency in learning expressive representations by employing SupCon loss at Ω_{new}. Simultaneously, it reutilizes prior representations at Ω_{old} and leverages Ω_{dAff} to derive a hybrid representation, encapsulating both forms of information.

The effect of the hyper-para Γ is shown in Fig. 7. It is evident that when Γ is not 0, the model's performance remains relatively stable. Notably, the model achieves its peak performance when Γ is 15. Furthermore, the results demonstrate that our model exhibits robustness to variations in ϱ and Γ.

In Equation (36), the aggregation formula is controlled by four hyper-parameters $[a, b, c, d]$. To investigate the impact of different weight values on model performance, we selected and compared seven sets of weight values. As shown in Table 9, the values of a and b greatly influence the model's performance. Optimal results are achieved when the hyper-parameters $[a, b, c, d]$ are set to $[0.3, 0.5, 0.1, 0.1]$.

C. PRIVACY STUDIES

In the previous experiments, we did not use our local Bayesian differential privacy strategy. Therefore, in this part of the experiments, we will experiment with local Bayesian differential privacy and analyse the security and privacy of our scheme.

1) EFFECT OF DIFFERENTIAL PRIVACY ON ACCURACY

In this experiment, we evaluate the performance of our scheme in privacy preservation by fixing $\mathcal{T}$ (the number

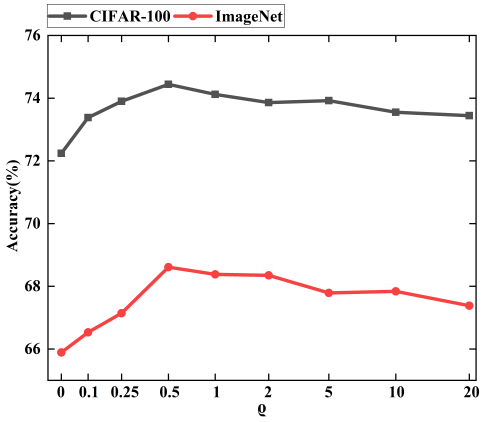

FIGURE 6. The effect of the hyper-parameter ϱ.

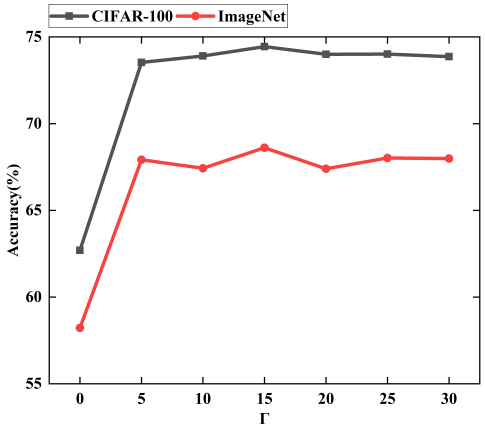

FIGURE 7. The effect of the hyper-parameter Γ.

of consecutive learning tasks) and K (the total number of clients). The benchmark experiments include "Ours1" and "DP". "Ours1" means no privacy-preserving methods were considered when training neural networks. This method provides maximum performance for the privacy-preserving model. "DP" refers to a content-level privacy-preserving method that uses classic differential privacy.

Moreover, to thoroughly investigate the applicability of LBDP in FCIL scenarios, we also compare it with the more advanced differential privacy schemes proposed in NbAFL [50] and Fed-SPA [51]. We use "LBDP" to represent the local Bayesian differential privacy mechanism used in our scheme, which is set to local differential privacy so that each client accumulates its privacy loss. We set the same parameters for all methods: a noise parameter sigma of 0.5 and a privacy budget epsilon of 8. We conducted experiments on the CIFAR-100 dataset, and the results are all average incremental accuracy.

As shown in Fig. 8, we fix $\mathcal{T}$ and compare the impact of each differential privacy method on the model performance for different numbers of clients. It can be seen that in the FCIL scenario, the effect of LBDP on the model performance is not much different from that of NbAFL and Fed-APS, which are

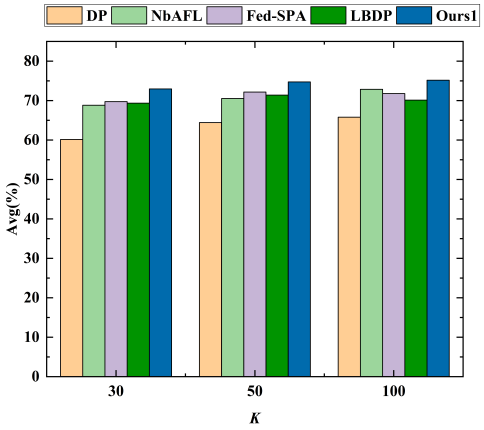

FIGURE 8. Effect of differential privacy on accuracy on CIFAR-100 ($\mathcal{T} = 5$).

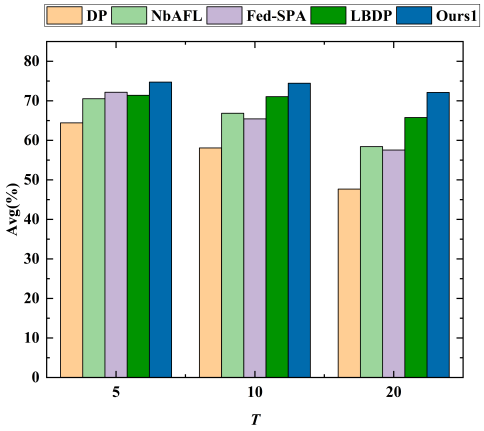

FIGURE 9. Effect of differential privacy on accuracy on CIFAR-100 ($K = 50$).

all very close to the baseline method Non-DP and only lag behind NbAFL and Fed-APS at $K = 100$.

Fig. 9 demonstrates the experimental results at various values of T when $K = 50$. It is easy to see that LBDP leads all baseline methods across the board when $\mathcal{T}$ increases to 20. This indicates that in more complex FCIL scenarios and with the same privacy budget, using LBDP can achieve better performance with less sacrifice.

2) PRIVACY ANALYSIS

We employ the local Bayesian differential privacy mechanism to address privacy issues at the content level. Within the differential privacy framework, the infusion of suitably calibrated noise into the dataset impedes algorithms from querying the dataset to extract precise information therein. As a result, an attacker cannot get the customer's personal data records or content directly from the compromised gradient. Local Bayesian differential privacy in our scheme provides a (ε, δ)-differential privacy guarantee, and its related proofs can be found in the literature [73]. Furthermore, in this paper, we made two improvements:

TABLE 10. Efficiency studies on CIFAR-100 ($\mathcal{T} = 10$, $K = 50$).

Methods	Training time	Compression time	Avg
Ours	53min	20min	71.06%
Ours-w/oMdwa	62min (↑ 16.98%)	23min (↑ 15.00%)	69.48%
Ours-w/oMc	68min (↑ 28.30%)	0	73.08%
Ours-w/oSingle	40min (↓ 24.53%)	0	58.32%

- Local Bayesian differential privacy (LBDP) adds noise based on data distribution. It introduces varying noise levels for datasets with different distributions, making the scale of noise addition more rational and thereby providing enhanced privacy protection.
- We adjusted Bayesian differential privacy into a local differential privacy model. We designed a new quantification method of privacy loss that allows each client to calculate its privacy loss locally in the iterative mechanism. Experiments demonstrate that under the same privacy budget, LBDP achieves superior model performance.

In addition, the old knowledge is compressed after we train the dual-branch model, which also protects the privacy of our clients to a certain extent. Besides, our experimental results show that LBDP can provide better privacy protection for FCIL with the same privacy budget. Thus, our scheme can provide at least content-level privacy protection.

D. EFFICIENCY STUDIES

1) EFFECT OF DIFFERENT COMPONENTS ON EFFICIENCY

We use multi-factor dynamic weighted aggregation strategy to improve the speed of model aggregation. Model compression also reduces the pressure on the local storage of old knowledge. To fully evaluate the complexity and performance, we have analyzed multiple dimensions, including training time, compression time and average incremental accuracy, as shown in Table 10, Ours-w/oMdwa, Ours-w/oMc represent the performance of training proposed dual-model without utilizing multi-factor dynamic weighted aggregation strategy and Model compression. In addition, we have set up a single-branch model (Ours-w/oSingle) for comparison. We conduct experiments on the CIFAR-100 dataset, setting $\mathcal{T} = 10$.

Compared to Ours-w/oMdwa, the training time and compression time were reduced by 9 minutes and 3 minutes, respectively, and Avg increased by 1.58%, indicating that our multi-factor dynamic weighted aggregation strategy effectively accelerates model aggregation.

In addition, the multi-factor dynamic weighted aggregation strategy also has some gains in terms of model compression time and Avg. Furthermore, comparing Ours-w/oMc, we achieved a 15 minute improvement in training time at the expense of 20 minutes of compression time, but only a 2.02% reduction in Avg. It is worth mentioning that by compressing

the dual-branch structure to a single-branch structure, local clients need only half the memory to save old knowledge. Hence, the sacrifice is worthwhile for clients with limited local resources.

Similarly, comparing Ours-w/oSingle, we sacrifice 13 minutes of training time and 20 minutes of compression time, respectively. Still, we can perform model compression at any time before the next incremental task, increasing the degree of training freedom. Meanwhile, compared to the 12.74% improvement in model performance, the additional computational cost is tolerable.

2) COMPUTATIONAL COMPLEXITY ANALYSIS

In FCIL, clients often have limited computing and storage resources. Therefore, for the proposed method to be of practical use, we have to make a good trade-off between improving the performance of the global model and increasing the complexity of the local model.

In this paper, we introduce a dual-model structure to address the catastrophic forgetting problem in FCIL. Given that the server's computational resources are typically abundant, we assume that executing Algorithm 2 on the server does not encounter computational bottlenecks. For the client, the Algorithm 3 needs to be executed continuously during the training process. It is evident that the primary computational pressure for clients stems from paradigm sample selection, model fusion, and model compression.

We use the Herding algorithm for paradigm samples selection. For each data point x, we utilize the feature extractor Ω_s to obtain its features. We assume the complexity of feature extraction to be $O(F_s)$. Therefore, for $|D_t^k|$, we easily conclude that the computational complexity is $O(F_s \times p^2 \times |D_t^k|)$, which implies that its computational complexity is mainly affected by the number of paradigm samples p to be selected. We believe that in our approach, this is a key step in constructing the dual-model, which can significantly alleviate the problem of forgetting old knowledge in FCIL. Therefore, it is perfectly acceptable compared to the improvement in model performance.

As for model fusion, analyzing Eq. (23) and Eq. (24), we can know that the key is that we use the channel attention mechanism, which has two fully-connected layers. Therefore, we can get its computational complexity as $O(C^2/r + C) \rightarrow O(C^2)$, where C is the number of channels. It can be learned from [78] that the computation time of Eq. (24) is usually at the millisecond level, so model fusion consumes very little of the client's computational resources.

According to Eq. (25), we know that the computational complexity of model compression is $(O(C \times H \times W))$, where H and W are the size of the matrix. Obviously, the number of data and the size of the model will have a direct impact on the time for model compression. However, clients can save half of the storage space, and we can perform model compression at any time before the next incremental task comes, thus increasing the freedom of training. Therefore, the computational cost due to model compression is also acceptable.

VII. CONCLUSION

In this paper, we focused on addressing a real-world FL challenge named Federated Class-Incremental Learning (FCIL), and we proposed a Privacy-Preserving Federated Class-Incremental Learning (PP-FCIL) approach, which is a pioneering exploration to tackle the real-world FCIL problem under privacy preservation. Compared with existing methods, we used both old and new knowledge to train new local models, ensuring that the models have better accuracy while alleviating the problem of catastrophic forgetting. At the same time, to ensure the privacy requirements of clients, we used local Bayesian differential privacy to adjust the privacy budget allocation for different datasets before the data left the client. This adjusted the degree of noise addition to improve the quality of service of the model while providing more fine-grained privacy protection. For global model aggregation, we proposed a multi-factor dynamic weighted aggregation strategy to improve the aggregation speed of global models and the performance of federated learning. The experimental results show that our method is effective and superior to the current state-of-the-art methods.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, vol. 54, 2017, pp. 1273–1282.

[2] U. Mohammad and S. Sorour, "Adaptive task allocation for asynchronous federated mobile edge learning," 2019, *arXiv:1905.01656*.

[3] J. S.-P. Díaz and Á. L. García, "Study of the performance and scalability of federated learning for medical imaging with intermittent clients," *Neurocomputing*, vol. 518, pp. 142–154, Jan. 2023.

[4] Y. Shen, A. Sowmya, Y. Luo, X. Liang, D. Shen, and J. Ke, "A federated learning system for histopathology image analysis with an orchestral stain-normalization GAN," *IEEE Trans. Med. Imag.*, vol. 42, no. 7, pp. 1969–1981, Jul. 2023.

[5] X. Zhou et al., "Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 82–89, Apr. 2023.

[6] C.-I. Huang et al., "Fed-HANet: Federated visual grasping learning for human robot handovers," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3772–3779, Jun. 2023.

[7] C. Wang, X. Chen, J. Wang, and H. Wang, "ATPFL: Automatic trajectory prediction model design under federated learning framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6553–6562.

[8] R. Du, K. Han, R. Gupta, S. Chen, S. Labi, and Z. Wang, "Driver monitoring-based lane-change prediction: A personalized federated learning framework," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–7.

[9] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Proc. 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 5451–5452.

[10] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. ACM NIPS*, 2015, pp. 2737–2745.

[11] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. ACM NIPS*, Long Beach, CA, USA, 2017, pp. 5330–5340.

[12] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, 2020, pp. 429–450.

[13] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[14] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Denver, CO, USA. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1310–1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[15] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 2512–2520.

[16] C. Ma et al., "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul. 2020.

[17] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[18] Y. Zhang, D. Zeng, J. Luo, Z. Xu, and I. King, "A survey of trustworthy federated learning with perspectives on security, robustness and privacy," in *Proc. Companion ACM Web Conf. (WWW)*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1167–1176.

[19] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 36–52.

[20] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 619–633.

[21] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[22] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proc. 24th ACM SIGACT-SIGMOD-SIGART Symp. Princ. Database Syst.*, C. Li, Ed. Baltimore, MD, USA, 2005, pp. 128–138.

[23] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Scottsdale, AZ, USA, Nov. 2014, pp. 1054–1067.

[24] N. Wang et al., "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Macao, China, Apr. 2019, pp. 638–649.

[25] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, 2016, pp. 308–318.

[26] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2020, pp. 1–18.

[27] A. Cheng, P. Wang, X. S. Zhang, and J. Cheng, "Differentially private federated learning with local regularization and sparsification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10112–10121.

[28] J. Lu, X. S. Zhang, T. Zhao, X. He, and J. Cheng, "APRIL: Finding the Achilles' heel on privacy for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10041–10050.

[29] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proc. 35th AAAI Conf. Artif.*, 2021, pp. 10165–10173.

[30] J. Hong, Z. Zhu, S. Yu, Z. Wang, H. H. Dodge, and J. Zhou, "Federated adversarial debiasing for fair and transferable representations," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 617–627.

[31] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[32] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. NIPS*, 2017, pp. 2990–2999.

[33] M. Yurochkin, M. Agarwal, S. Ghosh, K. H. Greenewald, T. N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. ICML*, in Proceedings of Machine Learning Research, vol. 97, 2019, pp. 7252–7261.

[34] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on Non-IID features via local batch normalization," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–27.

[35] H. Wang, M. Yurochkin, Y. Sun, D. S. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. ICLR*, 2020, pp. 1–16.

[36] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," in *Proc. ICLR*, 2021, pp. 1–2.

[37] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016, *arXiv:1602.05629*.

[38] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*.

[39] N. Shoham et al., "Overcoming forgetting in federated learning on Non-IID data," 2019, *arXiv:1910.07796*.

[40] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[41] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–19.

[42] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 1–17, Aug. 2021.

[43] J. Dong, Y. Cong, G. Sun, B. Zhong, and X. Xu, "What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4022–4031.

[44] L. Qu et al., "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10051–10061.

[45] Y. Ding et al., "Federated submodel optimization for hot and cold data features," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Red Hook, NY, USA: Curran Associates, 2022, pp. 1–13.

[46] P. Liu, X. Yu, and J. T. Zhou, "Meta knowledge condensation for federated learning," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, May 2023, pp. 1–16.

[47] H. Yang et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nature Commun.*, vol. 13, no. 1, p. 4269, Jul. 2022.

[48] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

[49] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," 2019, *arXiv:1905.02383*.

[50] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[51] R. Hu, Y. Gong, and Y. Guo, "Federated learning with sparsification-amplified privacy and adaptive optimization," in *Proc. 30th Int. Joint Conf. Artif. Intell. (IJCAI)*, Z.-H. Zhou, Ed., Aug. 2021, pp. 1463–1469.

[52] B. Wang, Y. Chen, H. Jiang, and Z. Zhao, "PPeFL: Privacy-preserving edge federated learning with local differential privacy," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15488–15500, Apr. 2023.

[53] A. Triastcyn and B. Faltings, "Federated learning with Bayesian differential privacy," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2587–2596.

[54] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, "SS-IL: Separated softmax for incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 824–833.

[55] J. Kim and D. Choi, "Split-and-bridge: Adaptable class incremental learning within a single neural network," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 8137–8145.

[56] T.-Y. Wu et al., "Class-incremental learning with strong pre-trained models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9591–9600.

[57] M. De Lange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[58] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 12365. Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 86–102.

[59] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," 2016, *arXiv:1612.00796*.
[60] Z. Li and D. Hoiem, "Learning without forgetting," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 614–629.
[61] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3420–3429.
[62] C. Wu, L. Herranz, X. Liu, Y. Wang, J. van de Weijer, and B. Raducanu, "Memory replay GANs: Learning to generate new categories without forgetting," in *Proc. NIPS*, Montréal, QC, Canada, Dec. 2018, 2018, pp. 5966–5976.
[63] X. Hu, K. Tang, C. Miao, X.-S. Hua, and H. Zhang, "Distilling causal effect of data in class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3956–3965.
[64] C. Simon, P. Koniusz, and M. Harandi, "On learning the geodesic path for incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1591–1600.
[65] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16050–16059.
[66] Y. Liu, B. Schiele, and Q. Sun, "Adaptive aggregation networks for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2544–2553.
[67] Y. Shi et al., "Mimicking the oracle: An initial phase decorrelation approach for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16701–16710.
[68] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "DyTox: Transformers for continual learning with dynamic token expansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9275–9285.
[69] Y. Wu et al., "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 374–382.
[70] J. Dong et al., "Federated class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10154–10163.
[71] J. Dong, Y. Cong, G. Sun, Y. Zhang, B. Schiele, and D. Dai, "No one left behind: Real-world federated class-incremental learning," 2023, *arXiv:2302.00903*.
[72] C. Liu, X. Qu, J. Wang, and J. Xiao, "FedET: A communication-efficient federated class-incremental learning framework based on enhanced transformer," in *Proc. 32nd Int. Joint Conf. Artif. Intell. (IJCAI)*, E. Elkind, Ed., Aug. 2023, pp. 3984–3992.
[73] A. Triastcyn and B. Faltings, "Bayesian differential privacy for machine learning," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, in Proceedings of Machine Learning Research, vol. 119, 2020, pp. 9583–9592.
[74] M. Welling, "Herding dynamical weights to learn," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 1121–1128.
[75] Z. Fu, Z. Wang, X. Xu, D. Li, and H. Yang, "Knowledge aggregation networks for class incremental learning," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109310.
[76] J. Ren et al., "Balanced meta-softmax for long-tailed visual recognition," in *Proc. NIPS*, 2020, pp. 4175–4186.
[77] P. Khosla et al., "Supervised contrastive learning," in *Proc. NIPS*, 2020, pp. 18661–18673.
[78] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.
[79] Y. J. Kim and C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2019, pp. 1–4.
[80] I. Omara, A. Hagag, G. Ma, F. E. A. El-Samie, and E. Song, "A novel approach for ear recognition: Learning Mahalanobis distance features from deep CNNs," *Mach. Vis. Appl.*, vol. 32, no. 1, p. 38, Jan. 2021.

JUE XIAO received the B.S. degree in automotive engineering from the Wuhan University of Technology, Wuhai, China, in 2020. He is currently pursuing the Ph.D. degree in cyberspace security with the Huazhong University of Science and Technology, Wuhan. His research interests include federated learning and artificial intelligence.

XUEMING TANG received the B.S. and M.S. degrees from the Department of Mathematics, Wuhan University, Wuhan, China, in 1995 and 1998, respectively, and the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), Wuhan, in 2006. He is currently an Associate Professor with the Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, HUST. His current research interests include cryptography, big data analytics, machine learning, and information networks.

SONGFENG LU received the Ph.D. degree in computer software and theory from the Huazhong University of Science and Technology. He was a Visiting Scholar with the University of York, U.K. He is currently a Professor with the Shenzhen Huazhong University of Science and Technology Research Institute and the Ph.D. Supervisor. His research interests include cybersecurity, artificial intelligence, intelligent manufacturing, industrial internet, and quantum computing.