Federated Class-Incremental Learning

Jiahua Dong[1, 2*], Lixu Wang[3*], Zhen Fang[4], Gan Sun[1†], Shichao Xu[3], Xiao Wang[3†], Qi Zhu[3†]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences.
[2]University of Chinese Academy of Sciences. [3]Northwestern University.
[4]DeSI Lab, AAII, University of Technology Sydney.
dongjiahua@sia.cn, lixuwang2025@u.northwestern.edu, {fzjlyt, sungan1412}@gmail.com
{shichaoxu2023@u., wangxiao@, qzhu@}northwestern.edu

Abstract

Federated learning (FL) has attracted growing attentions via data-private collaborative training on decentralized clients. However, most existing methods unrealistically assume object classes of the overall framework are fixed over time. It makes the global model suffer from significant catastrophic forgetting on old classes in real-world scenarios, where local clients often collect new classes continuously and have very limited storage memory to store old classes. Moreover, new clients with unseen new classes may participate in the FL training, further aggravating the catastrophic forgetting of global model. To address these challenges, we develop a novel Global-Local Forgetting Compensation (GLFC) model, to learn a global class-incremental model for alleviating the catastrophic forgetting from both local and global perspectives. Specifically, to address local forgetting caused by class imbalance at the local clients, we design a class-aware gradient compensation loss and a class-semantic relation distillation loss to balance the forgetting of old classes and distill consistent inter-class relations across tasks. To tackle the global forgetting brought by the non-i.i.d class imbalance across clients, we propose a proxy server that selects the best old global model to assist the local relation distillation. Moreover, a prototype gradient-based communication mechanism is developed to protect the privacy. Our model outperforms state-of-the-art methods by 4.4%∼15.1% in terms of average accuracy on representative benchmark datasets. The code is available at https://github.com/conditionWang/FCIL.

1. Introduction

Federated learning (FL) [4, 18, 42, 46] enables multiple local clients to collaboratively learn a global model while providing secure privacy protection for local clients. It successfully addresses the data island challenge without completely compromising clients' privacy [12, 22]. Recently, it has attracted significant interests in academia and achieved remarkable successes in various industrial applications, *e.g.*, autonomous driving [39], wearable devices [33], medical diagnosis [10, 52] and mobile phones [36].

Generally, most existing FL methods [16, 42, 46, 52] are modeled in a static application scenario, where data classes of the overall FL framework are fixed and known in advance. However, real-world applications are often dynamic, where local clients receive the data of new classes in an online manner. To handle such a setting, existing FL methods typically require storing all training data of old classes at the local clients' side so that a global model can be obtained via FL, however the high storage and computation overhead may render the FL unrealistic when new classes arrive dynamically [33, 36, 47, 52]. And if these methods [42, 52] are required to learn new classes continuously with very limited storage memory, they may suffer from significant performance degradation (*i.e.*, catastrophic forgetting [20, 37, 40]) on old classes. Moreover, in real-world scenarios, new local clients that collect the data of new classes in a streaming manner may want to participate in the FL training, which could further exacerbate the catastrophic forgetting on old classes in the global model training.

To address these practical scenarios, we consider a challenging FL problem named *Federated Class-Incremental Learning (FCIL)* in this work. In the FCIL setting, each local client collects the training data continuously with its own preference, while new clients with unseen new classes could join in the FL training at any time. More specifically, the data distributions of the collected classes across the current and newly-added clients are non-independent and identically distributed (non-i.i.d.). FCIL requires these local clients to collaboratively train a global model to learn new classes continuously, with constraints on privacy preservation and limited memory storage [37, 49]. To better com-

*Equal contributions (ordered alphabetically). †Corresponding authors.

prehend the FCIL problem, we here use COVID-19 diagnosis among different hospitals as a possible example [6]. Imagine that before the pandemic, there could be hundreds of hospitals working collaboratively to train a global infectious disease diagnosis model via FL. Due to the sudden emergence of COVID-19, these hospitals will collect a large amount of new data related to COVID-19 and add them into the FL training as new classes. Moreover, new hospitals whose main focus is not infectious diseases may join the fight against COVID-19, where they have little data of the old infectious diseases, and all hospitals should learn to diagnose the old diseases and new COVID-19 variants. In such scenarios, most existing FL methods will likely suffer from catastrophic forgetting on old diseases diagnosis under the sudden emergence of new COVID-19 variants data.

An intuitive way to address new classes (*e.g.*, learning new COVID-19 variants) continuously in the FCIL setting is to simply integrate FL [4, 32, 42] and class-incremental learning (CIL) [17, 37, 50] together. However, such strategy needs the central server to know when and where the data of new classes arrives (privacy-sensitive information), which violates the requirement of privacy preservation in FL. In addition, although local clients can utilize conventional CIL [17, 37] to address their local catastrophic forgetting, the non-i.i.d. class imbalance across clients may still cause heterogeneous forgetting on different clients, and this simple integration strategy could further exacerbate local catastrophic forgetting due to the heterogeneous global catastrophic forgetting on old classes across clients.

To tackle these challenges in FCIL, we propose a novel *Global-Local Forgetting Compensation (GLFC)* model in this paper, which effectively addresses local catastrophic forgetting occurred on local clients and global catastrophic forgetting across clients. Specifically, we design a class-aware gradient compensation loss to alleviate the local forgetting brought by the class imbalance at the local clients via balancing the forgetting of different old classes, and propose a class-semantic relation distillation loss to distill consistent inter-class relations across different incremental tasks. To overcome the global catastrophic forgetting caused by the non-i.i.d. class imbalance across clients, we design a proxy server to select the best old global model for the class-semantic relation distillation at the local side. Considering the privacy preservation, the proxy server collects perturbed prototype samples of new classes from local clients via a prototype gradient-based communication mechanism, and then utilizes them to monitor the performance of the global model for selecting the best one. Our model achieves 4.4%∼15.1% improvement in terms of average accuracy on several benchmark datasets, when compared with a variety of baseline methods. The major contributions of this paper are summarized as follows:

- We address a practical FL problem, namely Federated Class-Incremental Learning (FCIL), in which the main challenges are to alleviate the catastrophic forgetting on old classes brought by the class imbalance at the local clients and the non-i.i.d class imbalance across clients.
- We develop a novel Global-Local Forgetting Compensation (GLFC) model to tackle the FCIL problem, alleviating both local and global catastrophic forgetting. To our best knowledge, this is the first attempt to learn a global class-incremental model in the FL settings.
- We design a class-aware gradient compensation loss and a class-semantic relation distillation loss to address local forgetting, by balancing the forgetting of old classes and capturing consistent inter-class relations across tasks.
- We design a proxy server to select the best old model for class-semantic relation distillation on the local clients to compensate global forgetting, and we protect the communication between this proxy server and clients with a prototype gradient-based mechanism for privacy.

2. Related Work

Federated Learning (FL) is a decentralized learning framework that can train a global model by aggregating local model parameters [24, 44, 51, 53]. To collaboratively learn a global model, [32] proposes to aggregate local models via a weight-based mechanism. [38] introduces a proximal term to help local model approximate the global ones. [42] focuses on minimizing the model discrepancies across clients via an improved EWC. Moreover, [4] designs a layer-wise aggregation strategy to reduce computation overhead [55, 56]. [18] sacrifices the local optimality for rapid convergence, while [12,22] aim to improve the performance of local models. [34] integrates unsupervised domain adaptation [5, 9, 27, 28, 57] into federated learning framework [14, 31]. However, these existing FL methods cannot effectively learn new classes continuously, due to the limited memory to store old classes at the local clients' side.

Class-Incremental Learning (CIL) aims to learn new classes continuously while tackling forgetting on old classes [1, 19, 54]. Without access to the data of old classes, [20] designs new regulators for balancing the biased model optimization caused by new classes, and [25,41] use the knowledge distillation to surmount catastrophic forgetting. [40, 48] introduce generative adversarial networks to produce synthetic data of old classes. As claimed in [11, 30, 37, 49], the class imbalance between old and new classes is a crucial challenge for exemplar replay methods. [29, 50] design a self-adaptive network to balance biased predictions. [17] uses the causal effect on knowledge distillation to rectify class imbalance. [43] introduces geodesic path to traditional knowledge distillation. [1] combines task-wise knowledge distillation and separated softmax for bias compensation. These CIL methods however cannot be applied to tackle our

FCIL problem, due to their strong assumptions on when and where the data of new classes arrive.

3. Problem Definition

In the standard class-incremental learning [37, 41, 43], there are a sequence of streaming tasks $\mathcal{T} = \{\mathcal{T}^t\}_{t=1}^{T}$, where T denotes the task number, and the t-th task $\mathcal{T}^t = \{\mathbf{x}_i^t, \mathbf{y}_i^t\}_{i=1}^{N^t}$ consists of N^t pairs of samples $\mathbf{x}_i^t$ and their one-hot encoding labels $\mathbf{y}_i^t \in \mathcal{Y}^t$. $\mathcal{Y}^t$ represents the label space of the t-th task including C^t new classes that are different from $C^p = \sum_{i=1}^{t-1} C^i \subset \cup_{j=1}^{t-1}\mathcal{Y}^j$ old classes in previous $t-1$ tasks. Inspired by [30, 37, 49], we construct an exemplar memory $\mathcal{M}$ to select $\frac{|\mathcal{M}|}{C^p}$ exemplars for each old class in the t-th incremental task, and it satisfies $\frac{|\mathcal{M}|}{C^p} \ll \frac{N^t}{C^t}$.

We then extend conventional class-incremental learning to Federated Class-Incremental Learning (FCIL). Given K local clients $\{\mathcal{S}_l\}_{l=1}^{K}$ and a global central server $\mathcal{S}_G$, for the r-th global round ($r = 1, \cdots, R$), a set of local clients are randomly selected to participate in the gradient aggregation. Specifically, once the l-th client $\mathcal{S}_l$ is selected at each global round for the t-th incremental task, it will receive the latest global model $\Theta^{r,t}$, and train $\Theta^{r,t}$ on its privately accessible t-th incremental task $\mathcal{T}_l^t \cup \mathcal{M}_l \sim \mathcal{P}_l^{|\mathcal{T}_l^t|+|\mathcal{M}_l|}$, where $\mathcal{T}_l^t = \{\mathbf{x}_{li}^t, \mathbf{y}_{li}^t\}_{i=1}^{N_l^t} \subset \mathcal{T}^t$ is the training data of new classes, $\mathcal{M}_l$ denotes its exemplar memory, and $\mathcal{P}_l$ is the class distribution of the l-th client. $\{\mathcal{P}_l\}_{l=1}^{K}$ are non-independent and identically distributed (*i.e.*, non-i.i.d.) from each other. At the t-th incremental task, the label space $\mathcal{Y}_l^t \subset \mathcal{Y}^t$ of the l-th local client is a subset of $\mathcal{Y}^t = \cup_{l=1}^{K}\mathcal{Y}_l^t$, and it includes C_l^t new classes ($C_l^t \leq C^t$), different from $C_l^p = \sum_{i=1}^{t-1} C_l^i \subset \cup_{j=1}^{t-1}\mathcal{Y}_l^j$ old classes. After loading $\Theta^{r,t}$ and conducting the local training at the t-th incremental task, S_l can get a locally updated model $\Theta_l^{r,t}$. All locally updated models of selected clients are then uploaded to the global server S_G to be aggregated as the global model $\Theta^{r+1,t}$ of next round. The global server $\mathcal{S}_G$ then distributes parameters $\Theta^{r+1,t}$ to local clients for the next global round.

In the FCIL setting, we divide local clients $\{\mathcal{S}_l\}_{l=1}^{K}$ into three categories (*i.e.*, $\{\mathcal{S}_l\}_{l=1}^{K} = \mathbf{S}_o \cup \mathbf{S}_b \cup \mathbf{S}_n$) in each incremental task. Specifically, $\mathbf{S}_o$ consists of K_o local clients that cannot receive the new data of current task but have the exemplar memory stored via previous learned tasks; $\mathbf{S}_b$ includes K_b clients collecting the new data of current task and the exemplar memory of previous tasks; and $\mathbf{S}_n$ consists of K_n newly-added clients that receive the new data of current task, but have no any exemplar memory of old classes. These clients are dynamically changing as the incremental tasks arriving. Namely, we randomly determine $\{\mathbf{S}_o, \mathbf{S}_b, \mathbf{S}_n\}$ at each global round, and $\mathbf{S}_n$ are irregularly added at any global round in the FCIL. It causes the gradual increase of $K = K_o + K_b + K_n$ in streaming tasks.

Moreover, we have no any prior knowledge about the number of streaming tasks T, data distributions $\{\mathcal{P}_l\}_{l=1}^{K}$, when to collect new classes or add new local clients. The goal of FCIL is to effectively train a global model $\Theta^{R,T}$ to learn new classes consecutively while alleviating the catastrophic forgetting on old classes with the requirement of privacy preservation, via communicating the local model parameters with the global central server $\mathcal{S}_G$.

4. The Proposed GLFC Model

The overview of our model is depicted in Figure 1. To address the FCIL requirements, our model solves local forgetting via a class-aware gradient compensation loss and a class-semantic relation distillation loss (Section 4.1), while tackling global forgetting via a proxy server to select the best old model for local clients (Section 4.2).

4.1. Local Catastrophic Forgetting Compensation

At the t-th incremental task, given the l-th local client $\mathcal{S}_l \in \mathbf{S}_b$ with the training data $\mathcal{T}_l^t$ of new classes and exemplar memory $\mathcal{M}_l$, the classification loss $\mathcal{L}_{\mathrm{CE}}$ for a mini-batch $\{\mathbf{X}_{lb}^t, \mathbf{Y}_{lb}^t\} = \{\mathbf{x}_{li}^t, \mathbf{y}_{li}^t\}_{i=1}^{b} \subset \mathcal{T}_l^t \cup \mathcal{M}_l$ is:

$$\mathcal{L}_{\mathrm{CE}} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{D}_{\mathrm{CE}}(P_l^t(\mathbf{x}_{li}^t, \Theta^{r,t}), \mathbf{y}_{li}^t), \tag{1}$$

where b is the batch size, and $\Theta^{r,t}$ is the classification model at the r-th global round for the t-th task, which is transmitted from global server to local clients. $P_l^t(\mathbf{x}_{li}^t, \Theta^{r,t}) \in \mathbb{R}^{C^p+C^t}$ denotes the sigmoid probability predicted via $\Theta^{r,t}$, and $\mathcal{D}_{\mathrm{CE}}(\cdot,\cdot)$ is the binary cross-entropy loss.

As aforementioned, the class imbalance between old and new classes ($\mathcal{T}_l^t$ and $\mathcal{M}_l$) at the local side enforces the local training to suffer from significant performance degradation (*i.e.*, local catastrophic forgetting) on old classes. To prevent local forgetting, as shown in Figure 1, we develop a class-aware gradient compensation loss and a class-semantic relation distillation loss for local clients, which can correct imbalanced gradient propagation and ensure inter-class semantic consistency across incremental tasks.

• **Class-Aware Gradient Compensation Loss:** After $\mathcal{S}_G$ distributes $\Theta^{r,t}$ to local clients, the class-imbalanced distributions at local side cause imbalanced gradient back-propagation of the last output layer in $\Theta^{r,t}$. It forces the update of local model $\Theta_l^{r,t}$ to perform different learning paces within new classes and different forgetting paces within old classes after local training. This phenomenon heavily worsens the local forgetting on old classes, when new streaming data becomes part of old classes continuously.

As a result, we design a class-aware gradient compensation loss $\mathcal{L}_{\mathrm{GC}}$ to respectively normalize the learning paces of new classes and forgetting paces of old classes via re-weighting their gradient propagation. Specifically, inspired by [45, 46], for a single sample $(\mathbf{x}_{li}^t, \mathbf{y}_{li}^t)$ (its ground-truth

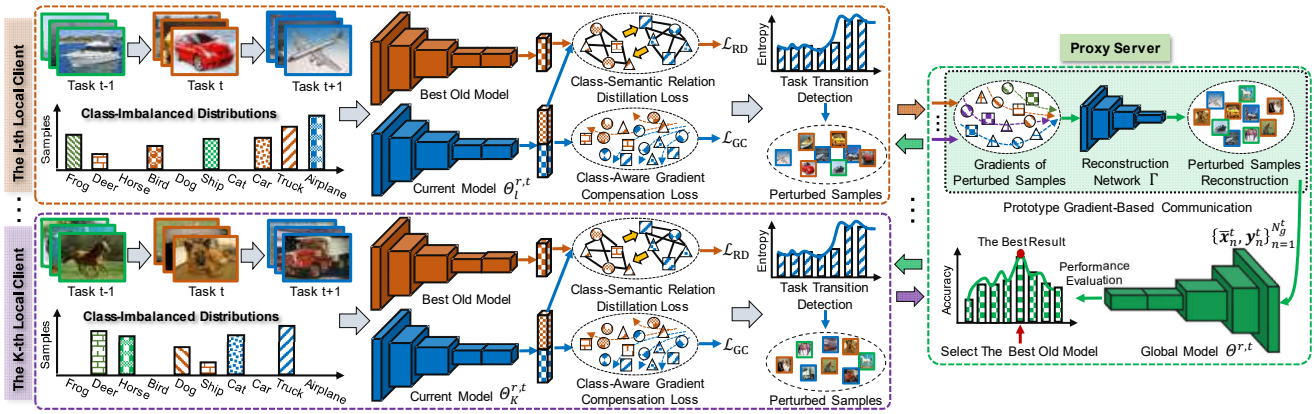

Figure 1. Overview of our GLFC model. It mainly consists of a *class-aware gradient compensation loss* $\mathcal{L}_{\mathrm{GC}}$ and a *class-semantic relation distillation loss* $\mathcal{L}_{\mathrm{RD}}$ to overcome local catastrophic forgetting caused by class imbalance at the local side, and a *proxy server* $\mathcal{S}_P$ to address global catastrophic forgetting brought by non-i.i.d. class imbalance across clients, where a prototype gradient-based communication mechanism between $\mathcal{S}_P$ and clients is developed for their private communication while selecting the best old model for $\mathcal{L}_{\mathrm{RD}}$.

label is y_{li}^t, the one-hot vector of y_{li}^t is $\mathbf{y}_{li}^t$), we obtain a gradient measurement $\mathcal{G}_{li}^t$ with respect to the y_{li}^t-th neuron $\mathcal{N}_{y_{li}^t}^t$ of the last output layer in $\Theta_l^{r,t}$:

$$\begin{aligned}\mathcal{G}_{li}^t &= \frac{\partial \mathcal{D}_{\mathrm{CE}}(P_l^t(\mathbf{x}_{li}^t, \Theta_l^{r,t}), \mathbf{y}_{li}^t)}{\partial \mathcal{N}_{y_{li}^t}^t} \\ &= P_l^t(\mathbf{x}_{li}^t, \Theta_l^{r,t})_{y_{li}^t} - 1,\end{aligned} \tag{2}$$

where $P_l^t(\mathbf{x}_{li}^t, \Theta_l^{r,t})_{y_{li}^t}$ is the y_{li}^t-th softmax probability of the i-th sample.

To normalize the learning paces of new classes and forgetting paces of old classes, we perform separate gradient normalization for old and new classes, and utilize it to re-weight $\mathcal{L}_{\mathrm{CE}}$. Given a mini-batch $\{\mathbf{x}_{li}^t, \mathbf{y}_{li}^t\}_{i=1}^b$, we define

$$\begin{aligned}\mathcal{G}_n &= \frac{1}{\sum_{i=1}^b \mathbb{I}_{\mathbf{y}_{li}^t \in \mathcal{Y}_l^t}} \sum\nolimits_{i=1}^b |\mathcal{G}_{li}^t| \cdot \mathbb{I}_{\mathbf{y}_{li}^t \in \mathcal{Y}_l^t}, \\ \mathcal{G}_o &= \frac{1}{\sum_{i=1}^b \mathbb{I}_{\mathbf{y}_{li}^t \in \cup_{j=1}^{t-1} \mathcal{Y}_l^j}} \sum\nolimits_{i=1}^b |\mathcal{G}_{li}^t| \cdot \mathbb{I}_{\mathbf{y}_{li}^t \in \cup_{j=1}^{t-1} \mathcal{Y}_l^j},\end{aligned} \tag{3}$$

as the gradient means for new and old classes, where $\mathbb{I}_{(\cdot)}$ is the indicator function that if the subscript condition is true, $\mathbb{I}_{(\mathrm{True})} = 1$; otherwise, $\mathbb{I}_{(\mathrm{False})} = 0$. Thus, the re-weighted $\mathcal{L}_{\mathrm{CE}}$ loss is formulated as follows:

$$\mathcal{L}_{\mathrm{GC}} = \frac{1}{b} \sum_{i=1}^b \frac{|\mathcal{G}_{li}^t|}{\bar{\mathcal{G}}_i} \cdot \mathcal{D}_{\mathrm{CE}}(P_l^t(\mathbf{x}_{li}^t, \Theta_l^{r,t}), \mathbf{y}_{li}^t), \tag{4}$$

where $\bar{\mathcal{G}}_i = \mathbb{I}_{\mathbf{y}_{li}^t \in \mathcal{Y}_l^t} \cdot \mathcal{G}_n + \mathbb{I}_{\mathbf{y}_{li}^t \in \cup_{j=1}^{t-1} \mathcal{Y}_l^j} \cdot \mathcal{G}_o$. For instance, when the i-th sample $\mathbf{x}_{li}^t$ belongs to new classes, $\bar{\mathcal{G}}_i$ will be $\mathcal{G}_n$, otherwise $\mathcal{G}_o$.

• **Class-Semantic Relation Distillation Loss:** During the training of local model $\Theta_l^{r,t}$ initialized as the current global model $\Theta^{r,t}$, the probability predicted by $\Theta_l^{r,t}$ indicates the inter-class semantic similarity relations. To ensure the inter-class semantic consistency across different incremental tasks, we design a class-semantic relation distillation loss $\mathcal{L}_{\mathrm{RD}}$ by considering the underlying relations among old and new classes. As depicted in Figure 1, we respectively forward a mini-batch dataset $\{\mathbf{X}_{lb}^t, \mathbf{Y}_{lb}^t\}$ into the stored old model Θ_l^{t-1} and current local model $\Theta_l^{r,t}$, and obtain the corresponding predicted probabilities $P_l^{t-1}(\mathbf{X}_{lb}^t, \Theta_l^{t-1}) \in \mathbb{R}^{b \times C^p}$ of old classes and $P_l^t(\mathbf{X}_{lb}^t, \Theta_l^{r,t}) \in \mathbb{R}^{b \times (C^p + C^t)}$ of old and new classes. These probabilities reflect the inter-class relations between old and new classes. Different from existing knowledge distillation strategies [1, 3, 8, 17, 26] that only ensure old classes' semantic consistency among Θ_l^{t-1} and $\Theta_l^{r,t}$, we consider inter-class relations between old and new classes simultaneously via optimizing $\mathcal{L}_{\mathrm{RD}}$. Namely, we utilize a variant of one-hot encoding labels $\mathbf{Y}_{lb}^t \in \mathbb{R}^{b \times (C^p + C^t)}$ by replacing the first C^p dimensions of $\mathbf{Y}_{lb}^t$ with $P_l^{t-1}(\mathbf{X}_{lb}^t, \Theta_l^{t-1})$, and denote this variant as $Y_l^t(\mathbf{X}_{lb}^t, \Theta_l^{t-1}) \in \mathbb{R}^{b \times (C^p + C^t)}$. Obviously, $Y_l^t(\mathbf{X}_{lb}^t, \Theta_l^{t-1})$ effectively indicates the inter-class semantic similarity relations for both old and new classes via smoothing the one-hot labels. Thus, we formulate $\mathcal{L}_{\mathrm{RD}}$ as follows:

$$\mathcal{L}_{\mathrm{RD}} = \mathcal{D}_{\mathrm{KL}}(P_l^t(\mathbf{X}_{lb}^t, \Theta_l^{r,t}) || Y_l^t(\mathbf{X}_{lb}^t, \Theta_l^{t-1})), \tag{5}$$

where $\mathcal{D}_{\mathrm{KL}}(\cdot||\cdot)$ is the Kullback-Leibler divergence. Overall, the optimization objective for the l-th local client is:

$$\mathcal{L}_l = \lambda_1 \mathcal{L}_{\mathrm{GC}} + \lambda_2 \mathcal{L}_{\mathrm{RD}}, \tag{6}$$

where λ_1, λ_2 are hyper-parameters. We update local model $\Theta_l^{r,t}$ via optimizing Eq. (6), and then aggregate all local models in global server S_G to obtain the global model $\Theta^{r+1,t}$ of the next round. When $t = 1$, there is no old model Θ_l^{t-1} to perform $\mathcal{L}_{\mathrm{RD}}$, and we set $\lambda_1 = 1.0, \lambda_2 = 0$, otherwise, $\lambda_1 = 0.5, \lambda_2 = 0.5$. Note that local clients in $\mathbf{S}_o$ and $\mathbf{S}_n$ have the same objective (*i.e.*, Eq. (6)) with the clients in

$\mathbf{S}_b$, except for the definition of $\bar{\mathcal{G}}_i$ in Eq. (4). $\bar{\mathcal{G}}_i$ is always set as $\mathcal{G}_o$ for $\mathbf{S}_o$ and $\mathcal{G}_n$ for $\mathbf{S}_n$.

• **Task Transition Detection:** When optimizing Eq. (6), it is essential for local clients to know when new classes arrive, then update exemplar memory $\mathcal{M}_l$ and store old classification model Θ_l^{t-1} used for $\mathcal{L}_{\mathrm{RD}}$. However, in the FCIL, we have no prior knowledge about when local clients receive the data of new classes. To tackle this issue, a trivial solution is to identify whether the labels of training data have been seen before. However, it cannot determine if the newly received labels are from new classes or the old classes observed by other local clients due to non-i.i.d. setting of class distributions. Another intuitive solution is to use performance degradation as the signal of collecting new classes. This solution is infeasible in the FCIL, since the random selection of $\{\mathbf{S}_o, \mathbf{S}_b, \mathbf{S}_n\}$ and their non-i.i.d. class distributions can cause sharp performance degradation, even though without receiving new classes.

To this end, we propose a task transition detection mechanism to accurately identify when local clients receive new classes. Specifically, at the r-th global round, every client computes the average entropy $\mathcal{H}_l^{r,t}$ via the received global model $\Theta^{r,t}$ on its current training data $\mathcal{T}_l^t$:

$$\mathcal{H}_l^{r,t} = \frac{1}{N_l^t} \sum_{i=1}^{N_l^t} \mathcal{I}(P_l^t(\mathbf{x}_{li}^t, \Theta^{r,t})), \tag{7}$$

where $\mathcal{I}(\cdot) = \sum_i p_i \log p_i$ is the entropy function. When $\mathcal{H}_l^{r,t}$ encounters a sudden rise and satisfies $\mathcal{H}_l^{r,t} - \mathcal{H}_l^{r-1,t} \geq r_h$, we argue that the local clients are receiving new classes, and update t by $t \leftarrow t+1$. Then they can update memory $\mathcal{M}_l$ and store old model Θ_l^{t-1}. We empirically set $r_h = 1.2$.

4.2. Global Catastrophic Forgetting Compensation

Although Eq. (6) could tackle local catastrophic forgetting brought by the class imbalance at the local side, it cannot address heterogeneous forgetting from other local clients (*i.e.*, global catastrophic forgetting). In other words, the non-i.i.d. class-imbalanced distributions across local clients result in certain global catastrophic forgetting on old classes, worsening the local catastrophic forgetting further. Thus, it is necessary to address the heterogeneous forgetting across clients from the global perspective. As aforementioned, the proposed class-semantic relation distillation loss $\mathcal{L}_{\mathrm{RD}}$ in Eq. (5) requires a stored old classification model Θ_l^{t-1} of previous tasks to distill inter-class relations. A better Θ_l^{t-1} can globally increase the distillation gain from previous tasks, strengthening the memory of old classes in a global view. As a result, the selection of Θ_l^{t-1} plays an important role in global catastrophic forgetting compensation, which should be considered from a global perspective.

However, in the FCIL, it is difficult to select the best Θ_l^{t-1} due to the privacy protection. The intuitive solution is that every client stores its best old model $\{\Theta_l^{t-1}\}_{t=2}^T$ for each task during the $(t-1)$-th task with the training data $\mathcal{T}_l^{t-1}$. Unfortunately, this solution considers the selection of Θ_l^{t-1} from a local perspective, and cannot guarantee the selected Θ_l^{t-1} has the best memory for all old classes, since each local client has only a subset of old classes (non-i.i.d.). To this end, we employ a proxy server $\mathcal{S}_P$ to select the best Θ^{t-1} for all clients from a global perspective, as depicted in Figure 1. Specifically, when local clients have identified new classes (*i.e.*, $\mathcal{T}_l^t$) at the beginning of the t-th task via task transition detection, they will transmit perturbed prototype samples of new classes to $\mathcal{S}_P$ via a prototype gradient-based communication mechanism. After receiving these gradients, $\mathcal{S}_P$ reconstructs the perturbed prototype samples, and utilizes them to monitor the performance of global model $\Theta^{r,t}$ (received from $\mathcal{S}_G$) until the best one is found. When stepping at the next task $(t+1)$, $\mathcal{S}_P$ will distribute the best $\Theta^{r,t}$ to local clients, and local clients regard it as the best old model to perform $\mathcal{L}_{\mathrm{RD}}$.

• **Prototype Gradient-Based Communication:** Given the l-th local client $\mathcal{S}_l \in \mathbf{S}_b \cup \mathbf{S}_n$ that receives the training data $\mathcal{T}_l^t$ of new classes for the t-th task, $\mathcal{S}_l$ identifies new classes via task transition detention. Then $\mathcal{S}_l$ selects only one representative prototype sample $\mathbf{x}_{lc^*}^t$ from $\mathcal{T}_l^t$ for each new class ($c = C_l^p+1, \cdots, C_l^p+C_l^t$), where the feature of $\mathbf{x}_{lc^*}^t$ is closest to the mean embedding of all samples belonging to the c-th class in the latent feature space. We then feed these prototype samples and their labels $\{\mathbf{x}_{lc^*}^t, \mathbf{y}_{lc^*}^t\}_{c=C_l^p+1}^{C_l^p+C_l^t} \subset \mathcal{T}_l^t$ into a L-layer gradient encoding network $\Gamma = \{\mathcal{W}_i\}_{i=1}^L$ to compute the gradient $\nabla\Gamma_{lc}$, where Γ is much shallower than $\Theta_l^{r,t}$ for communication efficiency, and $\mathcal{W}_i$ is the parameters of the i-th layer. The i-th element $\nabla_{\mathcal{W}_i}\Gamma_{lc}$ of $\nabla\Gamma_{lc}$ is defined as $\nabla_{\mathcal{W}_i}\Gamma_{lc} = \nabla_{\mathcal{W}_i}\mathcal{D}_{\mathrm{CE}}(P_l^t(\mathbf{x}_{lc^*}^t, \Gamma), \mathbf{y}_{lc^*}^t)$, where $P_l^t(\mathbf{x}_{lc^*}^t, \Gamma)$ is the probability predicted via Γ. Then $\mathcal{S}_l$ transmit C_l^t gradients $\{\nabla\Gamma_{lc}\}_{c=C_l^p+1}^{C_l^p+C_l^t}$ to $\mathcal{S}_P$ for the reconstruction of prototype samples.

$\mathcal{S}_P$ randomly shuffles all received gradients from selected clients of this global round to construct a gradient pool $\nabla\Gamma^t = \cup_l\{\nabla\Gamma_{lc}\}_{c=C_l^p+1}^{C_l^p+C_l^t}$, and we assume there are N_g^t gradients in this pool. This shuffling operation prevents $\mathcal{S}_P$ from tracking certain selected clients by remarking special gradient distributions. For the n-th element $\nabla\Gamma_n^t$ of $\nabla\Gamma^t$, we can obtain its corresponding ground-truth label y_n^t (with one-hot encoding label $\mathbf{y}_n^t$) via observing the gradient symbol of the last layer in $\nabla\Gamma_n^t$ (proposed in [46, 58]). Given a dummy sample $\bar{\mathbf{x}}_n^t$ initialized by a standard Gaussian $\mathcal{N}(0,1)$, we forward all pairs of $\{\bar{\mathbf{x}}_n^t, \nabla\Gamma_n^t, \mathbf{y}_n^t\}_{n=1}^{N_g^t}$ into $\Gamma = \{\mathcal{W}_i\}_{i=1}^L$ that is same as the gradient encoding network used by local clients, to recover prototype samples for each new class. The reconstruction loss $\mathcal{L}_{\mathrm{RT}}$ and the

Table 1. Performance comparisons between our model and other baseline methods on CIFAR-100 [21] with 10 incremental tasks.

Methods	10	20	30	40	50	60	70	80	90	100	Avg.	△
iCaRL [37] + FL	89.0	55.0	57.0	52.3	50.3	49.3	46.3	41.7	40.3	36.7	51.8	⇑15.1
BiC [49] + FL	88.7	63.3	61.3	56.7	53.0	51.7	48.0	44.0	42.7	40.7	55.0	⇑11.9
PODNet [11] + FL	89.0	71.3	69.0	63.3	59.0	55.3	50.7	48.7	45.3	45.0	59.7	⇑7.2
DDE [17] + iCaRL [37] + FL	88.0	70.0	67.3	62.0	57.3	54.7	50.3	48.3	45.7	44.3	58.8	⇑8.1
GeoDL [43] + iCaRL [37] + FL	87.0	76.0	70.3	64.3	60.7	57.3	54.7	50.3	48.3	46.3	61.5	⇑5.4
SS-IL [1] + FL	88.3	66.3	54.0	54.0	44.7	54.7	50.0	47.7	45.3	44.0	54.9	⇑12.0
Ours-w/oCGC	89.0	80.0	75.0	70.0	63.3	62.0	57.0	54.7	50.3	46.0	60.1	⇑6.8
Ours-w/oCRD	89.0	80.3	76.0	71.0	64.0	65.0	57.7	56.0	51.0	48.3	65.8	⇑1.1
Ours-w/oPRS	88.0	80.3	75.0	70.3	62.0	63.0	58.0	54.3	49.0	45.7	64.6	⇑2.3
Ours	90.0	82.3	77.0	72.3	65.0	66.3	59.7	56.3	50.3	50.0	**66.9**	–

Figure 2. Visualization of original prototype samples (top row), perturbed prototype samples (middle row), and reconstructed prototype samples via proxy server (bottom row) in CIFAR-100 [21].

update of $\bar{\mathbf{x}}_t^n$ are expressed as follows:

$$\mathcal{L}_{\mathrm{RT}} = \sum_{i=1}^{L} \|\nabla_{\mathcal{W}_i} \mathcal{D}_{\mathrm{CE}}(P^t(\bar{\mathbf{x}}_n^t, \Gamma), \mathbf{y}_n^t) - \nabla_{\mathcal{W}_i} \Gamma_n^t\|^2, \quad (8)$$

$$\bar{\mathbf{x}}_n^t \leftarrow \bar{\mathbf{x}}_n^t - \eta \nabla_{\bar{\mathbf{x}}_n^t} \mathcal{L}_{\mathrm{RT}}, \quad (9)$$

where $P^t(\bar{\mathbf{x}}_n^t, \Gamma)$ is the probability predicted via Γ. η denotes the learning rate to update $\bar{\mathbf{x}}_n^t$.

• **Selection of The Best Old Model:** $\mathcal{S}_P$ can only receive gradients from local clients at the first round of the t-th task, when they detect new classes. Then $\mathcal{S}_P$ reconstructs N_g^t prototype samples of new classes and their labels (*i.e.*, $\{\bar{\mathbf{x}}_n^t, \mathbf{y}_n^t\}_{n=1}^{N_g^t}$) via optimizing Eq. (9). At the t-th task, $\mathcal{S}_P$ forwards these reconstructed samples into the global model $\Theta^{r,t}$ (received from $\mathcal{S}_G$) to select the best Θ^t via evaluating which model has the best accuracy, until receiving gradients of new classes from the next task. At every global round starting from the second task, $\mathcal{S}_P$ distributes the best models of the last task and current task (*i.e.*, Θ^{t-1} and Θ^t), to all selected clients. If these selected clients detect new classes from $\mathcal{T}_l^{t+1}$ at the t-th task, they will set Θ^t as the old model Θ_l^{t-1}, otherwise, Θ^{t-1} is set as Θ_l^{t-1} to perform $\mathcal{L}_{\mathrm{RD}}$.

• **Perturbed Prototype Samples Construction:** Although the network Γ is only privately accessible to $\mathcal{S}_P$ and local clients, malicious attackers may steal Γ and these gradients to reconstruct raw prototype sample $\{\mathbf{x}_{lc^*}^t, \mathbf{y}_{lc^*}^t\} \in \mathcal{T}_l^t$ of the l-th local client. To achieve privacy preservation, we propose to add perturbations to these prototype samples. The attackers can get little useful information from the perturbed prototype samples even if they can reconstruct them.

To be specific, given a prototype sample $\{\mathbf{x}_{lc^*}^t, \mathbf{y}_{lc^*}^t\} \in \mathcal{T}_l^t$, we forward it into the local model $\Theta_l^{r,t}$ that has been trained via Eq. (6), and apply back-propagation to update this sample. In order to produce perturbed prototype sample, we introduce a Gaussian noise into the latent feature of prototype sample, and then update $\mathbf{x}_{lc^*}^t$ via Eq. (11):

$$\mathcal{L}_{\mathrm{GP}} = \mathcal{D}_{\mathrm{CE}}(P_l^t(\Phi(\mathbf{x}_{lc^*}^t) + \gamma\mathcal{N}(0, \sigma^2), \Theta_l^{r,t}), \mathbf{y}_{lc^*}^t), \quad (10)$$

$$\mathbf{x}_{lc^*}^t \leftarrow \mathbf{x}_{lc^*}^t - \eta \nabla_{\mathbf{x}_{lc^*}^t} \mathcal{L}_{\mathrm{GP}}, \quad (11)$$

where $\Phi(\mathbf{x}_{lc^*}^t)$ denotes the latent feature of $\mathbf{x}_{lc^*}^t$, and $P_l^t(\Phi(\mathbf{x}_{lc^*}^t) + \gamma\mathcal{N}(0, \sigma^2), \Theta_l^{r,t})$ is the probability predicted via $\Theta_l^{r,t}$ when adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to $\Phi(\mathbf{x}_{lc^*}^t)$. σ^2 represents the variance of features of all samples belonging to $\mathbf{y}_{lc^*}^t$, and we empirically set $\gamma = 0.1$ to control the effect of Gaussian noise in this paper. Some reconstructed prototype samples are visualized in Figure 2.

4.3. Optimization Pipeline of Our GLFC Model

Starting from the first incremental task, all clients are required to compute the average entropy of their private training data via Eq. (7) at the beginning of each global round, and follow iCaRL [37] to update their exemplar memory $\mathcal{M}_l$. For each global training round, the central server $\mathcal{S}_G$ randomly selects a set of local clients to conduct local training. After that, when the selected clients identify new classes via the task transition detection strategy, they will construct perturbed prototype samples of these new classes and share the corresponding gradients to the proxy server $\mathcal{S}_P$ via the prototype gradient-based communication mechanism. After receiving these gradients, $\mathcal{S}_P$ reconstructs these prototype samples, and utilizes them to select the best global model Θ^t until collecting gradients next time. Starting from the second task ($t = 2$), $\mathcal{S}_P$ will distribute best models of the last and current task (*i.e.*, Θ^{t-1}, and Θ^t) to selected clients. Then the l-th client uses Θ^{t-1} as its Θ_l^{t-1} to update the current local model $\Theta_l^{r,t}$ via optimizing Eq. (6), when it doesn't detect new classes via task transition detection. Otherwise, it uses Θ^t to train the current local model $\Theta_l^{r,t}$. Finally, $\mathcal{S}_G$ aggregates the updated local models $\Theta_l^{r,t}$ to get the global model $\Theta^{r+1,t}$ of next ground. The optimization pipeline is provided in supplementary material.

Table 2. Performance comparisons between our model and other baseline methods on ImageNet-Subset [7] with 10 incremental tasks.

Methods	10	20	30	40	50	60	70	80	90	100	Avg.	△
iCaRL [37] + FL	74.0	62.3	56.3	47.7	46.0	40.3	37.7	34.3	33.3	32.7	46.5	⇑10.5
BiC [49] + FL	74.3	63.0	57.7	51.3	48.3	46.0	42.7	37.7	35.3	34.0	49.0	⇑8.0
PODNet [11] + FL	74.3	64.0	59.0	56.7	52.7	50.3	47.0	43.3	40.0	38.3	52.6	⇑4.4
DDE [17] + iCaRL [37] + FL	76.0	57.7	58.0	56.3	53.3	50.7	47.3	44.0	40.7	39.0	52.3	⇑4.7
GeoDL [43] + iCaRL [37] + FL	74.0	63.3	54.7	53.3	50.7	46.7	41.3	39.7	38.3	37.0	50.0	⇑7.0
SS-IL [1] + FL	69.7	60.0	50.3	45.7	41.7	44.3	39.0	38.3	38.0	37.3	46.4	⇑10.6
Ours-w/oCGC	74.0	67.0	61.0	60.0	57.0	53.7	50.0	47.0	42.0	39.3	55.1	⇑1.9
Ours-w/oCRD	76.0	56.0	53.7	45.0	46.0	43.0	42.0	39.3	37.0	35.3	47.3	⇑9.7
Ours-w/oPRS	73.0	64.0	62.3	57.3	54.0	50.3	46.7	43.0	40.0	37.3	52.8	⇑4.2
Ours	73.0	69.3	68.0	61.0	58.3	54.0	51.3	48.0	44.3	42.7	**57.0**	–

Table 3. Comparisons of the first 10 tasks on TinyImageNet [35] with 20 tasks, where the rest comparisons are in supplementary material.

Methods	10	20	30	40	50	60	70	80	90	100	Avg.	△
iCaRL [37] + FL	67.0	59.3	54.0	48.3	46.7	44.7	43.3	39.0	37.3	33.0	47.3	⇑7.6
BiC [49] + FL	67.3	59.7	54.7	50.0	48.3	45.3	43.0	40.7	38.0	33.7	48.1	⇑6.8
PODNet [11] + FL	69.0	59.3	55.0	51.7	50.0	46.7	43.7	41.0	39.3	38.0	49.4	⇑5.5
DDE [17] + iCaRL [37] + FL	70.0	59.3	53.3	51.0	48.3	45.7	42.3	40.0	38.0	36.3	48.4	⇑6.5
GeoDL [43] + iCaRL [37] + FL	66.3	56.7	51.0	49.7	44.7	42.3	41.0	39.0	37.3	35.0	46.3	⇑8.6
SS-IL [1] + FL	66.7	54.0	47.7	45.3	42.3	42.0	40.7	38.0	36.0	34.3	44.7	⇑10.2
Ours-w/oCGC	67.7	60.3	57.7	55.0	51.0	49.0	48.0	45.7	44.3	42.0	52.1	⇑2.8
Ours-w/oCRD	68.0	60.0	53.0	47.3	42.0	39.0	37.3	35.3	33.7	32.0	44.8	⇑10.1
Ours-w/oPRS	67.3	59.7	55.0	51.3	50.7	48.0	46.3	43.3	41.7	40.3	50.3	⇑4.6
Ours	68.7	63.3	61.7	57.3	56.0	53.0	50.3	47.7	46.3	45.0	**54.9**	–

5. Experiments

5.1. Implementation Details

We use three datasets: CIFAR-100 [13, 21], ImageNet-Subset [7], and TinyImageNet [35] in our experiments. For a fair comparison with baseline class-incremental learning methods [1, 11, 17, 37, 43, 49] in the FCIL setting, we follow the same protocols proposed by [37, 49] to set incremental tasks, utilize the identical class order generated from iCaRL [37], and employ the same backbone (*i.e.*, ResNet-18 [15]) as the classification model [2]. The SGD optimizer whose learning rate is 2.0 is used to train all models. The exemplar memory $\mathcal{M}_l$ of each client is set as 2,000 for all streaming tasks. A shallow LeNet [23] with only 4 layers is used as the network Γ. We employ a SGD optimizer with the learning rate as 0.1 to construct perturbed samples for local clients, while utilizing a L-BFGS optimizer with the learning rate as 1.0 to reconstruct prototype samples for proxy server. We initialize the number of local clients as 30 in the first incremental task, introduce 10 additional new local clients as the learning tasks arrive consecutively. At each global round, we randomly select 10 clients to conduct 20-epoch local training. Each client randomly receives 60% classes from the label space of its seen task. We run our experiments for 3 times with 3 random seeds (2021, 2022, 2023) and report the averaged results. Please refer to more details in supplementary material.

5.2. Performance Comparison

This section shows comparison experiments to illustrate the effectiveness of our GLFC model, as shown in Tables 1, 2, 3, where '△' denotes the improvements of our model compared with other comparison methods. We observe that our model outperforms existing class-incremental methods [1, 11, 17, 37, 43, 49] in the FCIL setting by a margin of 4.4%∼15.1% in terms of average accuracy. It validates that our model could enable local clients to collaboratively train a global class-incremental model. Moreover, our model has stable performance improvement in comparison with other methods for all incremental tasks, which verifies the effectiveness of our model to address the forgetting in FCIL.

5.3. Ablation Studies

As shown in Tables 1, 2, 3, we investigate the effects of each module in our model via ablation studies. Ours-w/oCGC, Ours-w/oCRD and Ours-w/oPRS denote the performance of our model without using $\mathcal{L}_{\mathrm{GC}}$, $\mathcal{L}_{\mathrm{RD}}$ and the proxy server $\mathcal{S}_P$, where Ours-w/oCGC and Ours-w/oCRD utilize $\mathcal{L}_{\mathrm{CE}}$ and the knowledge distillation proposed in iCaRL [37] for a replacement. Compared with Ours, the performance of Ours-w/oCGC, Ours-w/oCRD and Ours-w/oPRS degrades evidently with a range of 1.1%∼10.1%. It verifies the effectiveness of all modules to cooperate together. The ablation performance verifies all modules are essential to train a global class-incremental model. The proxy server is also essential to select the best old model via evaluating reconstructed samples (shown in Figure 2).

5.4. Qualitative Analysis of Incremental Tasks

In this section, we conduct qualitative analysis of various incremental tasks ($T = 5, 10, 20$) on benchmark datasets to validate the superior performance of GLFC, as shown

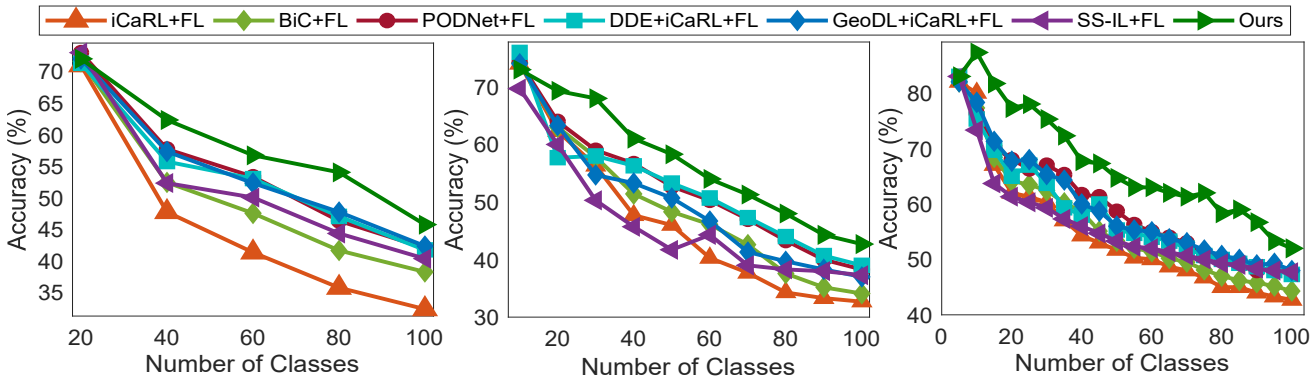

Figure 3. Qualitative analysis of different incremental tasks on CIFAR-100 [21] when $T = 5$ (left), $T = 10$ (middle) and $T = 20$ (right).

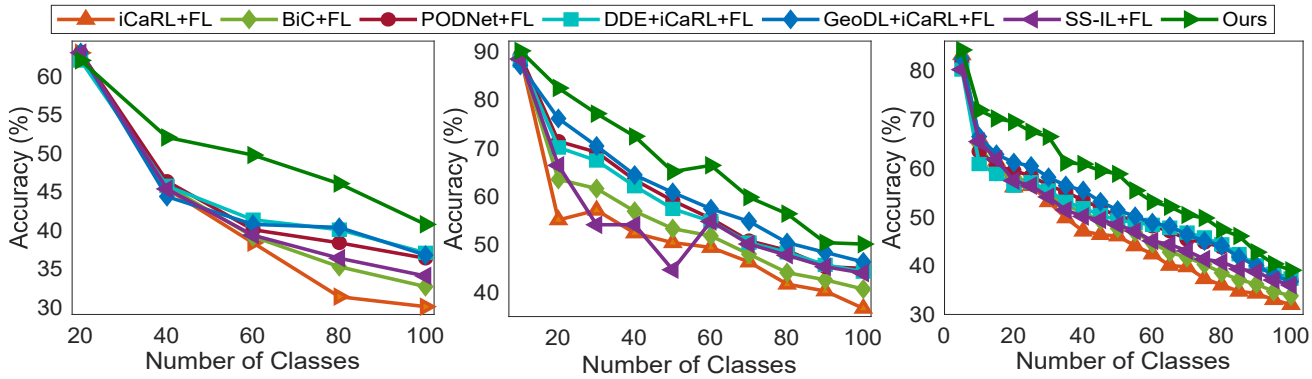

Figure 4. Qualitative analysis of incremental tasks on ImageNet-Subset [7] when $T = 5$ (left), $T = 10$ (middle) and $T = 20$ (right).

Table 4. Qualitative analysis of different exemplar memories in local clients on CIFAR-100 [21] when $T = 5$.

$\mathcal{M}_l$	20	40	60	80	100	Avg.
500	71.3	52.7	47.0	45.0	38.3	50.9
1000	71.0	56.3	52.7	50.0	42.3	54.5
1500	71.0	58.0	54.3	53.0	44.0	56.1
2000	72.0	62.3	56.7	54.0	45.7	58.1

in Figures 3, 4. According to these curves, we can easily observe that our model performs better than other competing baseline methods [1, 11, 17, 37, 43, 49] for all incremental tasks, in the settings with different number of tasks ($T = 5, 10, 20$). It demonstrates that the GLFC model enables multiple local clients to learn new classes in a streaming manner while addressing local and global forgetting.

5.5. Qualitative Analysis of Exemplar Memory

As shown in Table 4, we study the effects of different exemplar memories on the performance of our GLFC model, by respectively setting $\mathcal{M}_l$ as $\{500, 1000, 1500, 2000\}$ on CIFAR-100 [21]. From the results in Table 4, we can conclude that the better performance of GLFC model on learning new classes in a streaming manner will be, the larger size of exemplar memory $\mathcal{M}_l$ is. It validates that storing more training data of old classes at the local side could strengthen the memory ability of our proposed GLFC model for old classes. Moreover, the presented results also illustrate the effectiveness of our GLFC model about identifying new classes via the task transition detection mechanism and updating the exemplar memory.

6. Conclusion

In this paper, we propose a practical Federated Class-Incremental Learning (FCIL) problem, and develop a novel Global-Local Forgetting Compensation (GLFC) model to address the local and global catastrophic forgetting in FCIL. Specifically, a class-aware gradient compensation loss and a class-semantic relation distillation loss are designed to locally address the catastrophic forgetting, by correcting the imbalanced gradient propagation and ensuring consistent inter-class relations across tasks. We also employ a proxy server to tackle global forgetting by selecting the best old model for preserving the memory of old classes. Extensive experiments on representative benchmark datasets demonstrate the effectiveness of our proposed GLFC model.

Acknowledgments

This work was partially supported by National Nature Science Foundation of China under Grant 62003336; National Science Foundation of US under Grants 1834701, 2016240; and research awards from Facebook, Google, PlatON Network, and General Motors.

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *ICCV*, pages 844–853, October 2021. 2, 4, 6, 7, 8

[2] Chen Chen, Haobo Wang, Weiwei Liu, Xingyuan Zhao, Tianlei Hu, and Gang Chen. Two-stage label embedding via neural factorization machine for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3304–3311, 2019. 7

[3] Feilong Chen, Xiuyi Chen, Can Xu, and Daxin Jiang. Learning to ground visual objects for visual dialog. *arXiv preprint arXiv:2109.06013*, 2021. 4

[4] Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4229–4238, 2020. 1, 2

[5] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. Denoised maximum classifier discrepancy for source-free unsupervised domain adaptation. *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, 2022. 2

[6] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, 27(10):1735–1743, 2021. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 7, 8

[8] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4091–4101, 2021. 4

[9] Jiahua Dong, Yang Cong, Gan Sun, Zhen Fang, and Zhengming Ding. Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2

[10] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4022–4031, June 2020. 1

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020. 2, 6, 7, 8

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, volume 33, pages 3557–3568, 2020. 1, 2

[13] Zhen Fang, Jie Lu, Anjin Liu, Feng Liu, and Guangquan Zhang. Learning bounds for open-set learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 3122–3132. PMLR, 2021. 7

[14] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *ICML*, 2020. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[16] Junyuan Hong, Zhuangdi Zhu, Shuyang Yu, Zhangyang Wang, Hiroko H Dodge, and Jiayu Zhou. Federated adversarial debiasing for fair and transferable representations. In *KDD*, pages 617–627, 2021. 1

[17] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *CVPR*, 2021. 2, 4, 6, 7, 8

[18] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *ICML*, pages 5132–5143, 2020. 1, 2

[19] Jong-Yeong Kim and Dong-Wan Choi. Split-and-bridge: Adaptable class incremental learning within a single neural network. In *AAAI*, pages 8137–8145, 2021. 2

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 2

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, pages 32–33, 2009. 6, 7, 8

[22] Matthias De Lange, Xu Jia, Sarah Parisot, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Unsupervised model personalization while preserving privacy and scalability: An open problem. In *CVPR*, 2020. 1, 2

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[24] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *ICLR*, 2021. 2

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, pages 614–629, 2016. 2

[26] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML*, volume 119, pages 6316–6326. PMLR, 2020. 4

[27] Yahao Liu, Jihong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2021. 2

[28] Yahao Liu, Jinhong Deng, Jiale Tao, Tong Chu, Lixin Duan, and Wen Li. Undoing the damage of label shift for cross-domain semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022. 2

[29] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021. 2

[30] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, pages 12245–12254, 2020. 2, 3

[31] Lingjuan Lyu and Chen Chen. A novel attribute reconstruction attack in federated learning. *arXiv preprint arXiv:2108.06910*, 2021. 2

[32] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016. 2

[33] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Minh Hoang Dang, N. Asokan, and Ahmad-Reza Sadeghi. Dïot: A crowdsourced self-learning approach for detecting compromised iot devices. *arXiv preprint arXiv:1804.07474*, 2018. 1

[34] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 2

[35] Pouransari Pouransari and Saman Ghili. Tiny imagenet visual recognition challenge. *CS231N course, Stanford Univ., Stanford, CA, USA*, 2015. 7

[36] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019. 1

[37] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2, 3, 6, 7, 8

[38] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 2

[39] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Mérouane Debbah. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Transactions on Communications*, 68(2):1146–1159, 2020. 1

[40] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, page 2994–3003, 2017. 1, 2

[41] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017. 2, 3

[42] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019. 1, 2

[43] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *CVPR*, 2021. 2, 3, 6, 7, 8

[44] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020. 2

[45] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Eavesdrop the composition proportion of training labels in federated learning. *arXiv preprint arXiv:1910.06044*, 2019. 3

[46] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. In *AAAI*, volume 35, pages 10165–10173, 2021. 1, 3, 5

[47] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *International Conference on Learning Representations*, 2022. 1

[48] Chenshen Wu, Luis Herranz, Xialei Liu, yaxing wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: Learning to generate new categories without forgetting. In *NeurIPS*, pages 5962–5972. 2018. 2

[49] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 1, 2, 3, 6, 7, 8

[50] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021. 2

[51] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *ICLR*, 2021. 2

[52] Qian Yang, Jianyi Zhang, Weituo Hao, Gregory Spell, and Lawrence Carin. FLOP: federated learning on medical datasets using partial networks. In *KDD*, 2021. 1

[53] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *ICML*, pages 7252–7261, 2019. 2

[54] Duzhen Zhang, Xiuyi Chen, Shuang Xu, and Bo Xu. Knowledge aware emotion recognition in textual conversations via multi-task incremental transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4429–4440, 2020. 2

[55] Xinbang Zhang, Jianlong Chang, Yiwen Guo, Gaofeng Meng, Shiming Xiang, Zhouchen Lin, and Chunhong Pan. DATA: differentiable architecture approximation with distribution guided sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(9):2905–2920, 2021. 2

[56] Xinbang Zhang, Zehao Huang, Naiyan Wang, Shiming Xiang, and Chunhong Pan. You only search once: Single shot neural architecture search via direct sparse optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(9):2891–2904, 2021. 2

[57] Yiyang Zhang, Feng Liu, Zhen Fang, Bo Yuan, Guangquan Zhang, and Jie Lu. Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation. In Christian Bessiere, editor, *IJCAI*, pages 2526–2532. ijcai.org, 2020. 2

[58] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 5