# Forget Less, Learn More: Contrastive-Based Federated Class Incremental Learning with a Low-Dimensional Projection Layer

Ensieh Khazaei Dimitrios Hatzinakos
University of Toronto
ensieh.khazaei@mail.utoronto.ca, dimitris@comm.utoronto.ca

## Abstract

*Federated Class Incremental Learning (FCIL) extends Federated Learning (FL) to dynamic environments where clients continually encounter new classes over time, but past data becomes inaccessible. This leads to catastrophic forgetting of previous classes. Therefore, maintaining a balance between learning new classes (plasticity) and retaining past knowledge (stability) is crucial. To address these challenges, we propose a contrastive-based FCIL framework with a low-dimensional projection layer to enhance both stability and plasticity. A low-dimensional projection layer is introduced to improve the generalization capability of the feature extractor and stability-plasticity trade-off. In this regard, we employ supervised contrastive learning in the projection layer during the initial task to strengthen the feature extractor for better adaptation to new classes. Additionally, we train a data-free generator on the server and distribute it to clients to replay synthetic samples from past tasks. To balance stability and plasticity, we design a novel loss function that integrates cross-entropy loss, feature-level knowledge distillation loss, contrastive loss in the projection layer, and classification head refinement. Extensive experiments demonstrate that our framework outperforms state-of-the-art FCIL baselines, achieving higher accuracy and lower forgetting. The code is available at* https://github.com/EnsiehKhazaei/FCIL-LD.

## 1. Introduction

Federated learning (FL) is an emerging distributed learning paradigm that is designed to collaboratively train a global model among multiple clients without sharing their raw data. FL improves the privacy and security of clients and attracts a lot of attention specifically in privacy-sensetive applications like healthcare [1], finance [3], and IoT [6, 10, 11]. In a typical FL algorithm, it is assumed that the data distribution is fixed during the training process. However, in realistic scenarios, data is shown up sequentially, and the distribution of data changes over time. Therefore, data samples from new classes may appear, while the previous data may become inaccessible due to privacy constraints or memory limitations. In this regard, algorithms are required to perform well on current data while preserving the knowledge of previous data, introducing the need for continual learning approaches in federated settings [23, 29].

Continual learning (CL) aims to adapt to new data (plasticity) without forgetting previously learned knowledge (stability). CL can be categorized into different types [30], such as task incremental learning, where each task comes with a unique identifier, making it easier to distinguish among tasks; domain incremental learning, where the same set of classes is observed across different domains; and class incremental learning, where new classes are learned incrementally without task identifiers. Among these, class incremental learning is more challenging, as it requires the model to distinguish between all classes learned so far without task identifiers. The catastrophic forgetting issue in class incremental problems is exacerbated in federated settings by data heterogeneity, local resource limitations, and privacy considerations. The combination of these challenges makes federated class incremental learning (FCIL) a particularly difficult problem to address [23, 29].

In this work, we propose a novel contrastive-based FCIL framework with a low-dimensional projection layer (FCIL-LD) to enhance the learning of new classes (plasticity) and minimize the forgetting of previously learned tasks (stability). The general overview of our proposed framework is shown in Figure 1. We emphasize the importance of the feature extractor's quality and generalization capability in the initial task, as it significantly impacts the performance and forgetting in subsequent tasks. In this regard, we employ supervised contrastive (SupCon) loss [12] in a low-dimensional projection layer during the initial task. This enhances the feature extractor's generalization ability, enabling it to extract more informative features for classification and adapt more effectively to new tasks. To mitigate catastrophic forgetting, our approach trains a generator on the server side at the end of each task using data-free knowl-

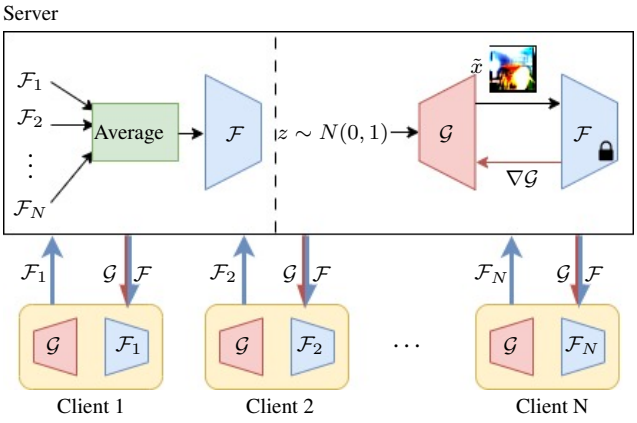


Figure 1. Overview of FCIL-LD framework. Local clients transmit their models ($\mathcal{F}_k$) to the server, where model aggregation is performed to update the global model $\mathcal{F}$. Subsequently, the global model is frozen, and the generator $\mathcal{G}$ is updated in a data-free manner. The updated global model $\mathcal{F}$ and generator $\mathcal{G}$ are then distributed back to the clients for training..

edge distillation techniques. This generator is then broadcast to the clients, where it generates data samples from previous tasks for replay. Additionally, a new loss function is introduced for the subsequent tasks to improve performance while reducing forgetting. The proposed loss function includes a cross-entropy loss for new data samples, a feature-level knowledge distillation loss on generated samples of previous tasks to alleviate catastrophic forgetting, a contrastive loss in the low-dimensional projection layer to balance stability and plasticity, and a classification head refinement loss to minimize the distribution gap between real and generated samples.

The main contributions of this work are summarized as follows.

- We propose a novel FCIL framework that efficiently adapts to new tasks while minimizing catastrophic forgetting.
- We introduce a low-dimensional projection layer to improve the generalization capability of the feature extractor and its power to adapt to new tasks while preserving knowledge from previous tasks.
- We design a new loss function to effectively balance the trade-off between stability and plasticity during client training.
- We conduct comprehensive experiments on four public datasets of CIFAR100, CIFAR10, PPMI, and VOC2012 to demonstrate the superiority of our proposed framework. Our framework improves last task accuracy by up to 48%, average accuracy by up to 55%, and reduces forgetting by up to 11% compared to the state-of-the-art FCIL baselines, highlighting its effectiveness in balancing the stability-plasticity trade-off.

## 2. Related Work

### 2.1. Class Incremental Learning

Class-Incremental Learning (CIL) [20] is a challenging continual learning problem where models incrementally learn new classes without forgetting previous knowledge, even in the absence of task identifiers. Various approaches have been proposed in the literature to address this problem, which can be categorized into replay-based methods [16, 26], structure-based methods [33], and regularization-based methods [5, 35]. Replay-based methods maintain a small memory buffer of representative samples from previous tasks or use generative models to synthesize data from earlier distributions. For instance, iCaRL [24] employs knowledge distillation alongside replay buffers to mitigate forgetting, while DGR [27] uses data generators instead of stored examples to simulate previous tasks, making it suitable for scenarios where data storage is restricted. Structure-based methods modify or expand the model's architecture to accommodate new tasks without interfering with older ones. PackNet [19] and DEN [33] are examples of this approach, utilizing selective capacity expansion or pruning to optimize resource usage. Regularization-based methods aim to preserve prior knowledge by constraining model parameter updates. EWC [13] achieves this by penalizing changes to parameters important for previous tasks and LwF [17] employs knowledge distillation to transfer information from previous class distributions to the current model, enabling it to remember old knowledge while adapting to new data.

### 2.2. Federated Continual Learning

Federated Continual Learning (FCL) [4, 8] merges the principles of CL and FL to address the challenges of learning from non-stationary and dynamic data distributions in decentralized systems. In real-world scenarios, federated systems encounter evolving data streams, such as changing user preferences or device behaviors, where new tasks or classes emerge over time. FCL seeks to enable models to adapt to new data while preserving knowledge of previous tasks. The distributed nature of federated learning, combined with the dynamic and sequential nature of the data streams, makes FCL a complex problem. Despite its practical importance, FCL remains relatively underexplored with few solutions proposed in the literature.

In [34], FedWeIT is introduced as an FCL framework to address the challenges of minimizing interference from irrelevant tasks while enabling effective knowledge transfer across clients. Despite its promising results as an early effort in FCL, it focuses on improving the performance of individual clients while the FCIL-LD framework prioritizes the development of a unified global model that incrementally learns from all participating clients. Moreover, FedWeIT [34] is in the category of task incremental learning

that relies on task identifiers during inference, whereas our work addresses the more complex CIL problem that aims to classify all encountered classes using a single global model without task identifiers. FedLwF-2T [29] is a distillation-based approach that transfers knowledge from two teacher models—the client's past model and the global model—to the client's current model. Although FedLwF-2T is proposed for the complex FCIL problem, it is applied exclusively to sensor signals rather than vision tasks. Dong *et al.* [4] propose the GLFC model to address both local and global forgetting in the FCIL problem. To mitigate local forgetting, it incorporates a class-aware gradient compensation loss and a class-semantic relation distillation loss, and a proxy server is employed to tackle global forgetting. However, GLFC assumes a memory buffer for each client, which is impractical in realistic scenarios where clients have limited storage resources. FedCIL [23] is another work that trains generators locally to synthesize data for replay in future tasks and alleviate forgetting. These generators are shared with the server to consolidate the global model. However, training generators locally imposes computational and memory demands, which are unavailable in many clients, and sharing these generators with the server raises privacy concerns. MFCL [2] addresses the limitations of FedCIL [23] by leveraging data-free knowledge distillation techniques to train a generator on the server side and the ABD loss [28] function is employed for local client training. Similarly, our proposed framework adopts a data-free generator training approach inspired by MFCL [2] but modifies the local client training process to achieve a better balance between stability and plasticity.

## 3. Methodology

Class incremental learning in the federated setting aims to train a global model collaboratively across multiple clients, where each client receives data incrementally across $T$ tasks. Each task introduces a new set of classes, and there is no overlap between the classes of different tasks. The dataset for client $k$ and task $t$ is denoted as $D_t^k = \{(x_i, y_i) \mid x_i \in \mathcal{X}_t^k, y_i \in \mathcal{Y}_t^k\}$. Here, $\mathcal{X}_t^k$ represents the input space, and $\mathcal{Y}_t^k$ corresponds to the label space for task $t$ on client $k$. The overall label space for task $t$ is defined as $\mathcal{Y}_t = \bigcup_{i=1}^{N} \mathcal{Y}_i^k$, where $N$ is the number of clients. By the end of task $t$, the total number of classes is $q = \sum_{i=1}^{t} |\mathcal{Y}_i|$. The goal is to train a global model $\mathcal{F}$ that can generalize across clients with different data distributions while minimizing catastrophic forgetting of previous tasks.

To address the above problem, we propose the FCIL-LD framework with better performance among all tasks and less catastrophic forgetting for previous tasks. Our proposed framework includes three main stages: 1) the initial task is trained using the SupCon loss [12] in a low-dimensional projection layer to enhance the generalization capability of the feature extractor, 2) at the end of each task, after integration of local models $\mathcal{F}_k$ from all clients, a generator model $\mathcal{G}$ is trained using the frozen global model $\mathcal{F}$ and data-free knowledge distillation methods [28, 32]. This generator synthesizes samples from previous tasks, which addresses catastrophic forgetting and ensures privacy by eliminating the need to store real client data, 3) the server broadcasts the updated global model $\mathcal{F}$ and generator $\mathcal{G}$ to all clients. On the client side, the local models $\mathcal{F}_k$ are updated using a novel loss function. This loss improves learning on the current task while mitigating forgetting of previous tasks by leveraging the samples generated by $\mathcal{G}$.

In the following, we elaborate on the key operations at the server and client sides of the FCIL-LD framework.

### 3.1. Server-Side Data-Free Generator Training

In a typical FL framework, at the end of each communication round, local models' weights are sent to the server and they are integrated by averaging among all of them. The updated global model is sent back to clients to transfer global knowledge to them. However, in the CIL, we only have access to data of the current task and previous classes' data is unavailable. The unavailability of previous data is challenging and causes catastrophic forgetting. In some research, memory is defined to store a selected set of samples from previous tasks, but this approach is not applicable in the federated setting since local clients have limited resources to store such samples. On the other hand, if we want to store those samples on the server, it raises privacy concerns since local clients are reluctant to reveal their local data to an external server. Another line of work explores training generators locally, using data available on individual clients. However, this approach needs substantial computational resources, which are often unavailable on local devices.

To alleviate these challenges, we follow [28] and employ a data-free knowledge distillation method on the server to train a generator that synthesizes samples from the previous tasks. The generator is then shared with the clients to generate samples during local training and reduce catastrophic forgetting. This approach has two advantages: 1) the server is powerful and has sufficient computational units to train the generator, and 2) it does not raise privacy concerns since no data sample is required for generator training.

At the end of each task, after traditional aggregation of local models' weights, the global model becomes frozen and the generator starts training by the algorithm proposed in [28]. The generator is trained using four loss components, as described below, to generate images similar to the previous tasks' distributions.

- Cross-entropy loss: This component forces the generator to produce images that can be classified correctly using the frozen global model. The cross-entropy loss is defined as:

$$\mathcal{L}_{CE} = CE(\text{argmax}(z[:q]), \mathcal{F}(\tilde{x})) \quad (1)$$

where $z$ is the noise vector input to the generator and $\tilde{x}$ is the synthetic sample generated from $z$, where $\tilde{x} = \mathcal{G}(z)$. $q$ denotes the total number of classes at the end of task $t$.

- Diversity loss: Since $\mathcal{G}$ has challenge to produce synthetic data from different classes uniformly, the information entropy is used to promote class diversity in each batch. In this regard, the entropy of the average of class prediction vector for a batch of synthetic data is maximized as follows:

$$\mathcal{L}_{div} = -\mathcal{H}_{\text{info}}(\frac{1}{B}\sum_{b=1}^{B}\mathcal{F}(\tilde{x}_b)) \quad (2)$$

where $\mathcal{H}_{\text{info}}$ is the information entropy and $B$ is the batch size. By maximizing the information entropy, the generator is guided to uniformly synthesize samples from all classes.

- Batch normalization loss: Prior work has demonstrated that in model inversion, the feature distribution of the discriminator model on real samples can differ significantly from that on synthetic samples. In [32], it is shown that minimizing the distance between the statistics (mean and standard deviation) of the global model's batch normalization layers for real and synthetic samples effectively addresses this distribution shift. To enforce similarity in feature distributions, a batch statistics loss is defined as:

$$\begin{aligned}\mathcal{L}_{BN} &= \frac{1}{L}\sum_{l=1}^{L} KL(\mathcal{N}(\mu_l, \sigma_l^2)||\mathcal{N}(\tilde{\mu}_l, \tilde{\sigma}_l^2)) \\ &= \frac{1}{L}\sum_{l=1}^{L}\log\frac{\tilde{\sigma}_l}{\sigma_l} - \frac{1}{2}(1 - \frac{\sigma_l^2 + (\mu_l - \tilde{\mu}_l)^2}{\tilde{\sigma}_l^2})\end{aligned} \quad (3)$$

where $L$ is the number of batch normalization layers and $KL$ is the Kullback-Leibler divergence. The $\mu_l$ and $\sigma_l$ correspond to the mean and standard deviation of $l$-th batch normalization layer for real samples and $\tilde{\mu}_l$ and $\tilde{\sigma}_l$ denote the respective statistics for synthetic samples.

- Smoothness loss: Natural images typically exhibit smooth transitions between neighboring pixels. To enhance the smoothness of synthetic images, a smoothness prior loss is added to the generator's loss function. This loss minimizes the discrepancy between synthetic images and their smoothed versions as follows:

$$\mathcal{L}_{prior} = ||\tilde{x} - \tilde{x}_{\text{smooth}}||_2^2 \quad (4)$$

where $\tilde{x}_{\text{smooth}}$ is the generated image after applying a Gaussian kernel to smooth it.

Finally, the training loss function for training the generator on the server side can be written as:

$$\min_{\mathcal{G}} \mathcal{L}_{CE} + \alpha_{div}\mathcal{L}_{div} + \alpha_{BN}\mathcal{L}_{BN} + \alpha_{prior}\mathcal{L}_{prior} \quad (5)$$

here, $\alpha_{div}$, $\alpha_{BN}$, and $\alpha_{prior}$ work as hyperparameters to calibrate the contributions of the diversity, batch statistics, and smoothness prior loss terms, respectively.

## 3.2. Client-Side Training using Contrastive Loss

In the client-side training, a low-dimensional projection layer is designed. While this layer does not directly contribute to the classification phase, it enhances the generalization of the feature extractor for the initial task. Additionally, it improves the balance between stability and plasticity when adapting to subsequent tasks. Details regarding the local loss function and its training procedure are provided in the following.

### 3.2.1 Initial Task: Enhancing Feature Extractor Generalization

While the quality of the feature extractor in the initial task is important in addressing the CIL problem, it is not thoroughly investigated in the literature. Some prior works [7, 18, 22] in CIL train the feature extractor during the initial task and freeze it for the later tasks. These approaches aim to maintain the knowledge of the first task and adapt to future tasks by manipulating the rest of the model, such as the classification head. However, they overlook the importance of the feature extractor's quality and generalization during the initial task and its impact on overall system performance. We believe improving the generalization of the feature extractor in the initial task has two advantages: 1) it enhances performance on the initial task due to increased robustness to input variations, and 2) it facilitates better adaptation in subsequent tasks since it has better generalization.

To improve the generalization of the feature extractor in the initial task, we use the SupCon loss [12] in a low-dimensional projection layer $\psi$, as shown in Figure 2. Although this projection layer is not used directly for the classification, it enhances the quality of the feature extractor, leading to performance improvement in the classification phase. The high-quality feature extractor ensures better feature representations are passed to the classification head. The projection layer operates in a lower-dimensional space ($\mathcal{R}^{D_P}$) compared to the output of the feature extractor ($\mathcal{R}^{D_F}$), where $D_P < D_F$. This lower dimensionality reduces the computational cost of applying SupCon loss and focuses on more important patterns to extract more generalized features.

To compute the SupCon loss in each batch $D_b^{org} = \{(x_i, y_i)\}_{i=1}^{BS}$, the samples are randomly augmented to create $D_b^{aug} = \{(x_i^{aug}, y_i)\}_{i=1}^{BS}$. These two sets are then concatenated to form $D_b = D_b^{org} \cup D_b^{aug}$ as a richer set of samples. The model processes this augmented batch, and the SupCon loss contrasts all samples from the same class (pos-

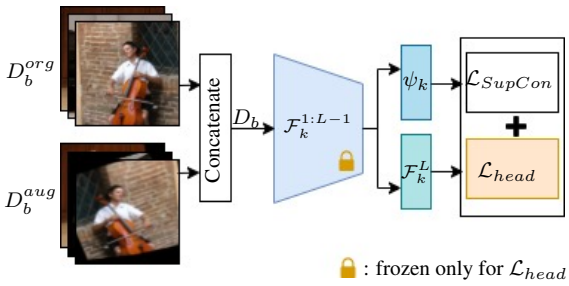


Figure 2. Initial task training of the $k$-th client. The original batch $D_b^{org}$ and its augmented version $D_b^{aug}$ are concatenated and fed to the model. The $\mathcal{L}_{SupCon}$ is applied to the projection layer ($\psi_k$) output to align representations of samples with the same label and $\mathcal{L}_{head}$ updates the classification head ($\mathcal{F}_k^L$).

itives) against the negatives (samples from different classes) within $D_b$. The SupCon loss for a batch from the $k$-th client is defined as follows:

$$\mathcal{L}_{SupCon} = \sum_{i=1}^{|D_b|} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(cos(z_i, z_p)/\tau)}{\sum_{j \in A(i)} \exp(cos(z_i, z_j)/\tau)} \tag{6}$$

Where $P(i)$ is the set of sample indices with the same label as sample $x_i$, and $A(i)$ is the set of all sample indices except $i$. The output of the low-dimensional projection layer for $i$-th sample is shown by $z_i = \psi_k(\mathcal{F}_k^{1:L-1}(x_i))$, where $\mathcal{F}_k^{1:L-1}$ is the feature extractor. The parameter $\tau$ is the temperature, which controls the penalty strength on hard negative samples, while $\cos(.,.)$ is the cosine similarity metric.

The main goal of the SupCon loss is to align the representations of different samples in the same class. This alignment facilitates the training of a more robust feature extractor and improves its generalization while it has negligible memory usage and computational cost. To train the classification head, the gradient propagation is blocked from the classification head back to the feature extractor, and the cross-entropy loss is applied to the linear head as follows:

$$\mathcal{L}_{head} = CE(\mathcal{F}_k^*(x), y) \tag{7}$$

where $\mathcal{F}_k^*$ represents that updates restricted to the classification head of $k$-th model.

The loss function for the initial task is defined using a combination of SupCon loss and cross-entropy one as:

$$\mathcal{L}_{init} = \mathcal{L}_{head} + \tau \mathcal{L}_{SupCon} \tag{8}$$

#### 3.2.2 Subsequent Tasks: Balancing Stability and Plasticity

In the FCIL, new tasks are introduced over time while data from previous tasks is no longer available. This scenario poses a critical challenge in addressing the stability-plasticity tradeoff during client training. To mitigate the forgetting of previous tasks, we leverage the generator described in Section 3.1 to synthesize samples representative of earlier tasks. However, training the local models using a standard cross-entropy loss applied to the combined set of real and synthesized samples is ineffective due to the inherent distribution shift between these two types of data.

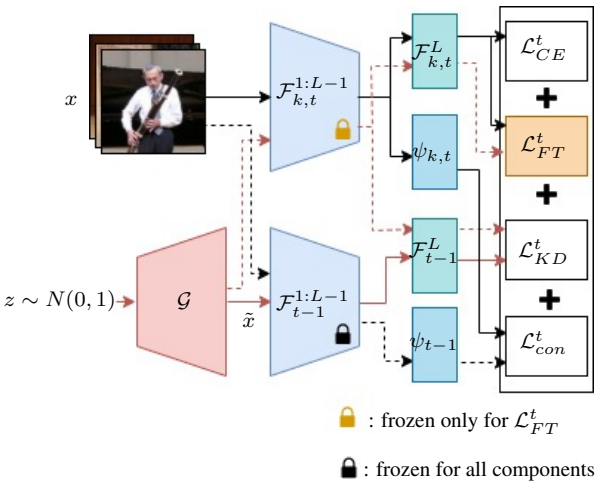


Figure 3. Training process of the $k$-th client for subsequent tasks. The model $\mathcal{F}_{k,t}$ is updated using new task samples $x$ and generated samples from previous tasks $\tilde{x}$. The optimization includes four loss components: cross-entropy loss $\mathcal{L}_{CE}^t$ for new data, knowledge distillation loss $\mathcal{L}_{KD}^t$ for transferring knowledge from the previous global model, contrastive loss $\mathcal{L}_{con}^t$ in the low-dimensional projection layer to balance stability and plasticity, and classification head refinement loss $\mathcal{L}_{FT}$.

To overcome these limitations, we are inspired by the loss function proposed in [28] and add a novel contrastive term in the low-dimensional projection layer. This term is designed to improve the stability-plasticity tradeoff, ensuring better adaptability to new tasks without compromising performance on earlier ones. In the following, a detailed explanation of the four components of the local loss and their contributions to the training process is described. The training process of local clients is shown in Figure 3.

- Cross-entropy loss: This component focuses on updating the feature extractor and the new part of the classification head by applying cross-entropy loss exclusively to the new data samples:

$$\mathcal{L}_{CE}^t = \mathcal{L}_{CE}(\mathcal{F}_{k,t}(x), y) \tag{9}$$

  where $\mathcal{F}_{k,t}$ is the model for $k$-th client at task $t$. With this loss component, the model is adapted to the new data by updating only the new portion of the classification head, while excluding any modifications to the past classes.
- Knowledge distillation loss: To address catastrophic

forgetting of previously learned classes, a feature-level knowledge distillation component is applied exclusively to the synthetic samples of previous tasks, unlike prior works [28] that apply it on both new and past data. This component specifically updates the feature extractor to keep the most important aspects of past task feature representations while allowing less relevant ones to adapt to new tasks. The knowledge distillation loss is defined as:

$$\mathcal{L}_{KD}^{t} = ||\mathcal{F}_{t-1}^{L}(\mathcal{F}_{k,t}^{1:L-1}(\tilde{x})) - \mathcal{F}_{t-1}^{L}(\mathcal{F}_{t-1}^{1:L-1}(\tilde{x}))||_2^2 \tag{10}$$

where $\mathcal{F}_{k,t}^{1:L-1}$ represents the feature extractor of the $k$-th client at task $t$, $\mathcal{F}_{t-1}^{1:L-1}$ is the feature extractor of the global model from the previous task, and $\mathcal{F}_{t-1}^{L}$ denotes the classification head of the global model from the previous task. Features with higher magnitudes in $\mathcal{F}_{t-1}^{L}$ are more critical for preserving past knowledge, while less relevant features can accommodate new task learning.

- Contrastive loss: Instead of applying feature-level knowledge distillation to new data, we introduce an additional contrastive learning term on the current task samples to balance between stability and plasticity. Applying knowledge distillation to new class feature representations besides past synthetic samples would force the feature extractor to mimic both important and unimportant aspects of past feature space, limiting its adaptability. In contrast, contrastive learning with cosine similarity only aligns the direction of representations while allowing flexibility in amplitude, resulting in better stability-plasticity balance. This loss component is expressed as:

$$\mathcal{L}_{con}^{t} = -\log \frac{\exp(cos(z_i^t, z_i^{t-1}))}{\sum_{j=1}^{BS} \exp(cos(z_i^t, z_j^{t-1}))} \tag{11}$$

where $z_i^t = \psi_{k,t}(\mathcal{F}_{k,t}^{1:L-1}(x_i))$ and $z_i^{t-1} = \psi_{t-1}(\mathcal{F}_{t-1}^{1:L-1}(x_i))$.
This contrastive component aims to align the representations of current task samples in the low-dimensional projection layer of the current model with those of the previous model, ensuring knowledge retention while enabling adaptation. Applying contrastive learning directly to the high-dimensional feature space can make alignment difficult due to the complexity of the feature distribution. Furthermore, if contrastive learning loss is applied directly at the feature extractor's output, the model may preserve the irrelevant part of past representations, restricting its ability to learn new tasks. By introducing a low-dimensional projection layer, we allow the model to selectively retain critical information while having greater flexibility to adjust representations for new classes.
- Fine-tuning loss: Finally, to prevent the model from distinguishing between real and synthetic data only based on their distribution shift, the feature extractor of the model is frozen and the classification head is fine-tuned by the conventional cross-entropy loss on the union of real and synthetic samples:

$$\mathcal{L}_{FT}^{t} = CE(\mathcal{F}_{k,t}^{*}([x, \tilde{x}]), y) \tag{12}$$

where $\mathcal{F}_{k,t}^{*}$ represents that updates are restricted to the classification head of the local model at current task $t$.

In summary, the final optimization objective combines the loss components as follows:

$$\min_{\mathcal{F}_t} \mathcal{L}_{CE}^{t} + \alpha_{KD}\mathcal{L}_{KD}^{t} + \alpha_{con}\mathcal{L}_{con}^{t} + \alpha_{FT}\mathcal{L}_{FT}^{t} \tag{13}$$

where $\alpha_{KD}$, $\alpha_{con}$, and $\alpha_{FT}$ are hyperparameters to regulate the contributions of each loss component.

# 4. Experiments

## 4.1. Experimental Setup

**Datasets.** To evaluate the efficacy of the FCIL-LD, we use four commonly-used vision datasets: CIFAR-10 [14], CIFAR-100 [14], VOC2012, and PPMI [31]. CIFAR-10 and CIFAR-100 are image classification datasets. VOC2012 and PPMI are image-based human activity recognition (HAR) datasets where PPMI includes interaction images with 12 different musical instruments, and VOC2012 contains 10 action classes.The data for each task is distributed among clients using a Dirichlet distribution with a concentration parameter of $\alpha = 1.0$ [25]. To distribute data among tasks, we follow the non-overlapping method proposed in [2], which introduces greater difficulty compared to [23], where tasks from different clients are highly overlapped. The number of tasks is set to 5 for CIFAR-10, CIFAR-100, and VOC2012, and 6 for PPMI in all experiments unless otherwise specified.

**Baselines.** We conduct a comprehensive comparative analysis with state-of-the-art baselines, including FedAvg [21], FedProx [15], FedLwF-2T [29], FedCIL [23], and MFCL [2], as well as the Oracle approach, as the upper bound for the performance, where each client in each task has access to the training data of all previous tasks. FedAvg and FedProx are standard FL approaches. FedLwF-2T is a knowledge distillatin based method without exemplar memory. FedCIL employs GANs to address FCIL problem, while MFCL is a data-free generator method designed to improve performance without relying on local data.

**Settings.** We use the Stochastic Gradient Descent (SGD) optimizer with a constant learning rate of 0.01 and a weight decay of 0.00001 across all settings. The batch size is uniformly set to 32. Following [9, 23], the number of clients is set to 5 where 60% of clients are randomly involved in each communication round. Each task is conducted over 50 communication rounds, with the number of local epochs set to 5 for CIFAR-10, VOC2012, and PPMI datasets, and 10

Table 1. Performance of the different baselines in terms of last task accuracy, average accuracy, and average forgetting for all datasets.

| | PPMI | | | VOC2012 | | | CIFAR-10 | | | CIFAR-100 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{A}_6$ | $\bar{\mathcal{A}}$ | $\bar{f}$ | $\mathcal{A}_5$ | $\bar{\mathcal{A}}$ | $\bar{f}$ | $\mathcal{A}_5$ | $\bar{\mathcal{A}}$ | $\bar{f}$ | $\mathcal{A}_5$ | $\bar{\mathcal{A}}$ | $\bar{f}$ |
| Oracle | 78.58 ±0.29 | 87.81 ±0.19 | – – | 66.47 ±1.11 | 77.41 ±1.14 | – – | 88.79 ±0.72 | 90.58 ±0.49 | – – | 61.38 ±0.55 | 65.20 ±0.34 | – – |
| FedAvg [21] | 15.76 ±0.15 | 39.20 ±0.09 | 95.92 ±0.43 | 17.57 ±0.14 | 40.60 ±0.37 | 90.26 ±0.89 | 19.38 ±0.07 | 43.33 ±0.02 | 94.33 ±0.17 | 16.20 ±0.08 | 33.61 ±0.17 | 74.09 ±0.31 |
| FedProx [15] | 15.67 ±0.23 | 39.22 ±0.20 | 95.97 ±0.53 | 17.56 ±0.17 | 40.58 ±0.16 | 89.44 ±1.30 | 19.42 ±0.09 | 43.13 ±0.38 | 94.16 ±0.63 | 16.01 ±0.42 | 33.58 ±0.27 | 73.66 ±0.51 |
| FedLwF-2T [29] | 18.72 ±0.52 | 52.02 ±0.63 | 86.53 ±1.16 | 18.80 ±0.57 | 48.55 ±1.75 | 79.32 ±1.18 | 19.58 ±0.24 | 45.89 ±0.33 | 78.94 ±4.33 | 28.86 ±0.67 | 46.47 ±1.20 | 41.35 ±1.36 |
| FedCIL [23] | 8.28 ±0.15 | 22.27 ±0.06 | 27.08 ±8.09 | 10.26 ±0.42 | 26.42 ±2.48 | 30.37 ±1.31 | 16.17 ±2.57 | 37.03 ±1.56 | 26.61 ±2.37 | 2.87 ±0.14 | 10.83 ±0.89 | 11.86 ±1.04 |
| MFCL [2] | 51.07 ±3.56 | 70.67 ±4.99 | 45.03 ±2.40 | 46.03 ±3.77 | 67.00 ±1.85 | 49.03 ±5.26 | 53.57 ±2.44 | 67.54 ±5.92 | 36.21 ±3.06 | 40.30 ±0.62 | 53.18 ±0.49 | 26.53 ±0.41 |
| FCIL-LD | **57.21** ±3.46 | **78.01** ±1.68 | **34.20** ±4.82 | **49.21** ±2.68 | **68.74** ±0.99 | **40.27** ±3.52 | **56.64** ±3.27 | **72.55** ±0.48 | **30.79** ±3.37 | **47.19** ±0.61 | **60.34** ±0.71 | **23.35** ±1.40 |

for CIFAR-100. We employ the ResNet-18 as the backbone architecture in all experiments.

In our proposed approach, $\tau$ is set to 0.1, while $D_F$ and $D_P$ are equal to 512 and 128, respectively. The generator design varies by dataset: for CIFAR-10 and CIFAR-100, we utilize a ConvNet with three convolutional layers and batch normalization, while for VOC2012 and PPMI, the generator comprises six convolutional layers with batch normalization. The input noise is a vector with a length 200. The data-free generator is trained for 5000 epochs using the Adam optimizer with a learning rate of 0.005. The $\alpha_{div}$, $\alpha_{BN}$, and $\alpha_{prior}$ are set to 1.0, 75.0, and 0.001 for all experiments. We tune $\alpha_{KD}$ and $\alpha_{con}$ using $\{5.0, 1.0, 0.5, 0.1\}$, and $\alpha_{FT}$ by $\{1.0, 0.5, 0.1\}$, reporting the best results for each configuration. Each experiment is conducted three times, and we report the mean and standard deviation among them.

**Metrics.** There are three commonly used metrics to evaluate approaches for the CIL problem:

1) Last task accuracy ($\mathcal{A}_T$): The global model's accuracy at the end of the final task $T$, evaluated over all classes to that point.

2) Average accuracy ($\bar{\mathcal{A}}$): The average of accuracy across all tasks is calculated by $\bar{\mathcal{A}} = \frac{1}{T}\sum_{i=1}^{T}\mathcal{A}_i$, where $\mathcal{A}_i$ represents the accuracy of the model on task $i$.

3) Average forgetting ($\bar{f}$): Forgetting for a given task $t$ is defined as the difference between the highest accuracy achieved on task $t$ and its final accuracy after completing all tasks. Therefore, the average forgetting among all tasks is defined as

$$\bar{f} = \frac{1}{T-1}\sum_{i=1}^{T-1}\max_{j\in\{i,\dots,T-1\}}\mathcal{A}_{j,i} - \mathcal{A}_{T,i} \qquad (14)$$

where $\mathcal{A}_{i,j}$ is the accuracy on test samples from task $i$ after training of $j$-th task.

## 4.2. Performance Comparison

Three evaluation metrics for all datasets are summarized in Table 1. The FCIL-LD framework outperforms all FL baselines across all evaluation metrics. The Oracle model serves as an upper bound for performance. Among the baseline methods, FedAvg [21] and FedProx [15] exhibit significantly lower performance, with up to a 38% decrease in average accuracy and up to a 60% increase in forgetting compared to FCIL-LD. This highlights the need for frameworks specifically designed to address the CIL problem in the federated setting. FedLwF-2T [29] achieves better performance and lower forgetting than FedAvg and FedProx by employing two teachers to mitigate catstrophic forgetting. However, it still underperforms FCIL-LD due to its lack of access to previous task data.

FedCIL [23], a generative-based FCIL baseline, demonstrates even lower performance than FedAvg. Since FedCIL trains generators on local clients, it requires extensive local training epochs to generate high-quality synthetic samples. Despite increasing the number of local epochs by a factor of four in FedCIL compared to other baselines, its performance remains limited due to the inherent difficulty in fine-tuning GANs. Additionally, in FedCIL, the global model is updated using samples generated by client generators, making its performance highly dependent on the quality of local generators. As a result, poor-quality generators negatively impact the global model, leading to overall performance degradation. Furthermore, FedCIL struggles with datasets that have a large number of classes, such as CIFAR-100,

Table 2. Performance comparison on CIFAR-100 with 10 and 20 tasks.

| | $N = 10$ | | | $N = 20$ | | |
|---|---|---|---|---|---|---|
| | $\mathcal{A}_{10}$ | $\bar{\mathcal{A}}$ | $\bar{f}$ | $\mathcal{A}_{20}$ | $\bar{\mathcal{A}}$ | $\bar{f}$ |
| Oracle | 61.26 ±0.40 | 65.96 ±0.34 | – – | 60.54 ±0.17 | 65.93 ±0.12 | – – |
| FedAvg [21] | 8.79 ±0.13 | 23.22 ±0.18 | 80.44 ±0.49 | 4.59 ±0.09 | 14.87 ±0.14 | 85.32 ±0.63 |
| FedProx [15] | 8.81 ±0.04 | 23.07 ±0.17 | 80.33 ±0.54 | 4.43 ±0.23 | 14.80 ±0.11 | 84.79 ±0.41 |
| FedLwF-2T [29] | 11.71 ±0.44 | 30.24 ±0.33 | 67.58 ±2.09 | 5.21 ±0.44 | 18.21 ±0.59 | 79.52 ±0.62 |
| FedCIL [23] | 1.74 ±0.08 | 8.25 ±0.21 | 10.71 ±0.43 | 1.08 ±0.03 | 7.23 ±0.13 | 9.83 ±0.29 |
| MFCL [2] | 29.69 ±1.80 | 45.83 ±0.71 | 37.73 ±2.01 | 17.14 ±0.17 | 33.71 ±0.38 | 53.41 ±1.10 |
| FCIL-LD | **35.63** ±0.79 | **52.86** ±0.92 | **33.80** ±0.65 | **20.35** ±0.82 | **37.52** ±0.88 | **50.61** ±1.22 |

Table 3. Performance impact of removing each loss component and the low-dimensional projection layer on CIFAR-10.

| | $\mathcal{A}_5$ | $\bar{\mathcal{A}}$ | $\bar{f}$ |
|---|---|---|---|
| Ours - w/o $\mathcal{L}_{SupCon}$ (Eq 8) | 52.26 ±0.77 | 69.18 ±2.28 | 32.35 ±1.07 |
| Ours - w/o $\mathcal{L}_{KD}$ (Eq 13) | 46.53 ±2.57 | 63.89 ±2.08 | 41.71 ±6.62 |
| Ours - w/o $\mathcal{L}_{con}$ (Eq 13) | 55.17 ±1.34 | 71.02 ±0.28 | 36.46 ±1.56 |
| Ours - w/o $\mathcal{L}_{FT}$ (Eq 13) | 51.96 ±2.21 | 66.47 ±1.5 | 20.04 ±4.76 |
| Ours - w/o low-dimensional projection layer | 45.62 ±1.87 | 63.81 ±1.14 | 47.57 ±3.07 |
| FCIL-LD (Ours) | 56.64 ±3.27 | 72.55 ±0.48 | 30.79 ±3.37 |

as well as those with high-resolution images, such as PPMI and VOC2012. Although FedCIL utilizes generative replay, it fails to achieve competitive performance due to difficulty of generator optimization in GANs. It is worth to note that the lower forgetting in FedCIL is a consequence of its poor overall performance not due to its effectiveness. MFCL [2] is the most similar baseline to our proposed approach, as it also employs a data-free generator on the server. However, our proposed method outperforms MFCL by up to 6.9% in last task accuracy and 7.3% in average accuracy. Additionally, in terms of average forgetting, our approach reduces forgetting by up to 10.8% compared to MFCL. This demonstrates the critical role of designing the low-dimensional projection layer and novel loss components in enhancing the generalization capability of the feature extractor while effectively balancing between stability and plasticity.

### 4.3. Scalability

To demonstrate the scalability of the FCIL-LD framework in the large number of tasks, we evaluate CIFAR-100 dataset with 10 and 20 tasks, as presented in Table 2. It is worth noting that CIFAR-100 is selected for this scenario due to the limited number of classes in other datasets. As expected, the performance of Oracle remains relatively fixed as the number of tasks increases while all FL baselines exhibit performance degredation and increased average forgetting. As shown in Table 2, the FCIL-LD consistently outperforms baselines, even in scenarios with a large number of tasks. The higher accuracy and lower average forgetting of the FCIL-LD demonstrates its ability to effectively handel the large number of tasks in the FCIL problem.

### 4.4. Ablation Studies

To analyze the contribution of each loss component in the FCIL-LD, we conduct an ablation study by removing each component and evaluating its impact. Table 3 presents the results of removing individual loss components, as well as the low-dimensional projection layer. The effect of the low-dimensional projection layer is examined in the fifth row of Table 3, where the projection layer is removed, and SupCon and contrastive learning loss are applied directly to the output of the feature extractor. Removing the low-dimensional projection layer results in a decline in accuracy and an increase in forgetting, highlighting the crucial role of the low-dimensional projection layer in our proposed framework. Additionally, removing the fine-tuning loss leads to reduced forgetting but at the cost of decreased accuracy. Therefore, careful selection of the coefficient $\alpha_{FT}$ is essential to achieve an optimal balance between stability and plasticity.

## 5. Conclusion

In this work, we propose FCIL-LD, a novel contrastive-based FCIL framework with a low-dimensional projection layer to effectively balance learning new classes while preserving past knowledge. The FCIL-LD enhances the generalization capability of the feature extractor through supervised contrastive learning in the initial task and trains a data-free generator on the server to replay synthetic samples without data sharing, addressing privacy constraints. To further balance the stability-plasticity trade-off, we design a novel loss function for client training in subsequent tasks. Extensive experiments on four datasets confirm the superiority of FCIL-LD, with up to 48% improvement in last task accuracy, 55% higher average accuracy, and 11% reduction in forgetting compared to state-of-the-art FCIL baselines.

## References


[1] Tahir Abbas, Areej Fatima, Tariq Shahzad, Meshal Alharbi, Muhammad Adnan Khan, and Arfan Ahmed. Multidisciplinary cancer disease classification using adaptive fl in healthcare industry 5.0. *Scientific Reports*, 14(1):18643, 2024. 1

[2] Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 6, 7, 8

[3] Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics*, 2023. 1

[4] 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. 2, 3

[5] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102. Springer, 2020. 2

[6] Alireza Esmaeilzehi, Ensieh Khazaei, Kai Wang, Navjot Kaur Kalsi, Pai Chet Ng, Huan Liu, Yuanhao Yu, Dimitrios Hatzinakos, and Konstantinos Plataniotis. Harwe: A multi-modal large-scale dataset for context-aware human activity recognition in smart working environments. *Pattern Recognition Letters*, 2024. 1

[7] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[8] Parisa Hamedi, Roozbeh Razavi-Far, and Ehsan Hallaji. Federated continual learning: Concepts, challenges, and solutions. *arXiv preprint arXiv:2502.07059*, 2025. 2

[9] Sean M Hendryx, Dharma Raj KC, Bradley Walls, and Clayton T Morrison. Federated reconnaissance: Efficient, distributed, class-incremental learning. *arXiv preprint arXiv:2109.00150*, 2021. 6

[10] Ensieh Khazaei, Alireza Esmaeilzehi, Bilal Taha, and Dimitrios Hatzinakos. Cdfl: Efficient federated human activity recognition using contrastive learning and deep clustering. *IEEE Sensors Journal*, 2024. 1

[11] Ensieh Khazaei, Bilal Taha, Alireza Esmaeilzehi, and Dimitrios Hatzinakos. Osr: Toward developing efficient federated learning-based human activity recognition using optimal server representations. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 1

[12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 1, 3, 4

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

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

[15] 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. 6, 7, 8

[16] Xingyu Li, Bo Tang, and Haifeng Li. Adaer: An adaptive experience replay approach for continual lifelong learning. *Neurocomputing*, 572:127204, 2024. 2

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[18] Chunwe Ma, Zhanghexua Ji, Ziyu Huang, Yan Shen, Mingche Gao, and Jinhui Xu. Progressive voronoi diagram subdivision enables accurate data-free class-incremental learning. 2023. 4

[19] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2

[20] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022. 2

[21] 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. 6, 7, 8

[22] Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18820–18830, 2023. 4

[23] Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*, 2023. 1, 3, 6, 7, 8

[24] 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

[25] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020. 6

[26] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual

learning. *Advances in neural information processing systems*, 32, 2019. 2

[27] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 2

[28] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9374–9384, 2021. 3, 5, 6

[29] Anastasiia Usmanova, François Portet, Philippe Lalanda, and German Vega. A distillation-based approach integrating continual learning and federated learning for pervasive services. *arXiv preprint arXiv:2109.04197*, 2021. 1, 3, 6, 7, 8

[30] Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022. 1

[31] Bangpeng Yao and Li Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9–16. IEEE, 2010. 6

[32] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8715–8724, 2020. 3, 4

[33] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 2

[34] 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. 2

[35] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2