

# Task-Aware Prompt Gradient Projection for Parameter-Efficient Tuning Federated Class-Incremental Learning

Hualong Ke<sup>1</sup>, Jiangming Shi<sup>1,2</sup>, Yachao Zhang<sup>1\*</sup>, Fangyong Wang<sup>3</sup>, Yuan Xie<sup>2,4</sup>, Yanyun Qu<sup>1\*</sup>

<sup>1</sup> Key Laboratory of Multimedia Trusted Perception and Efficient Computing,  
Ministry of Education of China, School of Informatics, Xiamen University,

<sup>2</sup> Shanghai Innovation Institute, <sup>3</sup> Hanjiang National Laboratory, <sup>4</sup> East China Normal University

hualongke@stu.xmu.edu.cn, jiangming.shi@outlook.com, yachaozhang@xmu.edu.cn

hzwfy@163.com, yxie@cs.ecnu.edu.cn, yyqu@xmu.edu.cn

## Abstract

Federated Continual Learning (FCL) has recently garnered significant attention due to its ability to continuously learn new tasks while protecting user privacy. However, existing Data-Free Knowledge Transfer (DFKT) methods require training the entire model, leading to high training and communication costs, while prompt pool-based methods with access to other task-specific prompts in the pool may pose a privacy leakage risk. To address these challenges, we propose a novel method: Task-aware Prompt gradient Projection and Replay (TPPR), which leverages visual prompts to build a parameter-efficient tuning architecture. Specifically, we propose the Task-Aware Prompt Gradient Projection (TAPGP) mechanism to balance the learning of task-agnostic and task-specific knowledge. In practice, we make the gradient of the deep prompts orthogonal to the virtual data and prompts of preceding tasks, which prevents the erosion of old task knowledge while allowing the model to learn new information. Additionally, we introduce Dual-Level Prompt Replay (DLPR) based on the exponential moving average to facilitate knowledge review at both inter-task and intra-task levels, effectively inheriting learned knowledge. Extensive experimental results demonstrate that our method effectively reduces model communication overhead and alleviates forgetting while fully protecting privacy. With only 1% of the training parameters, we achieve more than 5% accuracy improvements in all settings than SOTA with the same backbone.

## 1. Introduction

Federated Learning (FL) is a decentralized, privacy-focused technology that enables multiple entities to collaborate

\*Corresponding author.

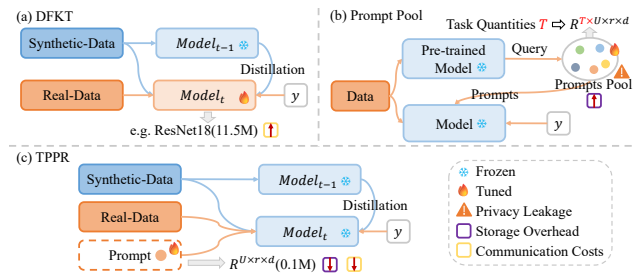


Figure 1. The brief overview of classic methods DFKT (a), prompt pool (b), and ours (c).

by model transmission instead of directly sharing data [1, 19, 21, 53]. In recent years, federated learning has garnered significant attention in fields such as healthcare [1, 50], IoT [22, 34] and autonomous driving [10, 11, 26], all of which prioritize data decentralization and information privacy. Traditional federated learning research often assumes that data categories and distributions are static, which means that the categories and domains of the training data remain unchanged throughout the entire process. In real-world applications, data categories and domains evolve over time [27, 36, 42, 58], while client devices have limited storage, hindering the handling of large datasets. These factors pose challenges for continual learning in the FL context.

Recent research has integrated continual learning into the federated learning framework to adapt to evolving data streams. A prominent setting is Federated Class-Incremental Learning (FCIL) [9, 14, 56], which aims to continuously learn new categories in the federated learning scenario. However, catastrophic forgetting remains a significant challenge for FCIL. The heterogeneity of data across different clients in federated learning complicates the extraction of robust features in a distributed environment, exacerbating knowledge forgetting when new tasks

are learned. Transfer learning [4, 5, 9, 18, 32, 40] has proven to be an effective method for mitigating knowledge forgetting. Data-Free Knowledge Transfer (DFKT) [31, 43, 46, 59, 62] has emerged as a key technique in FCIL because it allows knowledge transfer from the old task model to address the forgetting issue while avoiding direct access to raw training data, thus protecting privacy. Nevertheless, as shown in Figure 1 (a), these methods typically require training the entire model on the client-side and communicating with the server, leading to significant training overhead and transmission delays.

With the rise of foundational vision models, prompt-based learning has emerged as a parameter-efficient strategy for fine-tuning pre-trained models [23, 54, 60, 61]. This method requires training the prompt parameters to adapt the model to specific downstream tasks, making it highly suitable for FL scenarios with limited resources. Prompt-based approaches generally use a prompt pool to store task-specific prompts, aiming to balance the learning of task-agnostic and task-specific features [41, 47, 48]. Nevertheless, using a prompt pool introduces additional storage, communication overhead, and privacy concerns [35], depicted in Figure 1 (b). Specifically, in prompt learning, pre-trained models are frozen, and task-specific information is encoded in learnable prompts. Storing these learned prompts in the prompt pool retains all learned knowledge, which increases potential privacy leakage risks. Furthermore, from a conceptual standpoint, the essence of continual learning is to enable the learning of multiple tasks using the same set of learnable parameters. Essentially, storage-based methods such as the prompt pool store all knowledge, with task-specific prompts serving as identifiers, which doesn't solve the core problem of inheriting and transferring knowledge from previous tasks.

To address the aforementioned challenges, we propose a novel prompt-based method for FCIL called **Task-aware Prompt gradient Projection and Replay (TPPR)**. To begin with, we leverage visual prompts to reduce training and transmission costs in a parameter-efficient manner. Unlike the prompt pool approach, which assigns specific prompts to each task, we simultaneously learn both task-agnostic and task-specific knowledge in the same set of prompts, which poses a huge challenge to the model's ability to resist forgetting. Therefore, we address the issue of catastrophic forgetting from the following two perspectives: **(1) Protecting learned knowledge.** We introduce Task-Aware Prompt Gradient Projection (TAPGP), which ensures that the gradient of the deep prompts is orthogonal to the virtual data and prompts of previous tasks. This orthogonality helps alleviate knowledge interference between tasks and mitigates the damage to learned knowledge. **(2) Inheriting learned knowledge.** We propose Dual-Level Prompt Replay (DLPR), which carries the exponential moving aver-

age at both the inter-task and intra-task levels. Intra-task replay alleviates the model shift caused by data heterogeneity between clients, and inter-task replay efficiently integrates knowledge between new and old tasks, which improves stability and anti-forgetting ability. In summary, our contributions are as follows:

- We propose a novel prompt-based FCIL method that effectively addresses catastrophic forgetting in a parameter-efficient manner while fully protecting privacy.
- TAPGP effectively balances the learning of task-agnostic and task-specific knowledge, fully protecting previously learned knowledge when learning a new task. Besides, DLPR reviews knowledge at both the inter-task and intra-task levels, significantly promoting training stability and the heritage of learned knowledge.
- Extensive experiments on multiple datasets fully demonstrate the efficiency and effectiveness of our method.

## 2. Related Work

### 2.1. Federated Learning (FL)

Federated learning aims to collaboratively train a high-performance model while ensuring privacy protection. FedAvg [33] allows multiple clients to train models locally and send their updates to a central server, which aggregates them into a global model and returns it to the clients for further training. A key challenge in FL is data heterogeneity, where the data across clients is often non-independent and identically distributed (non-IID). This can manifest as label heterogeneity (imbalanced class distribution) [29, 49, 57] and feature heterogeneity (domain discrepancy) [6, 8, 20]. In federated learning, it's critical to consider the generalization ability of models across unseen domains and categories. Visual foundation models, with their robust generalization capacity, offer a potential solution to this problem. Approaches like PromptSRC [24], SGPT [8], and pFedPrompt [15] leverage prompts with pre-trained visual foundation models to enhance generalization. FedPGP [7], on the other hand, employs low-rank decomposition adaptation to mitigate the overfitting of pre-trained models to local datasets. While these methods have made strides in improving model generalization in federated learning, they still operate under the assumption that model data categories and distributions remain static, which fails to address the practical challenges posed by constantly evolving client data.

### 2.2. Continual Learning (CL)

Continual learning aims to enable the continuous acquisition of new tasks without forgetting previously learned knowledge. Leveraging the rich prior knowledge and robust generalization ability of visual foundation models, several prompt-based methods [13, 45, 48, 51, 52] have demonstrated strong performance in CL. L2P [48] employs a key-

query method to select the most suitable prompts for different tasks from a prompt pool. DualPrompt [47] divides the prompt pool into task-invariant and task-specific sections. CODAPrompt [41] introduces a novel attention-based, end-to-end key query mechanism. While these methods achieve impressive results, the use of the prompt pool may introduce privacy risks in federated learning settings. Additionally, GPM [39] and PGP [37] integrate gradient projection into CL, which ensures that gradients remain orthogonal to the data and features of preceding tasks. Nevertheless, due to privacy and security concerns, federated learning involves data being distributed across multiple clients, rendering these methods incompatible with such settings.

### 2.3. Federated Continual Learning (FCL)

Federated continual learning integrates federated learning and continual learning to address scenarios where distributed clients continuously learn from dynamically evolving data streams while ensuring privacy preservation. Many existing methods leverage DFKT techniques to mitigate catastrophic forgetting between old and new tasks. For instance, TARGET [58] pioneers used a generator to produce virtual data that simulates the global data distribution, effectively alleviating catastrophic forgetting in FCIL. Building on this, LANDER [42] introduces Label-Text Embedding (LTE) as an intermediary to align the features of real and virtual data around the LTE, significantly narrowing the distribution gap and enhancing the quality of generated virtual data. Further advancing the field, DDDR [30] proposes federated class inversion, which employs a diffusion model to reproduce data in FCIL. Additionally, several works explore prompt-based learning to address challenges in FCL. For example, Fed-CPrompt [3] introduces two key components: asynchronous prompt learning and contrastive continual loss, designed to handle asynchronous task arrivals and heterogeneous data distributions, respectively. Similarly, Powder [35] facilitates the transfer of knowledge encapsulated in prompts across sequentially learned tasks and clients. However, these prompt-based methods typically rely on a prompt pool, which introduces storage overhead and potential privacy leakage risks.

### 3. Problem Setting and Preliminary

Federated class-incremental learning seeks to address the challenge of incrementally learning new classes over time within the context of federated learning. The FCIL architecture comprises a central server, denoted as  $\mathcal{S}$ , and multiple distributed clients ( $\mathcal{C}_1, \dots, \mathcal{C}_m$ ). Throughout training, each client’s data remains local and is neither shared with the server nor other clients, thereby preserving data privacy. Each client is responsible for learning a sequence of  $T$  tasks, where the  $t$ -th task involves a set of non-overlapping classes. For each task  $t$ , the dataset of client  $k$  is repre-

sented as  $\mathcal{D}_k^t (1 \leq k \leq m)$ , comprising  $N_k^t$  pairs of samples and their corresponding labels  $\{(x_i, y_i)\}_{i=1}^{N_k^t}$ . The labels  $y_i$  belong to distinct subsets of classes  $\mathcal{Y}_k^t \subset \mathcal{Y}$ , where  $\mathcal{Y}$  denotes the set of all class labels. To distinguish their roles, the models for task  $t$  are denoted as  $S^t$  for the server and  $(\mathcal{C}_1^t, \dots, \mathcal{C}_m^t)$  for the clients, respectively.

Data-Free Knowledge Transfer (DFKT) is a critical strategy for mitigating catastrophic forgetting and addressing privacy concerns, making it a prominent approach in FCIL. In the context of data-free FCIL, we adopt the framework outlined in [2, 42, 58]. For task  $t$ , the process begins by performing DFKT on the server model up to task  $t - 1$  (i.e.,  $S^{t-1}$ ), which generates data  $\mathcal{M}^{t-1}$ . Subsequently, the server engages in communication with the clients for  $R$  rounds, where each round consists of two phases: client-side training and server-side aggregation.

In the client-side training phase, each client  $k$  trains its model  $\mathcal{C}_k^t$  using both the new task data  $\mathcal{D}_k^t$  and the synthetic data  $\mathcal{M}^{t-1}$  generated from previous tasks. In the server aggregation phase, the server aggregates the client models to update the global model in the FedAvg [33] manner:  $S^t = \frac{1}{m} \sum_{k=1}^m \mathcal{C}_k^t$ .

## 4. Methodology

### 4.1. Parameter-Efficient Tuning Architecture

In the context of federated learning, data is distributed across various clients for decentralized training. Recently, several methods [2, 42, 58] for FCIL require training the whole model, which requires a huge number of parameters for training and incurs high transmission overhead and time delay during model aggregation. To solve this problem, we adopt an efficient model fine-tuning method, prompt tuning, to reduce training and transmission costs. Taking inspiration from VPT [23], we utilize learnable visual prompts to adapt the pre-trained ViT model to our task, as shown in Figure 2. To achieve better fine-tuning effects, we adopt the VPT-Deep structure, as shown below:

$$[cls_u, e_{u,-}] = L_u([cls_{u-1}, e_{u-1}, p_{u-1}]) \quad u = 1, \dots, U.$$

$$\hat{y} = \text{Head}(cls_U).$$

where  $U$  is the number of layers;  $cls_u \in \mathbb{R}^d$  denotes the classification embeddings at  $L_u$ ’s output space; and  $e_{u-1}$  is a collection of image patch embeddings as the inputs to the  $u$ -th Transformer layer  $L_u$ .  $p_{u-1} \in \mathbb{R}^{r \times d}$  presents the learnable prompts as the inputs to the  $L_u$ .  $d$  is the embedding dimension, and  $r$  denotes the number of prompts.  $\text{Head}(\cdot)$  is the classifier. The colors  $\bullet$  and  $\bullet$  indicate learnable and frozen parameters, respectively.

Building on the VPT-Deep model structure, we follow the latest SOTA methods [2, 42, 58] for FCIL and incorporate DFKT to mitigate the problem of catastrophic forgetting. To ensure the quality of the generated data, we adopt

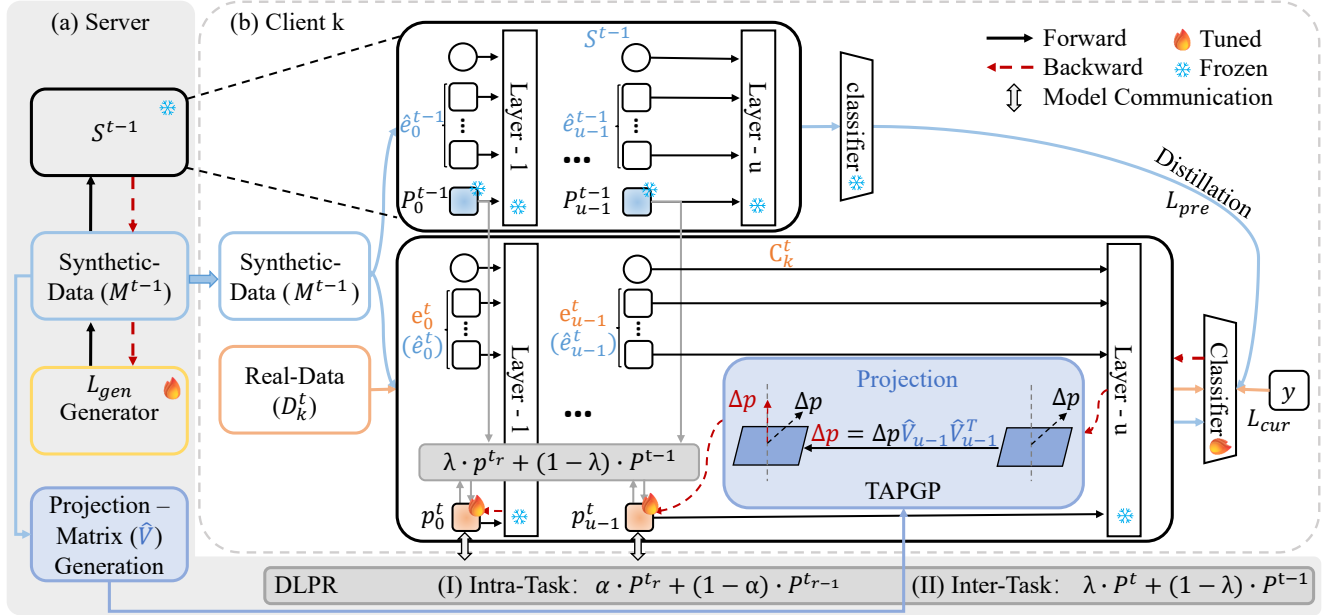


Figure 2. The framework of the proposed method. (a) Server Side. The server generates data  $\mathcal{M}^{t-1}$  using the global model  $\mathcal{S}^{t-1}$  and transfers it to the client of the subsequent task for knowledge distillation. Simultaneously, the generated data  $\mathcal{M}^{t-1}$  is employed to compute a projection matrix  $\hat{V}$  for prompt gradient projection. (b) Client Side. During forward propagation, distillation and dual-level prompt replay are utilized to inherit knowledge from previous tasks. During backpropagation, task-aware prompt gradient projection is applied to protect learned knowledge. (The symbol meanings are summarized in the supplementary for reference.)

LANDER [42] as the baseline and the training loss formulated as (detailed in the supplementary):

$$\mathcal{L}_{client} = \mathcal{L}_{cur} + \mathcal{L}_{pre}. \quad (1)$$

$$\mathcal{L}_{server} = \mathcal{L}_{gen}. \quad (2)$$

Clients focus on learning new data while minimizing forgetting.  $\mathcal{L}_{cur}$  denotes the training loss on the client  $k$ 's current data  $\mathcal{D}_k^t$ .  $\mathcal{L}_{pre}$  mitigates forgetting by distilling knowledge from the global model  $\mathcal{S}^{t-1}$  using synthetic data  $\mathcal{M}^{t-1}$ . The server primarily aggregates models and generates data for past tasks, with  $\mathcal{L}_{gen}$  representing the generator loss. Since we only train learnable prompts, the aggregation model essentially aggregates prompts (i.e., the server prompts  $P^t = \sum_{k=1}^m \frac{n_k}{n} p^{t^k}$  where  $p^{t^k}$  are prompts of  $k$ -th client in task  $t$ ), greatly reducing transmission overhead. However, addressing catastrophic forgetting remains a major challenge. Therefore, the subsequent TAPGP and DLPR modules will improve the model's anti-forgetting ability from the perspectives of protecting and inheriting learned knowledge. The overall process is detailed in Algorithm 1.

## 4.2. Task-Aware Prompt Gradient Projection

Current prompt learning methods often design shared prompts, task-specific prompts, or client-specific prompts to capture both shared and task/client-specific features. However, excessive specialization of components makes the

network complex, cumbersome, and lacking in scalability. Besides, storing these specific prompts brings additional storage overhead and privacy leakage risks. In this work, we propose a more elegant approach to balance the learning of shared and specific knowledge using the same prompts. We start from the perspective of model depth and learn both task-agnostic and task-specific knowledge simultaneously.

**Task-Agnostic Knowledge Learning.** As shown earlier, we utilize learnable visual prompts to activate specific knowledge within the pre-trained model, allowing it to better adapt to our task. Previous works [8, 16] have demonstrated that the shallow layers of the network primarily learn fundamental features of the image, which are common across tasks and clients and exhibit strong generalization capability. Inspired by this observation, we consider using the shallow layer prompts to capture task-agnostic and client-agnostic shared knowledge fully.

**Task-Specific Knowledge Learning.** The deeper layers of the model focus on processing task- and client-specific information, which is also the main stage where forgetting occurs. When the model overfits the features of specific data, it tends to forget previously learned knowledge, leading to the well-known problem of catastrophic forgetting. To better balance the relationship between new and old data knowledge, we apply gradient projection to the deep-layer prompts, alleviating the forgetting of previous knowl-

edge while learning task-specific knowledge for the new task. Specifically, recent research [37, 39] has observed that learning does not forget if the gradient is updated in a direction orthogonal to the subspace spanned by previous tasks. Orthogonal design prevents damage to old task knowledge. Based on the proof of PGP [37], for task  $t + 1$ , it is sufficient to satisfy the equation  $s^t \Delta p^T = 0$  to prevent forgetting knowledge from old tasks.  $\Delta p$  denotes the gradient of prompts and the element-wise sum space  $s^t = (e^t + p^t)$  between input space  $e^t$  and prompt space  $p^t$  from task  $t$ .  $e^t$  is a collection of image patch embeddings of data  $x^t$  for the task  $t$ . Under the FCIL, we are unable to obtain the data  $x^t$  due to the data being scattered across various clients. Thanks to the effective use of the generator, we can replace  $x^t$  with the synthetic data  $\hat{x}^t$  from  $\mathcal{M}^t$ . Therefore, we utilize image patch embeddings  $\hat{e}^t$  derived from  $\hat{x}^t$  and the aggregated prompt  $P^t$  to redefine anti-forgetting condition as:

$$s^t \Delta p^T = (\hat{e}^t + P^t) \Delta p^T = 0. \quad (3)$$

To solve Eq. (3), we can decompose  $s^t$  with Singular Value Decomposition (SVD):  $s^t = U_t \Sigma_t V_t^T$ . Here,  $U_t$  and  $V_t$  contain singular vectors corresponding to singular values in  $\Sigma_t$ . The diagonal matrix  $\Sigma_t$  can be further divided as:  $\Sigma_t = \begin{bmatrix} \Sigma_{t,1} & O \\ O & \Sigma_{t,0} \end{bmatrix}$  where  $\Sigma_{t,1}$  presents the non-zero elements of  $\Sigma_t$  (non-zero singular values) and  $\Sigma_{t,0}$  denotes the near-zero elements of  $\Sigma_t$ . Accordingly,  $V_t$  can be split into two components along the column dimension:  $V_t = [V_{t,1}, V_{t,0}]$ . Therefore, we obtain:

$$s^t [V_{t,1}, V_{t,0}] = U_t \begin{bmatrix} \Sigma_{t,1} & O \\ O & \Sigma_{t,0} \end{bmatrix} \Rightarrow s^t V_{t,0} = U_t \begin{bmatrix} O \\ \Sigma_{t,0} \end{bmatrix} \approx O. \quad (4)$$

Let  $\Delta p = \Delta p V_{t,0} V_{t,0}^T$ , we can get:

$$s^t \Delta p^T = s^t (\Delta p V_{t,0} V_{t,0}^T)^T = s^t V_{t,0} V_{t,0}^T \Delta p^T = O. \quad (5)$$

By using  $\hat{V} = V_{t,0}$  as the projection matrix, we can project the prompt gradient onto the orthogonal direction of the old task, thereby protecting learned knowledge.

Overall, in the process of backpropagation, it is implemented through the following form:

$$\begin{cases} p_{u-1} = p_{u-1} - \eta \Delta p_{u-1}, & \text{if } u < u_s, \\ p_{u-1} = p_{u-1} - \eta \phi(\Delta p_{u-1}), & \text{otherwise.} \end{cases} \quad (6)$$

where  $u_s$  is the layer at which gradient projection begins, and  $\phi(\Delta p_{u-1}) = \Delta p_{u-1} \hat{V}_{u-1} \hat{V}_{u-1}^T$  represents the prompt gradient projection operation.  $\Delta p_{u-1}$  and  $\hat{V}_{u-1}$  respectively denote the gradient of prompts and projection matrix at the  $u$ -th layer.  $\eta$  is learning rate. Therefore, after the server generates the virtual dataset  $\mathcal{M}^t$ , we need to further use the synthetic data to obtain the projection matrix  $\hat{V}$  for subsequent tasks.

### 4.3. Dual-Level Prompt Replay

Knowledge replay is a common technique used to mitigate catastrophic forgetting. Previous methods [17, 38, 44] often involve replaying data or prototypes, which may pose a risk of privacy breaches in the FL context. In our model, the pre-trained model parameters are frozen, and only the prompts are learned to activate specific knowledge within the model, allowing it to adapt to our specific task. Since the knowledge we learn is stored in the prompts, to better inherit learned knowledge and enhance training stability, we propose dual-level prompt replay.

- **Intra-Task Prompt Replay.** Due to the severe data heterogeneity across different clients, local models may experience drift. To reduce the interference of local model drift on the global model, we propose intra-task prompt replay to enhance the stability of the training process. As shown below:

$$P^{tr} = \alpha P^{tr} + (1 - \alpha) P^{tr-1}. \quad (7)$$

where  $P^{tr}$  represents the aggregated prompt for the  $r$ -th round in task  $t$ . We use  $\alpha$  to balance their proportions and set  $\alpha = 0.5$ .

- **Inter-Task Prompt Replay.** To better inherit previously learned knowledge by the model, we propose inter-task prompt replay. Essentially, this involves integrating the previously learned server prompts  $P^{t-1}$  with the current task prompts  $p^t$ . We discuss two ways to implement: (I) Mixing prompts. Specifically, for each training round  $r$ , we perform the following operations:

$$\bar{p}_{u-1}^{tr} = \lambda p_{u-1}^{tr} + (1 - \lambda) P_{u-1}^{t-1}, \quad (8)$$

$$L_u ([cls_{u-1}, e_{u-1}, \bar{p}_{u-1}^{tr}]) \quad u = 1, \dots, U. \quad (9)$$

where  $\lambda$  is the weight coefficient.  $p_{u-1}^{tr}$  is the prompt for the  $u$ -th layer of the model in the  $t$ -th task and the  $r$ -th training round. This approach ensures that the optimization process is always conditional on  $P^{t-1}$ , thereby alleviating forgetting. After  $R$  rounds of training, we finally update  $P^t = \lambda P^t + (1 - \lambda) P^{t-1}$ .

(II) Calculating attention. We use the attention operation to achieve merging, as shown below:

$$L_u ([cls_{u-1}, e_{u-1}, p_{u-1}^{tr}, P_{u-1}^{t-1}]) \quad u = 1, \dots, U. \quad (10)$$

Through experiments, it is found that the weighted recombination method of (I) is more effective, which is adopted in our other experiments as well.

## 5. Experiments

### 5.1. Experimental Setting

We evaluate the performance of our proposed method on the CIFAR-100 and Tiny-ImageNet datasets. Using LANNER [42] as the baseline, we ensure experimental fairness

---

**Algorithm 1:** TPPR

---

**Input:**  $T$ : number of tasks,  $R$ : number of communication rounds,  $E$ : local epochs.

```
1 for task  $t = 1$  to  $T$  do
2   if  $t \neq 1$  then
3      $\mathcal{M}^{t-1} = \text{DataGeneration}(\mathcal{S}^{t-1});$ 
4     for layer  $u = u_s$  to  $U$  do
5        $s_{u-1}^{t-1} = \hat{c}_{u-1}^{t-1} + P_{u-1}^{t-1};$ 
6        $\hat{V}_{u-1} =$ 
7          $\text{ProjectionMatrixGeneration}(s_{u-1}^{t-1});$ 
8     for round  $r = 1$  to  $R$  do
9       for client  $k = 1$  to  $m$  do
10         $p_r^{t,k} = P_r^{t-1};$ 
11         $\bar{p}_r^{t,k} = \lambda p_r^{t,k} + (1 - \lambda)P_r^{t-1};$ 
12         $p_r^{t,k} = \text{ClientUpdate}(\hat{V});$ 
13         $P_r^{t,k} = \sum_{k=1}^m \frac{n_k}{n} p_r^{t,k};$ 
14        // intra-task prompt replay;
15        if  $r \neq 1$  then  $P_r^{t,k} = \alpha P_r^{t,k} + (1 - \alpha)P_r^{t-1};$ 
16        // inter-task prompt replay;
17        if  $t \neq 1$  then  $P^t = \lambda P^t + (1 - \lambda)P^{t-1};$ 
18 ClientUpdate( $\hat{V}$ )
19   for  $E$  epochs do
20     for layer  $u = 1$  to  $U$  do
21       // task-aware prompt gradient projection;
22       if  $t \neq 1$  and  $u \geq u_s$  then
23          $p_{u-1} = p_{u-1} - \eta \Delta p_{u-1} \hat{V}_{u-1} \hat{V}_{u-1}^T;$ 
24       else
25          $p_{u-1} = p_{u-1} - \eta \Delta p_{u-1};$ 
26   return  $p;$ 
27 ProjectionMatrixGeneration( $s^t$ )
28    $s^t = U_t \Sigma_t V_t^T;$  // decomposing  $s^t$  with SVD;
29    $\Sigma_t = \begin{bmatrix} \Sigma_{t,1} & O \\ O & \Sigma_{t,0} \end{bmatrix};$   $V_t = [V_{t,1}, V_{t,0}];$ 
30   return  $V_{t,0};$ 
```

---

by maintaining the same settings and evaluation metrics. We simulate incremental learning by dividing the dataset classes into 5 and 10 tasks under both IID and non-IID conditions. A frozen pre-trained vision transformer is used as the backbone, and only the learnable prompts are trained.

## 5.2. Main Results

**Comparison with SOTA Methods.** Following [42], we conduct experiments under both IID and non-IID settings, with 5 and 10 tasks, respectively. Tables 1 and 2 present the average accuracy and forgetting measure across all tasks under various experimental configurations on the CIFAR-100

and Tiny-ImageNet datasets. As shown in Table 1, compared to state-of-the-art methods, our approach achieves an accuracy improvement of over 23% across all settings on the CIFAR-100 dataset. Even more surprised, as indicated in Table 2, on Tiny-ImageNet, our method demonstrates a performance enhancement exceeding 43% across all configurations. Furthermore, our experiments cover all possible combinations of the degree of data heterogeneity and task numbers, comprehensively demonstrating the superiority of our method and providing a reference for future research.

**Comparison in Same Backbone.** To eliminate the influence of the backbone on performance evaluation, we conduct experimental comparisons under a consistent backbone. We integrate the SOTA method with VPT-Deep and evaluate their performance across different data distributions and task quantities. As shown in Tables 1 and 2, on the CIFAR-100 and Tiny-ImageNet datasets, the introduction of VPT-Deep significantly improves performance compared to the SOTA method. This suggests that the pre-trained model provides rich and beneficial prior knowledge, making it an effective approach for improving accuracy. Importantly, compared to LANDER+VPT-Deep, our method achieves an additional over 8% improvement on CIFAR-100 across most settings. On the Tiny-ImageNet dataset, our approach further improves performance by 10% at  $T = 5$  and by 20% at  $T = 10$ , highlighting the effectiveness of our method.

**Comparison of Solving Catastrophic Forgetting.** As shown in Table 1, although the accuracy of LANDER+VPT-Deep surpasses that of the baseline LANDER, the forgetting degree  $\mathcal{F}$  is also significantly higher than that of LANDER under IID and  $\beta = 1$ . This suggests that while pre-trained models offer rich prior knowledge and can quickly capture key features in images, they still face significant catastrophic forgetting issues when learning new tasks. Our method effectively alleviates forgetting from the perspective of protecting and inheriting old task knowledge, reducing  $\mathcal{F}$  to 10%–20% compared to LANDER+VPT-Deep.

**Comparison in Less Heterogeneous Settings.** In Table 1, LANDER+VPT-Deep experience a decline in accuracy when there is less heterogeneity in certain cases. Pre-trained models incorporate a vast amount of prior knowledge. However, when task heterogeneity decreases, the model becomes more susceptible to overfitting to new tasks. This overfitting exacerbates the forgetting of previously learned knowledge, ultimately leading to a decline in the average accuracy across all tasks. Figure 3 shows the performance of LANDER+VPT-Deep and TPPR under the settings of  $T = 5$ . As shown in Figure 3 (a), on the 5-th task, although the highest accuracy is significantly higher when  $\beta = 0$  compared to  $\beta = 1$ , the previously learned knowledge has been severely forgotten, resulting in a lower average accuracy across all tasks than when  $\beta = 1$ . In contrast,

Data partition		IID				NIID(1)				NIID(0.5)			
Tasks		$T = 5$		$T = 10$		$T = 5$		$T = 10$		$T = 5$		$T = 10$	
Methods		Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )
FedEWC [25]		16.51	71.12	8.01	65.06	16.06	68.02	8.84	62.14	16.86	62.40	8.04	65.23
FedWeIT [55]		28.45	52.12	20.39	43.18	28.56	49.84	19.68	45.82	24.57	45.96	15.45	48.54
FedLwF [28]		30.61	45.32	23.27	37.71	30.94	42.71	21.16	41.03	27.59	41.25	17.98	45.23
TARGET (ICCV'23) [58]		36.31	32.23	24.76	35.45	34.89	34.48	22.85	38.25	33.33	39.23	20.71	42.23
MFCL (NeurIPS'23) [2]		42.67	-	31.35	-	-	-	-	-	41.19	-	28.99	-
DDDR (ECCV'24) [30]		51.04	-	43.45	-	-	-	-	-	48.45	-	41.27	-
FedProK (CVPRW'24) [12]		-	-	-	-	41.02	-	26.94	-	39.61	-	26.07	-
LANDER (CVPR'24) [42]		52.60	18.03	40.21	25.56	51.78	18.92	37.21	28.92	48.23	30.61	33.35	32.86
LANDER+VPT-Deep		68.61	27.74	57.57	33.93	69.49	21.84	51.72	36.02	64.58	23.52	54.53	31.17
TPPR (ours)		77.44	11.61	66.08	13.11	75.08	11.20	64.95	13.83	73.84	11.30	61.42	19.19

Table 1. Comparison with SOTA methods on CIFAR-100. The experiment is conducted under different task quantities ( $T = 5$  and  $T = 10$ ) with both IID and non-IID settings. NIID( $\beta$ ) indicates that the Dirichlet parameter is set to  $\beta$ , ‘Acc’ denotes average accuracy, and ‘ $\mathcal{F}$ ’ represents the forgetting measure [42]. The lower the  $\beta$ , the more severe the data heterogeneity. The results for the compared method are sourced from the SOTA method [30, 42, 58].

Data partition		IID				NIID(0.5)			
Tasks		$T = 5$		$T = 10$		$T = 5$		$T = 10$	
Methods		Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )
FedLwF [28]		24.32	34.57	-	-	21.76	37.23	-	-
TARGET (ICCV'23) [58]		26.25	23.43	-	-	23.95	24.58	-	-
MFCL (NeurIPS'23) [2]		15.11	-	10.13	-	13.35	-	8.54	-
DDDR (ECCV'24) [30]		25.47	-	19.01	-	23.96	-	16.65	-
LANDER (CVPR'24) [42]		30.29	21.65	-	-	27.98	23.03	-	-
LANDER+VPT-Deep		63.31	31.7	42.74	52.31	60.54	31.06	44.06	45.34
TPPR (ours)		74.63	13.04	65.96	21.78	71.16	10.80	63.23	19.33

Table 2. Comparison with SOTA methods on Tiny-ImageNet.

Method	Structure of Training	Trainable Parameter Quantities
LANDER (baseline) [42]	Whole Model	11.5 M
TPPR (ours)	Prompts	0.1 M ( $\downarrow$ 99%)

Table 3. Comparison of trainable parameter quantities.

our method, as shown in Figure 3 (b), mitigates overfitting to new tasks and performs well on all tasks.

Additionally, we exclude prompt pool-based methods from the comparison due to differing experimental settings and method types. Our approach achieves knowledge retention under the same parameters. In contrast, prompt pool methods improve performance by storing all learned knowledge (i.e., prompts), which not only fails to address knowledge inheritance and interference issues but also may pose privacy risks, making the comparison unfair to our method.

### 5.3. Comparison of Trainable and Transmission Parameter Quantities

Current SOTA methods necessitate training the entire model, leading to significant training overhead and model transmission latency. This makes them unsuitable for federated learning scenarios involving long distances and

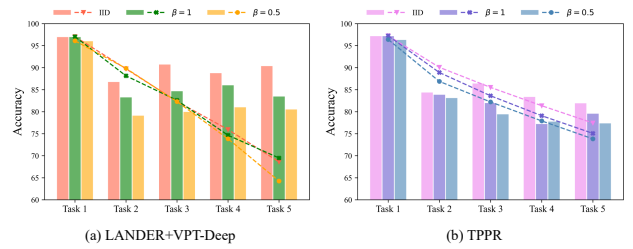


Figure 3. Performance analysis of Lander+VPT-Deep and TPPR. The bar chart illustrates the highest accuracy achieved on the  $t$ -th task, while the curve represents the average accuracy across the  $t$ -th task and all preceding tasks.

resource-constrained edge client devices. In contrast, our approach offers a more efficient solution by only training learnable prompts to adapt the pre-trained model to specific tasks, thereby substantially reducing both training and model communication costs. As illustrated in Table 3, the number of parameters we need to train is only 0.1 million, which is a mere 1% of the 11.5 million parameters required by the baseline method. This reduction in parameter quantity makes our approach more adaptable and efficient for deployment in demanding federated learning environments.

### 5.4. Ablation Studies and Analysis

In Table 4, we report the results of the proposed modules under task quantity  $T = 5$  and Dirichlet parameter  $\beta = 0.5$ . **Effectiveness of TAPGP.** TAPGP is grounded in a theoretical framework that ensures the gradient direction for prompt updates remains orthogonal to the prompts of previous tasks. This strategy effectively prevents interference between the knowledge of previous and subsequent tasks,

Methods	Components				NIID(0.5)	
	VPT-Deep	TAPGP	intra-	inter-	Acc( $\uparrow$ )	$\mathcal{F}$ ( $\downarrow$ )
LANDER [42]					48.23	30.61
TPPR	$\checkmark$				64.58	23.52
	$\checkmark$		$\checkmark$		66.95	19.61
	$\checkmark$			$\checkmark$	67.72	14.96
	$\checkmark$		$\checkmark$	$\checkmark$	70.45	15.44
	$\checkmark$	$\checkmark$			69.11	20.96
	$\checkmark$	$\checkmark$	$\checkmark$		71.28	16.83
	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	72.05	13.46
	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	73.84	11.30

Table 4. Effectiveness of proposed components.

addressing the catastrophic forgetting problem at its core. Building upon LANDER+VPT-Deep, TAPGP yields an additional improvement of approximately 4.5%, achieving an accuracy of 69.11%.

**Effectiveness of DLPR.** We employ a fully frozen pre-trained model, where knowledge learned from the current task is encoded within the prompts. Replay mechanisms for prompts can significantly enhance performance. Specifically, inter-task prompt replay effectively mitigates knowledge forgetting between tasks, resulting in an accuracy increase of about 3%, reaching 67.72%. Intra-task prompt replay helps alleviate the negative impact from data heterogeneity across different clients, improving model training stability and boosting accuracy by approximately 2.5%, reaching 66.95%. The combined use of both prompt replay mechanisms further enhances accuracy to 70.45%.

Importantly, the joint application of TAPGP and DLPR achieves a synergistic effect, delivering a remarkable accuracy of 73.84%. This demonstrates that the integration of these two modules effectively reinforces their complementary strengths. Moreover, any combination of proposed modules can further improve performance, indicating that our proposed modules have good compatibility and truly solve problems from the core of the problem. The data presented in the table strongly supports the claim that our proposed modules effectively mitigate catastrophic forgetting between tasks and data heterogeneity among clients, leading to outstanding performance.

**Analysis of Gradient Projection.** Considering that shallow layers of neural networks primarily capture low-level shared features such as edges and lines, while deeper layers focus on task-specific high-level features, we propose using task-aware prompt gradient projection (TAPGP). Figure 4 (a) illustrates the average accuracy of the model across all tasks when trained on the last task, with the task quantity set to  $T = 5$  and the Dirichlet parameter  $\beta = 0.5$ . The experimental data shown in the figure are obtained without utilizing the TAPGP module, which reduces interference and provides a more accurate reflection of its effective-

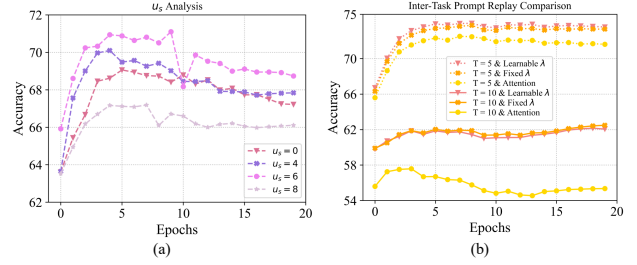


Figure 4. (a) Analysis of task-aware prompt gradient projection starting from different levels of the model. (b) Performance comparison of several different inter-task prompt replay methods.

ness. From the figure, we observe that as the layer of starting to projection increases (from  $u_s = 0$  to  $u_s = 6$ ), model performance gradually improves, reaching its peak at  $u_s = 6$ . This supports our initial hypothesis, demonstrating that deep prompt gradient projection effectively resolves conflicts between tasks and mitigates knowledge forgetting. However, our model consists of only 12 layers, and performance begins to decline when  $u_s = 8$ . This suggests that excessively high prompt gradient projection levels may not be sufficient for effective knowledge protection. In summary, our experiments indicate that the optimal performance is achieved when  $u_s = 6$ .

**Analysis of Prompt Replay.** We evaluate two different methods for inter-task prompt replay: mixing prompts and calculating attention. As shown in Figure 4 (b), under the setting of Dirichlet parameter  $\beta = 0.5$  and task quantities  $T = 5$  and  $T = 10$ , mixing prompts outperform calculating attention. Mixing prompts can be further categorized into two types: learnable  $\lambda$  and fixed  $\lambda$ , but their performance differences are not significant. The parameter  $\lambda$  balances the prompts from the previous task and the current task. Since our model consists of only 12 layers and thus has only 12 learnable  $\lambda$  parameters, this number is probably insufficient to capture complex relationships. However, this operation of inheriting old task knowledge is an effective means of alleviating forgetting.

## 6. Conclusion

We have developed a parameter-efficient turning method for FCIL based on visual prompts. TAPGP prevents parameter interference by ensuring that the gradients of new tasks remain orthogonal to the knowledge of preceding tasks. The proposed DLPR enables the review of knowledge at both inter-task and intra-task levels, significantly enhancing model stability and knowledge transfer. Our approach tackles FCIL through universal prompts, which not only reduces the cost of training and model communication but also ensures full privacy protection. Extensive experiments have validated the effectiveness and efficiency of our method.

## 7. Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62176224, No. 62176092, No. 62222602, No. 62306165), Science and Technology on Sonar Laboratory (No. 2024-JCJQ-LB-32/07), and China Academy of Railway Sciences under (No. 2023Y1357).

## References

- [1] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific reports*, 12(1):1953, 2022. 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. *NIPS*, 36, 2024. 3, 7
- [3] Gaurav Bagwe, Xiaoyong Yuan, Miao Pan, and Lan Zhang. Fed-cprompt: Contrastive prompt for rehearsal-free federated continual learning. *arXiv preprint arXiv:2307.04869*, 2023. 3
- [4] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, pages 10925–10934, 2022. 2
- [5] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, pages 5008–5017, 2021. 2
- [6] Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *CVPR*, pages 12077–12086, 2024. 2
- [7] Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. Harmonizing generalization and personalization in federated prompt learning. *arXiv preprint arXiv:2405.09771*, 2024. 2
- [8] Wenlong Deng, Christos Thrampoulidis, and Xiaoxiao Li. Unlocking the potential of prompt-tuning in bridging generalized and personalized federated learning. In *CVPR*, pages 6087–6097, 2024. 2, 4
- [9] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *CVPR*, pages 10164–10173, 2022. 1, 2
- [10] Eros Fani, Marco Ciccone, and Barbara Caputo. Feddrive v2: an analysis of the impact of label skewness in federated semantic segmentation for autonomous driving. *arXiv preprint arXiv:2309.13336*, 2023. 1
- [11] Lidia Fantauzzo, Eros Fani, Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11504–11511. IEEE, 2022. 1
- [12] Xin Gao, Xin Yang, Hao Yu, Yan Kang, and Tianrui Li. Fedprok: Trustworthy federated class-incremental learning via prototypical feature knowledge transfer. In *CVPRW*, pages 4205–4214, 2024. 7
- [13] Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *CVPR*, pages 28463–28473, 2024. 2
- [14] Haiyang Guo, Fei Zhu, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. Pilora: Prototype guided incremental lora for federated class-incremental learning. In *ECCV*, pages 141–159. Springer, 2024. 1
- [15] Tao Guo, Song Guo, and Junxiao Wang. Pfdprompt: Learning personalized prompt for vision-language models in federated learning. In *Proceedings of the ACM Web Conference 2023*, pages 1364–1374, 2023. 2
- [16] Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting, 2020. 4
- [17] Stella Ho, Ming Liu, Lan Du, Longxiang Gao, and Yong Xiang. Prototype-guided memory replay for continual learning. *IEEE transactions on neural networks and learning systems*, 2023. 5
- [18] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *NIPS*, 35:33716–33727, 2022. 2
- [19] Wenke Huang, Mang Ye, Zekun Shi, and Bo Du. Generalizable heterogeneous federated cross-correlation and instance similarity learning. *TPAMI*, 2023. 1
- [20] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, pages 16312–16322. IEEE, 2023. 2
- [21] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. A federated learning for generalization, robustness, fairness: A survey and benchmark. *TPAMI*, 2024. 1
- [22] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M Hadi Amini. A survey on federated learning for resource-constrained iot devices. *IEEE Internet of Things Journal*, 9(1):1–24, 2021. 1
- [23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727. Springer, 2022. 2, 3
- [24] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *ICCV*, pages 15190–15200, 2023. 2
- [25] 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. 7
- [26] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8423–8434, 2021. 1
- [27] Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *CVPR*, pages 12820–12829, 2024. 1

- [28] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 7
- [29] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *ICCV*, pages 5319–5329, 2023. 2
- [30] Jinglin Liang, Jin Zhong, Hanlin Gu, Zhongqi Lu, Xingxing Tang, Gang Dai, Shuangping Huang, Lixin Fan, and Qiang Yang. Diffusion-driven data replay: A novel approach to combat forgetting in federated class continual learning. In *ECCV*, pages 303–319. Springer, 2025. 3, 7
- [31] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 2
- [32] Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *IJCAI*, pages 2182–2188, 2022. 2
- [33] 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. 2, 3
- [34] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021. 1
- [35] Hongming Piao, Yichen Wu, Dapeng Wu, and Ying Wei. Federated continual learning via prompt-based dual knowledge transfer. In *ICML*. 2, 3
- [36] Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*, 2023. 1
- [37] Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *ICLR*, 2023. 3, 5
- [38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *NIPS*, 32, 2019. 5
- [39] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021. 3, 5
- [40] Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *CVPR*, pages 5055–5063, 2023. 2
- [41] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pages 11909–11919, 2023. 2, 3
- [42] Minh-Tuan Tran, Trung Le, Xuan-May Le, Mehrtash Harandi, and Dinh Phung. Text-enhanced data-free approach for federated class-incremental learning. In *CVPR*, pages 23870–23880, 2024. 1, 3, 4, 5, 6, 7, 8
- [43] Minh-Tuan Tran, Trung Le, Xuan-May Le, Mehrtash Harandi, Quan Hung Tran, and Dinh Phung. Nayer: Noisy layer data generation for efficient and effective data-free knowledge distillation. In *CVPR*, pages 23860–23869, 2024. 2
- [44] Guido M Van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):4069, 2020. 5
- [45] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *NIPS*, 36, 2024. 2
- [46] Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, Ming Gao, et al. Dfrd: Data-free robustness distillation for heterogeneous federated learning. *NIPS*, 36, 2024. 2
- [47] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, pages 631–648. Springer, 2022. 2, 3
- [48] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pages 139–149, 2022. 2
- [49] Nannan Wu, Li Yu, Xin Yang, Kwang-Ting Cheng, and Zengqiang Yan. Fediic: Towards robust federated learning for class-imbalanced medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 692–702. Springer, 2023. 2
- [50] An Xu, Wenqi Li, Pengfei Guo, Dong Yang, Holger R Roth, Ali Hatamizadeh, Can Zhao, Daguang Xu, Heng Huang, and Ziyue Xu. Closing the generalization gap of cross-silo federated medical image segmentation. In *CVPR*, pages 20866–20875, 2022. 1
- [51] Kunlun Xu, Zichen Liu, Xu Zou, Yuxin Peng, and Jiahuan Zhou. Long short-term knowledge decomposition and consolidation for lifelong person re-identification. *TPAMI*, 2025. 2
- [52] Kunlun Xu, Xu Zou, Gang Hua, and Jiahuan Zhou. Componential prompt-knowledge alignment for domain incremental learning. In *ICML*, 2025. 2
- [53] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. Efficient model personalization in federated learning via client-specific prompt generation. In *ICCV*, pages 19159–19168, 2023. 1
- [54] Seungryoung Yoo, Eunji Kim, Dahuin Jung, Jungbeom Lee, and Sungroh Yoon. Improving visual prompt tuning for self-supervised vision transformers. In *ICML*, pages 40075–40092. PMLR, 2023. 2
- [55] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *ICML*, pages 12073–12086. PMLR, 2021. 7
- [56] Hao Yu, Xin Yang, Xin Gao, Yan Kang, Hao Wang, Junbo Zhang, and Tianrui Li. Personalized federated continual learning via multi-granularity prompt. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4023–4034, 2024. 1

- [57] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *ICML*, pages 26311–26329. PMLR, 2022. [2](#)
- [58] Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *ICCV*, pages 4782–4793, 2023. [1](#), [3](#), [7](#)
- [59] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*, pages 10174–10183, 2022. [2](#)
- [60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. [2](#)
- [61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. [2](#)
- [62] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, pages 12878–12889. PMLR, 2021. [2](#)