A. Implementation Details

A.1. Datasets

CIFAR-100 [21] is composed of 60,000 color images from 100 classes. Each class has 500 training and 100 evaluation samples with the size 32×32.

ImageNet-Subset [7] is a subset of ImageNet [7], and it includes 100 classes sampled in the same way as [17]. We split each class into 500 training and 100 test samples with the size 224×224.

TinyImageNet [35] includes 100,000 samples for 200 classes, and each sample is downsized to 64×64. Each class has 500 training and 50 test samples.

A.2. Experimental Settings

For the federated learning setting, in the first task, there are total 30 local clients, and for each global round, we randomly select 10 clients to conduct 20-epoch local training. After the local training, these clients will share their updated models to participate in the global aggregation of this round. When the number of streaming tasks is $T = 10$, for CIFAR-100 and ImageNet-Subset, each task includes 10 new classes for 10 global rounds, and each task transition will introduce 10 additional new clients. For TinyImageNet, each task includes 20 new classes for the same 10 global rounds, and each task transition also includes 10 new clients. Therefore, in the last task of $T = 10$, the total number of local clients is 120, and we also randomly select 10 clients to perform global aggregation. For the case of $T = 20$, each task has 5 classes with 10 global rounds of training for CIFAR-100 and ImageNet-Subset, and the number of classes will be 10 for TinyImageNet. Note that the number of newly introduced clients is 5 now at each task transition. We also conduct experiments of $T = 5$, CIFAR-100 and ImageNet-Subset contain 20 classes for each task, while TinyImageNet has 40 classes per task. For all three datasets, each task covers 20 global rounds and there will be 20 new clients joining in the framework at each task transition. For building the non-i.i.d. setting, every client can only own 60% classes of the label space in the current task, and these classes are randomly selected. During the task transition global round, we assume 90% existing clients are from $\mathcal{S}_b$, while the resting 10% clients are from $\mathcal{S}_o$.

For fair comparisons with other class-incremental learning methods in the FCIL setting, we follow the same protocols proposed by [37, 49] to split classes into incremental tasks, utilize the identical class order generated from iCaRL [37]. Moreover, we use ResNet-18 as the classification model. As for the gradient encoding network, we use a shallow LeNet with only 4 layers. We use a SGD optimizer whose initial learning rate is 2.0 to train all classification models, and the learning rate is divided by 5, 25 and 125 when the accumulated local epochs of a task hit 100, 150 and 180, respectively. What's more, the optimizer of back-propagation perturbation generation is SGD with learning rate as 0.1, and the sample reconstruction optimization happened on the proxy server utilizes L-BFGS with learning rate as 1.0 for saving memory storage. The batch size is 128, and the exemplar memory $\mathcal{M}_l$ of each client has the sample size of 2,000 during all streaming tasks. As for the optimization iterations, the prototype perturbation generation has 100 iterations, and prototype sample reconstruction conducts 200 iterations for each gradient. We repeatedly run our experiments for three times with three random seeds (2021, 2022, 2023) and report the average results in our comparison experiments.

A.3. Comparison Methods

This paper is the first exploration to address the federated class-incremental learning (FCIL) problem, and there is not any baseline method that is built on similar settings. Therefore, for fair comparisons, we compare our GLFC model with several state-of-the-art class-incremental methods (*i.e.*, iCaRL [37], BiC [49], PODNet [11], DDE [17], GeoDL [43] and SS-IL [1] under the federated learning (FL) settings, to validate the effectiveness of our proposed GLFC model. Besides, top-1 accuracy metric is employed to evaluate the performance of other comparison methods and our proposed GLFC model.

B. 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 can use 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 detailed optimization pipeline is provided in Algorithm 1.

Algorithm 1: Optimization Pipeline of Our Model.

Given: At the r-th global round and the t-th task (assume $t \geq 2$), central server $\mathcal{S}_G$ randomly selects a set of clients $\{\mathcal{S}_{s_1}, \mathcal{S}_{s_2}, ..., \mathcal{S}_{s_m}\}$ with size as m; The selected clients have their local training data $\{\mathcal{T}_{s_1}^t, \mathcal{T}_{s_2}^t, \cdots, \mathcal{T}_{s_m}^t\}$ and local exemplar memories $\{\mathcal{M}_{s_1}, \mathcal{M}_{s_2}, \cdots, \mathcal{M}_{s_m}\}$; $\mathcal{S}_G$ sends the latest global classification model $\Theta^{r,t}$ to all selected clients; The gradient encoding model Γ and the proxy server $\mathcal{S}_P$;

All Clients:

for $\mathcal{S}_l$ *in* $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_K\}$ **do**
- Use $\mathcal{T}_l^t$ to compute average entropy $\mathcal{H}_l^{r,t}$ via Eq. (7);
- Apply iCaRL [37] on $\mathcal{T}_l^t$ to update $\mathcal{M}_l$;

Selected Clients:

Receive $\Theta^{r,t}$ from $\mathcal{S}_G$ as local classification model;

Receive Θ^{t-1}, Θ^t from $\mathcal{S}_P$;

for $\mathcal{S}_l$ *in* $\{\mathcal{S}_{s_1}, \mathcal{S}_{s_2}, \cdots, \mathcal{S}_{s_m}\}$ **do**
- *Task* = False;
- **if** $\mathcal{H}_l^{r,t} - \mathcal{H}_l^{r-1,t} \geq r_h$ **then**
  - *Task* = True;
- **if** *Task* = True **then**
  - $\Theta_l^{t-1} = \Theta^t$;
- **else**
  - $\Theta_l^{t-1} = \Theta^{t-1}$;
- **for** $\{\mathbf{X}_{lb}^t, \mathbf{Y}_{lb}^t\}$ *in* $\mathcal{T}_l^t \cup \mathcal{M}_l$ **do**
  - Update local model $\Theta_l^{r,t}$ via optimizing Eq. (6);
- **if** *Task* = True **then**
  - $\nabla\Gamma_l^t = \{\}$;
  - **for** c *in* $[C_l^p + 1, C_l^p + C_l^t]$ **do**
    - Generate perturbed sample $(\mathbf{x}_{lc^*}^t, \mathbf{y}_{lc^*}^t)$;
    - Compute gradient $\nabla\Gamma_{lc} = \cup_{\mathcal{W}_i} \nabla_{\mathcal{W}_i} \mathcal{D}_{\mathrm{CE}}(P_l^t(\mathbf{x}_{lc^*}^t, \Gamma), \mathbf{y}_{lc^*}^t)$;
    - $\nabla\Gamma_l^t \leftarrow \nabla\Gamma_l^t \cup \nabla\Gamma_{lc}$
  - Send $\nabla\Gamma_l^t$ to the proxy server $\mathcal{S}_P$;

Proxy Server:

Receive $\nabla\Gamma^t = \{\nabla\Gamma_{s_1}^t, \nabla\Gamma_{s_2}^t, ..., \nabla\Gamma_{s_m}^t\}$ from selected local clients, and there are N_g^t gradients in $\nabla\Gamma^t$;

Receive $\Theta^{r,t}$ from $\mathcal{S}_G$ as local classification model;

if $N_g^t \neq 0$ **then**
- Shuffle the gradient pool $\nabla\Gamma^t$;
- $\{\bar{\mathbf{X}}_P^t, \mathbf{Y}_P^t\} = \{\}$;
- **for** $n = 1, \cdots, N_g^t$ **do**
  - Reconstruct $\bar{\mathbf{x}}_n^t$ via optimizing Eq. (9);
  - $\{\bar{\mathbf{X}}_P^t, \mathbf{Y}_P^t\} \leftarrow \{\bar{\mathbf{X}}_P^t, \mathbf{Y}_P^t\} \cup (\bar{\mathbf{x}}_n^t, \mathbf{y}_n^t)$;

Forward $\{\bar{\mathbf{X}}_P^t, \mathbf{Y}_P^t\}$ to $\Theta^{r,t}$ and get the best Θ^t;

Distribute Θ^{t-1} and Θ^t to all selected local clients;

C. Experiments on TinyImageNet Dataset

C.1. Performance Comparison

As shown in Tables 5, 6, we present comparison experiments between our model and other baseline class-incremental learning methods on TinyImageNet dataset. The presented results show that our GLFC model significantly outperforms other state-of-the-art comparison methods by 4.7%∼11.0% in terms of average accuracy. It illustrates the effectiveness of our model to address both local and global catastrophic forgetting in the FCIL setting. Moreover, the performance of our model is the best among all incremental tasks, and there is a large performance improvement for each incremental task. This phenomenon validates that the proposed proxy server is effective to address global catastrophic forgetting brought by non-i.i.d. class imbalance across clients via prototype sample construction mechanism. Meanwhile, the proposed class-aware gradient compensation loss and class-semantic relation distillation loss guarantee that our model could effectively alleviate local catastrophic forgetting at local client side.

C.2. Ablation Studies

This subsection investigates the effectiveness of different variants of our model on TinyImageNet dataset, as presented in Tables 5, 6. When compared with Ours, Ours-w/oCGC degrades the performance of 2.7%∼2.8% in terms of average accuracy, which validates the effectiveness of the class-aware gradient compensation loss to compensate imbalanced gradient propagation. We observe that Ours performs better than Ours-w/oCRD by 10.1%∼10.2% in terms of average accuracy. The class-semantic relation distillation loss ensures inter-class semantic consistency across different incremental tasks to address local catastrophic forgetting. Moreover, the performance of Ours-w/oPRS is worse than Ours by 3.2%∼4.6% in terms of average accuracy. This performance degradation verifies that global catastrophic forgetting brought by non-i.i.d. class imbalance across clients could be effectively addressed via the proxy server. All proposed modules in our GLFC model could cooperate well to address the FCIL problem. When any one of proposed components is removed, as shown in Tables 5, 6, Ours-w/oCGC, Ours-w/oCRD and Ours-w/oPRS achieve significant performance degradation.

C.3. Effects of Incremental Tasks

As presented in Tables 7, 8, in this subsection, we introduce the qualitative analysis of various incremental tasks ($T = 5, 10$) on TinyImageNet dataset to validate the effectiveness of the proposed GLFC model. From the results in Tables 7, 8, we observe that the performance of our proposed model has a large improvement (3.2%∼10.0% in terms of average accuracy) over other state-of-the-art comparison methods for all incremental tasks. Even though there are different settings with different number of tasks ($T = 5, 10$), our proposed GLFC model still has the best performance, which verifies that our model could effectively tackle both local and global catastrophic forgetting in

Table 5. Comparisons of the first 10 tasks between our model and other baseline methods on TinyImageNet [35] with 20 incremental tasks.

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**	–

Table 6. Comparisons of the last 10 tasks between our model and other baseline methods on TinyImageNet [35] with 20 incremental tasks.

Methods	110	120	130	140	150	160	170	180	190	200	Avg.	△
iCaRL [37] + FL	32.0	30.3	28.0	27.0	26.3	25.3	24.7	24.0	22.7	22.0	26.2	⇑11.0
BiC [49] + FL	32.7	32.3	30.3	29.0	27.7	27.3	26.0	25.7	24.3	23.3	27.9	⇑9.3
PODNet [11] + FL	37.0	35.7	34.7	34.0	33.0	32.3	31.0	30.0	29.3	28.0	32.5	⇑4.7
DDE [17] + iCaRL [37] + FL	35.0	33.7	32.0	31.0	30.3	30.0	28.7	28.3	27.3	26.0	30.2	⇑7.0
GeoDL [43] + iCaRL [37] + FL	33.7	32.0	31.0	30.3	28.7	28.0	27.3	26.3	25.0	24.7	28.7	⇑8.5
SS-IL [1] + FL	33.0	31.0	29.3	28.3	27.7	27.0	26.3	26.0	25.0	24.3	27.8	⇑9.4
Ours-w/oCGC	40.7	38.3	37.3	36.0	35.3	33.7	33.0	31.7	30.3	29.0	34.5	⇑2.7
Ours-w/oCRD	30.7	29.7	29.3	28.0	27.7	27.0	25.7	25.0	24.0	22.7	27.0	⇑10.2
Ours-w/oPRS	39.0	38.0	37.3	36.3	34.7	33.3	31.7	31.0	30.3	28.7	34.0	⇑3.2
Ours	42.7	41.0	40.0	39.3	38.0	36.7	35.3	34.0	33.0	31.7	**37.2**	–

Table 7. Performance comparisons between our model and other baseline methods on TinyImageNet [35] with 10 incremental tasks.

Methods	20	40	60	80	100	120	140	160	180	200	Avg.	△
iCaRL [37] + FL	63.0	53.0	48.0	41.7	38.0	36.0	33.3	30.7	29.7	28.0	40.1	⇑7.8
BiC [49] + FL	65.3	52.7	49.3	46.0	40.3	38.3	35.7	33.0	31.7	29.0	42.1	⇑5.8
PODNet [11] + FL	66.7	53.3	50.0	47.3	43.7	42.7	40.0	37.3	33.7	31.3	44.6	⇑3.3
DDE [17] + iCaRL [37] + FL	69.0	52.0	50.7	47.0	43.3	42.0	39.3	37.0	33.0	31.3	44.5	⇑3.4
GeoDL [43] + iCaRL [37] + FL	66.3	54.3	52.0	48.7	45.0	42.0	39.3	36.0	32.7	30.0	44.6	⇑3.3
SS-IL [1] + FL	62.0	48.7	40.0	38.0	37.0	35.0	32.3	30.3	28.7	27.0	37.9	⇑10.0
Ours	66.0	58.3	55.3	51.0	47.7	45.3	43.0	40.0	37.3	35.0	**47.9**	–

the FCIL setting. Moreover, the significant performance improvement illustrates that our model enables multiple local clients to learn new classes consecutively, while addressing catastrophic forgetting on old learned classes under the privacy preservation and limited memory of local clients.

D. Qualitative Analysis of Exemplar Memory

In this subsection, as shown in Table 9, we further conduct extensive experiments ($T = 10$) on CIFAR-100 dataset to investigate the effects of different exemplar memories on the performance of our proposed GLFC model when setting $\mathcal{M}_l$ as $\{500, 1000, 1500, 2000\}$. From the presented results in Table 9, we easily observe that our model achieves the better performance for all incremental tasks, when local clients have large memory storage to store the exemplar samples of old classes. Moreover, storing more training data of old classes at the local side could promote the memory replay on old classes, which further addresses catastrophic forgetting at local clients' side for old classes. Besides, it validates that our proposed model is efficient to distinguish new classes via the task transition detection strategy and update the corresponding exemplar memory $\mathcal{M}_l$ at local side. The updated exemplar memory plays an essential role in tackling local catastrophic forgetting on old classes.

E. Limitation and Societal Impact

This section discusses the limitation for our proposed model and the potential societal impact of this paper.

E.1. Limitation

This paper mainly focuses on addressing Federated Class-Incremental Learning (FCIL) problem from the algorithm perspective. In the future, it is necessary to develop mathematical theoretical supports for understanding the FCIL problem and the proposed GLFC model. A possible way to develop mathematical theories for our model is considering existing mathematical explanations of federated learning (FL) and class-incremental learning (CIL)

Table 8. Performance comparisons between our model and other baseline methods on TinyImageNet [35] with 5 incremental tasks.

Methods	40	80	120	160	200	Avg.	△
iCaRL [37] + FL	65.0	48.0	42.7	38.7	35.0	45.9	⇑5.2
BiC [49] + FL	65.7	48.7	43.0	40.3	35.7	46.7	⇑4.4
PODNet [11] + FL	66.0	50.3	44.7	41.3	37.0	47.9	⇑3.2
DDE [17] + iCaRL [37] + FL	63.0	51.3	45.3	41.0	36.0	47.3	⇑3.8
GeoDL [43] + iCaRL [37] + FL	65.3	50.0	45.0	40.7	36.0	47.4	⇑3.7
SS-IL [1] + FL	65.0	42.3	38.3	35.0	30.3	42.2	⇑8.9
Ours	66.0	55.3	49.0	45.0	40.3	**51.1**	–

Table 9. Qualitative analysis of different exemplar memories in local clients on CIFAR-100 [21] when $T = 10$.

$\mathcal{M}_l$	10	20	30	40	50	60	70	80	90	100	Avg.
500	90.0	74.3	66.0	58.3	52.0	51.0	43.0	42.0	40.0	39.3	55.6
1000	89.0	78.3	72.0	64.3	59.7	59.0	52.3	49.3	48.7	47.7	62.0
1500	89.3	82.0	76.0	70.0	64.0	64.0	56.3	52.7	49.3	48.7	65.2
2000	90.0	82.3	77.0	72.3	65.0	66.3	59.7	56.3	50.3	50.0	**66.9**

simultaneously. However, as we know, there is rare theoretical analysis about CIL. Therefore, it might be tough to propose a brand-new theoretical support to analyze regular CIL problem. Instead, we will try to establish a theoretical analysis for the FCIL problem from a FL perspective in the future work.

E.2. Potential Societal Impact

The FCIL problem discussed in our paper doesn't have any negative societal impact. On the contrary, we believe our work can solve real-world problems and bring about extensive benefits. In comparison with standard federated learning (FL), the proposed Federated Class-Incremental Learning (FCIL) is more practical as we assume the data of new classes as well as new clients will indiscriminately and continuously participate in FCIL. To solve the FCIL problem, our proposed GLFC model can enable a global class-incremental learning model to be trained on decentralized devices without data sharing (uploading decentralized data to a central server or data exchange between participated devices). Compared to regular class-incremental learning methods that always need access to the training data, FCIL can protect the private information of participants by remaining the local data where it is collected.

We have faith that the proposed GLFC model can bring beneficial gains to a number of information-sensitive scenarios, such as medical diagnosis, smartphone applications, pharmaceutical companies, and high-technology enterprises, etc. In summary, this work is the first attempt to learn a global class-incremental model in the setting of FL, which expedites the development of FL-based applications with the requirement of privacy preservation.