# Towards Efficient Replay in Federated Incremental Learning

Yichen Li
Huazhong University of Science and Technology, China
ycli0204@hust.edu.cn

Qunwei Li
Ant Group, China
qunwei.qw@antgroup.com

Haozhao Wang
Huazhong University of Science and Technology, China
hz_wang@hust.edu.cn

Ruixuan Li*
Huazhong University of Science and Technology, China
rxli@hust.edu.cn

Wenliang Zhong
Ant Group, China
yice.zwl@antgroup.com

Guannan Zhang
Ant Group, China
zgn138592@antgroup.com

## Abstract

*In Federated Learning (FL), the data in each client is typically assumed fixed or static. However, data often comes in an incremental manner in real-world applications, where the data domain may increase dynamically. In this work, we study catastrophic forgetting with data heterogeneity in Federated Incremental Learning (FIL) scenarios where edge clients may lack enough storage space to retain full data. We propose to employ a simple, generic framework for FIL named Re-Fed, which can coordinate each client to cache important samples for replay. More specifically, when a new task arrives, each client first caches selected previous samples based on their global and local importance. Then, the client trains the local model with both the cached samples and the samples from the new task. Theoretically, we analyze the ability of Re-Fed to discover important samples for replay thus alleviating the catastrophic forgetting problem. Moreover, we empirically show that Re-Fed achieves competitive performance compared to state-of-the-art methods.*

## 1. Introduction

Federated learning (FL) is a distributed framework that allows multiple edge clients to learn a unified deep learning model cooperatively while preserving the data privacy of the local clients [23, 31, 44]. Recently, FL has attracted growing attention and been applied to various fields such as recommendation systems [7, 24] and smart healthcare [35, 46].

Typically, FL has been actively studied in a static setting, where the number of training samples does not change over time. However, in a realistic FL application, each client may continue collecting new data. It is difficult to learn new data while retaining previous information in machine learning due to the notorious phenomenon known as catastrophic forgetting [9], leading to performance degradation on previous tasks. This challenge is further compounded in FL settings, where the data in a client remains inaccessible to other ones and clients lack enough storage to retain full previous samples.

To address this issue, researchers have studied federated incremental learning (FIL), which enables each client to continuously learn from a local private and incremental task stream [4, 20]. The authors in [48] aim to personalize the models for each client by decomposing the model parameters into shared parameters and adaptive parameters, facilitating the transfer of common knowledge across similar tasks among clients. FCIL is proposed in [6] which specifically focuses on the federated class-incremental learning scenario and a global model is developed by incorporating additional class-imbalance losses. It is studied in [29] to utilize extra distilled data at both the server and client sides, where knowledge distillation is employed to mitigate catastrophic forgetting. FedCIL is proposed in [38] to learn a generative network and reconstruct past samples for replay, improving the retention of previous information.

While these approaches may be effective at learning new tasks, it is essential to also consider the constraints of privacy concerns and data heterogeneity. For example, data-reconstruction techniques with gradient or adversarial training are employed in FCIL and FedCIL to alleviate catastrophic forgetting, but it may cause privacy leakage of local data. Moreover, existing works simply assume that each client collects incremental data of different tasks in an independently and identically distributed (IID) manner, ignoring the issue of data heterogeneity in real-world scenarios.

*Ruixuan Li is the corresponding author.

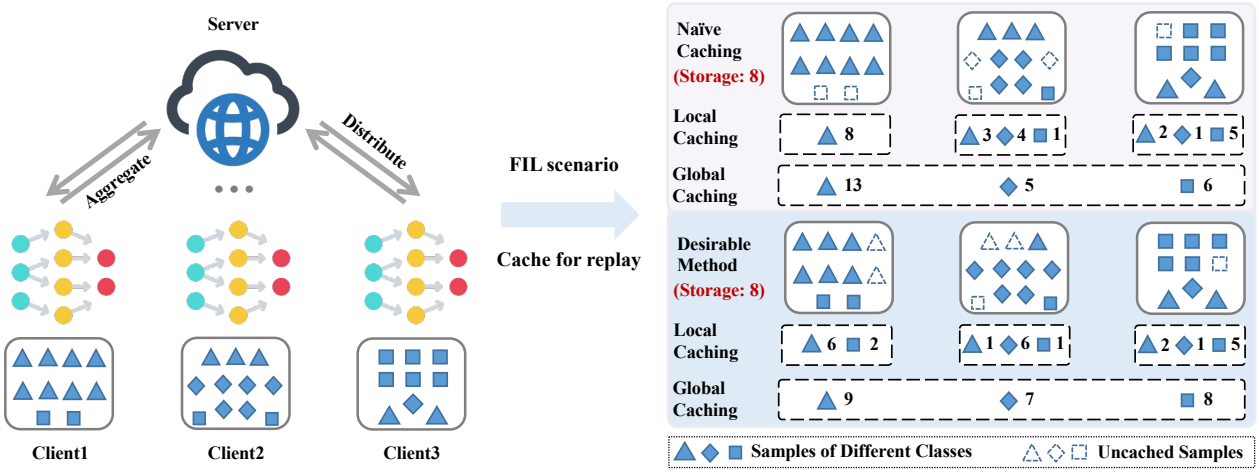


Figure 1. The motivation for our method: an example of 3-client in FIL scenario. When a new task arrives, each client needs to cache previous samples with limited storage for replay, alleviating catastrophic forgetting. Global caching represents all the samples cached by all clients collectively. With a naive caching method, the client may ignore the sample's correlation across clients which increases the statistical data heterogeneity in global caching. With a desirable method, the client tends to cache samples which both considers the distribution of local samples and reduces the statistical data heterogeneity in the global caching.

In this paper, we investigate a simple and efficient method for catastrophic forgetting in FIL. We consider classification task in FIL and first identify two types of task for newly collected data: (1) Class-Incremental Task: the newly arrived data has different labels from previous data, and thus the label space of data is growing. (2) Domain-Incremental Task: the new data has a domain shift from previous data, and it does not change the label space of data. Then, we assume data heterogeneity as the data of each task in clients is Non-IID.

To tackle the catastrophic forgetting problem in the non-federated environment, data replay [32, 39] based methods have demonstrated great effectiveness by caching the important samples from previous tasks and replaying them when learning the new task. However, existing replay-based methods fail to consider the correlation between clients in FL, and the cached samples may not be globally optimal for tasks. Essentially, the cached samples should be strongly correlated with the statistical heterogeneity of all data across clients. The intuitive explanation can also be found in a simple 3-client example illustrated in Figure 1.

To explore this idea, we propose an efficient FIL framework named Re-Fed that can alleviate catastrophic forgetting by allowing all clients to synergistically cache data samples. More specifically, in Re-Fed, each client caches samples based not only on their importance in the local dataset but also on their correlation to the global dataset. Here we fist employ an additional personalized informative model (PIM) for each client which can incorporate knowledge of data from global point of view in local caching such that the cached samples contribute to both local and global understanding of the data. Then, we quantify the sample importance by calculating the gradient norm of previous local samples during the update of the PIM. Finally, the client caches the samples with higher importance scores, and trains the local model with both the cached samples from previous tasks and the samples from the new task.

Through extensive experiments on various datasets and two types of newly collected data (Class-Incremental Task and Domain-Incremental Task), we show that Re-Fed significantly improves the model accuracy compared to state-of-the-art approaches. The major contributions of this paper are summarized as follows:

- We are the first to study the problem of catastrophic forgetting with data heterogeneity in FIL. To address this problem, we propose a novel framework named Re-Fed which can be seen as an off-the-shelf personalization add-on for standard FIL and it inherits privacy protection and efficiency properties as traditional FL applications in FIL scenarios.
- Next, we theoretically show that Re-Fed can efficiently discover the important samples for data replay, with guaranteed convergence.
- Finally, we carry out extensive experiments on various datasets and different FIL task scenarios. Experimental results illustrate that our proposed model outperforms the state-of-the-art methods by up to 19.73% in terms of final accuracy on different tasks.

## 2. Related Work

**Federated Learning** FL is a technique to train a shared global model by aggregating models from multiple clients that are trained on their own local private datasets [23, 31,

44]. One effective architecture for FL is FedAvg [31], which optimizes the global model by aggregating the parameters of local models trained on private local data. However, traditional FL algorithms like FedAvg face challenges due to data heterogeneity, where the datasets in clients are Non-IID, resulting in degradation in model performance [15, 26]. To tackle the Non-IID issue in FL, a proximal term is introduced in optimization in [22] to mitigate the effects of heterogeneous and Non-IID data distribution across participating devices. Another approach of federated distillation [14], aims to distill the knowledge from multiple local models into the global model by aggregating only the soft predictions generated by each model. The authors in [25] proposed a knowledge distillation method that utilizes unlabeled training samples as a proxy dataset. Recently, there has been growing interest in data-free knowledge distillation methods that leverage adversarial approaches in generating data [45, 50, 51]. However, the aforementioned methods follow a framework designed to handle Non-IID static data with spatial heterogeneity, overlooking the potential challenges posed by incremental tasks with temporal heterogeneity in FL scenarios.

**Incremental Learning** Incremental Learning (IL) is a machine learning technique that allows a model to learn continuously from a incremental sequence of tasks while retaining knowledge gained from previous tasks [12, 43], including task-incremental learning [5, 30], class-incremental learning [39, 49], and domain-incremental learning [2, 33]. Existing approaches in IL can be classified into three main categories: replay-based methods [27, 39], regularization-based methods [16, 47], and parameter isolation methods [8, 28]. Replay-based methods select representative old samples to retain previously learned knowledge when training on a new task. Regularization-based methods protect existing knowledge from being overwritten with new knowledge by imposing constraints on the loss function of new tasks. Parameter isolation methods typically introduce additional parameters and computations to learn new tasks. Here we focus on the federated incremental learning scenario, which can be viewed as a combination of federal learning and incremental learning.

## 3. Methodology

We first formulate two FIL scenarios and propose a simple and scalable framework Re-Fed. Then, we present a scalable algorithm and provide rigorous analytical results to show the efficiency of the proposed method.

### 3.1. Problem Formulation

In the standard IL (non-federated environment), a model learns from a sequence of streaming tasks $\{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^n\}$ where $\mathcal{T}^t$ denotes the $t$-th task of the dataset. Here $\mathcal{T}^t = \sum_{i=1}^{N^t}(x_t^{(i)}, y_t^{(i)})$, which has $N^t$ pairs of sample data $x_t^{(i)} \in \mathcal{X}^t$ and corresponding label $y_t^{(i)} \in \mathcal{Y}^t$. We use $\mathcal{X}^t$ and $\mathcal{Y}^t$ to represent the domain space and label space for the $t$-th task, which has $|\mathcal{Y}^t|$ classes and $\mathcal{Y} = \bigcup_{t=1}^{n} \mathcal{Y}^t$ where $\mathcal{Y}$ denotes the total classes of all time. Similarly, we use $\mathcal{X} = \bigcup_{t=1}^{n} \mathcal{X}^t$ to denote the total domain space for tasks of all time. In this paper, we focus on two types of IL scenarios: (1) Class-Incremental Task: all tasks share the same domain space, i.e., $\mathcal{X}^1 = \mathcal{X}^t, \forall t \in [n]$. As the sequence of learning tasks arrives, the number of the classes may change, i.e, $\mathcal{Y}^1 \neq \mathcal{Y}^t, \forall t \in [n]$. (2) Domain-Incremental Task: all tasks share the same number of classes i.e., $\mathcal{Y}^1 = \mathcal{Y}^t, \forall t \in [n]$. As the sequence of tasks arrives, the client needs to learn the new task while their domain and data distribution changes, i.e., $\mathcal{X}^1 \neq \mathcal{X}^t, \forall t \in [n]$.

We further consider IL in a federated setting. We aim to train a global model for $K$ total clients and assume that client $k$ can only access the local private streaming tasks $\{\mathcal{T}_k^1, \mathcal{T}_k^2, \cdots, \mathcal{T}_k^n\}$. When the $t$-th task comes, while clients can cache all samples from previous tasks without forgetting, the goal is to train a global model $w^t$ over all $t$ tasks $\mathcal{T}^t = \{\sum_{n=1}^{t}\sum_{k=1}^{K}\mathcal{T}_k^n\}$, which can be formulated as :

$$w^t = \arg\min_{w} \sum_{n=1}^{t}\sum_{k=1}^{K}\sum_{i=1}^{N_k^n} \frac{1}{|\mathcal{T}^t|} l\left(f_{w_k}(x_{k,n}^{(i)}), y_{k,n}^{(i)}\right). \quad (1)$$

where $f_{w_k}(\cdot)$ is the output of the model $w_k$ in client $k$ and $l(\cdot)$ is the cross-entropy loss. Then, due to poor storage in common edge devices, each client caches partial samples for replay. Here we assume each client can only store total $M$ samples and has to cache $M - N_k^t$ samples from $(t-1)$ previous tasks, which is denoted as $\mathcal{T}_{k,cached}^{t-1} = \sum_{i=1}^{M-N_k^t}(\bar{x}_{k,t-1}^{(i)}, \bar{y}_{k,t-1}^{(i)})$. The goal is to train a global model $w^t$ over both cached samples and the $t$-th new task, which can be formulated as:

$$w^t = \arg\min_{w} \sum_{k=1}^{K}\sum_{i=1}^{M} \frac{1}{|\mathcal{T}_{k,local}^t|} l\left(f_{w_k}(\tilde{x}_{k,t}^{(i)}), \tilde{y}_{k,t}^{(i)}\right). \quad (2)$$

where $\mathcal{T}_{k,local}^t = \mathcal{T}_{k,cached}^{t-1} + \mathcal{T}_k^t = \sum_{i=1}^{M}(\tilde{x}_{k,t}^{(i)}, \tilde{y}_{k,t}^{(i)})$.

### 3.2. Re-Fed: Framework for FIL

The key idea of Re-Fed is to identify the sample importance and coordinate clients to cache important previous samples with limited local storage when the new task arrives. Specifically, in each communication round, the clients train the local models with the private sequence of tasks and the server aggregates local models from all participated clients. Then, when new tasks come, each client first trains an extra personalized informative model on previous local samples with the regulation of both the global model and local model.

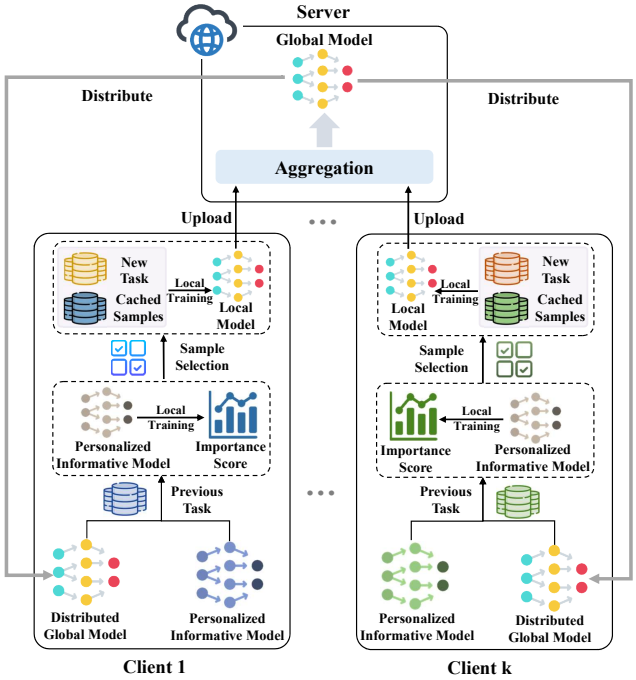


Figure 2. Illustration of the Re-Fed framework. When a new task arrives, each client first updates the personalized informative model on previous local samples with the distributed global model. Then, samples are selected to be cached by the sample importance scores that are calculated with the updated personalized informative model. Finally, each client trains the local model with both the new task and cached previous samples.

During the update of such model, gradient norms of individual samples are recorded to calculate sample importance scores. Finally, each client caches samples with higher importance scores based on local storage and continues training the local model on the cached samples and the new task. The workflow of the proposed framework is shown in Algorithm 1 and Figure 2 illustrates the Re-Fed framework.

**Personalized Informative Model.** In FIL with Non-IID client samples, sample importance should be defined based not only on their importance in the local dataset but also on their correlation to the global dataset across clients such that the model can be better trained with the cached samples. In a standard FL scenario, each client can only access its own local model and global model, which respectively contains local and global information. A straightforward idea here is to calculate two sample importance scores using local and global models and cache samples based on such two scores for data reply. Then, one can build upon such an idea by adding following capabilities: (1) The global model is aggregated by local models from participated clients and the gradient norm of the global model can be calculated locally without training the global model. (2) A control mechanism should be available to adjust the proportion of local and global information in the sample importance.

**Algorithm 1:** Re-Fed

**Input:** $T$: communication round; $K$: client number; $\eta$: learning rate; $\{\mathcal{T}^t\}_{t=1}^n$: distributed dataset with $n$ tasks; $w$: parameter of the model; $v_k$: personalized informative model in client $k$.

Initialize the parameter $w$;
**for** $c = 1$ ***to*** $T$ **do** `// the t-th new task`
  Server randomly selects a subset of devices $S_t$ and send $w^{t-1}$
  **for** *each selected client* $k \in S_t$ **in parallel do**
    Update $v_k^{t-1}$in $s$ local iterations with (3).
    **for** *During the update of* $v_k^{t-1}$ **do**
      Calculate the importance score after total $s$ iterations for the sample $(\tilde{x}_{k,t-1}^{(i)}, \tilde{y}_{k,t-1}^{(i)})$ with (5).
    **end**
    Cache previous samples with higher importance scores;
    Training the local model with cached samples and the new task with (6);
    Send the model $w_k^t$ back to the server.
  **end**
  $w^t \leftarrow \text{ServerAggregation}(\{w_k^t\}_{k \in S_t})$
**end**

Toward the above goals, here we introduce an additional personalized informative model (PIM) for each client, which digests both the information from the local model and the global model. Then, we propose to adopt a ratio factor to adjust the information proportion from local and global sides. Finally, we can record the gradient norms of the samples to calculate the importance scores during the update of PIM, resulting in sample importance scores with both local and global information. Suppose that the client $k$ receives the global model $w^{t-1}$ and then the $t$-th new task arrives, and the clients update PIM $v_k^{t-1}$ with previous local samples $\mathcal{T}_{k,local}^{t-1}$ in $s$ iterations as follows:

$$v_{k,s}^{t-1} = v_{k,s-1}^{t-1} - \eta\Bigg(\sum_{i=1}^{M} \nabla l\left(f_{v_{k,s-1}^{t-1}}(\tilde{x}_{k,t-1}^{(i)}), \tilde{y}_{k,t-1}^{(i)}\right) + q(\lambda)(v_{k,s-1}^{t-1} - w^{t-1})\Bigg). \tag{3}$$

where $q(\lambda) = \frac{1-\lambda}{2\lambda}, \lambda \in (0, 1)$, and $\eta$ is the rate to control the step size of the update. The hyper-parameter $\lambda$ adjusts the balance between the local and global information incorporated in the update.

To better understand the update, we can draw an analogy to momentum methods in optimization. Momentum-based

Table 1. Performance comparison of various methods in two incremental learning scenarios.

| Scenario | Dataset | FedAvg | FedProx | Fixed | DANN+FL | Shared | FCIL | FedCIL | Re-Fed |
|---|---|---|---|---|---|---|---|---|---|
| Class-Incremental | CIFAR10 ($\alpha = 1.0$) | 26.73±1.12 | 25.87±0.68 | 19.21±0.06 | 24.86±2.31 | 23.91±1.70 | 25.04±0.11 | 27.35±1.24 | **29.22±0.49** |
| | CIFAR100 ($\alpha = 5.0$) | 17.21±1.35 | 18.03±0.91 | 9.27±0.22 | 19.73±2.17 | 18.30±1.53 | 23.02±0.66 | 17.98±1.46 | **25.61±0.88** |
| | Tiny-ImageNet ($\alpha = 10$) | 27.58±0.74 | 21.82±0.90 | 12.34±0.23 | 20.77±1.31 | 22.19±0.54 | 29.58±0.15 | 24.41±0.95 | **32.07±0.27** |
| Domain-Incremental | Digit10 ($\alpha = 0.1$) | 77.59±0.39 | 79.09±0.58 | 71.26±0.04 | 76.44±1.05 | 74.77±0.23 | 77.59±0.39 | 83.85±0.80 | **85.96±0.14** |
| | Office31 ($\alpha = 1$) | 39.25±1.61 | 43.01±1.59 | 37.44±0.72 | 45.21±2.10 | 37.55±0.69 | 39.25±1.61 | 46.26±2.24 | **50.80±0.77** |
| | DomainNet ($\alpha = 10$) | 51.73±2.32 | 49.12±2.71 | 46.30±1.42 | 50.01±3.31 | 41.76±1.26 | 51.73±2.32 | 47.28±3.01 | **56.66±0.50** |

methods leverage the past updates to guide the current update direction [1]. Similarly, the term $q(\lambda)(v_{k,s-1}^{t-1} - w^{t-1})$ acts as a momentum component. It incorporates information from the global model $w^{t-1}$ to influence the update of PIM $v_k^{t-1}$. The hyper-parameter $\lambda$ controls the weight of this momentum component, and it lies within the range of (0,1). When $\lambda$ is close to 0, PIM primarily focuses on recovering the global model $w^{t-1}$. In other words, it will align itself with the global data. On the other hand, as $\lambda$ becomes larger, it leads to a stronger emphasis on local training.

**Theorem 3.1** (Convergence of PIM)**.** *Assuming that the global model $w^t$ converges to the optimal model $\hat{w}$ at communication round $t$ by $g(t)$ as: $\mathbb{E}\left[||w^t - \hat{w}||^2\right] \leq g(t)$, $\lim_{t\to\infty} g(t) = 0$ and $g(t+1) \leq g(t)$, there exists a constant $C < \infty$ such that for any client $k \in [K]$ the personalized informative model $v_k^t$ can converge to the optimal model $\hat{v}_k$ by $Cg(t)$.*

With Theorem 3.1, we ensure the convergence of PIM and thus can calculate the gradient norms of samples during the training stage. We provide the proof in Appendix F.1.

**Sample Importance.** To quantify the sample importance, we investigate the impact of samples on the generalization ability of the model, and allocate the samples that can enhance the generalization with higher importance. We calculate the gradient norm with respect to model parameters of PIM as the importance scores to samples. The gradient norm of samples during training is recorded, which can be regarded as the contribution of the sample to the model update. A similar flavor of such a method can be found in [36, 42]. Suppose that the client $k$ has converged on $(t-1)$-th tasks with local samples $\mathcal{T}_{k,local}^{t-1}$, when it comes to the incremental $t$-th task, the client should calculate the importance scores for all local samples. Here we denote $G^p(\tilde{x}_{k,t-1}^{(i)})$ is the gradient norm of the sample $(\tilde{x}_{k,t-1}^{(i)}, \tilde{y}_{k,t-1}^{(i)})$ in $p$-th iteration during the update of PIM $v_{k,p}^{t-1}$, which is:

$$G^p(\tilde{x}_{k,t-1}^{(i)}) = \left|\left|\nabla l\left(f_{v_{k,p}^{t-1}}(\tilde{x}_{k,t-1}^{(i)}), \tilde{y}_{k,t-1}^{(i)}\right)\right|\right|^2. \quad (4)$$

According to [42], the difference in the gradient of the loss function with and without a sample $(\tilde{x}_{k,t-1}^{(j)}, \tilde{y}_{k,t-1}^{(j)})$ is upper bounded and the bound is linearly dependent on the sample gradient norm defined in Eq. 4. Thus, caching samples based on sample gradient norms can least affect the gradient and best preserve the dynamics of training.

As PIM integrates both local and global models, a greater gradient norm of a sample with PIM indicates that such a sample drives PIM more to fitting the task with local and global knowledge. Such an effect could be more prominent at early training during $s$ iterations for PIM, where fluctuation around optima is rare than that later in the training. Thus, we accumulate the gradient norm during the training of PIM and emphasis on the early training stage to calculate the sample importance as:

$$I(\tilde{x}_{k,t-1}^{(i)}) = \sum_{p=1}^{s} \frac{1}{p} G^p(\tilde{x}_{k,t-1}^{(i)}). \quad (5)$$

We also provide illustrative experimental results with $I(\tilde{x}_{k,t-1}^{(i)}) = \sum_{p=1}^{s} G^p(\tilde{x}_{k,t-1}^{(i)})$ in the Appendix E and show the performance improvement with the sample importance adopted in Eq. 5.

**Local Training.** After caching important samples with higher importance scores, each client continues to train the local model $w_k^t$ with local samples $\mathcal{T}_{k,local}^t$ in iteration $p \in [1, s]$ as follows:

$$w_{k,p}^t = w_{k,p-1}^t - \eta \sum_{i=1}^{M} \nabla l\left(f_{w_{k,p-1}^t}(\tilde{x}_{k,t}^{(i)}), \tilde{y}_{k,t}^{(i)}\right). \quad (6)$$

**Modularity of Re-Fed.** From the Re-Fed framework and Algorithm 1, we can see that a key feature of Re-Fed is its unique modularity. One can readily use prior art developed for FIL algorithm, and employ Re-Fed as a useful off-the-shelf add-on. Our method has several advantages:

- **Optimization:** It is possible to plug in other aggregation methods beyond FedAvg [31] in Algorithm 1 to update the global model, and inherit the convergence benefits. In the subsequent experimental design, we investigate the performance of our Re-Fed framework with FedAvg algorithm and provide a detailed algorithm definition using FedAvg in Appendix D.
- **Privacy:** Re-Fed transmits no more extra information over the network than typical FL algorithms. This is dif-

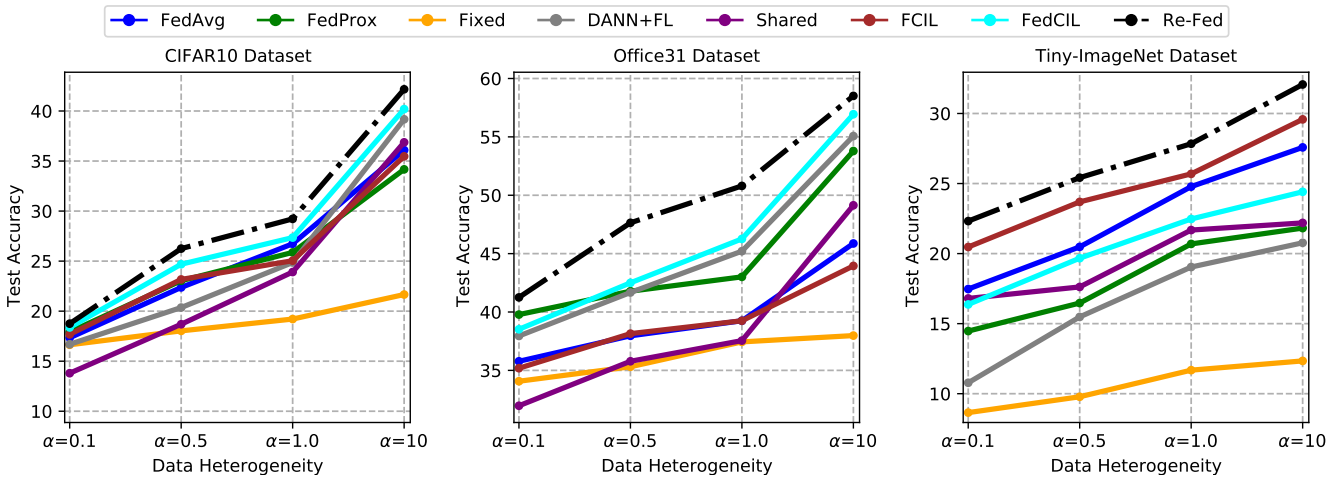


Figure 3. Performance w.r.t data heterogeneity $\alpha$ for three datasets.

Table 2. Test accuracy for Re-Fed w.r.t data heterogeneity $\alpha$ and hyper-parameter $\lambda$ on CIFAR10, Office31 and Tiny-ImageNet.

| Dataset | $\alpha = 0.1$ | | | $\alpha = 0.5$ | | | $\alpha = 1.0$ | | | $\alpha = 10$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda = 0.2$ | $\lambda = 0.5$ | $\lambda = 0.8$ | $\lambda = 0.2$ | $\lambda = 0.5$ | $\lambda = 0.8$ | $\lambda = 0.2$ | $\lambda = 0.5$ | $\lambda = 0.8$ | $\lambda = 0.2$ | $\lambda = 0.5$ | $\lambda = 0.8$ |
| CIFAR10 | **18.75±1.30** | 18.61±1.09 | 17.91±0.81 | **26.25±1.64** | 26.00±0.97 | 25.62±0.52 | 27.05±0.88 | 27.80±0.21 | **29.22±0.49** | 38.43±0.43 | 40.04±0.19 | **42.17±0.25** |
| Office31 | **41.25±1.01** | 39.29±1.34 | 38.18±0.68 | 46.86±0.91 | **47.64±0.53** | 47.13±1.16 | 43.81±0.73 | 48.67±0.99 | **50.08±0.77** | 52.79±1.28 | 55.92±0.38 | **58.51±0.46** |
| Tiny-ImageNet | **22.32±0.12** | 20.51±0.98 | 18.00±1.30 | 24.60±0.48 | **25.42±0.59** | 24.39±0.66 | 24.88±0.87 | 27.15±0.78 | **27.84±0.73** | 29.03±0.30 | 30.26±0.24 | **32.07±0.27** |

ferent from most other FIL methods where sample reconstruction methods are applied for data replay, which may raise privacy concerns.

- **Resource:** Re-Fed allows each client to train a backbone model using only its local training data, without employing additional distillation data or generated data, leading to extra either computation cost or storage overhead.

## 4. Experiments

In this section, we evaluate our proposed framework concerning two incremental learning scenarios. We investigate the relationship between the data heterogeneity and the balance between the local and global information incorporated in PIM. Additionally, we conduct parameter sensitivity analysis to verify the effectiveness of our method.

### 4.1. Experiment Setup

**Dataset:** We conduct our experiments with heterogeneously partitioned datasets over two federated incremental scenarios on six datasets. (1) **Class-Incremental Learning:** CIFAR10[17], CIFAR100[17] and Tiny-ImageNet[18]. (2) **Domain-Incremental Learning:** Digit10, Office31[41] and DomainNet[37]. Among them, the Digit10 dataset contains 10 digit categories in four domains: MNIST[19], EMNIST[3], USPS[13] and SVHN[34]. Details of datasets and data processing can be found in Appendix A.

**Baseline:** For a fair comparison with other key works, we follow the same protocols proposed by [31, 39] to set up FIL tasks. We evaluate all the methods with two representative FL models **FedAvg** [31] and **Fedprox** [22], two models designed for federated class-incremental learning: **FCIL** [6] and **FedCIL** [38], and three customed methods of **Fixed**: we train the model only from the first task and evaluate it for all the coming sequence of tasks; **DANN+FL**: here we adopt the robust adversarial-based method DANN[9] in local training for domain adaption; **Shared**: we adopt all front layers before the last fully connected layer as shared layers, and use relevant different fully-connected layers to obtain outputs for different tasks. Details of datasets and data processing can be found in Appendix B.

**Configurations:** Unless otherwise mentioned, we set the number of local training epoch $E = 20$ and communication round $T = 150$ for each task, which ensures the convergence of previous tasks before the arrival of new task. We use the Dirichlet distribution Dir($\alpha$) to distribute local samples to yield data heterogeneity for all tasks where a smaller $\alpha$ indicates higher data heterogeneity. We employ ResNet18 [11] as the basic backbone model in all methods. We calculate the Top-10 accuracy for the Tiny-ImageNet and DomainNet datasets and Top-1 accuracy for others. Each experiment setting is run three times and we record accuracy in the final 10 rounds and report the average value and standard deviation. The total clients number is 20/10 with an active ratio $k = 0.4$ for {CIFAR10, CIFAR100,

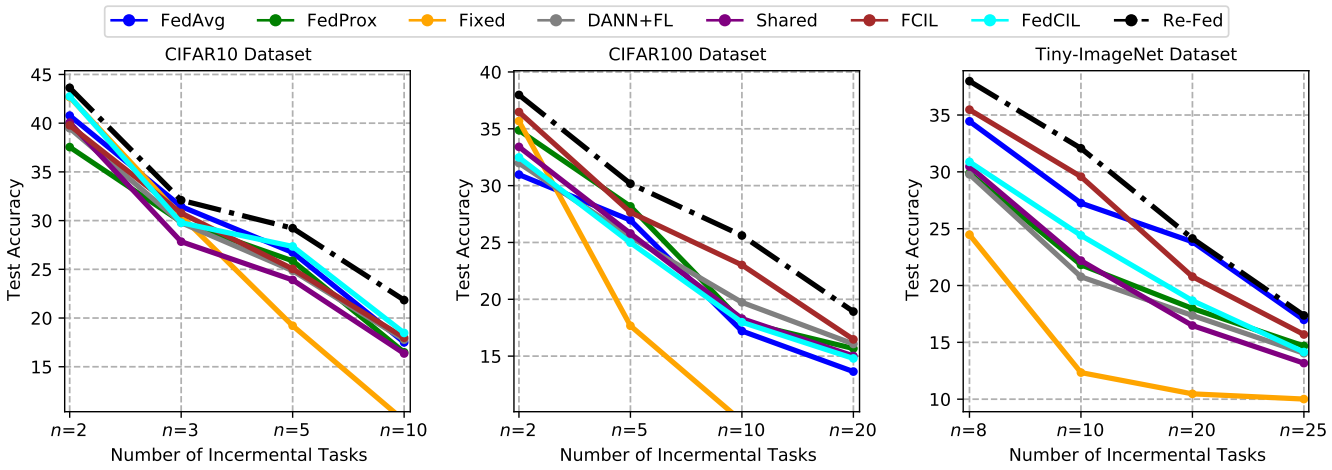


Figure 4. Performance w.r.t number of incremental tasks $n$ for three class-incremental datasets.

Table 3. Evaluation of various methods, in terms of the communication rounds to reach the best test accuracy (150 communication rounds each task). We report the sum of communication rounds required to achieve the best performance on each task.

| Scenario | Dataset | FedAvg | FedProx | Fixed | DANN+FL | Shared | FCIL | FedCIL | Re-Fed |
|---|---|---|---|---|---|---|---|---|---|
| Class-Incremental | CIFAR10 (Task:5) | 613±2.67 | 685±3.00 | 142±0.67 | 712±3.67 | 574±1.33 | 590±2.67 | 738±4.00 | **562±1.67** |
| | CIFAR100 (Task:10) | 1103±2.33 | 1246±3.00 | 137±2.00 | 1258±4.67 | 1154±3.33 | 1095±2.67 | 1311±5.67 | **1039±4.33** |
| | Tiny-ImageNet (Task:10) | 1197±2.67 | 1234±2.67 | 132±3.00 | 1305±3.67 | 1278±4.33 | 1185±2.33 | 1317±3.33 | **1128±3.67** |
| Domain-Incremental | Digit10 (Task:4) | 410±1.67 | 412±0.67 | 112±0.33 | 483±1.33 | 372±2.00 | 410±1.67 | 419±2.67 | **325±1.33** |
| | Office31 (Task:3) | 413±2.67 | 429±2.00 | 144±0.67 | 436±3.67 | 391±1.12 | 413±2.67 | 431±3.33 | **388±1.67** |
| | DomainNet (Task:6) | 726±3.33 | 767±2.67 | 141±1.67 | 752±4.00 | 694±2.67 | 726±3.33 | 791±3.67 | **661±2.33** |

Tiny-ImageNet, Digit10, DomainNet}/{Office31}. The maximum number of cached samples $M$ is 2000/1000/300 for {Digit10, DomainNet, Tiny-ImageNet}/{CIFAR10, CIFAR100}/{Office31}. We experiment with different numbers of incremental tasks and each task arrives with new classes: 10/10/5 tasks with 20/10/2 new classes in each task for {Tiny-ImageNet}/ {CIFAR100}/ {CIFAR10}. Details of configurations can be found in Appendix C.

### 4.2. Performance Overview

**Test Accuracy.** Table 1 shows the test accuracy of various methods with data heterogeneity across six datasets. We report the final accuracy of the global model when all clients finish their training on all tasks. FedCIL outperforms FedAvg and FedProx on CIFAR10, Digit10 and Office31 as the generator in FedCIL exhibits effective training on simple datasets, enabling the generation of high-quality samples for data replay. However, its performance experiences a significant decline with larger-scale datasets such as CIFAR100 and DomainNet, where the classes and domains become more complex. In the federated domain-incremental learning scenario, FCIL reverts to the FedAvg algorithm when there are no new incremental sample classes. Re-Fed achieves the best performance in all cases by a margin of 1.87%∼19.73% in terms of final accuracy. More discussions and results on model performance are available in Appendix E.

**Data Heterogeneity.** Figure 3 displays the test accuracy with different levels of data heterogeneity on three datasets. As shown in this figure, all methods achieve an improvement in test accuracy with the decline in data heterogeneity, and Re-Fed consistently achieves a leading improvement in performance with different levels of data heterogeneity.

Then, we conduct more research on the setting of hyper-parameter $\lambda$. In our framework, we modify the $\lambda$ value to adjust the global and local information proportion in PIM. As shown in Table 2, we select three different $\lambda$ values with four different data heterogeneity settings and evaluate the final test accuracy on three datasets. Experimental results show that the value of $\lambda$ should be chosen accordingly under different data heterogeneity. Nevertheless, results exhibit the same trend: as the degree of data heterogeneity increases, our Re-Fed framework performs better while $\lambda$ decreases as PIM contains more global information. Empirically, striking a balance between global and local information is the key to addressing the data heterogeneity in FIL. Vice versa, as $\alpha$ increases, the data distribution on the client side becomes more IID. At this point, the clients

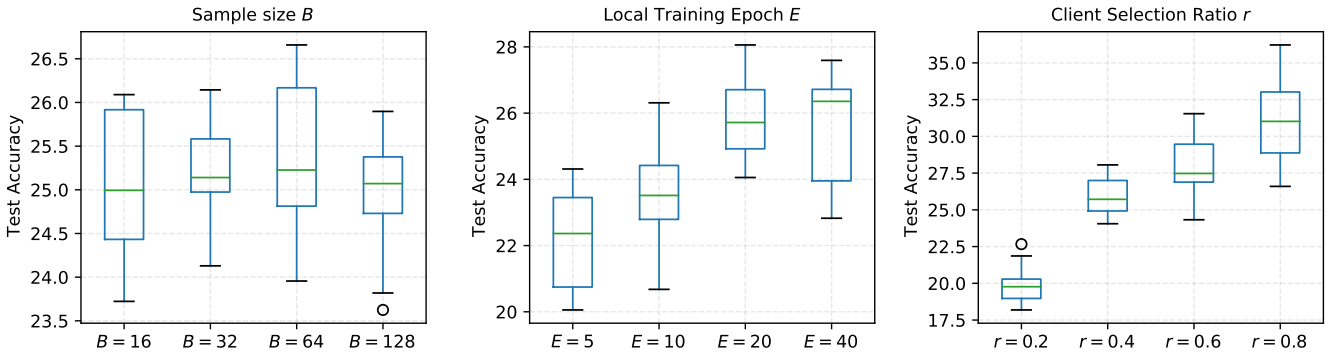


Figure 5. Performance of Re-Fed under different configurations (a) local training epoch $E$, (b) sample size $B$ in the classifier, (c) client selection ratio $r$ of all clients on CIFAR100 with $\alpha = 5.0$.

Table 4. Test accuracy for Re-Fed and other baselines w.r.t memory size $M$ on Digit10, CIFAR10, CIFAR100 and Tiny-ImageNet.

| Dataset | FedAvg | | | DANN+FL | | | FCIL | | | Re-Fed | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M = 1000$ | $M = 2000$ | $M = 3000$ | $M = 1000$ | $M = 2000$ | $M = 3000$ | $M = 1000$ | $M = 2000$ | $M = 3000$ | $M = 1000$ | $M = 2000$ | $M = 3000$ |
| Digit10 | 77.05±0.24 | 77.59±0.39 | 81.64±0.31 | 75.16±0.86 | 76.44±1.05 | 83.34±0.97 | 77.50±0.24 | 77.59±0.39 | 81.64±0.31 | **82.30±0.16** | **85.96±0.14** | **86.32±0.04** |
| CIFAR10 | 26.73±1.12 | 30.19±1.63 | 33.41±0.92 | 24.86±2.31 | 27.65±1.34 | 30.09±1.06 | 25.04±0.11 | 29.14±0.19 | 31.77±0.28 | **29.22±0.49** | **32.83±0.20** | **33.41±0.74** |
| CIFAR100 | 17.21±1.35 | 23.58±1.82 | 29.90±2.11 | 19.73±2.17 | 25.56±2.68 | 28.49±1.98 | 23.02±0.66 | 26.12±0.54 | 29.06±0.89 | **25.61±0.88** | **28.41±0.24** | **29.90±0.71** |
| Tiny-ImageNet | 24.42±0.59 | 27.58±0.74 | 31.50±0.63 | 18.06±1.08 | 20.77±1.31 | 26.93±0.77 | 25.20±1.12 | 29.58±0.15 | 33.44±0.38 | **28.31±0.19** | **32.07±0.27** | **35.66±0.42** |

require less global information and can rely more on their local information for caching important samples.

**Quantitative Analysis**. Figure 4 shows the qualitative analysis of the number of incremental tasks $n$ on three class-incremental datasets. According to these curves, we can easily observe that our model performs better than other baselines across all tasks, with varying numbers of incremental tasks. It demonstrates that Re-Fed enables clients to learn new incremental classes better than other methods.

**Communication Efficiency.** Table 3 shows the evaluation of various methods in terms of the communication rounds to reach the test accuracy reported in Table 1. Here we show the sum of communication rounds required to achieve the performance on each task. Re-Fed requires the least communication rounds to achieve the reported test accuracy on all datasets with its caching method. Thus, the local model is easier to converge. More details of the communication round on each task are available in Appendix E.

**Parameter Sensitivity Analysis.** Figure 5 shows the performance of Re-Fed under different configurations with standard boxplots from ten trails with different random seeds. Re-Fed achieves similar performance with different sample sizes $B$, and it achieves a better result when we increase the local training epochs. However, Re-Fed has a comparable performance when the $E$ is set to 20 and 40. In addition, a larger client selection ratio $r$ contributes to higher test accuracy.

For the FIL scenario, we conduct additional research on the client storage $M$. When $M$ is large enough for clients to cache samples from all tasks, our framework degrades to a normal FedAvg algorithm with a naive caching method. As shown in Table 4, we select three different $M$ values to conduct experiments. Experimental results demonstrate that larger memory size $M$ contributes to the training and Re-Fed outperforms other baselines in all cases. To conclude, Re-Fed is only sensitive to few parameters and still robust to most parameters in a large range.

## 5. Conclusion and Future Work

We proposed Re-Fed, a simple framework, to address the catastrophic forgetting with data heterogeneity in federated incremental learning. Re-Fed can be thought of as a lightweight personalization add-on for any federated learning algorithms with global aggregation, which maintains privacy and communication efficiency. Extensive experiments conducted on various settings and baselines show that Re-Fed achieves significant improvement in test accuracy.

Although existing works and our method have demonstrated great effectiveness over the FIL scenarios, none have studied the dynamic requirements of the edge clients. To deploy the FL system in practical settings, it is necessary to consider personalized local factors such as storage, computation and even the different arrival time of the new task. In the future, we seek to work a step forward in this field.

## Acknowledgements

This work is supported by National Natural Science Foundation of China under grants 62376103, 62302184, 62206102, Science and Technology Support Program of Hubei Province under grant 2022BAA046, and CCF-AFSG Research Fund.

## References


[1] Louis KC Chan, Narasimhan Jegadeesh, and Josef Lakonishok. Momentum strategies. *The journal of Finance*, 51(5): 1681–1713, 1996. 5

[2] Nikhil Churamani, Ozgur Kara, and Hatice Gunes. Domain-incremental continual learning for mitigating bias in facial expression and action unit recognition. *ArXiv*, abs/2103.08637, 2021. 3

[3] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 6, 1

[4] Marcos F Criado, Fernando E Casado, Roberto Iglesias, Carlos V Regueiro, and Senén Barro. Non-iid data and continual learning processes in federated learning: A long road ahead. *Information Fusion*, 88:263–280, 2022. 1

[5] Neil T. Dantam, Zachary K. Kingston, Swarat Chaudhuri, and Lydia E. Kavraki. Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and Systems*, 2016. 3

[6] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10164–10173, 2022. 1, 6

[7] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society*, 1:45–61, 2020. 1

[8] Chrisantha Fernando, Dylan S. Banarse, Charles Blundell, Yori Zwols, David R Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *ArXiv*, abs/1701.08734, 2017. 3

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 6

[10] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *CoRR*, abs/2002.05516, 2020. 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[12] Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *ArXiv*, abs/1810.12488, 2018. 3

[13] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994. 6, 1

[14] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479, 2018. 3

[15] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *ArXiv*, abs/1811.11479, 2018. 3

[16] Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. *arXiv: Learning*, 2020. 3

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[18] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[19] Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 2010. 6, 1

[20] Yuan Lei, Shir Li Wang, Minghui Zhong, Meixia Wang, and Theam Foo Ng. A federated learning framework based on incremental weighting and diversity selection for internet of vehicles. *Electronics*, 11(22):3668, 2022. 1

[21] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Federated multi-task learning for competing constraints. *CoRR*, abs/2012.04221, 2020. 6

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 3, 6

[23] Xin-Chun Li, Yi-Chu Xu, Shaoming Song, Bingshuai Li, Yinchuan Li, Yunfeng Shao, and De-Chuan Zhan. Federated learning with position-aware neurons. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10072–10081. 1, 2

[24] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8423–8434, 2021. 1

[25] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020. 3

[26] Lumin Liu, Jun Zhang, S. H. Song, and Khaled Ben Letaief. Edge-assisted hierarchical federated learning with non-iid data. *ArXiv*, abs/1905.06641, 2019. 3

[27] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 226–227, 2020. 3

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. *ArXiv*, abs/1502.02791, 2015. 3

[29] Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages

2182–2188. International Joint Conferences on Artificial Intelligence Organization, 2022. Main Track. 1

[30] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural networks : the official journal of the International Neural Network Society*, 116:56–73, 2018. 3

[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 5, 6

[32] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013. 2

[33] M Jehanzeb Mirza, Marc Masana, Horst Possegger, and Horst Bischof. An efficient domain-incremental learning approach to drive in all weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3001–3011, 2022. 3

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6, 1

[35] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3): 1–37, 2022. 1

[36] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *CoRR*, abs/2107.07075, 2021. 5

[37] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 6

[38] Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*, 2023. 1, 6

[39] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 3, 6

[40] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 1

[41] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010. 6

[42] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. *CoRR*, abs/1812.05159, 2018. 5

[43] Gido M. van de Ven and Andreas Savas Tolias. Three scenarios for continual learning. *ArXiv*, abs/1904.07734, 2019. 3

[44] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. ATPFL: automatic trajectory prediction model design under federated learning framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6553–6562. 1, 3

[45] Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20412–20421, 2023. 3

[46] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5: 1–19, 2021. 1

[47] Dong Yin, Mehrdad Farajtabar, and Ang Li. Sola: Continual learning with second-order loss approximation. *ArXiv*, abs/2006.10974, 2020. 3

[48] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pages 12073–12086. PMLR, 2021. 1

[49] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6980–6989, 2020. 3

[50] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10164–10173. 3

[51] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021. 3

# Towards Efficient Replay in Federated Incremental Learning

Supplementary Material

## A. Dataset

**Class-Incremental Task Dataset:** New classes are incrementally introduced over time. The dataset starts with a subset of classes, and new classes are added in subsequent stages, allowing models to learn and adapt to an increasing number of classes.

- **CIFAR10:** A dataset with 10 object classes, including various common objects, animals, and vehicles. It consists of 50,000 training images and 10,000 test images.
- **CIFAR100:** Similar to CIFAR10, but with 100 fine-grained object classes. It has 50,000 training images and 10,000 test images.
- **Tiny-ImageNet:** A subset of the ImageNet dataset with 200 object classes. It contains 100,000 training images, 10,000 validation images, and 10,000 test images.

**Domain-Incremental Task Dataset:** New domains are introduced gradually. The dataset initially contains samples from a specific domain, and new domains are introduced at later stages, enabling models to adapt and generalize to new unseen domains.

- **Digit10:** Digit-10 dataset contains 10 digit categories in four domains: **MNIST**[19], **EMNIST**[3], **USPS**[13], **SVHN**[34].Each dataset is a digit image classification dataset of 10 classes in a specific domain, such as handwriting style.
  - **MNIST:** A dataset of handwritten digits with a training set of 60,000 examples and a test set of 10,000 examples.
  - **EMNIST:** An extended version of MNIST that includes handwritten characters (letters and digits) with a training set of 240,000 examples and a test set of 40,000 examples.
  - **USPS:** The United States Postal Service dataset consists of handwritten digits with a training set of 7,291 examples and a test set of 2,007 examples.
  - **SVHN:** The Street View House Numbers dataset contains images of house numbers captured from Google Street View, with a training set of 73,257 examples and a test set of 26,032 examples.
- **Office31:** A dataset with images from three different domains: Amazon, Webcam, and DSLR. It consists of 31 object categories, with each domain having around 4,100 images.
- **DomainNet:** A large-scale dataset with images from six different domains: Clipart, Painting, Real, Sketch, Quickdraw, and Infograph. It contains over 0.6 million images across 345 categories.

## B. Baseline

- **Representative FL models:**
  - **FedAvg:** : It is a representative federated learning model, which aggregates client parameters in each communication. It is a simply yet effective model for federated learning.
  - **FedProx:** It is also a representative federated learning model, which is better at tackling heterogeneity in federated networks than FedAvg
- **Custom methods:**
  - **Fixed:** we train the model only from the first task and evaluate it for all the coming sequence of tasks.
  - **DANN+FL:** Here we adopt the robust adversarial-based method DANN[9]. This baseline mainly follows the domain adaptation paradigm which is different from the incremental learning setting and are often prone to issues like catastrophic forgetting.
  - **Shared:** Inspired by the multi-task learning scenario[40], we adopt all front layers before the last fully connected layer as shared layers, and use relevant different fully-connected layers to get outputs for different tasks.
- **Models for federated class-incremental learning:**
  - **FCIL:** This approach addresses the federated class-incremental learning and trains a global model by computing additional class-imbalance losses. A proxy server is introduced to reconstruct samples to help clients select the best old models for loss computation.
  - **FedCIL:** This approach employs the ACGAN backbone to generate synthetic samples to consolidate the global model and align sample features in the output layer. Authors conduct experiments in the FCIL scenario, and here we adopt it to our FDIL setting.

## C. Configurations

For local training, the batch size is 64, learning rate for our models is 0.01/0.001 for {Office31, CIFAR10, CIFAR100}/{Digit10, DomainNet, Tiny-ImageNet}. For the update of the personalized informative model, the epoch is set to 40 for each client. For the multi-task learning structure in our approach, we treat all previous layers before the last fully-connected layer as share layers, and we use two different fully-connected layers to get outputs as the auxiliary classifier result and target classification result. We build the Virtual Machine(VM) to simulate the experiment environment and set up different processes to simulate different clients. The VM is configured with 8 RTX4090 and 6 2.3GHz Intel Xeon CPUs.

## D. Detailed Re-Fed Framework with FedAvg

**Algorithm 2:** Re-Fed for FIL with FedAvg Algorithm

**Input:** $T$: communication round; $K$: number of clients; $\eta$: learning rate; $\{\mathcal{T}^t\}_{t=1}^n$: distributed dataset with $n$ tasks; $w$: parameter of the model; $v_k$: personalized informative model in client $k$; $\lambda$: factor of information proportion.

Initialize the parameter $w$;

**for** $c = 1$ ***to*** $T$ **do** `// When the t-th new task arrives`

  Server randomly selects a subset of devices $S_t$ and send $w^{t-1}$ to them;

  **for** *each selected client* $k \in S_t$ **in parallel do**

    Receive the distributed global model $w^{t-1}$ and initializess the personalized informative model $v_k^{t-1}$;

    Update $v_k^{t-1}$in $s$ local iterations with previous local samples $\mathcal{T}_{k,local}^{t-1}$:

    $$v_{k,s}^{t-1} = v_{k,s-1}^{t-1} - \eta\left(\sum_{i=1}^{M} \nabla l\left(f_{v_{k,s-1}^{t-1}}(\tilde{x}_{k,t-1}^{(i)}), \tilde{y}_{k,t-1}^{(i)}\right) + q(\lambda)(v_{k,s-1}^{t-1} - w^{t-1})\right),\ q(\lambda) = \frac{1-\lambda}{2\lambda}, \lambda \in (0,1).$$

    **for** *During the update of* $v_k^{t-1}$ **do**

      Calculate the importance score for the sample $(\tilde{x}_{k,t-1}^{(i)}, \tilde{y}_{k,t-1}^{(i)})$ after total $s$ iterations:

      $I(\tilde{x}_{k,t-1}^{(i)}) = \sum_{p=1}^{s} \frac{G^p(\tilde{x}_{k,t-1}^{(i)})}{p}$.

    **end**

    Cache previous samples with higher importance scores;

    Train the local model with cached samples and the new task $(\tilde{x}_{k,t}^{(i)}, \tilde{y}_{k,t}^{(i)})$ in $s$ iterations:

    $$w_{k,s}^{t} = w_{k,s-1}^{t} - \eta\left(\sum_{i=1}^{M} \nabla l\left(f_{w_{k,s-1}^{t}}(\tilde{x}_{k,t}^{(i)}), \tilde{y}_{k,t}^{(i)}\right)\right).$$

    Send the model $w_k^t$ back to the server.

  **end**

  The server aggregates the local models: $w^t = \sum_{k \in S_t} \frac{1}{|S_t|} w_k^t$.

**end**

## E. Experimental Results

In this section, we further provide more details about the experiment results on the test accuracy and communication rounds. We record the test accuracy of the global model at training stage of each task and the communication rounds required to achieve the corresponding performance. Then, as we use a form of "Early-Emphasis" to accumulate the gradient norms and calculate the sample importance scores in Re-Fed, we compare and show results with two other methods of calculation of sample importance scores.

### E.1. Detailed Results of Test Accuracy.

Table 5, 6, 7, 8 and 9 show the results of test accuracy on each incremental task in the **Acc** (Accuray) line, where "$\Delta$" denotes the improvement of our method with other baselines. Here we measure average accuracy over all tasks on each client in the **Acc** line and highlight the best test accuracy in **bold**.

### E.2. Detailed Results of Communication Round.

Table 5, 6, 7, 8 and 9 show the detailed results of communication round on each incremental task in the **CoR** (Communication Round) line and highlight the results of fewest number of communication rounds in underline.

### E.3. Different Weighting Methods for Gradient Norms.

Table 10 shows the impact of using different methods to calculate the sample importance score with gradient norms in the update of personalized informative models. Here we adopt three methods: **Average Weighting:** we assign an equal weight to gradient norms from different iterations; **Early-Emphasis:** a higher weight to gradient norms in the early-training as

adopted by Re-Fed; and **Late-Emphasis**: the sorting of samples with the sample importance score obtained by the method of **Early-Emphasis** is reversed.

Table 5. Performance comparisons of various methods on CIFAR10 with 5 incremental tasks (2 new classes for each task).

| **CIFAR10** ($\alpha$ = 1.0) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Target | 2 | 4 | 6 | 8 | 10 | Avg | $\Delta(\uparrow)$ |
| FedAvg | Acc | 92.65 | 76.67 | 42.90 | 40.46 | 26.73 | 55.88 | 2.57↑ |
| | CoR | 142 | 123 | 125 | 98 | 119 | 122 | 10↑ |
| FedProx | Acc | 92.39 | 74.18 | 39.84 | 37.55 | 25.87 | 53.97 | 4.48↑ |
| | CoR | 153 | 137 | 141 | 123 | 132 | 137 | 25↑ |
| Fixed | Acc | 92.65 | 62.48 | 36.54 | 24.20 | 19.21 | 47.02 | 11.43↑ |
| | CoR | 142 | 0 | 0 | 0 | 0 | / | / |
| DANN+FL | Acc | 93.07 | 77.81 | 44.32 | 36.98 | 24.86 | 55.41 | 3.04↑ |
| | CoR | 151 | 140 | 150 | 126 | 145 | 142 | 30↑ |
| Shared | Acc | 92.65 | 76.19 | 42.15 | 38.24 | 23.91 | 54.63 | 3.82↑ |
| | CoR | 142 | 117 | 116 | <u>83</u> | 118 | 115 | 3↑ |
| FCIL | Acc | 92.65 | 78.07 | 43.66 | 40.28 | 25.04 | 55.94 | 2.51↑ |
| | CoR | 142 | 125 | <u>108</u> | 92 | 121 | 118 | 6↑ |
| FedCIL | Acc | **94.05** | **80.22** | 46.19 | 35.50 | 27.35 | 56.66 | 1.79↑ |
| | CoR | 148 | 150 | 146 | 147 | 150 | 148 | 36↑ |
| Re-Fed | Acc | 92.65 | 79.23 | **47.41** | **43.75** | **29.22** | **58.45** | / |
| | CoR | <u>142</u> | <u>109</u> | 116 | 85 | <u>106</u> | <u>112</u> | |

Table 6. Performance comparisons of various methods on CIFAR100 with 10 incremental tasks (10 new classes for each task).

| **CIFAR100** ($\alpha$ = 5.0) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Target | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | Avg | $\Delta(\uparrow)$ |
| FedAvg | Acc | 58.70 | 43.72 | 48.69 | 38.28 | 30.81 | 26.16 | 24.90 | 20.72 | 18.97 | 17.21 | 32.82 | 5.57↑ |
| | CoR | 137 | 121 | <u>76</u> | 135 | 102 | 143 | 90 | <u>86</u> | 132 | 75 | 110 | 6↑ |
| FedProx | Acc | 56.51 | 42.02 | 48.03 | 39.11 | 32.33 | 27.24 | 26.50 | 20.88 | 19.67 | 18.03 | 33.03 | 5.36↑ |
| | CoR | 146 | 134 | 112 | 139 | 119 | 140 | 125 | 105 | 132 | 99 | 125 | 21↑ |
| Fixed | Acc | 58.70 | 34.52 | 35.09 | 30.37 | 27.01 | 23.96 | 18.18 | 14.78 | 11.47 | 9.27 | 26.34 | 12.05↑ |
| | CoR | 137 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | / | / |
| DANN+FL | Acc | 58.82 | 44.12 | 46.84 | 39.66 | 31.54 | 27.93 | 24.21 | 24.03 | 21.32 | 19.73 | 33.82 | 4.57↑ |
| | CoR | 145 | 129 | 123 | 138 | 124 | 134 | 112 | 109 | 129 | 121 | 126 | 22↑ |
| Shared | Acc | 58.70 | 42.53 | 48.49 | 39.10 | 31.88 | 27.39 | 25.85 | 25.74 | 24.35 | 18.30 | 34.23 | 4.16↑ |
| | CoR | 137 | 117 | 82 | 137 | 113 | 137 | 103 | 97 | 135 | 89 | 115 | 11↑ |
| FCIL | Acc | 58.70 | 45.65 | 51.87 | 42.37 | 37.32 | 32.01 | 29.00 | 28.47 | 24.99 | 23.02 | 37.33 | 1.06↑ |
| | CoR | 137 | 123 | 77 | 134 | 105 | 140 | 96 | 88 | 130 | <u>73</u> | 110 | 6↑ |
| FedCIL | Acc | **61.20** | **47.05** | 49.66 | 38.14 | 32.69 | 24.11 | 23.90 | 23.99 | 19.89 | 17.98 | 33.86 | 4.53↑ |
| | CoR | 146 | 138 | 123 | 131 | 125 | 143 | 122 | 129 | 130 | 126 | 131 | 27↑ |
| Re-Fed | Acc | 58.70 | 43.66 | **53.53** | **40.17** | **38.71** | **35.96** | **31.25** | **28.77** | **27.53** | **25.61** | **38.39** | / |
| | CoR | <u>137</u> | <u>104</u> | 80 | <u>105</u> | <u>93</u> | <u>121</u> | <u>85</u> | 105 | <u>120</u> | 87 | <u>104</u> | |

Table 7. Performance comparisons of various methods on Tiny-ImageNet with 10 incremental tasks (20 new classes for each task).

| **Tiny-ImageNet** ($\alpha$ = 10.0) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Target | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 | 200 | Avg | $\Delta(\uparrow)$ |
| FedAvg | Acc | 85.80 | 68.58 | 57.22 | 43.75 | 40.52 | 41.13 | 34.10 | 29.59 | 28.40 | 27.58 | 45.67 | 5↑ |
| | CoR | 132 | 143 | 139 | 125 | 107 | 97 | 128 | 121 | 109 | 98 | 120 | 7↑ |
| FedProx | Acc | 82.02 | 66.15 | 54.32 | 40.57 | 38.80 | 38.99 | 30.59 | 24.12 | 22.76 | 21.82 | 42.01 | 8.66↑ |
| | CoR | 127 | 140 | 142 | 134 | 120 | 113 | 114 | 121 | 110 | 108 | 123 | 10↑ |
| Fixed | Acc | 85.80 | 51.07 | 30.94 | 28.11 | 25.30 | 24.26 | 19.48 | 17.18 | 14.66 | 12.34 | 30.91 | 19.76↑ |
| | CoR | 132 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | / | / |
| DANN+FL | Acc | 85.24 | 68.16 | 55.32 | 41.11 | 36.45 | 35.38 | 28.83 | 24.54 | 21.09 | 20.77 | 41.69 | 8.98↑ |
| | CoR | 138 | 140 | 141 | 131 | 124 | 126 | 137 | 128 | 121 | 123 | 131 | 18↑ |
| Shared | Acc | 85.80 | 67.21 | 56.49 | 42.05 | 40.17 | 37.59 | 28.61 | 25.90 | 23.89 | 22.19 | 42.99 | 7.68↑ |
| | CoR | 132 | 135 | 145 | 125 | 119 | 127 | 129 | 116 | 130 | 125 | 128 | 15↑ |
| FCIL | Acc | 85.80 | 71.94 | 61.02 | 50.73 | 44.25 | **42.40** | 36.96 | 34.51 | 31.36 | 29.58 | 48.86 | 1.81↑ |
| | CoR | 132 | 130 | 127 | 112 | 106 | 109 | 124 | 122 | 121 | 108 | 119 | 6↑ |
| FedCIL | Acc | **86.43** | 69.39 | 58.11 | 45.74 | 41.02 | 38.93 | 31.29 | 27.65 | 25.17 | 24.41 | 44.81 | 5.86↑ |
| | CoR | 146 | 144 | 137 | 121 | 117 | 126 | 132 | 140 | 124 | 129 | 132 | 19↑ |
| Re-Fed | Acc | 85.80 | **72.06** | **65.29** | **52.39** | **45.93** | 42.15 | **38.88** | **36.95** | **35.19** | **32.07** | **50.67** | / |
| | CoR | 132 | 120 | 126 | 121 | 91 | 103 | 110 | 114 | 112 | 92 | 113 | |

Table 8. Performance comparisons of various methods on Digit10 with 4 domains and Office-31 with 3 domains.

| | | **Digit10** ($\alpha$ = 0.1) | | | | | | **Office-31** ($\alpha$ = 1.0) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Target | MNIST | EMNIST | USPS | SVHN | Avg | $\Delta(\uparrow)$ | Amazon | Dlsr | Webcam | Avg | $\Delta(\uparrow)$ |
| FedAvg | Acc | 92.82 | 88.62 | 84.02 | 77.59 | 85.76 | 3.99↑ | 58.08 | 31.62 | 39.25 | 42.98 | 8.76↑ |
| | CoR | 112 | 82 | 96 | 122 | 103 | 22↑ | 144 | 136 | 135 | 138 | 9↑ |
| FedProx | Acc | 93.07 | 87.43 | 85.67 | 79.09 | 86.32 | 3.43↑ | 58.69 | 34.25 | 43.01 | 45.32 | 6.42↑ |
| | CoR | 114 | 93 | 89 | 118 | 103 | 22↑ | 145 | 146 | 139 | 143 | 14↑ |
| Fixed | Acc | 92.82 | 85.35 | 82.11 | 71.26 | 82.48 | 7.27↑ | 58.08 | 24.56 | 37.44 | 40.03 | 11.71↑ |
| | CoR | 112 | 0 | 0 | 0 | / | / | 144 | 0 | 0 | / | / |
| DANN+FL | Acc | **96.07** | 87.30 | 82.81 | 76.44 | 85.66 | 4.09↑ | **59.95** | 42.21 | 45.21 | 49.12 | 2.62↑ |
| | CoR | 132 | 107 | 116 | 129 | 120 | 39↑ | 149 | 144 | 141 | 145 | 16↑ |
| Shared | Acc | 92.82 | 82.10 | 80.36 | 74.77 | 82.51 | 7.24↑ | 58.08 | 35.33 | 37.55 | 43.65 | 8.09↑ |
| | CoR | 112 | 76 | 84 | 103 | 93 | 12↑ | 144 | 122 | 124 | 130 | 1↑ |
| FCIL | Acc | 92.82 | 88.62 | 84.02 | 77.59 | 85.76 | 3.99↑ | 58.08 | 31.62 | 39.25 | 42.98 | 8.76↑ |
| | CoR | 112 | 82 | 96 | 122 | 103 | 22↑ | 144 | 136 | 135 | 138 | 9↑ |
| FedCIL | Acc | 94.61 | 90.24 | 87.55 | 83.85 | 89.06 | 0.69↑ | 59.37 | 45.91 | 46.26 | 50.51 | 1.23↑ |
| | CoR | 118 | 86 | 92 | 125 | 105 | 24↑ | 146 | 139 | 148 | 144 | 15↑ |
| Re-Fed | Acc | 92.82 | **91.64** | **88.57** | **85.96** | **89.75** | / | 58.08 | **47.07** | **50.80** | **51.74** | / |
| | CoR | 112 | 68 | 73 | 71 | 81 | | 144 | 125 | 118 | 129 | |

Table 9. Performance comparisons of various methods on DomainNet with 6 domains.

| **DomainNet** ($\alpha$ = 10) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Target | Clipart | Infograph | Painting | Quickdraw | Real | Sketch | Avg | $\Delta(\uparrow)$ |
| FedAvg | Acc | 52.07 | 36.22 | 45.09 | 46.59 | 49.36 | 51.73 | 46.84 | 3.39↑ |
| | CoR | 141 | 128 | 97 | 108 | 136 | 115 | 121 | 11↑ |
| FedProx | Acc | 50.31 | 33.64 | 41.77 | 45.04 | 47.44 | 49.12 | 44.55 | 5.68↑ |
| | CoR | 136 | 131 | 115 | 130 | 137 | 116 | 128 | 1↑ |
| Fixed | Acc | 52.07 | 29.58 | 32.24 | 38.91 | 40.09 | 46.30 | 39.87 | 10.36↑ |
| | CoR | 141 | 0 | 0 | 0 | 0 | 0 | / | / |
| DANN+FL | Acc | **55.66** | 36.44 | 42.02 | 38.84 | 45.89 | 50.01 | 44.81 | 5.42↑ |
| | CoR | 142 | 126 | 109 | 112 | 137 | 121 | 125 | 15↑ |
| Shared | Acc | 52.07 | 35.22 | 37.83 | 35.19 | 40.52 | 41.76 | 40.43 | 9.80↑ |
| | CoR | 141 | 113 | 98 | 125 | 120 | 96 | 116 | 6↑ |
| FCIL | Acc | 52.07 | 36.22 | 45.09 | 46.59 | 49.36 | 51.73 | 46.84 | 3.39↑ |
| | CoR | 141 | 128 | 97 | 108 | 136 | 115 | 121 | 11↑ |
| FedCIL | Acc | 54.52 | 38.98 | 40.45 | 41.77 | 45.09 | 47.28 | 44.68 | 5.55↑ |
| | CoR | 148 | 136 | 128 | 112 | 142 | 125 | 132 | 22↑ |
| Re-Fed | Acc | 52.07 | **42.26** | **48.11** | **48.98** | **53.34** | **56.66** | **50.23** | / |
| | CoR | 141 | 103 | 97 | 109 | 118 | 91 | 110 | |

Table 10. Performance comparisons of three weighting methods for gradient norms in two incremental scenarios.

| Dataset | Class-Incremental Scenario | | | Domain-Incremental Scenario | | |
|---|---|---|---|---|---|---|
| | CIFAR10 | CIFAR100 | Tiny-ImageNet | Digit10 | Office31 | DomainNet |
| Early-Emphasis | **29.22** | **25.61** | **32.07** | **85.96** | **50.80** | **56.66** |
| Average-Weighting | 28.73 | 24.88 | 30.42 | 85.71 | 48.95 | 56.04 |
| Late-Emphasis | 26.57 | 22.18 | 28.08 | 84.36 | 47.29 | 53.90 |

## F. Analysis of the Federated Incremental-Learning Framework: Re-Fed

In this section, we prove the convergence of personalized informative models. To simplify the notation, here we conduct an analysis on a fixed task while the convergence does not depend on the IL setting. We first define following standard assumptions.

**Assumption 1** ($L_2$ Distance.) The $L_2$ distance between the optimal local models $\hat{w_k}:= \arg\min_{w_k}\{f(w_k)\}$ and the optimal global model $\hat{w}:= \arg\min_{w}\{\frac{1}{K}\sum_{k=1}^{K}\nabla f(w_k)\}$ is bounded by:

$$||\hat{w_k} - \hat{w}|| \leq M,\ \forall k \in [K]. \tag{7}$$

**Assumption 2** (Gradient Variance.) The variance of stochastic gradients is finite and bounded at all clients by:

$$\mathbb{E}\Big[||\nabla f(\hat{w_k})||^2\Big] \leq \sigma^2,\ \forall k \in [K]. \tag{8}$$

**Assumption 3** (Strong Convexity.) There exists $\mu_k \in \mathbb{R}_+$ and a unique solution $\hat{w_k}$:

$$f(w_k) - f(\hat{w_k}) \geq \langle \nabla f(\hat{w_k}), \hat{w_k} - w_k\rangle + \frac{\mu_k}{2}||w_k - \hat{w_k}||^2. \tag{9}$$

### F.1. Proof of Theorem 3.1

**Definition 1** (Personalized Informative Model Formulation.) Denote the objective of personalized informative model $v_k$ on client $k$ while $f(\cdot)$ is strongly convex as:

$$\begin{aligned} \hat{v_k}(\lambda) &:= \arg\min_{v_k}\Big\{f(v_k) + \frac{q(\lambda)}{2}||v_k - \hat{w}||^2\Big\} \\ q(\lambda) &:= \frac{1-\lambda}{2\lambda},\ \lambda \in (0,1) \end{aligned} \tag{10}$$

where $\hat{w}$ denotes the global model.

**Lemma 1** (Proportion of Global and Local Information.) *For all $\lambda \in (0,1)$ and $\lambda \to f(\lambda_k)$ is non-increasing:*

$$\begin{aligned} &\frac{\partial f(\hat{v_k}(\lambda))}{\partial \lambda} \leq 0 \\ &\frac{\partial ||\hat{v_k}(\lambda) - \hat{w}||^2}{\partial \lambda} \geq 0. \end{aligned} \tag{11}$$

Then, for $k \in [K]$, we can get:

$$\lim_{\lambda \to 0} \hat{v_k}(\lambda) := \hat{w}. \tag{12}$$

*Proof.* The proof here directly follows the proof in Theorem 3.1 [10]. As $\lambda$ declines and $q(\lambda)$ grows, the objective of Eq. 10 tends to optimize $||v_k - \hat{w}||^2$ and increase the local empirical training loss $f(v_k)$, leading to the convergence on the global model. Hence we can modify the $\lambda$ value to adjust the optimization direction of our model $v_k$ thus the dominance of local and global model information.

**Theorem 3.1** (Personalized Informative Model.) *Assuming the global model $w^t$ converges to the optimal model $\hat{w}$ with $g(t)$ for any client $k \in [K]$ at each communication round $t$: $\mathbb{E}\Big[||w^t - \hat{w}||^2\Big] \leq g(t)$ and $\lim_{t\to\infty} g(t) = 0$, then there exists a constant $C < \infty$ such that the personalized informative model $v_k^t$ can converge to the optimal model $\hat{v_k}$ with $Cg(t)$.*

*Proof.* Here we first introduce the Lemma 2 here proved by [21] Lemma 13.

**Lemma 2** ([21] Lemma 13.) Under assumptions above, $f(v_k)$ is $\mu_k$-strongly convex at each communication round $t$, we have:

$$\begin{aligned} \mathbb{E}\Big[||v_k^{t+1} - \hat{v_k}||^2\Big] \leq &(1 - \eta(\mu_k + q(\lambda)))\,\mathbb{E}\Big[||v_k^t - \hat{v_k}||^2\Big] + \eta^2\left(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k})\right)^2 + \eta^2 q(\lambda)^2\mathbb{E}\Big[||w^t - \hat{w}||^2\Big] \\ &+ 2\eta^2 q(\lambda)\left(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k})\right)\sqrt{\mathbb{E}\Big[||w^t - \hat{w}||^2\Big]} + 2\eta q(\lambda)\sqrt{\mathbb{E}\Big[||v_k^t - \hat{v_k}||^2\Big]\mathbb{E}\Big[||w^t - \hat{w}||^2\Big]}. \end{aligned} \tag{13}$$

Assume $g(t+1) \leq g(t)$ and let positive number $A$ be chosen such that $A(g(t)-g(t+1)) \leq g^2(t)$, and we arrive at $(1-\frac{g(t)}{A})g(t) \leq g(t+1)$. Then, we prove the **Theorem 3.2** by induction. Assuming that $\mathbb{E}\Big[||v_k^t - \hat{v_k}||^2\Big] \leq Cg(t)$ where $C > 0$ and $C \geq \frac{\mathbb{E}\Big[||v_k^0 - \hat{v_k}||^2\Big]}{g(0)}$, the learning rate $\eta = \frac{2g(t)}{A(\mu_k + q(\lambda))}$, here we can continue with **Lemma 2**:

$$\begin{aligned}\mathbb{E}\Big[||v_k^{t+1} - \hat{v_k}||^2\Big] \leq & (1 - \frac{2g(t)}{A})Cg(t) + \frac{4q(\lambda)\sqrt{C}g(t)}{A(\mu_k + q(\lambda))} \\ & + \frac{4g(t)^2}{A^2(\mu_k + q(\lambda))^2}\left((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\right).\end{aligned} \tag{14}$$

Therefore, if we let $C = \max\{\frac{\mathbb{E}\Big[||v_k^0 - \hat{v_k}||^2\Big]}{g(0)}, 16, \frac{4\Big((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\Big)}{A(\mu_k + q(\lambda))^2(1 - \frac{1}{(1 + \frac{\mu_k}{q(\lambda)})})}\}$, then we have:

$$\begin{aligned}&\frac{4q(\lambda)\sqrt{C}g(t)^2}{A(\mu_k + q(\lambda))} + \frac{4g(t)^2}{A^2(\mu_k + q(\lambda))^2}\left((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\right) \leq \\ &\frac{q(\lambda)Cg(t)^2}{A(\mu_k + q(\lambda))} + \frac{4g(t)^2}{A^2(\mu_k + q(\lambda))^2}\left((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\right) = \\ &\frac{Cg(t)^2}{A} \cdot \frac{1}{(1 + \frac{\mu_k}{q(\lambda)})} + \frac{4g(t)^2}{A^2(\mu_k + q(\lambda))^2}\left((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\right) \leq \\ &\frac{Cg(t)^2}{A} \cdot \frac{1}{(1 + \frac{\mu_k}{q(\lambda)})} + \frac{g(t)^2}{A^2} \cdot CA\left(1 - \frac{1}{(1 + \frac{\mu_k}{q(\lambda)})}\right) = \frac{Cg(t)^2}{A}.\end{aligned} \tag{15}$$

The first inequality uses the fact that $16 \leq C$ and consequently $4\sqrt{C} \leq C$. The second inequality results from the definition of $C$ as $\frac{4\Big((\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))^2 + q(\lambda)^2 g(t) + 2q(\lambda)\sqrt{g(t)}(\sigma + q(\lambda)(M + \frac{\sigma}{\mu_k}))\Big)}{A(\mu_k + q(\lambda))^2}\} \leq C(1 - \frac{1}{(1 + \frac{\mu_k}{q(\lambda)})})$. Hence, combining the results of 14 and 15 yields

$$\begin{aligned}\mathbb{E}\Big[||v_k^{t+1} - \hat{v_k}||^2\Big] &\leq (1 - \frac{2g(t)}{A})Cg(t) + \frac{Cg(t)^2}{A} \\ &= (1 - \frac{g(t)}{A})Cg(t) \\ &\leq Cg(t+1),\end{aligned} \tag{16}$$

and we have the desired result.