Diffusion-Driven Data Replay: A Novel Approach to Combat Forgetting in Federated Class Continual Learning

Jinglin Liang[1], Jin Zhong[1], Hanlin Gu[2], Zhongqi Lu[3], Xingxing Tang[2], Gang Dai[1], Shuangping Huang[1,5]⋆, Lixin Fan[4], and Qiang Yang[2,4]

[1]South China University of Technology,
[2]The Hong Kong University of Science and Technology,
[3]China University of Petroleum, [4]WeBank, [5]Pazhou Laboratory
eeljl@mail.scut.edu.cn, eehsp@scut.edu.cn

Abstract. Federated Class Continual Learning (FCCL) merges the challenges of distributed client learning with the need for seamless adaptation to new classes without forgetting old ones. The key challenge in FCCL is catastrophic forgetting, an issue that has been explored to some extent in Continual Learning (CL). However, due to privacy preservation requirements, some conventional methods, such as experience replay, are not directly applicable to FCCL. Existing FCCL methods mitigate forgetting by generating historical data through federated training of GANs or data-free knowledge distillation. However, these approaches often suffer from unstable training of generators or low-quality generated data, limiting their guidance for the model. To address this challenge, we propose a novel method of data replay based on diffusion models. Instead of training a diffusion model, we employ a pre-trained conditional diffusion model to reverse-engineer each class, searching the corresponding input conditions for each class within the model's input space, significantly reducing computational resources and time consumption while ensuring effective generation. Furthermore, we enhance the classifier's domain generalization ability on generated and real data through contrastive learning, indirectly improving the representational capability of generated data for real data. Comprehensive experiments demonstrate that our method significantly outperforms existing baselines. Code is available at https://github.com/jinglin-liang/DDDR.

Keywords: Federated Continual Learning · Diffusion Model

1 Introduction

Federated Learning (FL) [25, 33] is an emerging machine learning paradigm, offering a way to perform collective learning across decentralized devices while ensuring data privacy. Its ability to protect privacy makes it indispensable in domains such as healthcare [34] and finance [29], which are highly sensitive to data

⋆ Corresponding Author

security. However, real-world applications of FL also encounter challenges, such as the introduction of new data classes by clients over time and the variability of participants within the federation. This situation has led to the development of Federated Class Continual Learning (FCCL) [4, 8], a novel concept that necessitates models to incorporate new class information into the model during federated training without compromising the knowledge previously acquired.

The key challenge within FCCL is the issue of catastrophic forgetting [32], where the model loses previously acquired knowledge upon learning new tasks. While this problem has been explored to some extent in traditional Continual Learning (CL) [24], privacy preservation in FCCL introduces unique constraints that may prevent the direct application of existing strategies. For instance, experience replay [28, 42], a leading approach in CL for mitigating forgetting by retaining and rehearsing data from previous tasks, faces significant hurdles in a federated context. Specifically, in privacy-sensitive environments such as healthcare, the prolonged storage of historical data by users might not be permissible [49]. Moreover, the departure of federated participants can result in the loss of their stored data, further complicating data management and continuity.

To circumvent these limitations, the forefront works [4, 40, 60] in FCCL explore training generators to reproduce data from previous tasks. Specifically, FedCIL [40] employs a federated training of an improved version of ACGAN [35] to regenerate historical data. However, the training of GANs is known to be relatively unstable, a problem that becomes even more pronounced in federated settings [41]. Alternative methods [4, 60] utilize data-free knowledge distillation techniques [7] to train generators. Nevertheless, such methods tend to generate adversarial samples [12] for classifiers. These adversarial samples, due to their significant divergence from the distribution of real data, offer limited guidance capability for the model. To encapsulate, while current mainstream FCCL methods focus on mitigating model forgetting through generated replay, they encounter issues such as training instability and the inferior quality of generated data.

Inspired by the training stability and high quality of generated data characteristic of diffusion models in image generation [11, 38, 43], we propose the Diffusion-Driven Data Replay (DDDR), an innovative FCCL framework utilizing diffusion models for data replay. During the learning of each new task, DDDR employs our proposed Federated Class Inversion technique to extract Class embeddings for each new class. Specifically, we leverage a pre-trained conditional diffusion model for reverse engineering each class, which entails searching within the model's input space for a conditional embedding capable of guiding the model to generate images of the corresponding class. This Class embedding can be regarded as a condensed representation of the class, and by preserving this embedding, we can continuously generate data for the current task in subsequent tasks. The advantages of this approach are twofold. On one hand, leveraging powerful pre-trained diffusion models enables the generation of high-quality images. On the other hand, Federated Class Inversion requires optimization and communication of only the class embedding parameter, significantly reducing the computational, communication resources, and training time required compared to training the entire diffusion model. Subsequently, we employ the class embeddings derived from previous tasks to replay historical data for the classifier, the target model in continuous learning, to mitigate its forgetting. However, despite the high quality of this data, a certain distributional discrepancy from real data persists, potentially impacting model performance. To mitigate this, we introduce a contrastive learning constraint in the learning of new tasks, aiming to narrow the feature space gap between generated and real data within the same class. This enhances the classifier's generalization ability across both generated and real domains, indirectly boosting the representational capacity of the generated data for the real data.

Our contributions can be summarized as follows:

1) We propose DDDR, an innovative FCCL framework. This marks the first application of employing the diffusion model to reproduce data in FCCL, effectively mitigating catastrophic forgetting.
2) We propose Federated Class Inversion, achieving high-quality data generation in federated settings without consuming excessive additional resources.
3) By incorporating contrastive learning, we enhance the generalization ability of classifiers across generated and real domains, further strengthening the representational capacity of generated data towards real data.
4) Comprehensive experiments across various datasets demonstrate that our approach significantly outperforms existing methods, establishing a new state-of-the-art (SOTA) benchmark for FCCL.

2 Related Work

2.1 Continual Learning

Continual Learning (CL) aims to develop machine learning models that can learn from a stream of data over time without forgetting previously acquired knowledge. This field has seen the development of various strategies to mitigate catastrophic forgetting [32], broadly classified into four categories: regularization techniques, experience replay, dynamic architectural methods, and generative replay. Regularization techniques [2,21,24,59] are designed to prevent the model from significantly altering the weights important for previous tasks while learning new ones. Experience replay [3,28,39,42] involves storing a subset of previously encountered data and periodically retraining the model on this data alongside new information. Dynamic architectural methods [31,36,44,58] involve modifying the network architecture to accommodate new tasks, thereby preserving previous knowledge while expanding the model's capacity. Generative replay [27,48,50,53] leverages generative models to synthesize data for past tasks, which is then used to retrain the model alongside new data. However, given that the above methods were primarily designed for scenarios involving centralized training, they may not be well-suited for contexts where stringent privacy protection is paramount.

2.2 Federated Continual Learning

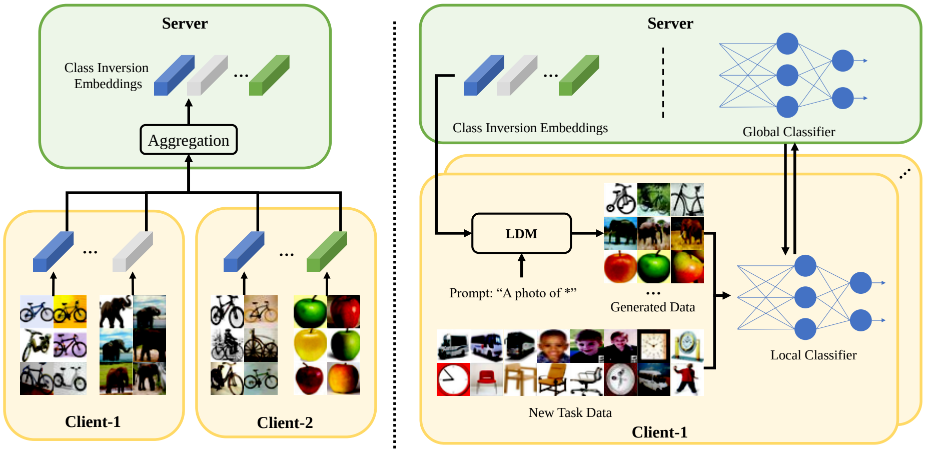
Federated Continual Learning (FCL) represents a fusion of Federated Learning (FL) and Continual Learning (CL), aimed at addressing the dual challenges of learning continuously from data streams across distributed devices while simultaneously safeguarding privacy and data locality [56]. Early FCL research [57] selectively activates model parameters associated with the current task through the input of a task ID, which necessitates the explicit notification of the model about the current task's ID, introducing additional complexity in task identification and parameter management. GLFC [10] demands clients store historical data, raising storage concerns. Park et al. [37] focuses on Federated Incremental Domain Learning scenarios, where the number of classes remains constant while domains incrementally evolve. Federated reconnaissance [14] emphasizes Few-Shot Learning settings within FCL. Some works [5, 17] focus on applying Federated Continual Learning to tasks beyond image classification. Ma et al. [30] distills the knowledge from old to new models using surrogate datasets. The effectiveness of this approach heavily relies on the similarity between the surrogate dataset and clients' local datasets. SOTA works [4, 40, 60] in FCCL train generators to replicate historical data. Specifically, FedCIL [40] introduces an enhanced version of ACGAN [35] for federated training, while others [4, 60] apply Data-Free Knowledge Distillation [7] in a federated context.

Building upon the concept of generative replay, our approach uniquely employs diffusion models for generation, achieving unprecedented training stability and superior quality of generated data.

2.3 Diffusion Models in Federated Learning

Despite the powerful generative capabilities of diffusion models being applied across various domains, their exploration in FL remains limited. Initial efforts [18, 46] have trained diffusion models within the FL framework. These methods, similar to those for training Generative Adversarial Networks (GANs) in a federated setting [26, 41, 52], adhere to the principle that privacy is preserved as long as the training process prevents unauthorized access to or inference of clients' local data. They suggest that generating data that is similar but not identical to a client's local data does not constitute a privacy breach, as long as the generated data does not replicate any specific local data. Conversely, other studies [54, 55, 61] have not directly pursued federated training of diffusion models. Instead, they exploit the pre-trained diffusion models to enable one-shot federated learning (OSFL), where the entire training process necessitates only a single round of communication between clients and the server. Clients guide the server's diffusion model by transmitting non-sensitive information such as image features [54], image descriptions [61], or classifiers [55], which is then used to generate client-specific data on the server. This generated data is subsequently utilized to train classification models server-side.

Diverging from the above methods, we introduce pre-trained diffusion models into the FCCL framework and propose Federated Class Inversion, thereby achieving high-quality generation with reduced resource consumption.



(a) Federated Class Inversion Phase

(b) Replay-Augmented Training Phase

Fig. 1: Overview of the DDDR Framework. (a) Federated Class Inversion Phase, in which a pre-trained diffusion model is utilized to reverse-engineer an embedding for each class. This embedding serves as a condensed representation of all images within the class, efficiently encapsulating the essence of the class in a compact vector. (b) Replay-Augmented Training Phase, in which clients employ the diffusion model along with previously obtained embeddings to regenerate data. Subsequently, clients train classifiers using the generated data and real data from new tasks.

3 Problem Formulation

In this section, we specifically formulate the task setting within the FCCL. FCCL employs a cooperative learning scheme involving a server and k clients collaboratively engaging in a sequence of n tasks, symbolized as $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}$. Each task $\mathcal{T}_t$ corresponds to a distinct dataset $\mathcal{B}_t$, which is distributed among the k clients. This dataset comprises image-label pairs for image classification and is formatted as $\mathcal{B}_t = \{(x_i, y_i)|i = 1, 2, ..., m\}$. Here, x_i represents the images, while y_i denotes their associated class labels, all of which are part of the label set $\mathcal{Y}_t$ for the task $\mathcal{T}_t$. A fundamental feature of this setup is the mutual exclusivity of label sets across tasks, guaranteed by $\mathcal{Y}_{t1} \cap \mathcal{Y}_{t2} = \emptyset$ for any $t1 \neq t2$. Additionally, during the learning of the current task, access to data from previous tasks and the Task ID is restricted. The objective is to equip the model after learning all tasks with the capability to accurately classify images across the cumulative label set $\bigcup_{t=1}^{n} \mathcal{Y}_t$.

4 Diffusion-Driven Data Replay

As illustrated in Figure 1, the DDDR framework encompasses two phases: the Federated Class Inversion Phase and the Replay-Augmented Training Phase. During the first phase, we extract and save an embedding for each class. In the second phase, we utilize the embeddings obtained from the first phase to replay data, and these generated data, along with real data, are used to train the classifier. Detailed descriptions of these phases are provided subsequently.

4.1 Federated Class Inversion Phase

Replaying historical data is an effective strategy to counter catastrophic forgetting [3,42]. To achieve this in contexts where data retention is not permissible, an intuitive solution is to train a generator capable of reproducing data from previous tasks. The advanced generative capabilities of diffusion models inspire their application in FCCL to generate historical data. However, training a diffusion model for each task is impractical due to the significant time and computational resources required, and the quality of generation may be compromised with limited client data.

To address this challenge, inspired by works in personalized generative models [11] and image editing [19], we propose a novel approach known as Federated Class Inversion. This method utilizes a frozen, pre-trained diffusion model to conduct reverse engineering on images from various classes, searching for conditional embeddings that can guide the diffusion model to generate images of the corresponding classes. This strategy negates the need for training a diffusion model for each task, substantially reducing the computational burden.

Subsequently, we will introduce Federated Class Inversion from three aspects: the pre-trained Latent Diffusion Model we employ, the local training for Class Inversion, and the global aggregation of Class Embeddings.

Latent Diffusion Model. Theoretically, any pre-trained conditional diffusion model can be used for Federated Class Inversion. In this work, we choose the Latent Diffusion Model (LDM) due to its fast inference speed and widely available pre-trained weights.

The Latent Diffusion Model comprises two primary components: an autoencoder and a diffusion model. The autoencoder [1,47] consists of an encoder, $\mathcal{E}$, and a decoder, $\mathcal{D}$. The encoder $\mathcal{E}$ maps the input image x to a low-dimensional latent code $z = \mathcal{E}(x)$ to reduce the computational load of subsequent denoising, while the decoder $\mathcal{D}$ is trained to perform the inverse mapping of the encoder.

The diffusion model, the second component, is tasked with denoising the latent codes. It is a conditional U-net [16], where the condition can be derived from various sources such as text, segmentation maps, etc. In our context, we focus solely on textual inputs. Its training objective is to predict the noise added to a latent code based on the input condition and a noise-corrupted version of the code. The mathematical formulation of this objective is given by:

$$\mathcal{L}_{LDM} = \mathbb{E}_{z\sim\mathcal{E}(x),p,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}z + \sqrt{1-\alpha_t}\epsilon, t, c_\theta(p))\|_2^2\right], \tag{1}$$

where z is the latent code of the input image x generated by the encoder $\mathcal{E}$, ϵ refers to noise sampled from the standard normal distribution $\mathcal{N}(0,1)$, t denotes the timestep in the diffusion process, and α_t is a hyperparameter related to t,

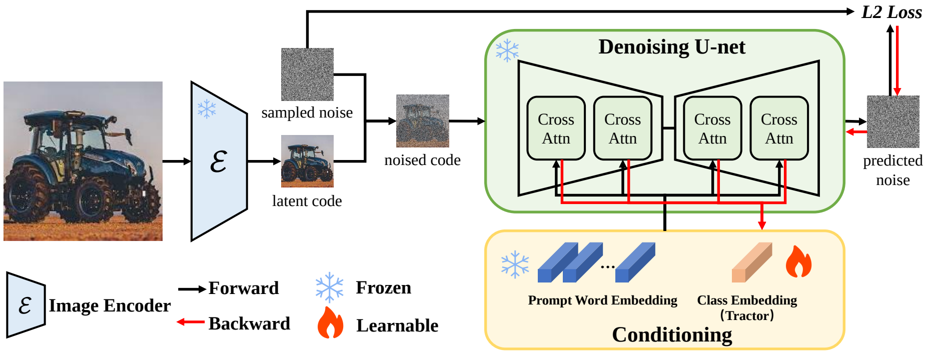


Fig. 2: Demonstrating the Local Class Inversion using the tractor class as an example. Initially, a tractor image is sampled from the client's local dataset and fed into the encoder $\mathcal{E}$, yielding the latent code. Concurrently, a frozen prompt's word embedding is concatenated with a learnable class embedding to form a guiding condition. This condition, along with the noise-added latent code, is inputted into the diffusion model to calculate the loss. The class embedding is then optimized using this loss.

p represents the text condition, $c_\theta(p)$ denotes the word embedding obtained by encoding p using the text encoder c_θ, and ϵ_θ is the model that predicts the noise.

Local Class Inversion. To capture class embeddings for further replay, we utilize a pre-trained text-to-image LDM for reverse engineering on images from every class, a process termed Class Inversion. Specifically, our objective is to search within the text embedding space, which serves as the input space for the LDM, for an embedding that can instruct the diffusion model to recreate images of a given class. To accomplish this, as illustrated in Figure 2, a prompt p, such as "a photo of", is encoded using a frozen text encoder c_θ into a word embedding $c_\theta(p)$. A learnable embedding v is then randomly initialized and concatenated with $c_\theta(p)$ to form the guiding condition $[c_\theta(p); v]$. This combined condition is employed to compute the LDM's loss function. Our optimization goal is formulated as:

$$v_i^* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(X_i),p,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}z + \sqrt{1-\alpha_t}\epsilon, t, [c_\theta(p); v])\|_2^2\right], \tag{2}$$

where X_i denotes the set of images belonging to the i^{th} class, and v_i represents the class embedding for class i.

During the Federated Class Inversion Phase, each client locally optimizes these class embeddings v_i utilizing Equation 2. This method strategically focuses on optimizing and communicating only the class embeddings with the server, significantly reducing the computational and communication resources required compared to training the entire generative model. We provide a detailed analysis of time and transmission efficiency in the supplementary materials.

Global Class Embedding Aggregation. In a federated setting, images of each class are distributed across various clients. To train a global class embedding for each class without transmitting data, we employ the FedAvg [33] algorithm to aggregate the class embeddings. The aggregation is represented as:

$$v_i = \frac{1}{k} \sum_{j=1}^{k} v_i^{(j)}, \tag{3}$$

where $v_i^{(j)}$ denotes the class embedding for class i uploaded by the j client, while v_i represents the global class embedding for class i.

By iteratively performing local training and global aggregation until the class embeddings converge, we obtain the class embeddings for each class. These embeddings are preserved for subsequent data replay.

Privacy. During the Federated Class Inversion Phase, akin to other distributed training approaches for generative models [26,46], our method trains by transmitting learnable parameters rather than disseminating data, inherently providing a layer of privacy protection. Moreover, we confine our optimization to the input space of the LDM without modifying the parameters of the pre-trained LDM, thereby not altering its output space. Consequently, the probability of generating images that are identical to the original data with our method is lower compared to those approaches that adjust the model's parameters. Should a higher level of privacy protection be required, methodologies such as differential privacy [51] can be seamlessly incorporated into our framework. We provide further analysis of privacy issues in the supplementary materials.

4.2 Replay-Augmented Training Phase

In this phase, we first utilize the class embeddings obtained during the Federated Class Inversion Phase to guide the diffusion model in generating data. Subsequently, both the generated and real data are employed together to train the classifier.

Data Generation. To mitigate the catastrophic forgetting of knowledge from previous tasks, we employ the class embeddings learned from previous tasks to generate historical classes images $\hat{\mathcal{X}}_p$ and their corresponding labels $\hat{\mathcal{Y}}_p$. Additionally, considering the non-IID challenge that can affect model training stability and performance, we also generate current task images $\hat{\mathcal{X}}_c$ and their labels $\hat{\mathcal{Y}}_c$ using the class embeddings of current task classes. By ensuring all clients share a similar distribution of generated data, this approach mitigates the extent of non-IID challenges.

Training Classifier. The training process for the classifier aims to fulfill two objectives: learning from the current task and revisiting previous tasks.

For learning the new task, we calculate the cross-entropy loss for both the real and generated data of the current task, formulated as:

$$\mathcal{L}_{CE} = \mathbb{E}_{x\sim\mathcal{X}_c\cup\hat{\mathcal{X}}_c, y\sim\mathcal{Y}_c\cup\hat{\mathcal{Y}}_c}[CE(\mathcal{F}(x), y)], \tag{4}$$

where $\mathcal{F}$ refers to the classifier, $\mathcal{X}_c$ denotes the real image set of the current task, and $\mathcal{Y}_c$ represents their corresponding labels.

Moreover, since the LDM was not tuned on the clients' data, the generated data may exhibit domain discrepancies from the real data. To address this challenge, we employ a supervised contrastive learning loss [9, 20] to constrain the classifier's feature space. By narrowing the feature representation gap between generated and real data within the same class, we enhance the model's generalization ability across both the generated data domain and real data. This approach indirectly strengthens the representational capability of the generated data towards real data. Specifically, we extract features before the final fully connected layer of the classifier, denoted as $e = \mathcal{F}_e(x)$, where $\mathcal{F}_e$ represents the feature extraction part of the classifier $\mathcal{F}$, and x signifies an individual image. These extracted features are then employed to compute the loss through the following formulation:

$$\mathcal{L}_{SCL} = \mathbb{E}_{e_i\sim\mathcal{F}_e(\mathcal{X}_c\cup\hat{\mathcal{X}}_c), e_p\sim P(e_i)}[\log \frac{\exp\left(sim\left(e_i, e_p\right)/\tau\right)}{\sum_{i\neq j}\exp\left(sim\left(e_i, e_j\right)/\tau\right)}], \tag{5}$$

where $P(e_i)$ denotes the set of positive samples that belong to the same class as e_i, $sim(e_i, e_j) = f_1(e_i)^T f_1(e_j)$ is the similarity function, f_1 represents a multi-layer perceptron (MLP) that maps features into an l_2-normalized feature space, and τ is a temperature coefficient.

For revisiting the old task, we employ two loss functions. Initially, we compute the cross-entropy loss directly using generated data from old tasks, expressed as:

$$\mathcal{L}_{PCE} = \mathbb{E}_{x\sim\hat{\mathcal{X}}_p, y\sim\hat{\mathcal{Y}}_p}[CE(\mathcal{F}(x), y)]. \tag{6}$$

Subsequently, to transfer the knowledge from historical tasks to the new model, we employ a knowledge distillation approach [15] on the generated dataset to migrate the knowledge of the model trained on the previous task to the current model. The knowledge distillation loss function can be represented as:

$$\mathcal{L}_{KD} = \mathbb{E}_{x\sim\hat{\mathcal{X}}_p}[KL(\mathcal{F}(x), \mathcal{F}'(x))], \tag{7}$$

where KL denotes the calculation of the Kullback-Leibler divergence, and $\mathcal{F}'$ represents the model preserved after training on the previous task.

In summary, the final objective function for training the classifier on the client side is formulated as:

$$\mathcal{F}^* = \arg\min_{\mathcal{F}} \mathcal{L}_{CE} + w_1\mathcal{L}_{SCL} + w_2\mathcal{L}_{PCE} + w_3\mathcal{L}_{KD}, \tag{8}$$

where w_1, w_2, and w_3 are hyperparameters used to balance the contributions of each term.

On the server side, we employ the FedAvg [33] algorithm to aggregate the parameters of the classifiers.

5 Experiments

5.1 Experimental Settings

Dataset. We conduct experiments on two datasets: Cifar-100 [22] and Tiny-ImageNet [23]. To simulate a class-incremental learning scenario, we employ a widely utilized data partitioning paradigm [4, 59]. Specifically, we divide the dataset classes evenly according to the predetermined task number, with each subset corresponding to the dataset for a particular task. In this work, we set the number of tasks to either 5 or 10. Additionally, we consider both IID and non-IID scenarios. For IID, we evenly distribute the data of each class among all clients. For non-IID, we adopt the widely accepted practice [25, 60] in FL of using the Dirichlet distribution to simulate an imbalanced label distribution across various clients. In our experiments, the Dirichlet parameter is set to 0.5.

Evaluation metric. We follow the principal works in FCCL [4, 60], employing two evaluation metrics: average accuracy and forgetting measure [6]. Average accuracy is the mean accuracy achieved by the model on all classes after the completion of training across all tasks. The forgetting measure is the mean difference between the peak accuracy and the final accuracy for each class, reflecting the extent to which the model forgets previously learned tasks.

Compared methods. We conducted comparisons with four methods: 1) **Finetune**, involving directly fine-tuning on subsequent tasks. 2) **FedEWC**, which implements the classical regularization-based continual learning strategy EWC [24] within the Fedavg [33] framework. 3) **Target** [60], an FCCL work that leverages data-free knowledge distillation for generative replay. 4) **MFCL** [4], a contemporaneous work with Target, follows a similar generative replay strategy as Target. However, it distinguishes itself by designing a more sophisticated loss function for the generator, achieving superior performance. It is the current SOTA in the field of FCCL.

Implementation detail. To ensure a fair comparison, all comparative methods and our approach utilize ResNet-18 [13] as the classifier, with all experiments conducted using 5 clients. During the Federated Class Inversion Phase, we employ the LDM pre-trained on the LAION-400M dataset [45], as proposed by Rombach et al. [43]. The training process consists of 10 communication rounds, with each client performing 50 local training steps per round. In the Replay-Augmented Training Phase, the number of communication rounds is increased to 100, with each client conducting training for 5 epochs locally per round. The coefficients for the losses, w_1, w_2, and w_3, are set to 1, 0.5, and 10, respectively.

5.2 Main Results

To validate the effectiveness of DDDR, we conducted comparative analyses between DDDR and four existing methods, with the results, averaged over three

Table 1: Results of the comparative experiments on the Cifar-100 dataset. 'T' indicates the task number. 'Acc' denotes average accuracy, with higher values indicating better performance, and 'FM' represents the forgetting measure, where lower values signify lesser forgetting of historical tasks. The best results are highlighted in bold.

Data partition	IID				non-IID			
Tasks	T=5		T=10		T=5		T=10	
Method	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)
Finetune	17.33	0.83	9.03	0.88	16.48	0.81	8.56	0.85
FedEWC	21.35	0.69	11.76	0.73	20.96	0.70	11.48	0.75
Target	34.40	0.48	22.95	0.49	34.35	0.48	21.71	0.51
MFCL	42.67	0.37	31.35	0.46	41.19	0.34	28.99	0.41
Ours	**51.04**	**0.29**	**43.45**	**0.32**	**48.45**	**0.26**	**41.27**	**0.26**

Table 2: Results of the comparative experiments on the Tiny-ImageNet dataset.

Data partition	IID				non-IID			
Tasks	T=5		T=10		T=5		T=10	
Method	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)
Finetune	12.29	0.60	6.80	0.67	11.68	0.57	6.58	0.64
FedEWC	13.27	0.49	8.22	0.56	12.55	0.47	7.66	0.52
Target	17.56	0.45	12.53	0.49	17.87	0.41	11.28	0.42
MFCL	15.11	0.52	10.13	0.54	13.35	0.48	8.54	0.51
Ours	**25.47**	**0.36**	**19.01**	**0.36**	**23.96**	**0.33**	**16.65**	**0.27**

experiments, presented in Tables 1 and 2, and Figure 3. From Tables 1 and 2, we can derive the following insights: 1) DDDR demonstrates improved performance over existing methods in all experimental settings across both datasets, in terms of average accuracy and forgetting measure. This suggests that our method effectively mitigates the issue of catastrophic forgetting in FCCL through high-quality data replay, thereby establishing a new SOTA for FCCL. 2) Finetune yields the poorest results, with its forgetting measure values indicating almost complete forgetting of historical tasks. 3) The regularization-based method FedEWC shows some improvement over Finetune but falls short when compared to generative replay-based methods like Target and MFCL. This suggests that while regularization-based methods can alleviate forgetting to some extent, their effectiveness is limited due to the lack of data-level guidance for the model. 4) The performance of the two generative replay-based methods, Target and MFCL, surpasses the other two baseline methods, yet there remains a discernible gap between them and our DDDR. This disparity is attributed to the lower quality of generated data, which constrains their performance.

Figure 3 depicts the changes in average accuracy as the model sequentially learns a series of tasks. We observe a monotonic decline in the average accu-

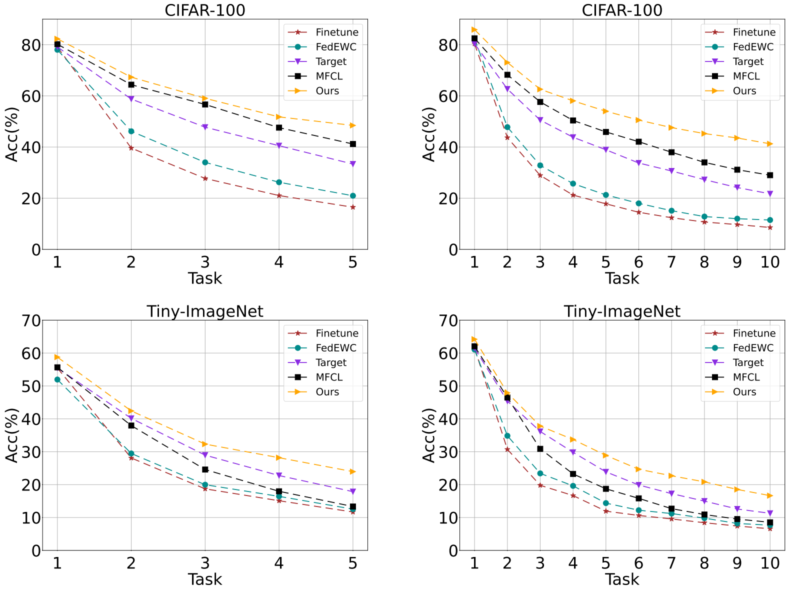


Fig. 3: Details of the variation in average accuracy as the learned task number increases across different methods on the CIFAR-100 and TinyImageNet datasets with a non-IID distribution of data.

racy of all methods as the model progressively learns more tasks, attributed to the forgetting of knowledge from old tasks. Notably, DDDR exhibits the least pronounced decrease in average accuracy across all settings, demonstrating its superior capability to mitigate the forgetting of historical tasks. Moreover, for the initial task, where no historical data exists, our method also achieves the highest accuracy. This indicates that DDDR effectively reduces the impact of non-IID distributions by generating data for the current task, thus improving performance.

5.3 Ablation Study

To evaluate the effectiveness of each component of DDDR, we performed ablation experiments, with the results detailed in Table 3. These experiments targeted three main components: generated data for historical tasks $(\hat{\mathcal{X}}_p, \hat{\mathcal{Y}}_p)$, generated data for the current task $(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$, and the contrastive learning loss $\mathcal{L}_{SCL}$. For the ablation of $(\hat{\mathcal{X}}_p, \hat{\mathcal{Y}}_p)$, we omitted the calculation of the historical task cross-entropy loss from Eq.(6) and the knowledge distillation loss from Eq.(7) during classifier training. In the case of ablation for $(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$, only real data from the current task were used to compute the cross-entropy loss from Eq.(4) and the contrastive learning loss from Eq.(5), excluding any generated data. Finally, for the ablation of the $\mathcal{L}_{SCL}$, we did not calculate the loss specified in Eq.(5). Subsequently, we elucidate the role of each component individually.

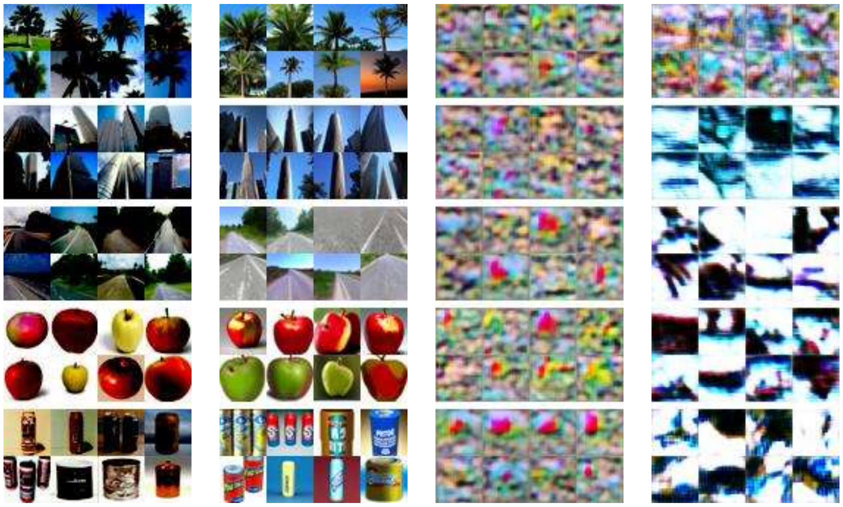
Table 3: Results of ablation experiments for DDDR on the Cifar-100 dataset. The number of tasks is set to 5, with a non-IID distribution of data. $(\hat{\mathcal{X}}_p, \hat{\mathcal{Y}}_p)$ and $(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$ respectively denotes the generated data for historical tasks and current task. $\mathcal{L}_{SCL}$ signifies the contrastive learning loss function. The symbols ✓ and ✗ respectively indicate the inclusion and ablation of the corresponding settings.

	$(\hat{\mathcal{X}}_p, \hat{\mathcal{Y}}_p)$	$(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$	$\mathcal{L}_{SCL}$	Acc(↑)	FM(↓)
1	✓	✓	✓	48.45	0.26
2	✗	✓	✓	17.63	0.84
3	✓	✗	✓	44.29	0.36
4	✓	✓	✗	45.34	0.28
5	✗	✗	✓	16.37	0.82
6	✓	✗	✗	45.13	0.29
7	✗	✓	✗	17.51	0.83
8	✗	✗	✗	16.48	0.81

The Impact of Generated Data for Historical Tasks. As illustrated in rows 2, 5, 7, and 8 of Table 3, the absence of generated data for historical tasks leads to a significant degradation in both average accuracy and the forgetting measure, with a particularly notable increase in the forgetting measure indicating that the model has almost completely forgotten the historical tasks. This highlights that DDDR's capability to alleviate catastrophic forgetting can be attributed to its ability to generate high-quality data for historical tasks.

The Impact of Generated Data for Current Task. The comparative analysis of rows 1 and 3, alongside rows 4 and 6 in Table 3, demonstrates that excluding generated data for the current task not only reduces the average accuracy but also increases the forgetting measure scores. This indicates the effectiveness of the strategy, which by ensuring all clients share a similarly distributed set of generated data, effectively mitigates the degree of non-IIDness in data distribution, thereby enhancing the model's performance.

The Impact of Contrastive Learning Loss. The observations from rows 1 and 4 in Table 3 demonstrate that the combined use of $\mathcal{L}_{SCL}$ and $(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$ enhances both the average accuracy and the forgetting measure. However, a contrasting review of rows 3, 6, and subsequently rows 5 and 8 reveals that applying $\mathcal{L}_{SCL}$ in the absence of $(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$ leads to a deterioration in both average accuracy and the forgetting metric. This indicates that the efficacy of $\mathcal{L}_{SCL}$ is not absolute. It yields significant performance gains when generated data for the current task is included in training, but surprisingly produces adverse effects when such data is omitted. This observation suggests that the performance improvements attributed to $\mathcal{L}_{SCL}$ stem from its ability to bridge the feature representation gap between generated and real data, thereby enhancing the classifier's generalization capability across the generated and real data domains.



(a) Real Data (b) DDDR (c) MFCL (d) Target

Fig. 4: Visualization of generated outcomes from three generative replay methods and the real data from the CIFAR-100 dataset.

5.4 Visualization of Generated Results

To analyze the generative capabilities of DDDR's Federated Class Inversion method, we conduct a visual analysis, with the results depicted in Figure 4 and supplementary material. It is evident that the generated images by DDDR closely approximate the data distribution of real images. This indicates that our Federated Class Inversion method can deliver excellent generative results while only requiring a reduced number of optimization parameters. In contrast, the other two methods based on data-free knowledge distillation, namely Target and MFCL, aim to generate images that mislead the classifier into categorizing them as a specific class. As a result, their generated samples often resemble adversarial examples [12] for the classifier, exhibiting a significant distribution gap from the real data. It can be inferred that the superior generation quality of DDDR is key to more effectively addressing catastrophic forgetting.

6 Conclusion

This work proposes DDDR, leveraging diffusion models to replay historical data and address the issue of catastrophic forgetting in FCCL. Compared to existing generative replay methods, our approach is capable of producing higher-quality data. Furthermore, we enhance the domain generalization ability of the classifier on both generated and real data, enabling more efficient utilization of generated data. Comprehensive experiments demonstrate that DDDR significantly outperforms existing approaches, establishing a new state-of-the-art in the FCCL domain.

Acknowledgement The research is partially supported by National Key Research and Development Program of China (2023YFC3502900), National Natural Science Foundation of China (No.62176093, 61673182), Key Realm Research and Development Program of Guangzhou (No.202206030001), Guangdong-Hong Kong-Macao Joint Innovation Project (No.2023A0505030016).

References

1. Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., Van Gool, L.: Soft-to-hard vector quantization for end-to-end learned compression of images and neural networks. arXiv preprint arXiv:1704.00648 **3** (2017)
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: ECCV. pp. 139–154 (2018)
3. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. In: NeurIPS. vol. 32 (2019)
4. Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M., Avestimehr, S.: A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. In: NeurIPS. vol. 36 (2023)
5. Chaudhary, Y., Rai, P., Schubert, M., Schütze, H., Gupta, P.: Federated continual learning for text classification via selective inter-client transfer. arXiv preprint arXiv:2210.06101 (2022)
6. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: ECCV. pp. 532–547 (2018)
7. Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., Tian, Q.: Data-free learning of student networks. In: ICCV. pp. 3514–3522 (2019)
8. Criado, M.F., Casado, F.E., Iglesias, R., Regueiro, C.V., Barro, S.: Non-iid data and continual learning processes in federated learning: A long road ahead. Information Fusion **88**, 263–280 (2022)
9. Dai, G., Zhang, Y., Wang, Q., Du, Q., Yu, Z., Liu, Z., Huang, S.: Disentangling writer and character styles for handwriting generation. In: CVPR. pp. 5977–5986 (2023)
10. Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., Zhu, Q.: Federated class-incremental learning. In: CVPR. pp. 10164–10173 (2022)
11. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: ICLR (2022)
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. Hendryx, S.M., KC, D.R., Walls, B., Morrison, C.T.: Federated reconnaissance: Efficient, distributed, class-incremental learning. arXiv preprint arXiv:2109.00150 (2021)
15. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
16. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)

17. Jiang, Z., Ren, Y., Lei, M., Zhao, Z.: Fedspeech: Federated text-to-speech with continual learning. arXiv preprint arXiv:2110.07216 (2021)
18. Jothiraj, F.V.S., Mashhadi, A.: Phoenix: A federated generative diffusion model. arXiv preprint arXiv:2306.04098 (2023)
19. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: CVPR. pp. 6007–6017 (2023)
20. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NeurIPS. vol. 33, pp. 18661–18673 (2020)
21. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
22. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
23. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
24. Lee, S.W., Kim, J.H., Jun, J., Ha, J.W., Zhang, B.T.: Overcoming catastrophic forgetting by incremental moment matching. In: NeurIPS. vol. 30 (2017)
25. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proceedings of Machine learning and systems. pp. 429–450 (2020)
26. Li, W., Chen, J., Wang, Z., Shen, Z., Ma, C., Cui, X.: Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation. IEEE Transactions on Neural Networks and Learning Systems (2022)
27. Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A.D., Jui, S., de Weijer, J.v.: Generative feature replay for class-incremental learning. In: CVPR. pp. 226–227 (2020)
28. Liu, Y., Schiele, B., Sun, Q.: Rmm: Reinforced memory management for class-incremental learning. In: NeurIPS. vol. 34, pp. 3478–3490 (2021)
29. Long, G., Tan, Y., Jiang, J., Zhang, C.: Federated learning for open banking. In: Federated Learning: Privacy and Incentive. pp. 240–254 (2020)
30. Ma, Y., Xie, Z., Wang, J., Chen, K., Shou, L.: Continual federated learning based on knowledge distillation. In: IJCAI. vol. 3 (2022)
31. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: CVPR. pp. 7765–7773 (2018)
32. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
33. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282 (2017)
34. Nguyen, D.C., Pham, Q.V., Pathirana, P.N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., Hwang, W.J.: Federated learning for smart healthcare: A survey. ACM Computing Surveys (CSUR) **55**(3), 1–37 (2022)
35. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: ICML. pp. 2642–2651 (2017)
36. von Oswald, J., Henning, C., Grewe, B.F., Sacramento, J.: Continual learning with hypernetworks. In: ICLR (2020)

37. Park, T.J., Kumatani, K., Dimitriadis, D.: Tackling dynamics in federated incremental learning with variational embedding rehearsal. arXiv preprint arXiv:2110.09695 (2021)
38. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. In: ICLR (2023)
39. Prabhu, A., Torr, P.H., Dokania, P.K.: Gdumb: A simple approach that questions our progress in continual learning. In: ECCV. pp. 524–540 (2020)
40. Qi, D., Zhao, H., Li, S.: Better generative replay for continual federated learning. In: ICLR (2023)
41. Rasouli, M., Sun, T., Rajagopal, R.: Fedgan: Federated generative adversarial networks for distributed data. arXiv preprint arXiv:2006.07228 (2020)
42. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. In: NeurIPS. vol. 32 (2019)
43. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
44. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
45. Schuhmann, C., Kaczmarczyk, R., Komatsuzaki, A., Katta, A., Vencu, R., Beaumont, R., Jitsev, J., Coombes, T., Mullis, C.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In: NeurIPS Workshop Datacentric AI. No. FZJ-2022-00923 (2021)
46. Tun, Y.L., Thwal, C.M., Yoon, J.S., Kang, S.M., Zhang, C., Hong, C.S.: Federated learning with diffusion models for privacy-sensitive vision tasks. In: 2023 International Conference on Advanced Technologies for Communications (ATC). pp. 305–310 (2023)
47. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: NeurIPS. vol. 30 (2017)
48. Van de Ven, G.M., Tolias, A.S.: Generative replay with feedback connections as a general strategy for continual learning. arXiv preprint arXiv:1809.10635 (2018)
49. Vizitiu, A., Niţă, C.I., Puiu, A., Suciu, C., Itu, L.M.: Towards privacy-preserving deep learning based medical imaging applications. In: 2019 IEEE international symposium on medical measurements and applications (MeMeA). pp. 1–6 (2019)
50. Wang, L., Yang, K., Li, C., Hong, L., Li, Z., Zhu, J.: Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In: CVPR. pp. 5383–5392 (2021)
51. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security **15**, 3454–3469 (2020)
52. Wijesinghe, A., Zhang, S., Ding, Z.: Pfl-gan: When client heterogeneity meets generative models in personalized federated learning. arXiv preprint arXiv:2308.12454 (2023)
53. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., Fu, Y.: Incremental classifier learning with generative adversarial networks. arXiv preprint arXiv:1802.00853 (2018)
54. Yang, M., Su, S., Li, B., Xue, X.: Exploring one-shot semi-supervised federated learning with a pre-trained diffusion model. arXiv preprint arXiv:2305.04063 (2023)
55. Yang, M., Su, S., Li, B., Xue, X.: One-shot federated learning with classifier-guided diffusion models. arXiv preprint arXiv:2311.08870 (2023)

56. Yang, X., Yu, H., Gao, X., Wang, H., Zhang, J., Li, T.: Federated continual learning via knowledge fusion: A survey. IEEE Transactions on Knowledge and Data Engineering (2024)
57. Yoon, J., Jeong, W., Lee, G., Yang, E., Hwang, S.J.: Federated continual learning with weighted inter-client transfer. In: ICML. pp. 12073–12086 (2021)
58. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: ICLR (2018)
59. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML. pp. 3987–3995 (2017)
60. Zhang, J., Chen, C., Zhuang, W., Lyu, L.: Target: Federated class-continual learning via exemplar-free distillation. In: ICCV. pp. 4782–4793 (2023)
61. Zhang, J., Qi, X., Zhao, B.: Federated generative learning with foundation models. arXiv preprint arXiv:2306.16064 (2023)