CGoFed: Constrained Gradient Optimization Strategy for Federated Class Incremental Learning

Jiyuan Feng, Xu Yang, Liwen Liang, Weihong Han, Binxing Fang, and Qing Liao, *Member, IEEE*

***Abstract*—Federated Class Incremental Learning (FCIL) has emerged as a new paradigm due to its applicability in real-world scenarios. In FCIL, clients continuously generate new data with unseen class labels and do not share local data due to privacy restrictions, and each client's class distribution evolves dynamically and independently. However, existing work still faces two significant challenges. Firstly, current methods lack a better balance between maintaining sound anti-forgetting effects over old data (stability) and ensuring good adaptability for new tasks (plasticity). Secondly, some FCIL methods overlook that the incremental data will also have a non-identical label distribution, leading to poor performance. This paper proposes CGoFed, which includes relax-constrained gradient update and cross-task gradient regularization modules. The relax-constrained gradient update prevents forgetting the knowledge about old data while quickly adapting to the new data by constraining the gradient update direction to a gradient space that minimizes interference with historical tasks. The cross-task gradient regularization also finds applicable historical models from other clients and trains a personalized global model to address the non-identical label distribution problem. The results demonstrate that the CGoFed performs well in alleviating catastrophic forgetting and improves model performance by 8% -23% compared with the SOTA comparison method.**

***Index Terms*—Federated learning, class incremental learning, catastrophic forgetting.**

I. Introduction

FEDERATED learning (FL) has emerged as a well-known distribution learning paradigm that aims to collaboratively train global models across multiple clients without sharing client data. FL provides a privacy-preserving solution to collaborative training across clients and is widely applied in various domains [1], including finance [2], [3], the Internet of Things (IoT) [4], [5], [6], and healthcare [7], [8], [9], as it only transfers model parameters without raw data [10], [11].

Federated learning (FL) commonly assumes that clients maintain a static label distribution of local data during the training and testing phases of the model [2], [12], [13]. However, in the real world, the label distributions of clients often change dynamically over time [14], [15], and each local client continuously introduces new classes to update the knowledge of the global model [14]. This scenario, known as federated class incremental learning (FCIL), also referred to as federated continuous learning or lifelong federated learning [16], reflects a more natural and realistic setting. Given that clients in federated learning are typically lightweight mobile devices with limited storage capacity, constantly storing new data can lead to a significant memory crisis. If clients only store and train the model on the new class, this can result in catastrophic forgetting of the old class, as the model may struggle to distinguish between the old class and the new data. Therefore, addressing catastrophic forgetting in FCIL requires extensive attention and exploration.

The key to addressing the catastrophic forgetting problem in federated class incremental learning (FCIL) lies in training a model that achieves a suitable balance between stability and plasticity. In FCIL, regularization techniques and memory-based replay techniques are commonly employed. However, current FCIL methods, particularly those based on regularization, often fall short of delivering satisfactory results, especially as the number of tasks increases. For instance, CFeD [17] is a regularization method that utilizes distillation learning to mitigate the forgetting problem. It achieves this by transferring knowledge from a previously converged model to the current model through a surrogate dataset. However, distillation-based methods like CFeD heavily rely on pseudo-labels assigned to the surrogate dataset. As the number of tasks continues to increase, the generated pseudo-labels may become increasingly biased, resulting in the model's diminished ability to learn new tasks and forget the knowledge of previous tasks.

In addition to addressing catastrophic forgetting, FCIL methods also face the problem of non-identical label distribution in incremental data across clients. Specifically, the incremental data contributed by each client at each task may belong to entirely different class distributions, resulting in suboptimal performance. For example, Dong et al. [16] propose GLFC, a

Received 24 June 2024; revised 7 January 2025; accepted 14 February 2025. Date of publication 24 February 2025; date of current version 25 March 2025. This work was supported by the Guangdong Major Project of Basic and Applied Basic Research under Grant 2019B030302002, in part by the the Major Key Project of PCL under Grant PCL2024A05, in part by the Shenzhen Science and Technology Program under grant ZDSYS20210623091809029, and in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant under Grant 2022B1212010005. Recommended for acceptance by Y. Tong. *(Corresponding author: Qing Liao.)*

Jiyuan Feng is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China, and also with Pengcheng Laboratory, Shenzhen 518055, China (e-mail: fengjy@stu.hit.edu.cn).

Xu Yang, Liwen Liang, and Binxing Fang are with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: xuyang97@stu.hit.edu.cn; LiwenLiang151101@outlook.com; fangbx@cae.cn).

Weihong Han is with the Department of New Networks, Pengcheng Laboratory, Shenzhen 518055, China (e-mail: hanwh@pcl.ac.cn).

Qing Liao is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China, and also with the Department of New Networks, Pengcheng Laboratory, Shenzhen 518055, China (e-mail: liaoqing@hit.edu.cn).

Digital Object Identifier 10.1109/TKDE.2025.3544605

rehearsal-based method that leverages a proxy server (memory buffer) to select the optimal model from previous tasks, thereby retaining prior knowledge and achieving high performance. However, GLFC assumes that the classes of new tasks for all clients are drawn from the same distribution, which is atypical in real-world scenarios. In a more general setting, the classes of new tasks for different clients originate from diverse distributions. This class incremental scenario presents a more demanding challenge for aggregating the global model, as it experiences significant drift due to the non-identical label distributions of new tasks across clients.

We summarize the above-mentioned issues as the trade-off problem between stability and plasticity and the problem of non-identical label distribution for incremental data in Federated Class Incremental Learning (FCIL). Our paper proposes a **C**onstrained-based **G**radient **o**ptimization strategy for **Fed**erated Learning, called **CGoFed**, to address these two problems. In CGoFed, we introduce a relaxed constrained gradient update module to address catastrophic forgetting in FCIL, balancing stability and plasticity. This module restricts the direction of gradient updates based on the gradient space of the previous task, ensuring that the updated model parameters do not interfere with the historical tasks. Additionally, we propose a cross-task gradient regularization module to handle non-identical label distributions in incremental data, using a historical task selection mechanism to identify the most relevant tasks for the current task. We also design two Class Incremental Settings to evaluate existing FCIL methods in general federated scenarios. CGoFed achieves promising results, balancing stability (anti-forgetting) and plasticity (task adaptation) across diverse datasets. The contributions of our work can be summarized as follows:

- We summarize two significant challenges in federated class incremental learning, including the trade-off problem between stability and plasticity and the problem of non-identical label distribution in incremental data. We propose the CGoFed to solve the above problems.
- We develop the relax-constrained gradient update module in CGoFed to ensure that the updated model parameters for new tasks do not interfere with historical tasks while effectively balancing the stability of old tasks and the plasticity of new tasks.
- We design a cross-task gradient regularization module of CGoFed to address the problem of non-identical label distribution in incremental data. This module leverages the historical models of other clients that are most relevant to each task, thereby enhancing the overall performance and adaptability of the system.
- We compare CGoFed with other FCIL methods under two different class incremental settings. Experimental results indicate that CGoFed is highly competitive in alleviating forgetting and enhancing the performance of the global model compared to existing methods.

II. Related Work

We categorize existing federated class-incremental learning methods into expansion-based, regularization-based, and memory-based methods [18].

A. Expansion-Based Methods

These methods often improve robustness against catastrophic forgetting by decoupling or reconstructing the model structure to expand the model capacity.

For instance, Yoon et al. [19] propose FedWeIT, which splits parameters into global, local base, and task-adaptive parameters. FedWeIT achieves knowledge transfer and addresses the catastrophic forgetting problem by sharing task-specific parameters between clients. Hendryx et al. [20] utilize prototypical networks to implement fast knowledge transfer without gradient-based learning, mitigating catastrophic forgetting when clients individually learn a growing set of classes. In FCIL, a fixed-capacity model is more practical because clients have limited resources.

B. Regularization-Based Methods

Regularization-based methods penalize significant parameter changes associated with previous tasks by incorporating a regularization term into the classification loss function. This approach effectively prevents catastrophic forgetting during the training of new tasks [21]. For instance, Kirkpatrick et al. [22] introduce Elastic Weight Consolidation (EWC), which constrains parameters near the old task parameters by incorporating a quadratic penalty term when learning a new task. Shoham et al. [23] propose FedCurv, which employs EWC to penalize parameters that significantly change during the new task and are critical in the old task using a diagonal Fisher information matrix. However, this method disregards or alters the change trajectory of the model in the parameter space, negatively impacting the model's performance on new tasks.

Other regularization-based methods employ knowledge distillation techniques that rely on surrogate datasets to transfer knowledge from old tasks. For example, Li et al. [24] propose Learning without Forgetting and its extended version [24], which design a knowledge distillation loss to retain knowledge from previous tasks while optimizing accuracy on new tasks. Xu et al. [25] propose FedReg, which mitigates forgetting by regularizing local model parameters using generated pseudo data that encode knowledge of past tasks. Lee et al. [26] design a distillation loss using the global model's prediction on locally available data to acquire new knowledge and prevent catastrophic forgetting. Ma et al. [17] propose CFeD, in which each client transfers the knowledge of old tasks to the new model through a surrogate dataset, and the server distills the knowledge of the previous round model into a new aggregation model to address catastrophic forgetting.

Knowledge distillation typically constructs the same sample-label mapping relationship, enabling the trained model for the old task to transfer knowledge to the new task model [27], [28]. However, as the number of tasks or the task offset increases, the mapping relationship for previous tasks can degrade or become incorrect. This degradation causes performance decline in prior tasks and increases noise in the learning process of new tasks [29], leading to a poor balance between plasticity and stability.

C. Memory-Based Methods

These methods typically retain a subset of data from previous tasks to mitigate forgetting by preserving knowledge from

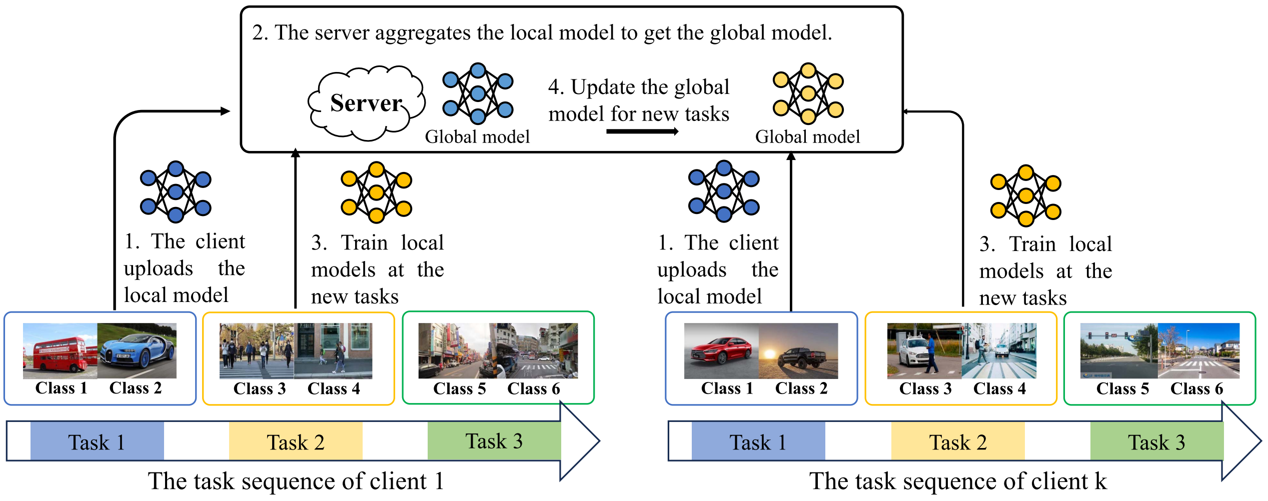

2. The server aggregates the local model to get the global model.
Server
Global model
4. Update the global model for new tasks
Global model
1. The client uploads the local model
3. Train local models at the new tasks
1. The client uploads the local model
3. Train local models at the new tasks
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Task 1
Task 2
Task 3
The task sequence of client 1
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Task 1
Task 2
Task 3
The task sequence of client k

Fig. 1. Illustration of FCIL. In federated class incremental learning, the client first trains the local model on task 1 and uploads it to the server. The server aggregates the local model to get the global model. As the class of data increases, the client trains the local model on task 2 and uploads it to the server. Finally, the server aggregates the global model suitable for the new task. It's noted that all clients train the local model independently on their task stream.

previous tasks in the form of data, gradients, parameters, models, and other representations.

For example, inspired by iCaRL [30], Zhang et al. [31] propose generating synthetic data from a generative model to capture the global data distribution and transfer knowledge of old tasks to current tasks, preventing catastrophic forgetting. Bakman et al. [32] improve the centralized continual learning method GPM to address global catastrophic forgetting in federated class-incremental learning, ensuring that updates for new tasks are orthogonal to the activation principal subspace of previous tasks. Dong et al. [16] propose GLFC, which designs a class-aware gradient compensation loss to solve the local forgetting problem caused by the continuous emergence of new classes in clients. GLFC constructs an exemplar memory mechanism using a proxy server to select a subset of previous tasks to retain prior knowledge. LGA [33], an extension of GLFC, relies on the server to reconstruct perturbed images and select the best old model.

These methods achieve an acceptable effect in preventing catastrophic forgetting. However, they still need to address the more challenging incremental scenario where new task data for all clients come from different label distributions, a problem we refer to as non-identical label distribution for incremental data.

III. Discussion on FCIL

We formally define federated class incremental learning and summarize two incremental-data settings in the client.

A. Federated Class Incremental Learning

We first introduce the concept of FCIL in terms of client, task, server, and optimization objective, as shown in Fig. 1.

Clients: We consider K clients, each indexed by k. Client k can only access a tasks sequence and is prohibited from accessing previous tasks while training the current task t. For task t, client k trains a local model Θ_k^t using the same convolutional neural network model with Γ layers.

Tasks: The sequence of tasks is represented as $\{D_k^1, D_k^2, \ldots, D_k^T\}$, where $D_k^t = \{(\mathbf{x}_{k,i}^t, \mathbf{y}_{k,i}^t) \mid 0 < i \leq n_k^t\}$ denotes the private dataset of the t-th task for the client k. This dataset comprises n_k^t instances, each consisting of features $\mathbf{x}_k^t$ and their corresponding labels $\mathbf{y}_k^t$.

Server: The server aggregates the local models of all the clients to obtain a personalized global model $\Theta_k^{t,g}$ exclusively for the client k. The model parameters are denoted as $\Theta = \{(\theta^l)_{l=1}^{\Gamma}\}$, where θ^l represents the parameters of the l-th layer.

Optimization Objective: Finally, FCIL expects to minimize the loss of all tasks among all samples. The optimization objective of FCIL can be written as follows:

$$\min_{\Theta} \sum_{t=1}^{T} \sum_{k=1}^{K} \frac{n_k^t}{N^t} \mathcal{L}\left(D_k^t; \Theta\right) \tag{1}$$

where n_k^t is the number of training samples of client k, N^t is the sum of all n_k^t at task t.

B. Catastrophic Forgetting

In neural networks, catastrophic forgetting refers to the significant memory loss for previously learned tasks or data when the model learns new data or tasks [34], [35]. Let Θ^{init} represent a neural network model in its initial state. After training on a series of tasks, the model reaches a state Θ^t. When the model learns a new task T_{new}, its state changes to Θ^{new}. If the model's performance on an old task T_{old} decreases significantly, from $P_{old}(\Theta^t)$ to $P_{old}(\Theta^{new})$, i.e., $P_{old}(\Theta^{new}) \ll P_{old}(\Theta^t)$, this is considered catastrophic forgetting. The root cause of catastrophic forgetting is that when the neural network model learns a new task, it updates weights through gradient descent, which can disrupt the weight distribution of the previous task. Consequently, learning new tasks leads to a sharp decline in the performance of old tasks. This phenomenon is illustrated in Fig. 2. New tasks (new classes of data) can significantly alter the model's parameters, causing the gradient update direction to move towards a space that is irrelevant to the old task.

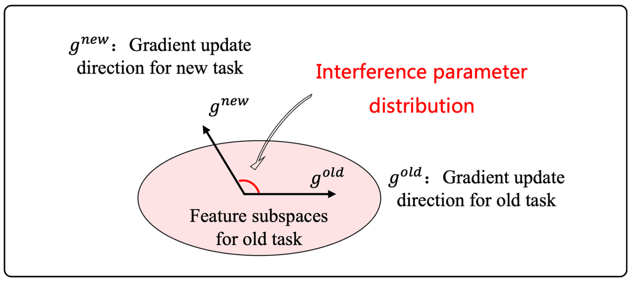

g^{new}: Gradient update direction for new task
Interference parameter distribution
g^{new}
g^{old}
g^{old}: Gradient update direction for old task
Feature subspaces for old task

Fig. 2. The gradient update direction of the old task keeps in its feature subspace, but after the new task appears, the gradient is updated to the space unrelated to the old task.

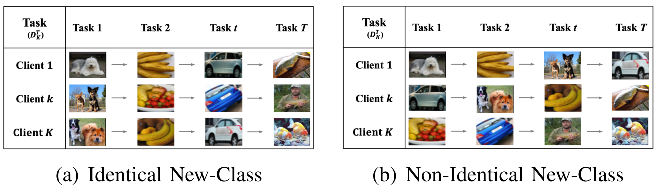


Fig. 3. Illustration of two different incremental data scenarios, where each row representing a task sequence for each client. Subfigure (a) shows the Identical New-Class scenario, where data for the same task (each column) across different clients are drawn from identical label distributions. Subfigure (b) shows the Non-Identical New-Class scenario, where data for the same task (each column) across different clients are drawn from non-identical label distributions.

C. Incremental-Data Definition

In this paper, we are the first to propose addressing the non-identical label distribution problem for incremental data in FCIL [36]. Here, we analyze the problem we solved and define in detail two different types of incremental data settings in federated learning.

Currently, most existing federated class incremental learning methods usually ideally assumes all clients have identical task sequences. In this scenario, all client data with the same task ID have the same class label distribution. We refer to this incremental data scenario as identically incremental data (Identical New-Class). However, in the real world, clients collect or produce local data independently, leading to different clients' incremental data derived from non-identical label distributions. We refer to this incremental data scenario as non- identicalincremental data (Non-Identical New-Class). Here, we provide a definition of the two incremental data scenarios.

Definition 1 (Identical New-Class): Define the t-th task of client k as D_k^t. The label of new tasks added by client k does not appear in the history tasks of any other client, i.e., $D_k^t \cap \bigcup_{\iota=1}^{t-1} D^\iota = \varnothing$. Fig. 3(a) shows.

Definition 2 (Non-Identical New-Class): The label of new tasks added by client k may have already appeared in a historical task for any other client, i.e., $D_k^t \cap \bigcup_{\iota=1}^{t-1} D^\iota \neq \varnothing$. Fig. 3(b) shows.

Section V-A will detail how to split the dataset to generate the above two incremental data scenarios.

IV. Our Method: CGoFed

In this section, we introduce our CGoFed method. We first introduce the conventional gradient-based method and notations in Section IV-A. Then, we discuss the relax-constrainted gradient update module of CGoFed to achieve a good balance between stability (anti-forgetting) and plasticity (adapting new tasks) in detail in Section IV-B. Finally, we present the cross-task gradient regularization module of CGoFed to alleviate the problem of non-identical label distribution in FCIL in Section IV-C.

A. Strong-Constrainted Gradient Update

Previous works propose the gradient-based method to prevent forgetting and realize forward knowledge transfer in continuous learning. Here, we briefly introduce conventional strong-constrainted gradient update method in three critical steps.

1) Constructing Representations Space: Gradient-based methods utilize the gradient obtained by training the model on a subset of samples to represent the gradient space of the entire task. Assume that task t is currently being trained. The representation matrix of the gradient space for task t is obtained through forward propagation of the model $\mathcal{F}$ as follows:

$$\mathbf{R}^t = \mathcal{F}(\Theta^t, X^t), \tag{2}$$

where $\mathbf{R}^t = [\mathrm{z}_1^t, \mathrm{z}_2^t, \ldots, \mathrm{z}_{n_s}^t]$ is the representation matrix for task t. $X^t = \{x_i\}_{i=1}^{n_s}$ contains n_s samples randomly sampled from the data D^t of the current task.

2) Solve for Orthogonal Basis: After computing the representation matrix for task t, gradient-based methods apply a matrix factorization algorithm to obtain basis vectors representing the gradient space of task t. The Singular Value Decomposition (SVD) algorithm is the most commonly used matrix decomposition method, and its formula is as follows:

$$\mathbf{R}^t = U^t \Sigma^t (V^t)^\top, \tag{3}$$

where U^t and V^t are orthogonal singular matrices, the left singular matrix U^t is a canonical orthogonal basis spanning $\mathbb{R}^n$. Multiplying the gradient of the new task by the left singular matrix U^t and its transpose matrix $(U^t)^\top$ projects the gradient into the orthogonal space of the gradient of the old task. The matrix Σ^t stores the singular values along its diagonal. To obtain a more compact representation matrix, we use the norm-based criteria to compute the κ-rank approximation $(R^t)_\kappa$ of R^t as follows:

$$\left\|(\mathbf{R}^t)_\kappa\right\|_F^2 \geq \epsilon \left\|\mathbf{R}^t\right\|_F^2, \tag{4}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm, and $(R^t)_\kappa$ contains the top-κ largest singular values along its main diagonal. The threshold ϵ controls the number of selected singular values and basis vectors. A larger ϵ results in more orthogonal basis vectors and a higher dimension of the orthogonal gradient space M^t. The representation space $S^t = span\{u_1^t, u_2^t, \ldots, u_\kappa^t\}$ for task t is spanned by the first κ column vectors in left singular matrix U^t. The basis vectors of S^t are stored in memory $\mathcal{M} = \{(M^t)_{t=1}^T\}$, where $M^t = [u_1^t, u_2^t, \ldots, u_\kappa^t]$. The singular values corresponding to each basis vector from a matrix $\Sigma^t = [\sigma_1^t, \sigma_2^t, \ldots, \sigma_\kappa^t]$.

3) Scaling Basis Vector: In conventional gradient-based methods, all basis vectors have the same weight when constructing the gradient space, offering stability to old tasks to some extent by maintaining uniform importance across tasks. However, recent models exhibit varying levels of forgetting across different old tasks, necessitating greater attention to tasks experiencing higher forgetting. Inspired by the non-uniform distribution of singular values, we calculate the importance of basis vectors using these singular values to bolster stability for crucial tasks. Terming the variable basis vector approach, we modify the process of deriving orthogonal bases in typical gradient-based methods using the singular values $\Sigma^t = [\sigma_1^t, \sigma_2^t, \ldots, \sigma_\kappa^t]$ obtained in (3), we transform these singular values into important coefficient $\Lambda^t = [\lambda_1^t, \lambda_2^t, \ldots, \lambda_\kappa^t]$ using the sigmoid function, preserving the original proportion of singular values Σ^t. The process is as follows:

$$\Lambda^t = \frac{1}{1+e^{-\Sigma^t}}. \tag{5}$$

Before training task $t+1$, we employ the importance coefficient Λ as the weight for each basis vectors $M^t = [u_1^t, u_2^t, \ldots, u_\kappa^t]$ to adjust the gradient space. This weighted adjustment transforms the basis vectors into $M^t = [\lambda_1^t \cdot u_1^t, \lambda_2^t \cdot u_2^t, \ldots, \lambda_\kappa^t \cdot u_\kappa^t]$ based on their singular value distribution. Following this strategy, as tasks accumulate, previous task's basis vectors receive higher relative weights, thereby enhancing the model's performance on those tasks.

However, the aforementioned gradient-based method restricts the gradient update direction of training new tasks to the orthogonal direction of the old task space, which is not always globally optimal. As the number of new tasks increases, the model's plasticity to new tasks diminishes within the limited gradient space, making it challenging to achieve a satisfactory balance between stability and plasticity.

B. Relax-Constrainted Gradient Update

In this subsection, we propose a relax-constrained gradient update module to improve the plasticity of the model for new tasks. We first describe the normal gradient constraint process and then introduce our proposed relaxation strategy.

1) Normal Gradient Constraint: After completing the training of task t, all clients will begin training on the new task $t+1$. The new task often brings many new classes, causing the descent direction of the model gradient to shift rapidly to the gradient space of the new task. The dramatic shift in descent direction is the root cause of the severe performance degradation on the old task. Therefore, we aim for the gradient update to occur in the gradient space orthogonal to the old task, as expressed in the following formula:

$$\nabla_{\Theta^{t+1}}\mathcal{L}^{t+1} = \nabla_{\Theta^{t+1}}\mathcal{L}^{t+1} - \left(\nabla_{\Theta^{t+1}}\mathcal{L}^{t+1}\right) M^t (M^t)^\top. \tag{6}$$

At the end of the new task training, we update the memory $\mathcal{M}$ that stores the basis vectors of all tasks. As mentioned above, this constraining process is too strict and limits the performance of the model on new tasks.

2) Variable Gradient Constraint: To enhance the model's plasticity for new tasks, we propose a technique called variable gradient constraint. This technique allows for flexible adjustment of the model's plasticity on new tasks. As depicted in Fig. 4, the strict constraint on the gradient update direction is identified as a significant factor contributing to poor plasticity. To address this issue, we expand the update range of the gradient, enabling the model to adapt more effectively to new tasks. This slight angle deviation is introduced by incorporating an adjustable constraint strength coefficient, denoted as μ, during the gradient update. We gradually reduce the constraint strength coefficient during the training process, providing strong constraints in the early tasks to ensure stability, and gradually relaxing the constraints as training progresses to improve plasticity on new tasks. Moreover, when forgetting reaches a certain threshold τ in the later stages of training, we automatically tighten the constraints to prevent significant forgetting. The relaxation process is as follows:

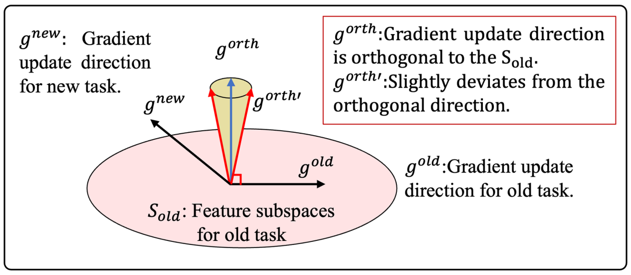

g^{new}: Gradient update direction for new task.
g^{orth}
g^{new}
$g^{orth\prime}$
g^{orth}:Gradient update direction is orthogonal to the S_{old}.
$g^{orth\prime}$:Slightly deviates from the orthogonal direction.
g^{old}
g^{old}:Gradient update direction for old task.
S_{old}: Feature subspaces for old task

Fig. 4. The diagram of Variable Gradient Constraint technique. When the update direction of the new task is not constrained, it only stays in the unrelated subspace of the old task, resulting in catastrophic forgetting. When the update direction is strictly constrained in the orthogonal direction, the adaptability of the new task is decreased due to the limited gradient space. When using the variable gradient constraint technique, the gradient update direction of the new task can be expanded and orthogonal to subspace of the old task.

$$f(\alpha, t) = \alpha^t, \tag{7}$$

$$\mu^t = \begin{cases} \mu^{\text{init}} f(\alpha, t) & \text{if } AF < \tau, \\ \mu^{\text{init}} f(\alpha, t - t_\tau) & \text{if } AF \geq \tau, \end{cases} \tag{8}$$

where $f(\alpha, t) = \alpha^t$ is used to model the exponential decay of the constraint strength with respect to the task index t. Here, μ^t represents the constraint strength coefficient for task t, controlling the degree of deviation from the orthogonal direction. We utilize a decay rate α to dynamically adjust the coefficient μ for each task based on the average forgetting (AF) observed during training. t_τ denotes the most recent task number at which AF exceeds the threshold τ. μ^{init} is the initial constraint strength coefficient, typically set to $\mu^{\text{init}} = 1$. According to (6) and (8), we reformulate the gradient update procedure as follows:

$$\nabla_{\Theta^{t+1}}\mathcal{L}^{t+1} = \nabla_{\Theta^{t+1}}\mathcal{L}^{t+1} - \mu^t(\nabla_{\Theta^{t+1}}\mathcal{L}^{t+1}) M^t (M^t)^\top. \tag{9}$$

We introduce a novel relax-constrained gradient update module designed to improve adaptability to new tasks and ensure stability for existing tasks using an adjustable constraint strength coefficient. To optimize the balance between plasticity and stability, we dynamically adjust the decay rate of the constraint strength coefficient and achieve a good balance between the stability of old tasks and plasticity for new tasks.

C. Cross-Task Gradient Regularization

In this subsection, we propose a cross-task gradient regularization module of CGoFed to solve the problem of non-identical label distribution for incremental data between the clients in federated class incremental learning.

Each client's task streams are independent, leading to unique class distributions in incremental data at each time step. This independence poses a challenge for server aggregation, as local models may interfere with each other. It is intuitive that the current task of client i may overlap with historical tasks of another client j. Therefore, leveraging valuable knowledge from historical models of other clients can mitigate catastrophic forgetting and improve the global model's performance on the current task. Next, we elaborate on our novel cross-task gradient regularization module, which integrates historical models from other clients into the optimization objective.

Firstly, the proposed cross-task gradient regularization module operates by computing and transmitting the representation matrix $\mathbf{R}^t$ of the current task t to the server, along with the uploaded model parameters at the conclusion of each task. To derive $\mathbf{R}^t$, each client randomly samples and computes it using (2) upon task t completion. This matrix shuffles feature representations across all sample batches, ensuring that previous task data is not directly shared and avoiding potential breaches of sample-level privacy. This method is computationally efficient, requiring no additional communication or computational overhead.

Secondly, for each client k, the server calculates the similarity matrix $\mathbf{\Psi_k}$ between the current task t and all tasks of other clients (indexed by i). Here, φ_i^t denotes the similarity between client k's current task and client i's t-th task. The computation of $\mathbf{\Psi_k}$ and φ_i^t is detailed below:

$$\mathbf{\Psi_k} = \begin{bmatrix} \varphi_1^1 & \cdots & \varphi_1^t \\ \vdots & \ddots & \vdots \\ \varphi_i^1 & \cdots & \varphi_i^t \end{bmatrix}, \varphi_i^t = \mathbf{D}\left(\mathbf{R}_i^t, \mathbf{R}_k^t\right). \tag{10}$$

The similarity function $\mathbf{D}$ employs the L2-norm distance, where $\mathbf{R}_i^t$ and $\mathbf{R}_k^t$ denote the representation matrices of client i and client k, respectively.

Thirdly, the server employs a task selection mechanism to choose π historical tasks with the lowest φ values for client k, based on its representation matrix $\mathbf{\Psi_k}$. These selected historical tasks have shown significant improvements for the global model on client k's current task through the computation of the cross-task cooperation loss, which is detailed as follows:

$$A\left(\Theta_k^t, \Theta^{\text{old}}\right) = \sum_{j=0}^{t-1}\sum_{i=0}^{\pi} \frac{\varphi_i^j}{\sum_{i\in\pi}\varphi_i^j}\left\|\Theta_k^t - \Theta_i^j\right\|^2, \tag{11}$$

where we select two other clients ($\pi = 2$) that exhibit the lowest φ values across historical tasks for each client k. This results in obtain $\pi * (t-1)$ historical task models are expressed by Θ^{old}, which are used to compute the regularization loss during training of the t-th task.

Lastly, the t-th column vector of the matrix signifies the similarity between the client k and other clients i for current task t. To enhance the model's performance, the server performs personalized aggregation for client k using the similarity between its model and these of other clients on the current task t as follows:

$$\Theta_k^{t,g} = \Theta_k^t + \sum_{i\in K_{\backslash k}} \frac{\varphi_i^t}{\sum_{i\in K_{\backslash k}}\varphi_i^t}\Theta_i^t. \tag{12}$$

The aggregate weights are computed based on the similarity of all representations from the previous training round. The ratio between the model of client k and the sum of the models from other clients can be set empirically.

To summarize, we rewrite the optimization objective of the local model for each client k at task t as follows:

$$CE_k(\Theta_k^t) = \frac{1}{n_k^t}\sum_{i=1}^{n_k^t}\ell(f(\mathbf{x}_{k,i}^t;\Theta_k^{t,g}), \mathbf{y}_{k,i}^t), \tag{13}$$

$$\min_{\Theta_k^{t,g}}\left[\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_k\left(\Theta_k^t\right) + A\left(\Theta_k^t, \Theta^{\text{old}}\right)\right], \tag{14}$$

where $\ell(\hat{y}, y)$ is the cross-entropy loss function. We combine the local loss function with the cross-task gradient regularization loss to obtain the complete optimization objective via the (14). Finally, the detailed process of CGoFed is described in Algorithm 1.

TABLE I
TIME COMPLEXITY ANALYSIS

		Time Complexity
Clients	Train local model	$O(N \cdot d)$
	Orthogonal gradient updates	$O(d^2)$
	Singular value decomposition	$O(d^2 \cdot \log k)$
Server	Calculate similarity matrix	$O(K^2 \cdot d)$
	Aggregate model	$O(K \cdot d)$

D. Time Complexity

Denote the number of clients, client samples, feature dimensions, and parameter dimensions as K, N, k, and d, respectively. The time complexity analysis of each module is shown in Table I. On the client side, all clients execute model training, orthogonal gradient updates, representation construction, and singular value decomposition (SVD) in parallel. The time complexity of client training is proportional to the data size and model parameter dimension, resulting in a complexity of $O(N \cdot d)$. The time complexity of orthogonal gradient updates is determined by projection calculations, which typically have a complexity of $O(d^2)$. Representation construction and SVD involve multiple iterations and matrix-matrix operations, with the primary computational cost arising from eigenvalue decomposition or iterative processes. This yields a complexity of $O(d^2 \cdot \log k)$. Therefore, the total time complexity on the client side is $O(N \cdot d + d^2 + d^2 \cdot \log k)$. On the server side, the model parameters and representations sent by the clients are aggregated. The time complexity for calculating the similarity matrix using client representations is $O(K^2 \cdot d)$, while the complexity for model aggregation is $O(K \cdot d)$. Hence, the total time complexity on the server side is $O(K^2 \cdot d + K \cdot d)$.

Algorithm 1: Constrained Gradient Optimization Strategy for Federated Class Incremental Learning.

```
Input: The number of all clients $K$, the number of tasks $T$.
Output: Global model $\Theta^{T,g}$
1  Initial the task sequence $\{D_k^1, D_k^2, ..., D_k^T\}$;
2  Initial the model parameters $\Theta_k^{init}$ for each client $k$;
3  for $t \in [T]$ do
4      ON CLIENTS:
       for $k \in [K]$ in parallel do
5          if $t = 1$ then
6              Train $\Theta_k^t$ via Eq.(1);
7              Update $\nabla_{\theta^{t+1}}\mathcal{L}^{t+1}$ on the orthogonal gradient space via Eq. (6);
8          else
9              Train $\Theta_k^t$ via Eq.(14);
10             Compute constraint strength coefficient $\mu_t$ for learned tasks via Eq. (8);
11             Update $\nabla_{\Theta^{t+1}}\mathcal{L}^{t+1}$ on the orthogonal gradient space via Eq. (9);
12         end
13         Constructed representation $\mathbf{R}_k^t$ via Eq. (2) ;
14         Update orthogonal basis with singular weight $M_k^t$ via Eq. (3,4);
15         Compute weight $\mathbf{\Lambda}^t$ of each basis via Eq. (5);
16         Send $\Theta_k^t$ and $\mathbf{R}_k^t$ to server;
17     end
18     ON SERVER:
       if $t = 1$ then
19         Aggregation $\Theta_k^{t,g}$ with $\Theta_k^t$;
20     end
21     Calculate similarity matrix $\mathbf{\Psi_k}$ between $\mathbf{R}_k^t$ and $\mathbf{R}_i^{t-1}$ via Eq. (10);
22     Aggregation $\Theta_k^{t,g}$ with $\varphi_i^t$, $\Theta_k^t$ and $\Theta_i^t$ via Eq. (12);
23 end
```

The computational complexity of our method mainly arises from performing SVD. To reduce this overhead, SVD is performed only once per task, reducing its impact on model training time. In our experiments, we provide a detailed comparison of time consumption on the CIFAR-100 and Tiny-ImageNet-200 datasets in Section VI-F. Overall, the results demonstrate that our approach achieves practical efficiency, making it suitable for real-world applications.

V. Experiments

In this section, we evaluate the proposed CGoFed method for two different class incremental learning scenarios on multiple datasets.

A. Datasets

We conduct experiments on image datasets, CIFAR-100 [37] and Tiny-ImageNet-200 [38], as well as graph datasets, CoraFull [39] and Reddit [40], to validate the effectiveness of our proposed CGoFed method across diverse datasets.

Image Datasets: The CIFAR-100 dataset contains 60,000 color images with 100 classes, each class has 600 images (32x32 pixels with 3 color channels). The Tiny-ImageNet-200 dataset contains 100,000 color images with 200 classes, each class has 500 images (64x64 pixels with 3 color channels). These datasets are extensively used to evaluate class incremental learning methods [32], [41].

Graph Datasets: The CoraFull and Reddit datasets are widely used benchmarks for graph-based machine learning tasks. The CoraFull dataset is a citation network consisting of 19,793 nodes and 126,842 edges, where nodes correspond to documents and edges represent citation relationships. Each node is associated with a feature vector of size 8,710, and the dataset includes 70 classes. The Reddit dataset is a large-scale social network graph with 232,965 nodes and 11,606,919 edges. Nodes represent posts, and edges connect posts shared by the same users. Each node has a feature vector of size 602, and the dataset contains 41 classes. We train node classification models on both graph datasets to evaluate the performance of the methods in diverse scenarios of federated class-incremental learning.

Split Incremental Data Scenario: In class incremental learning, the training data is split into task sequences, and the training data belonging to different tasks have different class labels. Therefore, class incremental learning necessitates the model to have the ability to differentiate between known classes from previous tasks and new classes for the current task. For instance, the model might initially learn about “fish” and “lion” from task 1 and subsequently about “rose” and “peony” from task 2. The class incremental learning method requires distinguishing between “rose” and “lion” among the different tasks. We partition the Identical New-Class and Non-Identical New-Class incremental data scenarios as defined in Section III.C. First, we divide the entire dataset into non-overlapping task sequence $\{D^1, D^2, \ldots, D^T\}$. For example, in the CIFAR-100 dataset, all classes are numbered from 0 to 99. Assuming there are 10 tasks, with each task containing 10 classes, the task sequences are as follows: $Task\ D^1 = \{0, 1, \ldots, 9\}$, $Task\ D^2 = \{10, 11, \ldots, 19\}$,..., $Task\ D^{10} = \{90, 91, \ldots, 99\}$. **For Identical New-Class**, each client samples from task sequence $\{D^1, D^2, \ldots, D^T\}$, such as $D_k^t \subset D^t$ is the t-th task of client k, to generate its private task sequence $\{D_k^1, D_k^2, \ldots, D_k^T\}$. The dimension of the model's classifier increases with the number of tasks. **For Non-Identical New-Class**, each client also samples from the task sequence to generate its private dataset. Then, each client independently shuffles its task sequence to obtain a new task sequence, such as $\{D_k^4, D_k^T, \ldots, D_k^1\}$. Finally, each client trains its local model following the shuffled order. The dimension of the model's classifier is fixed to the number of classes in the dataset.

B. Baseline Methods

We compare CGoFed with 7 baseline methods and clearly explain the reasons for choosing these baselines, including:

- *FedAvg [12]:* This naive FL method uses FedAvg to aggregate local models in class incremental tasks, serving as the theoretical lower bound for forgetting in FCIL due to its lack of forgetting mitigation mechanisms.

- *FedEWC [22]:* We include EWC as a regularization-based method that mitigates catastrophic forgetting by re-weighting the importance parameter of old tasks. We extend this widely cited method to federated learning and name it FedEWC.
- *FedLwF [24]:* We include LwF as a regularization-based method for its strong representative in continuous learning, where it mitigates catastrophic forgetting by preserving knowledge of previous tasks through knowledge distillation. As a widely recognized method balancing new and old task learning, we extend LwF to the federated learning setting and name it FedLwF.
- *CFeD [17]:* We include CFeD as a regularization-based method because of its effectiveness in mitigating catastrophic forgetting in FCIL. Using class-wise feature distillation, it preserves prior knowledge while adapting to new tasks; this is a strong benchmark in the federated setting.
- *TARGET [31]:* We include TARGET as a memory-based method because it employs a generative model to create synthetic data, effectively addressing catastrophic forgetting in FCIL. TARGET transfers knowledge from old tasks to new ones with synthetic data, making it a strong benchmark for ensuring stability and performance across sequential tasks.
- *GLFC [16]:* We include GLFC as a memory-based method because it selects the best model using a proxy server to preserve prior knowledge while learning new tasks in a federated setting. Replaying a portion of historical data effectively balances knowledge retention and adaptation, serving as a strong benchmark to mitigate forgetting and improve performance in FCIL.
- *FedWeIT [19]:* We include FedWeIT as a baseline for its role as a representative expansion-based method in FCIL. It mitigates catastrophic forgetting by dividing network weights into global and sparse task-specific parameters, re-weighting the latter to transfer knowledge across clients. Its innovative strategy for balancing model capacity growth and knowledge preservation makes it a strong benchmark for FCIL.

C. Implementation Details

We use the simple but efficient AlexNet [42] network for CIFAR100 and Tiny-imagenet by following [32], [41]. We set 10 clients, each client with 10 tasks, and trained each task in sequence. Each task will train for 20 communication rounds, and the local epochs of each client are 5 in each communication round, the learning rate is 0.01, and the mini-batch size is 64. For node classification tasks of graph datasets, we employ a two-layer Graph Convolutional Network (GCN) as the local model with a 256-dimensional hidden layer and an output layer whose size depends on the number of classes. The learning rate is set to 0.0001 for the CoraFull dataset and 0.01 for the Reddit dataset. All experiments were conducted on a T4 GPU, utilizing a fixed random seed to ensure the reproducibility and accuracy of the results.

D. Evaluation Metrics

In class incremental learning, we follow the *Average Accuracy* and *Average Forgetting* metrics commonly used in previous work [22], [43], [44], [45] to evaluate the effect of methods in mitigating catastrophic forgetting. To calculate the *Average Accuracy*, after training task t, we test the accuracy of the current model on the trained task and obtain the $a_{t,i}$ where $i \leq t$. The *Average Accuracy* of all the T tasks is defined as follows:

$$A_T = \frac{1}{T}\sum_{i=1}^{T} a_{t,i}, \tag{15}$$

The *Average Forgetting* is defined as the averaged disparity between minimum task accuracy during training all tasks. For T tasks, the *Average Forgetting* is computed as follows:

$$AF = \frac{1}{T-1}\sum_{i=1}^{T-1} \max_{t\in 1,\ldots,T-1} (a_{t,i} - a_{T,i}) \tag{16}$$

Note, $AF \in [-1, 1]$, lower F implies less forgetting on previous tasks. The forgetting metric represents the difference between the maximum knowledge about the previous tasks acquired throughout the learning process and the knowledge of previous tasks the model retains.

E. Experimental Results

We conduct experiments on two incremental data scenarios: Identical New-Class and Non-Identical New-Class. Table II shows the average accuracy and forgetting metric after learning all the federated class incremental learning tasks. The experiment results demonstrate that our proposed CGoFed method performs best on CIFAR-100 and Tiny-Inagenet-200 datasets.

GLFC and FedWeIT are two laudable federated class incremental learning methods. The *AF* of FedWeIT is comparable to our CGoFed, under CIFAR-100 and Tiny-ImageNet-200 datasets and two incremental data scenarios. However, the average accuracy of CGoFed is still higher than the above two methods, especially on the more challenging Tiny-ImageNet-200 dataset; our advantage is more significant, with an improvement of 21.35% .

Besides, we observe that FedAvg suffers from severe catastrophic forgetting, showing the highest forgetting regardless of which incremental data scenario. We also see that classical class incremental learning methods such as FedEWC and FedLwF perform poorly in federated learning. Due to the differences in the class distribution of clients in federated learning, the aggregation process of the global model will interfere with the model parameter distribution after each round of training, thus affecting the effect of these methods. In particular, FedEWC needs to estimate the weight parameters of previous tasks, and model aggregation seriously interferes with the weight distribution of previous tasks. Thus, FedEWC has had little improvement in FedAvg.

TABLE II
PERFORMANCE COMPARISONS BETWEEN CGOFED AND COMPARED METHOD WITH TWO DIFFERENT DATA INCREMENTAL SCENARIOS ON CIFAR-100 AND TINY-IMAGENET-200 DATASET AT 10 TASKS

Datasets	CIFAR-100				Tiny-ImageNet-200			
Scenarios	Identical New-Class		Non-identical New-Class		Identical New-Class		Non-identical New-Class	
Metrics	A_{10} (↑)	AF (↓)	A_{10} (↑)	AF (↓)	A_{10} (↑)	AF (↓)	A_{10} (↑)	AF (↓)
FedAvg	22.31%	0.6954	23.12%	0.7099	24.11%	0.8602	25.80%	0.8848
FedEWC (PNAS17 [22])	22.07%	0.6708	22.96%	0.6681	25.56%	0.7168	28.56%	0.7164
FedLwF (TPAMI17 [24])	28.53%	0.5356	31.10%	0.5109	26.39%	0.6052	30.40%	0.6082
TARGET (CVPR23 [31])	26.88%	0.3510	31.28%	0.3137	14.55%	0.4009	15.13%	0.3872
CFeD (IJCAI22 [17])	18.78%	0.6193	19.61%	0.6489	23.94%	0.7697	23.71%	0.7646
GLFC (CVPR22 [16])	54.78%	0.2207	57.09%	0.3494	21.70%	0.3978	22.5%	0.3644
FedWeIT (ICML22 [19])	54.25%	0.0789	54.19%	0.0292	21.53%	0.1714	20.46%	0.0518
CGoFed (Ours)	**66.88%**	**0.0051**	**66.03%**	**0.0041**	**43.18%**	**0.0307**	**43.85%**	**0.0333**

TABLE III
PERFORMANCE COMPARISON BETWEEN CGOFED AND BASELINE METHODS ON GRAPH DATASETS: CORAFULL WITH 7 TASKS AND REDDIT WITH 10 TASKS

Datasets	**CoraFull**		**Reddit**	
Metrics	A_7(↑)	AF(↓)	A_{10}(↑)	AF(↓)
FedAvg	21.09%	0.7269	17.79%	0.9059
FedEWC [22]	29.39%	0.7578	34.89%	0.7153
FedLwF [24]	25.45%	0.5336	38.43%	0.6762
CGoFed (Ours)	42.57%	0.4188	57.95%	0.1902

For two incremental data scenarios, in CIFAR-100, GLFC and TARGET have higher average accuracy in the Non-Identical New-Class scenario, 2.3% -4.4% more elevated than in the Identical New-Class scenario. The Identical New-Class scenario is the more challenging setting because each task brings more new classes, making it more difficult for the model to adapt to the new class. Our CGoFed method has achieved far better results than the comparison methods, no matter what incremental data scenario.

F. Result of Graph Learning

To evaluate the effectiveness of CGoFed on diverse datasets, we conducted experiments using graph datasets. Notably, most existing FCIL methods are specifically designed for image data and are incompatible with graph datasets and GNN models. Therefore, we can only adapt two regularization-based methods, FedEWC and FedLwF, for use with GNN models to enable a fair comparison. The results of these experiments are summarized in Table III. The results demonstrate that CGoFed consistently outperforms baseline methods (FedAvg, FedEWC, and FedLwF) on both the CoraFull and Reddit datasets in terms of A_7 and AF. On the two datasets, CGoFed achieves the highest accuracy of 76.38% and 69.20% , significantly outperforming the baseline methods. It also demonstrates the lowest AF of 0.2095 and 0.2831, respectively. These results highlight the adaptability of our method to different types of data and models. While the performance of the comparative methods is less satisfactory, it consistents with the the accepted benchmark of graph continual learning [46], [47].

G. Result of Task Accuracy

It is worth mentioning that the FedWeIT is the most effective method for mitigating the forgetting problem among all the compared methods in both incremental data scenarios, while CFeD exhibits high adaptability to new tasks (detailed in Section VI-B). Therefore, we plot the accuracy change curve for each task of CFeD, FedWeIT, and CGoFed, starting from the prior task of each task. As shown in Fig. 5, our proposed CGoFed method quickly adapts to each task at the beginning and maintains or even improves task accuracy after completion, particularly for Task 5 and Task 8. However, after training per task, each task of CFeD showed a severe accuracy drop during the new task training phase. We analytically consider that the reasons for the poor effect of CFeD include the following two aspects. Firstly, the CFeD method is only suitable for a small number of tasks (T=2), as mentioned by [17]. When the number of tasks increases, the distillation loss term of this method will consider more and more historical tasks, and the distillation effect of a single historical task or early task will decline rapidly. Second, distillation-based methods usually rely on particular surrogate datasets. Our investigation shows that the surrogate dataset provided by CFeD is only suitable for the simple CIFAR100, but severe forgetting still occurs for the more challenging Tiny-Imagenet-200. Its average forgetting metric is higher than EWC and LwF but lower than FedAvg. The fundamental reason is the limited capacity of previous tasks' knowledge carried by the distillation loss term, and it cannot play a good role when the number of tasks is large.

H. Result of Non-IID Scenario

To further verify that our CGoFed method can still perform well in the non-independent identically distributed (Non-IID) scenario popular in federated learning, we experiment with the Non-IID setting, the same as the TARGET method [31]. We set different Dirichlet parameters β to generate different Non-IID Settings, i.e., $\beta = \{0.05, 0.1, 0.5, 1\}$, and the Average Accuracy and the Average Forgetting metrics are shown in Fig. 7. The Average Accuracy and Average Forgetting metric of our CGoFed are better than those of the comparison methods, regardless of which Non-IID setting. The lower the Dirichlet parameter, the more severe the Non-IID data distribution between clients. The Average Forgetting metric of CGoFed decreases as the degree of Non-IID exacerbates. When the data distribution is extremely imbalanced ($\beta = 0.05$), CGoFed has the highest anti-forgetting ability, and the Average Accuracy is 21.89% higher than the optimal comparison method (LwF). When the Dirichlet parameter

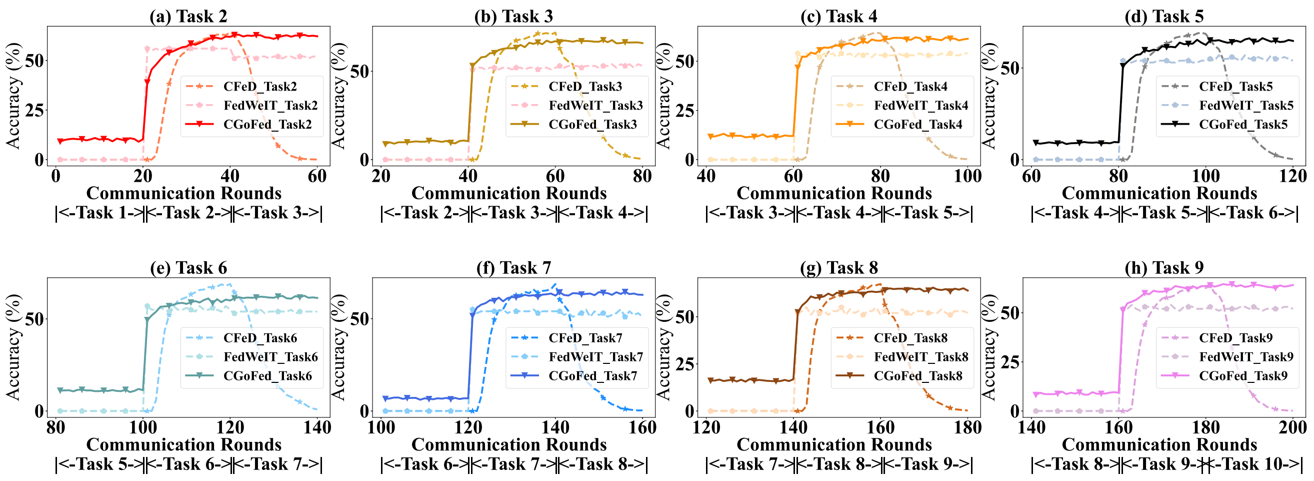


Fig. 5. Accuracy change curves of each task for FedWeIT, CFeD and CGoFed methods in the CIFAR-100 dataset, Identical New-Class scenario.

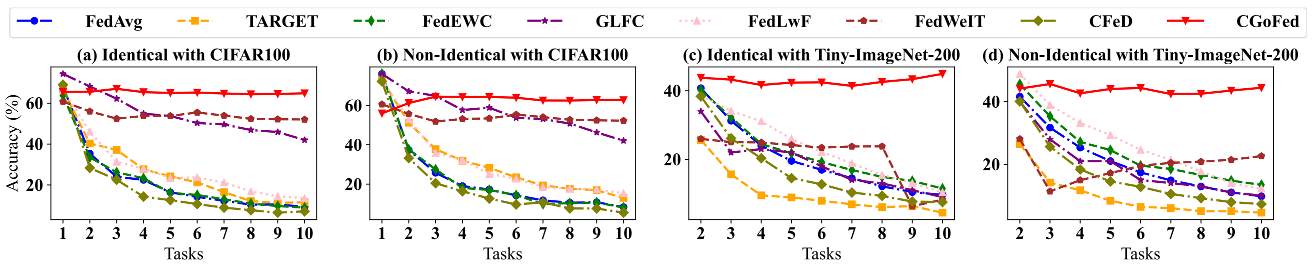


Fig. 6. Performance comparisons between CGoFed and other methods in Identical New-Class and Non-Identical New-Class scenarios with CIFAR-100 and Tiny-ImageNet-200 datasets.

is 1, the difference in Average Accuracy and Average Forgetting metrics is slight compared to the IID setting. The above results forcefully show that our method can alleviate the catastrophic forgetting problem when the client data is Non-IID in federated learning.

The severe performance degradation of these comparison methods is due to these regularization methods relying on a loss function based on predicted labels to prevent excessive updates to model parameters, thus mitigating catastrophic forgetting. However, the predicted labels are affected by the data distribution, which means that the predicted labels are difficult to align with the true labels under the extreme non-IID distribution, leading to severe bias in the loss function, and thus resulting in significant performance degradation of the regularization method. In contrast, CGoFed is independent of the distribution of predicted labels, making it less sensitive to variations in data distribution. Consequently, it outperforms the comparison methods in extreme Non-IID scenarios.

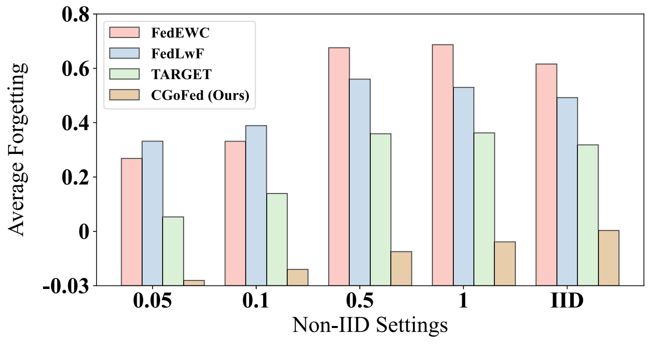
VI. Analysis

A. Forgetting Analysis

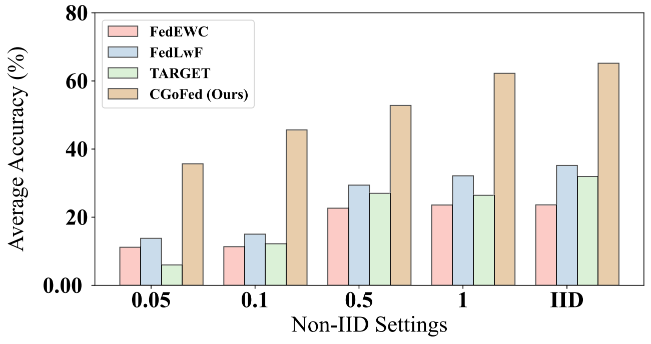
We plotted the variation curve of the historical task average accuracy at the end of each task during training as shown in Fig. 6. In the training process of all tasks, our CGoFed method always maintains an excellent anti-forgetting effect, and even when training the later tasks, CGoFed can not only balance the accuracy of historical tasks but also improve the performance of the current task. In the CIFAR-100 dataset, the anti-forgetting effect of GLFC and FedWeIT is second only to CGoFed but significantly higher than other comparison methods. The variation curve of the average accuracy of FedWeIT is smoother, but the average accuracy is 10.94% lower than that of CGoFed. The variation curve of the average accuracy of GLFC has a slight downward trend. However, it can still maintain good performance in the training process of each task, so it has a high average accuracy. The variation curve of the average accuracy of other comparison methods has an inevitable and severe decline. In the Tiny-ImageNet-200 dataset, The anti-forgetting effect of FedWeIT also shows a severe decrease and a sudden inflection point. The FedWeIT method preserves the knowledge in previous tasks by freezing the model parameters of historical tasks, and the model parameters with general knowledge cannot distinguish the new class well.

B. Adaptability Analysis of New Tasks

An important metric for evaluating federated class incremental learning methods is their ability to adapt to new tasks. Incremental learning methods often face the *stability-plasticity*



(a) Average Forgetting



(b) Average Accuracy

Fig. 7. The Average Accuracy and Average Forgetting for all learned tasks on CIFAR100 for 10 tasks under both Non-IID and IID settings.

dilemma [48], [49], including in federated class incremental learning. During the learning of new tasks, the model should not only have the ability to adapt to the new task quickly but also retain knowledge from previous tasks.

Therefore, we plotted the accuracy variation of the current task during the first five training rounds for the last task. We focus on the model's ability to rapidly adapt to new tasks in the early rounds rather than the final accuracy achieved on these tasks. The adaptability of all compared methods to new tasks is shown in Fig. 8. In the CIFAR-100 dataset, CGoFed demonstrates superior performance with minimal tuning, achieving over 50% accuracy after the first round of new tasks. While FedWeIT and GLFC also adapt quickly to new tasks, their accuracy is lower than that of CGoFed. Notably, CFeD requires re-learning for new tasks but shows good adaptability. This result aligns with the stability-plasticity dilemma in incremental learning: CFeD exhibits high plasticity (quick adaptation to new tasks) but low stability (quick forgetting of old tasks). In contrast, our proposed CGoFed method balances stability and plasticity.

C. Impact of Constraint Strength Coefficient α

The hyper-parameter α controls the constraint strength of the task gradient. To investigate the impact of the hyper-parameter, we conduct experiments on the CIFAR-100 dataset with a Non-Identical New-Class task distribution, focusing on the impact of constraint strength α on the plasticity of new tasks. Accordingly, we measure the accuracy of new tasks after each task is completed and illustrate the accuracy variation as the number of tasks increases. The results are depicted in Fig. 9. When $\alpha = 1$, the constraint strength of the gradient update direction remains unchanged, resulting in a strictly orthogonal update direction and the lowest accuracy for each task at the end of training. By gradually reducing the constraint strength, i.e., decreasing α, we observe an increase in the accuracy of each task. This indicates enhanced plasticity for new tasks, and the enhancement is more pronounced as the number of tasks increases. When $\alpha = 0.98$, our method significantly improves the plasticity of the new task compared to strict constraint gradient update direction ($\alpha = 1$). However, when $\alpha < 0.98$, the improvement in plasticity diminishes. Therefore, we present the accuracy on each task and the Average Forgetting (AF) after training all tasks for values of α less than 1 and greater than 0.98in Table IV. The results indicate that our proposed method does not cause substantial forgetting. The AF is close to the optimal comparison method ($AF = 0.0292$) and significantly lower than other methods ($AF = 0.3494$). Overall, we enhance the model's plasticity on new tasks and final performance with minimal negative impact.

D. Impact of Threshold τ

The threshold τ in (7) and (8) determine when to reset the constraint strength coefficient to prevent further forgetting. To examine the impact of τ on final accuracy, we conduct experiments on the CIFAR-100 dataset using a Non-Identical New-Class task distribution. These experiments are carried out under the constraint strength with decay rate $\alpha = 0.98$ and various values of $\tau = \{0, 0.001, 0.01, 0.05\}$, with the results presented in Fig. 10. According to our strategy in (7) and (8), whenever the Average Forgetting (AF) exceeds the threshold, resetting the constraint strength helps reduce AF and avoid forgetting. When $\tau = 0$, the reset occurs on task 0,at $\tau = 0.001$, it happens on task 3; at $\tau = 0.01$, it happens on task 4; and at $\tau = 0.05$, it occurs on task 6. Selecting an appropriate threshold allows the constraint strength to be reset at the appropriate time, such as with $\tau = 0.01$, where AF fluctuates minimally in later tasks(The distance between the dashed lines is the smallest). However, when $\tau > 0.01$, AF increases more, leading to a significant drop in anti-forgetting performance (AF up to 0.1076). On the other hand, when $\tau < 0.01$, the constraint strength is reset too early during training, fixing it at an initial value for the rest of the process, as a result, the constraint cannot be effectively relaxed and the AF fluctuates greatly, making it difficult to find a good balance between keeping old tasks stable and adapting to new tasks.

E. Ablation Studies

Table V presents the results of ablation studies. "Ours-w/o Regular" represents the results when our method does not use the cross-task gradient regularization module. The results demonstrate that the cross-task gradient regularization module can effectively improve the accuracy of the final global model by 3.17% . We did not remove the relax-constrained gradient update module because it is essential for addressing the forgetting problem. Our method performs similarly to FedAvg

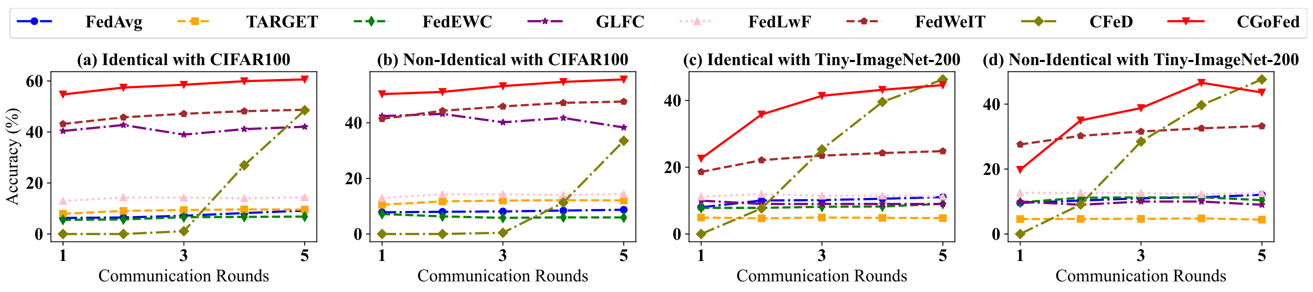


Fig. 8. Comparison of adaptability to new tasks between CGoFed and other methods in Identical New-Class and Non-Identical New-Class scenario with CIFAR-100 and Tiny-ImageNet-200 datasets.

TABLE IV
ACCURACY (ACC) OF EACH TASK AT THE END OF WHOLE TRAINING FOR DIFFERENT CONSTRAINT STRENGTH COEFFICIENT α ON CIFAR100 DATASET AND UNDER NON-IDENTICAL NEW-CLASS SCENARIO

α	Task 1		Task 2		Task 3		Task 4		Task 5		Task 6		Task 7		Task 8		Task 9		Task 10	
	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)	ACC(↑)	AF(↓)
0.98	59.70%	0.00	53.90%	0.00	66.80%	0.0033	62.50%	0.0407	65.30%	0.0352	61.90%	0.0312	58.20%	0.0341	63.60%	0.0397	69.20%	0.0371	68.50%	0.0552
0.99	63.60%	0.00	59.90%	0.00	71.10%	0.00	65.40%	0.0275	69.00%	0.0416	65.40%	0.0217	59.00%	0.019	64.90%	0.0212	70.00%	0.0216	65.40%	0.0287
1	69.80%	0.00	65.70%	0.00	74.10%	0.00	66.50%	0.0045	67.30%	0.0048	65.50%	0.0037	58.20%	0.0044	66.10%	0.0061	64.90%	0.0026	62.20%	0.0037

AF is the average forgetting over all trained tasks after training each task.

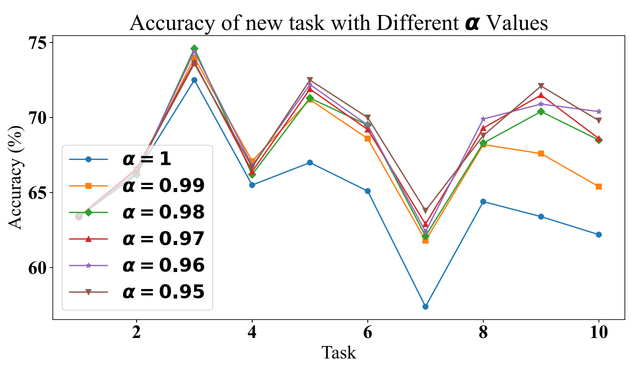


Fig. 9. The accuracy of new tasks with different α values.

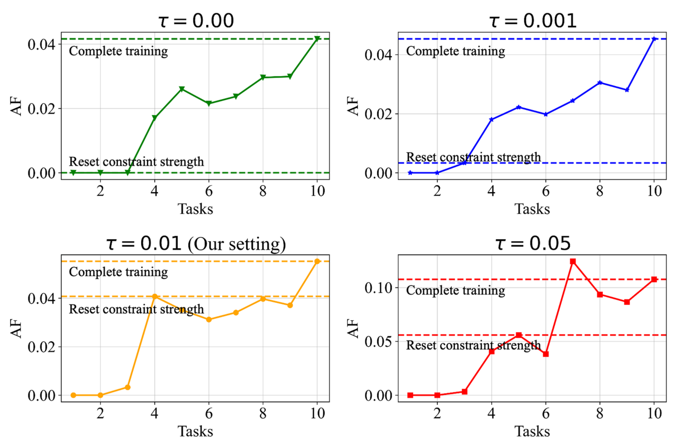


Fig. 10. Average Forgetting (AF) trend under different thresholds (τ) with a constraint strength decay rate of $\alpha = 0.98$, on the CIFAR-100 dataset with Non-Identical New-Class task distribution.

TABLE V
ABLATION RESULT (%) ON CIFAR-100

Scenario	Identical New-Class		Non-Identical New-Class	
metric	AA(↑)	AF (↓)	AA (↑)	AF (↓)
Strict Constraint	60.26%	0.1961	59.41%	0.2143
Ours-w/o Regular	62.49%	0.0311	61.19%	0.4867
Ours	65.66%	0.0332	65.37%	0.0319

Experimental results based on two incremental data scenarios. AA(↑) is the average accuracy of all tass, and AF (↓) is the average forgetting.

when the relax-constrained gradient update module is not used. It should be noted that the cross-task gradient regularization module primarily enhances the performance of the global model. However, as a regularization term, it has a very limited effect on mitigating forgetting, as verified in the field of continual learning. Additionally, we compared a "Strict Constraint" gradient update strategy [50], which performed worse than "Ours-w/o Regular". This result demonstrates that our relaxed strategy not only improves average accuracy but also better mitigates forgetting, effectively achieving a balance between stability and plasticity.

F. Efficiency Analysis

To validate the efficiency of CGoFed, we compared the training times of CGoFed with other methods on the CIFAR-100 and Tiny-ImageNet-200 datasets under Identical New-Class and Non-Identical New-Class scenarios. Fig. 11 demonstrates that the training time of CGoFed is significantly lower compared to the current state-of-the-art methods, GLFC and FedWeIT, across two different datasets; thus, CGoFed has a great advantage in terms of efficiency. Similar trends are observed for Tiny-ImageNet-200, where CGoFed maintains efficient training times even as the dataset complexity increases, significantly outperforming most baselines. Although our training time is slightly higher than CFeD, FedLwF, and FedEWC, our performance is substantially better. Overall, CGoFed achieves an acceptable performance while maintaining optimal time efficiency.

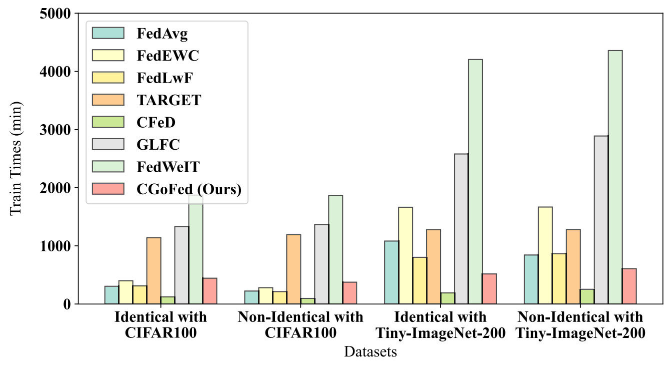


Fig. 11. Performance comparisons between CGoFed and other methods in Identical New-Class and Non-Identical New-Class scenarios with CIFAR-100 and Tiny-ImageNet-200 datasets.

VII. Conclusion

In conclusion, this paper presents the CGoFed method to address the trade-off problem between stability and plasticity in federated class incremental learning by focusing on gradient updates. Our method relax-constrained the gradient update direction of the local model to preserve knowledge from previous tasks without revisiting historical task data or relying on auxiliary datasets. Additionally, we designed a regularization loss to facilitate client cross-task cooperation from the server's perspective. Extensive experimental results demonstrate the effectiveness of our method compared to existing federated class incremental learning methods.

References

[1] F. Liu, Z. Zheng, Y. Shi, Y. Tong, and Y. Zhang, "A survey on federated learning: A perspective from multi-party computation," *Front. Comput. Sci.*, vol. 18, no. 1, 2024, Art. no. 181336.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[3] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf. USENIX Assoc.*, 2020, pp. 493–506.

[4] N. Sun, W. Wang, Y. Tong, and K. Liu, "Blockchain based federated learning for intrusion detection for Internet of Things," *Front. Comput. Sci.*, vol. 18, no. 5, Art. no. 185328, 2024.

[5] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1759–1799, Third Quarter 2021.

[6] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, Third Quarter 2021.

[7] D. C. Nguyen et al., "Federated learning for smart healthcare: A survey," *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–37, 2022.

[8] J. Li et al., "A federated learning based privacy-preserving smart healthcare system," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 2021–2031, Mar. 2022.

[9] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthc. Inform. Res.*, vol. 5, pp. 1–19, 2021.

[10] P. Zhou, K. Wang, L. Guo, S. Gong, and B. Zheng, "A privacy-preserving distributed contextual federated online learning framework with Big Data support in social recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 824–838, Mar. 2021.

[11] W. Guo, F. Zhuang, X. Zhang, Y. Tong, and J. Dong, "A comprehensive survey of federated transfer learning: Challenges, methods and applications," *Front. Comput. Sci.*, vol. 18, no. 6, 2024, Art. no. 186356.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[13] S. Guo, X. Wang, S. Long, H. Liu, L. Hai, and T. H. Sam, "A federated learning scheme meets dynamic differential privacy," *CAAI Trans. Intell. Technol.*, vol. 8, no. 3, pp. 1087–1100, 2023.

[14] Y. Shi, Y. Tong, Y. Zeng, Z. Zhou, B. Ding, and L. Chen, "Efficient approximate range aggregation over large-scale spatial data federation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 418–430, Jan. 2023.

[15] X. Yang, H. Yu, X. Gao, H. Wang, J. Zhang, and T. Li, "Federated continual learning via knowledge fusion: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 8, pp. 3832–3850, Aug. 2024.

[16] J. Dong et al., "Federated class-incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*,, 2022, pp. 10164–10173.

[17] Y. Ma, Z. Xie, J. Wang, K. Chen, and L. Shou, "Continual federated learning based on knowledge distillation," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 2182–2188.

[18] 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.

[19] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 12073–12086.

[20] S. M. Hendryx, D. R. Kc, B. Walls, and C. T. Morrison, "Federated reconnaissance: Efficient, distributed, class-incremental learning," 2021, *arXiv:2109.00150*.

[21] Y. Wang et al., "Distribution-regularized federated learning on non-IID data," in *Proc. Int Conf. Data. Eng.*, 2023, pp. 2113–2125.

[22] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

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

[24] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[25] C. Xu, Z. Hong, M. Huang, and T. Jiang, "Acceleration of federated learning with alleviated forgetting in local training," 2022, *arXiv:2203.02645*.

[26] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 38461–38474.

[27] J. Zhang, S. Guo, J. Guo, D. Zeng, J. Zhou, and A. Y. Zomaya, "Towards data-independent knowledge transfer in model-heterogeneous federated learning," *IEEE Trans. Comput.*, vol. 72, no. 10, pp. 2888–2901, Oct. 2023.

[28] Y. Chen, W. Lu, X. Qin, J. Wang, and X. Xie, "MetaFed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16671–16682, Nov. 2024.

[29] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.

[30] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.

[31] J. Zhang, C. Chen, W. Zhuang, and L. Lyu, "TARGET: Federated class-continual learning via exemplar-free distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2023, pp. 4782–4793.

[32] Y. F. Bakman, D. N. Yaldiz, Y. H. Ezzeldin, and S. Avestimehr, "Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning," 2023, *arXiv:2309.01289*.

[33] J. Dong, H. Li, Y. Cong, G. Sun, Y. Zhang, and L. V. Gool, "No one left behind: Real-world federated class-incremental learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2054–2070, Apr. 2024.

[34] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[35] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychol. Rev.*, vol. 97, no. 2, Art. no. 285, 1990.

[36] X.-C. Li et al., "Map: Model aggregation and personalization in federated learning with incomplete classes," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 6560–6573, Nov. 2024.

[37] A. Krizhevsky and G. Hinton, *Learn. Mult. Layers Features Tiny Images*, vol. 1, p. 7, 2009.

[38] S. University, "Tiny imagenet visual recognition challenge," 2015. [Online]. Available: https://tiny-imagenet.herokuapp.com/

[39] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Inf. Retrieval*, vol. 3, pp. 127–163, 2000.

[40] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[41] G. Saha, I. Garg, and K. Roy, "Gradient projection memory for continual learning," 2021, *arXiv:2103.09762*.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 25, pp. 84–90, 2012.

[43] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 532–547.

[44] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3390–3398.

[45] S. I. Mirzadeh, M. Farajtabar, R. Pascanu, and H. Ghasemzadeh, "Understanding the role of training regimes in continual learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7308–7320.

[46] Y. Liu, R. Qiu, and Z. Huang, "CaT: Balanced continual graph learning with graph condensation," in *Proc. 2023 IEEE Int. Conf. Data Mining*, 2023, pp. 1157–1162.

[47] X. Zhang, D. Song, and D. Tao, "CGLB: Benchmark tasks for continual graph learning," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 13006–13021.

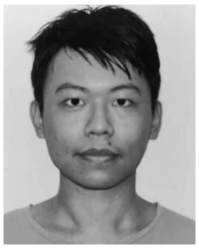
[48] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. V. De Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, May 2023.

[49] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Front. Psychol.*, vol. 4, 2013, Art. no. 504.

[50] G. Saha and K. Roy, "Continual learning with scaled gradient projection," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 9677–9685.

Jiyuan Feng received the ME degree in computer technology from the Department of Cyberspace Security, Guangzhou University, Guangzhou, China, in 2022. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include federated learning, continual learning, and data mining.

Xu Yang received the ME degree in computer science and technology from the Department of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai, China, in 2021. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include federated learning, data mining, and machine learning.



Liwen Liang received the BS degree in computer science and technology from the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China, in 2023. He is currently working toward the ME degree with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include federated learning, domain generalization, and machine learning.

Weihong Han received the MS degree from the National University of Defense Technology, China, in 1997, and the PhD degree from the National University of Defense Technology, in 2000. She is member of the Cyber Security Association of China. From 2002 to 2017, she worked with the National University of Defense Technology. Now, she is a researcher with the Department of New Networks, Peng Cheng Laboratory, China. Her current research interests include computer network and network security.

Binxing Fang received the MS degree in computer science and technology from Tsinghua University, Beijing, in 1984, and the PhD degree in computer science and technology from the Harbin Institute of Technology, Harbin, in 1989. He is currently a member of the Chinese Academy of Engineering. His current research interests include computer networks, information and network security, and artificial intelligence security.

Qing Liao (Member, IEEE) received the PhD degree from the Hong Kong University of Science and Technology, in 2016. She is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include data mining, artificial intelligence and information security, etc.