Federated Continual Learning via Prompt-based Dual Knowledge Transfer

Hongming Piao [*] [1] **Yichen Wu** [*] [1] **Dapeng Wu** [1] **Ying Wei** [2]

Abstract

In Federated Continual Learning (FCL), the challenge lies in effectively facilitating knowledge transfer and enhancing the performance across various tasks on different clients. Current FCL methods predominantly focus on avoiding interference between tasks, thereby overlooking the potential for positive knowledge transfer across tasks learned by different clients at separate time intervals. To address this issue, we introduce a **P**rompt-based kn**ow**le**d**ge transf**er** FCL algorithm, called **Powder**, designed to effectively foster the transfer of knowledge encapsulated in prompts between various sequentially learned tasks and clients. Furthermore, we have devised a unique approach for prompt generation and aggregation, intending to alleviate privacy protection concerns and communication overhead, while still promoting knowledge transfer. Comprehensive experimental results demonstrate the superiority of our method in terms of reduction in communication costs, and enhancement of knowledge transfer. Code is available at https://github.com/piaohongming/Powder.

1. Introduction

Federated learning (FL) is a distributed training framework that allows for learning from tasks on different clients without transmitting raw data, while continual learning (CL) aims to enable a model to continuously learn new tasks without catastrophic forgetting. From spatial dimension and temporal dimension respectively, FL and CL try to tackle the non independent and identically (non-iid) distribution between tasks, which poses challenges to the model's memory stability and knowledge transfer between tasks. However, existing federated learning algorithms are based on the assumption that there is a stable data distribution on clients, thus struggle to adapt to the continuous changes of data distribution in the environment of clients. Meanwhile, Clients running centralized continual learning cannot utilize information from other clients under the premise of privacy protection, facing issues of insufficient and biased data. To address these problems, recent studies have begun focusing on federated continual learning (FCL). Federated continual learning retains the challenges of both federated learning and continual learning: knowledge transfer, catastrophic forgetting, privacy protection and communication overhead.

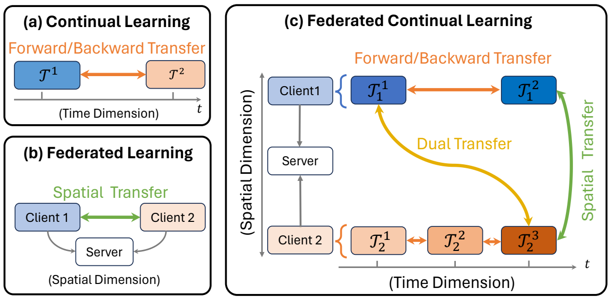

Figure 1. Various knowledge transfer types in different settings, where $\mathcal{T}_c^{t_c}$ denotes the t_c-th task on the c-th client.

Table 1. Transfer under different task correlation, measures by overlapped level between tasks.

METHOD	OVERLAPPED LEVEL		
	50%	70%	90%
FEDSPACE	-5.95	-3.11	-8.26
FED-CODAP	-1.18	-0.59	-0.73
OURS	6.04	6.40	6.68

Research in federated continual learning can be broadly categorized into **rehearsal-based** and **rehearsal-free** approaches. **rehearsal-based** methods use a memory buffer to save raw samples or prototypes of previous tasks, then replay them with model decomposition such as FedWEIT (Yoon et al., 2021), with knowledge distillation such as GLFC (Dong et al., 2022) and CFeD (Ma et al., 2022b), or with regularization such as Fedspace (Shenaj et al., 2023). However, memory buffer leads to additional storage overhead and privacy issue on clients, which are always edge devices with limited resource. To alleviate this problem, **rehearsal-free** methods such as TARGET (Zhang et al., 2023) and Fed-CIL (Qi et al., 2023) are proposed. They generate pseudo-

[*] Equal contribution [1] City University of Hong Kong [2] Nanyang Technological University. Correspondence to: Hongming Piao <hpiao6-c@my.cityu.edu.hk>, Dapeng Wu <dapengwu@cityu.edu.hk>, Ying Wei <ying.wei@ntu.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

samples by training additional generative models to replace the memory buffer. However, research on model inversion attacks (Carlini et al., 2023) shows that additional generative models pose privacy risks. Besides, methods above transmit the entire model parameters between server and clients, leading to unacceptable communication overhead. With the development of vision foundation models, **prompt-based** methods, which concatenate trainable parameters to multi-head self-attention layers, have gained increasing attention in federated continual learning. This is not only because their parameter-efficient characteristics, but also the capability of prompts to continuously enhance foundation models in a federated way. Fed-CPrompt (Bagwe et al., 2023) applies a global prompt pool consisting of task-specific prompts following CODAPrompt (Smith et al., 2023)(Sec.3), and designs a contrastive continual loss to address non-iid distribution among tasks. Similarly, HePCo (Halbe et al., 2023) introduces CODAPrompt and further trains a pseudo-representation generator on the server to address forgetting during global aggregation. However, existing prompt-based methods solve catastrophic forgetting by avoiding the impact between tasks, overlooking **dual knowledge transfer** over spatial and temporal dimension in federated continual learning, as shown in Fig.1. Furthermore, in real-world applications, there are different degrees of correlation between different tasks. we argue that knowledge transfer between tasks with high correlation brings better performance. Table 1 verifies this intuition by comparing the combined transfer (Sec.5.1) of Fedspace, Fed-CODAP and our method under different overlapped level between tasks. The knowledge transfer of ours improves with increased task correlation, but existing methods fail to achieve positive transfer despite task correlation. Additionally, as the number of tasks increases, the prompt pool size grows continuously. Communicating the entire prompt pool, which contains task-specific parameters, not only boosts **communication overhead**, but also leads to **privacy problem**.

In order to achieve positive dual knowledge transfer under acceptable communication overhead and privacy protection, we propose a **P**rompt-based dual kn**ow**le**d**ge transf**er** method in federated continual learning (**Powder**). Our main contributions are as follows:

- For federated continual learning, we first take into account the dual knowledge transfer which includes both spatial transfer and forward/backward transfer. And we construct a benchmark that can effectively evaluate the knowledge transfer under controllable different task correlation.

- We propose a prompt-based method that can 1) estimate task correlation for selective knowledge transfer; 2) transfer the most related knowledge among dual dimensions via two-step prompt aggregation, with the consideration of communication overhead and privacy. 3) Avoid the erasure of transferred knowledge via dual distillation loss.

- We conduct comprehensive experiments at different scales, with out-of-distribution dataset ImageNet-R and DomainNet commonly used for continual learning, providing empirical evidence for the effectiveness of Powder.

2. Related Work

Federated Learning. Federated learning is a distributed training framework that operates under the premise of privacy protection. In this framework, individual clients train parameters on their private data and upload them to a central server for global aggregation. This process allows for a better global model without the transmission of raw data. FedAvg (McMahan et al., 2016), a cornerstone of federated learning, aggregates models trained on multiple clients by computing a weighted average based on the number of samples on clients. The three crucial challenges in federated learning are forgetting alleviation, knowledge transfer among spatial dimension and communication overhead under non-iid distribution. For the first challenge, common methods involve adding regularization to models' weight (Li et al., 2020; Shoham et al., 2019) or output (Lee et al., 2021) to control their update directions. For the second challenge, beyond transferring with simple global aggregation in FedAvg, some personalized federated learning methods (Ma et al., 2022a; Chen et al., 2023) attempt to transfer the most related information by designing different client-specific aggregation algorithms. For the third challenge, the research focus is on training and transmitting only a subset of parameters (Chen et al., 2019) and accelerating convergence (Karimireddy et al., 2020). With the development of vision foundation model (e.g., Vision Transformer (Dosovitskiy et al., 2020)), recent research suggests that using parameter-efficient transfer learning methods, such as Visual Prompt Tuning (Jia et al., 2022) can effectively address these challenges (Feng et al., 2023; Yang et al., 2023). However, in real-world applications, client environments are constantly changing. Existing federated learning methods, based on the assumption of static client data distribution, can not adequately address this challenge.

Continual Learning. Continual Learning aims to enable a model to continuously learn new tasks while avoiding forgetting previous tasks. The two crucial challenges in continual learning are catastrophic forgetting and knowledge transfer among temporal dimensions. For the first challenge, Replay-based methods (Rebuffi et al., 2017; Aljundi et al., 2019; Wang et al., 2023; Wu et al., 2024) optimize the use of a memory buffer $\mathcal{M}$ storing samples from previous tasks, and design sampling algorithms to select and store the most representative of these samples. For the more challenging rehearsal-free scenario, regularization-based methods (Kirkpatrick et al., 2017; Zenke et al., 2017) limit or penalize the update of parameters with higher importance for previous tasks, while optimization-based methods explicitly control the optimization procedure of current tasks by performing gradient projection (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2020), meta-learning (Javed & White, 2019; Gupta et al., 2020), or finding robust representation (Mirzadeh et al., 2020b;a). For the second challenge, PR (Henning et al., 2021) utilizes the Bayesian strategy to learn task-specific posteriors with a shared meta-model. CUBER (Lin et al., 2022) categorizes regularization into different transfer types based on the angles between the tasks gradient and the sample space. D-TS (Ye & Bors, 2022) selectively distills knowledge for new tasks from a dynamically expanding teacher module. With the development of visual foundation models with robust representations, as well as the emergence of Visual Prompt Tuning (Jia et al., 2022), some prompt-based methods have shown remarkable effectiveness. L2P (Wang et al., 2022b) introduces a key-query similarity method to select prompts from a prompt pool and uses them to adapt the visual foundation model to different tasks in continual learning. Built upon L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a) divides the prompt pool into task-specific and task-invariant parts. CODAPrompt (Smith et al., 2023) transforms prompt selection into a differential process with an attention mechanism and adds a portion of prompts for each task in the prompt pool, achieving superior performance. However, existing prompt-based methods do not discuss knowledge transfer between tasks. Additionally, centralized continual learning faces data scarcity and biased data due to device environments and privacy concerns, but current methods are insufficient to address this issue.

3. Preliminaries

Problem Statement. Suppose there are N tasks $\mathcal{T} = \{\mathcal{T}^1, ..., \mathcal{T}^N\}$. In the scenario of asynchronous FCL with C clients sharing a single server, each client $c \in \{1, 2, ..., C\}$ can learn tasks sequentially at their own pace. We denote $\mathcal{T}_c^{t_c}$ as the current t_c-th task being learned by the c-th client.The model parameters of the c-th client during training the t_c-th task are denoted as $\boldsymbol{\theta}_c^{t_c}$. To clarify, we define the parameters of all clients on previous tasks $\mathcal{T}^{pre}$ as $\boldsymbol{\theta}^{\text{pre}}$ and the parameters of all clients on current task $\mathcal{T}^{cur}$ as $\boldsymbol{\theta}^{\text{cur}}$ as:

$$\boldsymbol{\theta}^{\text{pre}} \triangleq \boldsymbol{\theta}^{\text{pre}}_{[1:C]} = \cup_{c=1}^{C} \boldsymbol{\theta}_c^{[1:t_c-1]}, \;\; \boldsymbol{\theta}^{\text{cur}} \triangleq \boldsymbol{\theta}^{\text{cur}}_{[1:C]} = \cup_{c=1}^{C} \boldsymbol{\theta}_c^{t_c}$$

Besides, we use $|\cdot|$ to represent the size of sets, $[\,\cdot\,]_{i,j}$ to represent the ij-th entry of a matrix.

CODAPrompt. For prompt-tuning methods for ViT, We illustrate them and discuss which one has the best transfer capability in Appendix A. For the generation of prompts, existing prompt-based FCL methods (Bagwe et al., 2023; Halbe et al., 2023) mainly follow CODAPrompt (Smith et al., 2023), a state-of-the-art prompt-based CL method. During the training of $\mathcal{T}_c^{t_c}$, there is a set of task-specific prompts $\mathbf{P}_c^{t_c} \in \mathbb{R}^{M \times L \times D}$ for $\mathcal{T}_c^{t_c}$, where M is the length of the set, L is the length of a prompt, D is the output dimension of a ViT encoder. The prompts from previous tasks $\mathcal{T}^{pre}$ and current tasks $\mathcal{T}^{cur}$ collectively form the global prompt pool $\mathbf{P}_g \in \mathbb{R}^{M_g \times L \times D}$, where $M_g = M \times N$. $N = |\{\mathcal{T}^{pre}, \mathcal{T}^{cur}\}|$ is the number of existing tasks. For each sample x, its prompt $\mathbf{p} \in \mathbb{R}^{L \times D}$ is generated by a weighted sum of the prompts,

$$\mathbf{p} = \sum\nolimits_{m}^{M_g} \alpha_m [\mathbf{P}_g]_m \;, \tag{1}$$

where the weights $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{M_g}\}$ are achieved through query-key similarity:

$$\alpha = \{\gamma(q(x) \odot [\mathbf{A}_g]_1, [\mathbf{K}_g]_1), \gamma(q(x) \odot [\mathbf{A}_g]_2, [\mathbf{K}_g]_2), \\ \ldots, \gamma(q(x) \odot [\mathbf{A}_g]_{M_g}, [\mathbf{K}_g]_{M_g})\}, \tag{2}$$

where $\mathbf{K}_g \in \mathbb{R}^{M_g \times D}$ and $\mathbf{A}_g \in \mathbb{R}^{M_g \times D}$ are trainable keys and trainable attention weights corresponding to each prompt in $\mathbf{P}_g$ respectively. $\odot$ is the Hadamard product. $\gamma(\cdot, \cdot)$ is the cosine similarity. Due to the one-to-one correspondence between $\mathbf{A}_g$, $\mathbf{K}_g$, $\mathbf{P}_g$, we use $\mathbf{P}_g \in \mathbb{R}^{M_g \times (2+L) \times D}$ to represent them together as the global prompt pool for simplicity.

4. Method

4.1. Prompt Generation with Two-step Aggregation

We divide the prompt generation with query-key similarity (Sec.3) into two steps, aiming to transfer more useful knowledge with fewer parameters and privacy protection.

First-step. Initially, we estimate a dual task correlation matrix $\mathbf{G}_g^{\text{task}} \in \mathbb{R}^{M_g \times M_g}$ to represent the dual dimension correlations between existing tasks, where $[\mathbf{G}_g^{\text{task}}]_{i,j}$ represents the similarity between i-th task and j-th task. We update this matrix when new tasks emerge in the system. The specific estimation and update algorithms are detailed in Sec.4.3. With $\mathbf{G}_g^{\text{task}}$, the prompts corresponding to each task in the global prompt pool $\mathbf{P}_g \in \mathbb{R}^{M_g \times (2+L) \times D}$ are aggregated with other prompts based on task correlation to transfer the most relevant knowledge, that is:

$$\hat{\mathbf{P}}_g = \mathbf{G}_g^{\text{task}} \cdot \mathbf{P}_g, \tag{3}$$
$$\text{where } [\hat{\mathbf{P}}_g]_n = \sum\nolimits_{m}^{M_g} [\mathbf{G}_g^{\text{task}}]_{n,m} [\mathbf{P}_g]_m,$$

Second-step. Since transferring the entire prompt pool in each round would incur increasing communication and computational overhead, we select the top-k relevant tasks (including itself) for each task $\mathcal{T}_c^{t_c}$ based on the dual task

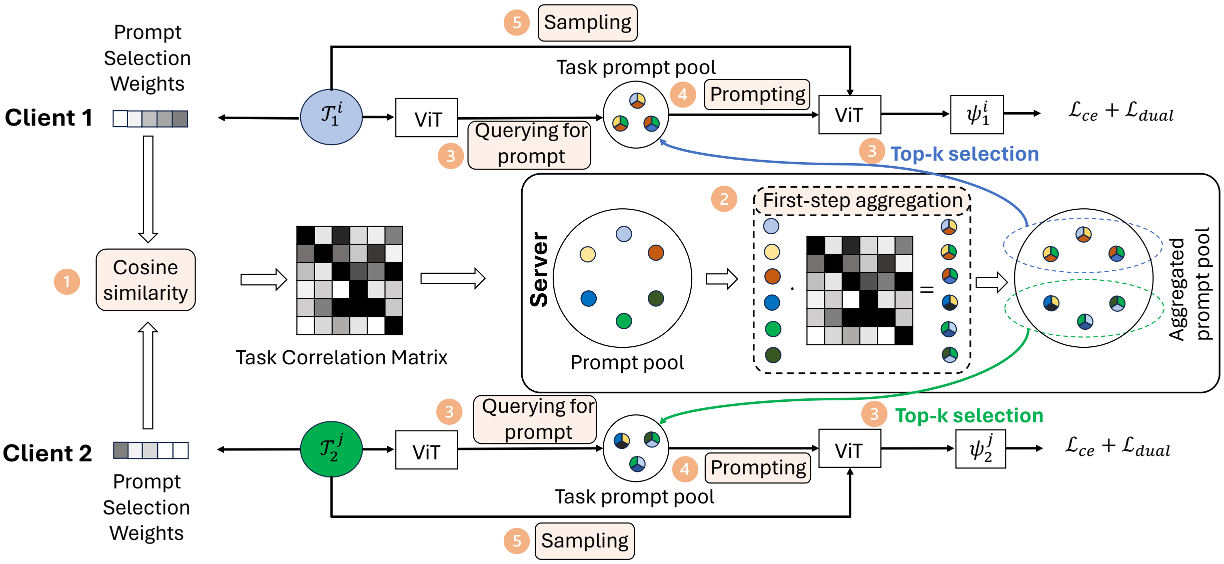

Figure 2. The proposed Powder. When new tasks $\mathcal{T}_1^i$ and $\mathcal{T}_2^j$ appears, the server requests for their prompt selection weights and update the task correlation matrix (Sec.4.3). During the training of $\mathcal{T}_1^i$ and $\mathcal{T}_2^j$, the server first operates first-step aggregation on global prompt pool by aggregating the most relevant task-specific prompts with the task correlation matrix (Sec.4.1). A task and its task-specific prompts in the server prompt pool are represented in the same color. In the second-step aggregation, the server selects smaller local prompt pool and transmits it to clients, which generate prompt for each sample via query-key similarity. (Sec.4.1). During the local training procedure, $\mathcal{L}_{dual}$ is proposed to retain the transferred knowledge.

correlation matrix $\mathbf{G}_g^{\text{task}}$, that is:

$$\mathbf{P}_{t_c}^{local} = \{ \, [\hat{\mathbf{P}}_g]_s \mid s \in \text{top-}k([\mathbf{G}_g^{\text{task}}]_{t_c}) \, \}, \tag{4}$$

where $[\mathbf{G}_g^{\text{task}}]_{t_c}$ is the correlation of $\mathcal{T}_c^{t_c}$ with other tasks. Then, the server transmits the aggregated prompts to the client where task $\mathcal{T}_c^{t_c}$ is located. The prompts for sample x of task $\mathcal{T}_c^{t_c}$ are calculated by aggregation with query-key similarity introduced in Sec.3, but with the local prompt pool $\mathbf{P}_{t_c}^{local}$.

Although only a subset of the global prompt pool is transmitted between the client and server to **mitigate the communication overhead**, the existence of the first-step aggregation allows the client to **forward transfer** knowledge from the entire global prompt pool, which has also been filtered through task correlation. Additionally, the task correlation matrix is continuously updated (Sec.4.3), and as new tasks emerge, previous tasks continuously gain knowledge from the most relevant new tasks, thus achieving **backward transfer**. Moreover, due to the first-step aggregation, we avoid transmitting prompts containing task-specific knowledge unaltered to the client, and the transmitted prompts have a high relevance to the tasks on the client, which **alleviates privacy concerns** to some extent.

4.2. Dual Distillation Loss

Although we have transferred knowledge from tasks with high relevance through a two-step aggregation, due to the non-iid distribution of tasks, tasks can easily overfit to the current task data during training, leading to the erasure of transferred knowledge. The mainstream methods to address this issue include adding regularization to model weights (Kirkpatrick et al., 2017; Li et al., 2020), knowledge distillation from previous models (Dong et al., 2022; Ma et al., 2022b; Shenaj et al., 2023), avoiding the impact on other tasks through gradient projection (Lin et al., 2022), etc. However, these methods limit the performance on individual tasks, which contradicts our goal of fully utilizing transferred knowledge to improve the performance of each task. Further, since there is dual correlation in terms of spatial and temporal dimensions between classes of different tasks, some classes of the current task may have little relevance to the knowledge transferred through prompts while other classes have high relevance with the knowledge. Therefore, for classes with little relevance, restricting their learning only has a negative effect. For classes with high relevance, we need to increase their constraints to align with the transferred prompts, namely learn and retain knowledge transferred from outside the local data distribution. Based on the analysis above, we propose a dual distillation loss as follows:

$$\mathcal{L}_{dual}(\hat{y}_{cu}, \hat{y}_{tr}) = -\beta \sum_{k=0,k\neq y}^{K} [\hat{y}_{tr}]_k \log \frac{[\hat{y}_{cu}]_k}{[\hat{y}_{tr}]_k}, \tag{5}$$

$$\text{where } \beta = \frac{\exp(\sum_i [\mathbf{G}_g^{\text{class}}]_{y,i})}{\sum_{(x',y')\in\mathcal{T}} \exp(\sum_i [\mathbf{G}_g^{\text{class}}]_{y',i})}.$$

In $\mathcal{L}_{dual}$, $\hat{y}_{cu}$ represents the output logits of the current model. $\hat{y}_{tr}$ represents the output logits of the model at the beginning of current round, which contains the newest knowledge transferred from other tasks. $\mathbf{G}_g^{\text{class}}$ represents the dual class correlation matrix after t-th tasks. We use the dual class correlation matrix $\mathbf{G}_g^{\text{class}}$ to control the degree of distillation for different samples. Specifically, $[\mathbf{G}_g^{\text{class}}]_{y,i}$ represents the correlation between class y and i, thus $\sum_i [\mathbf{G}_g^{\text{class}}]_{y,i}$ represents the overall correlation between class y and other classes. In this way, if class y has higher correlation with other classes, samples of class y will get a larger β. Additionally, to further mitigate the negative impact of distillation on sample learning, we follow FedNTD (Lee et al., 2021) by masking the logits corresponding to the sample labels ($k \neq y$) during the distillation process.

4.3. The Proposed Powder Algorithm

Task & Class Correlation Update. $\mathbf{G}_g^{\text{class}}$ is updated in the same way as $\mathbf{G}_g^{\text{task}}$, thus we take $\mathbf{G}_g^{\text{task}}$ as an example here. When task $\mathcal{T}_c^{t_c}$ first appears in the FCL system, the server initializes a set of prompts $\mathbf{P}_c^{t_c}$ in the global prompt pool $\mathbf{P}_g$. At the same time, the server initializes a new row and a new line in current task correlation matrix $\mathbf{G}_g^{\text{task}}$, denoted by the t_c-th row and line. In initialized $[\mathbf{G}_g^{\text{task}}]_{t_c,:}$ and $[\mathbf{G}_g^{\text{task}}]_{:,t_c}$, the relevance of task $\mathcal{T}_c^{t_c}$ with itself is 1 and with other tasks is 0. Subsequently, the server performs the first step aggregation as described in Sec.4.1, and sends the aggregated global prompt pool $\hat{\mathbf{P}}_g$ to the client c where task $\mathcal{T}_c^{t_c}$ is located. Client c calculates the average prompt selection weight $\alpha_c^{t_c}$ from the global prompt pool for task $\mathcal{T}_c^{t_c}$ and sends it back by

$$\alpha_c^{t_c} = \frac{1}{|\mathcal{T}_c^{t_c}|} \sum\nolimits_{x \in \mathcal{T}_c^{t_c}} \alpha^x, \tag{6}$$

where α^x is the prompt selection weights of sample x. The server determines the new task correlation matrix $\mathbf{G}_g^{\text{task}}$ between tasks by comparing the cosine similarity between different prompt selection weights:

$$[\mathbf{G}_g^{\text{task}}]_{n,m} = \left(\frac{\alpha_*^n \cdot \alpha_*^m}{\|\alpha_*^n\| \|\alpha_*^m\|} \right)^p, n, m \in [0, N], \tag{7}$$

where N represents the number of existing tasks and α^n represents the prompt selection weights of n-th tasks in the FCL system. p is a hyperparameter to increase the difference between different cosine similarity. It is noteworthy that to ensure the accuracy of correlation estimation, we transmit P_g to the clients when a new task appears in the system. However, in real-world applications, the same task often lasts for a relatively long time, and the rounds in which new tasks appear in the FCL system account for only a small part of the entire FCL process. Therefore, these communication overheads are minimal. We conduct experiments with new tasks appearing every 3 rounds, which is enough to show our advantage in communication overhead.

Algorithm 1 The training procedure of Powder.

```
Input Dataset $\{\mathcal{T}_{(1:c)}^{(1:t_c)}\}$, Pre-trained ViT encoder $\boldsymbol{\xi}$, hyperparameters $k, \lambda, p$, global prompt pool $\mathbf{P}_g \leftarrow \{\}$
Output global prompt pool $\mathbf{P}_g = \{\mathbf{P}_c^{t_c} | \mathcal{T}_c^{t_c} \in \mathcal{T}\}$, task-specific classification head $\boldsymbol{\psi}_g = \{\boldsymbol{\psi}_c^{t_c} | \mathcal{T}_c^{t_c} \in \mathcal{T}\}$, dual task correlation matrix $\mathbf{G}_g^{\text{task}}$, dual class correlation matrix $\mathbf{G}_g^{\text{class}}$
for round $r = 1, 2, \ldots$ do
    if new tasks set $|\mathcal{T}_r^{new}| > 0$ then
        for $\mathcal{T}_c^{t_c} \in \mathcal{T}_r^{new} \subset \mathcal{T}_r^{cur}$ do
            Initialize task-specific parameters $\mathbf{P}_c^{t_c} \in \mathbf{P}_g$, $\boldsymbol{\psi}_c^{t_c}$, $[\mathbf{G}_g^{\text{task}}]_{t_c,:}$, $[\mathbf{G}_g^{\text{task}}]_{:,t_c}$, $[\mathbf{G}_g^{\text{class}}]_{t_c,:}$, $[\mathbf{G}_g^{\text{class}}]_{:,t_c}$
        end for
        Server transmits $\hat{\mathbf{P}}_g$ calculated in Eq. (3) to clients.
        Clients calculate prompt weights for current tasks $\{\alpha_c^{t_c} | \mathcal{T}_c^{t_c} \in \mathcal{T}_r^{cur}\}$ and tasks just finished $\{\alpha_c^{t_c} | \mathcal{T}_c^{t_c} \in \mathcal{T}_r^{fin}\}$ by Eq.(6) and transmit to server.
        Server updates $\mathbf{G}_g^{\text{task}}$, $\mathbf{G}_g^{\text{class}}$ by Eq.(7)
    end if
    for current tasks $\mathcal{T}_c^{t_c} \in \mathcal{T}_r^{cur}$ do
        Server calculates $\hat{\mathbf{P}}_g$ by the first-step aggregation with Eq.(3)
        Server selects and transmits $\mathbf{P}_{t_c}^{local}$ to client $c$ by the second-step aggregation with Eq.(4)
        Solve the problem in Eq.(8) for local CL
        Client $c$ transmits optimized $\mathbf{P}_c^{t_c}$ to server and server updates $\mathbf{P}_g$
    end for
end for
```

Training & Inference In conclusion, we denote the parameters used for the inference of task $\mathcal{T}_c^{t_c}$ as $\boldsymbol{\theta}_c^{t_c}$, which includes fixed ViT encoder $\boldsymbol{\xi}$, local prompt pool $\mathbf{P}_{t_c}^{local}$ at the beginning of the round, task-specific classification head $\boldsymbol{\psi}_c^{t_c}$. During the training of task $\mathcal{T}_c^{t_c}$, we learn $\mathbf{P}_c^{t_c}$ and $\boldsymbol{\psi}_c^{t_c}$ by optimizing the following objective:

$$\min_{\mathbf{P}_c^{t_c}, \boldsymbol{\psi}_c^{t_c}} \mathcal{L}_{ce}(\boldsymbol{\theta}_c^{t_c}; \mathcal{T}_c^{t_c}) + \lambda \mathcal{L}_{dual}(\boldsymbol{\theta}_c^{t_c}; \mathcal{T}_c^{t_c}, \mathbf{P}_{t_c}^{local}), \tag{8}$$

where $\mathcal{L}_{ce}$ is cross entropy and λ is a hyperparameter to control the effect of distillation. We describe the Powder Algorithm in Algorithm1.

5. Experiment

5.1. Evaluation Benchmarks

Dataset: We construct our benchmarks based on two image datasets commonly used for prompt-based continual learning: **ImageNet-R** and **DomainNet**. ImageNet-R

(Hendrycks et al., 2021; Wang et al., 2022a) contains 30,000 samples from 200 categories, including hard samples from ImageNet (Deng et al., 2009) and newly collected samples with various styles. This dataset is more distant from the pre-trained distribution of ViT, making it suitable for FCL research based on vision foundation models. Following DualPrompt (Wang et al., 2022a), we split the dataset into a training set with 24,000 images and a test set with 6,000 images. To search for more suitable values of k and λ, we selected 20% of the training set as a validation set. DomainNet (Peng et al., 2019) includes 600,000 images from six domains and 345 classes, making it an important dataset for transfer learning research. Although DomainNet is relatively close to the pre-trained distribution of ViT, its prominent class diversity still reflects the effectiveness of our method to a certain extent. To fully validate our method and to better align with real-world applications, we considered several dimensions when constructing the benchmark:

- **Transferability**: In the real world, tasks encountered by different edge devices often have varying degrees of similarity. Therefore, we hope that the classes included in different tasks within the benchmark have varying levels of overlap, as better transfer performance might be achieved between tasks with higher overlap. Additionally, samples of the same class encountered by different edge devices in the real world are often biased, so we expect that each task contains only a small portion of the total data volume for each class.

- **Asynchrony**. Following (Yoon et al., 2021; Qi et al., 2023; Shenaj et al., 2023), in real-world FCL, edge devices switch tasks asynchronously, which presents additional challenges to avoid negative impacts between tasks.

Based on the above considerations, we design our benchmark. For ImageNet-R, each task contains 20 randomly selected classes from ImageNet-R, with each class containing 20% of the total samples of that class. These tasks are randomly distributed across all clients, with each task lasting a different number of rounds. For DomainNet, each task contains 35 randomly selected classes. Since DomainNet is closer to the pre-trained distribution, we randomly sample 2% of the total samples for each class, while the rest is the same as ImageNet-R. Our benchmark controls the overlap between tasks by controlling random selection of classes with the least overlap τ, in order to more thoroughly study the performance of FCL systems under different overall task correlation. It is worth noting that we did not use the common method of constructing tasks using a Dirichlet distribution in FL, as this assumes that all clients will learn the same class set in order to better control task similarity. However, in the FCL scenario, the class sets learned by different tasks on each client can vary greatly, even being disjoint, making it difficult to control task similarity using Dirichlet distribution sampling directly. To achieve asynchrony, after every 3 rounds, we randomly select 40% clients and switch their tasks.

Evaluation metrics: We evaluate the effectiveness, transfer capacity, and communication overhead by adapting six metric to FCL scenario, including Average Incremental Accuracy (AIA) (Douillard et al., 2020), Forgetting Measure (FM) (Chaudhry et al., 2018), Forward Transfer (FT) (Lopez-Paz & Ranzato, 2017), Backward Transfer (BT) (Lopez-Paz & Ranzato, 2017), Combined Transfer (CT), Total Communication Parameter Size between Client and Server(C2S&S2C).

- **Average Incremental Accuracy (AIA):** This metric measures the average accuracy over the FCL process, computed as AIA $= \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\sum_{\mathcal{T}_c^t\in\mathcal{T}_r} a_{c,r}^t$, where $\mathcal{R}$ denotes set of round with task switch, $\mathcal{T}_r$ denotes the set of existing tasks at round r, $a_{c,r}^t$ denotes the accuracy of $\mathcal{T}_c^t$ at round r.

- **Forgetting Measure (FM):** Forgetting is measured by the difference between the highest historical accuracy and the current accuracy of a task. This metric quantifies the model's memory stability by the average forgetting over the FCL process, computed as FM $= \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\sum_{\mathcal{T}_c^t\in\mathcal{T}_r} a_{c,r}^t - \widetilde{a}_{c,r}^t$, where $\widetilde{a}_{c,r}^t$ denotes the max accuracy of $\mathcal{T}_c^t$ before round r.

- **Forward Transfer (FT):** This metric assesses the model's ability to transfer knowledge into a task, from both previously learned tasks and other currently learned tasks, computed as FT $= \frac{1}{|\mathcal{T}|}\sum_{\mathcal{T}_c^t\in\mathcal{T}} \dot{a}_c^t - \hat{a}_c^t$, where $\mathcal{T}$ denotes all tasks during the FCL process, $\dot{a}_c^t$ denotes the accuracy of $\mathcal{T}_c^t$ when it finished and $\hat{a}_c^t$ denotes the accuracy of single task training.

- **Backward Transfer (BT):** This metric evaluates the model's ability to transfer knowledge from new tasks back to previously learned tasks, computed as BT $= \frac{1}{|\mathcal{T}|}\sum_{\mathcal{T}_c^t\in\mathcal{T}} a_{c,\max(\mathcal{R})}^t - \dot{a}_c^t$, where $a_{c,\max(\mathcal{R})}^t$ denotes the final accuracy of $\mathcal{T}_c^t$.

- **Combined Transfer (CT):** This metric is a combination of FT and BT, evaluating the amount of information that a task $\mathcal{T}_c^t$ acquires from other tasks. The other tasks can have any sequence relationship with task $\mathcal{T}_c^t$ in terms of temporal dimension. It is computed as CT $= \frac{1}{|\mathcal{T}|}\sum_{\mathcal{T}_c^t\in\mathcal{T}} a_{c,\max(\mathcal{R})}^t - \hat{a}_c^t$.

Baselines: To the best of our knowledge, we are the first to investigate dual task knowledge transfer in FCL based on visual foundation models. Therefore, our baselines are selected as follows: 1) We adapt state-of-the-art non-ViT-based FCL methods to ViT-based and our benchmark, including: FedWEIT (Yoon et al., 2021), CFeD (Ma et al., 2022b), GLFC (Dong et al., 2022), Fedspace (Shenaj et al., 2023). Due to the fact that prompt-tuning itself has a certain effect on improving the model's performance, and non-ViT-

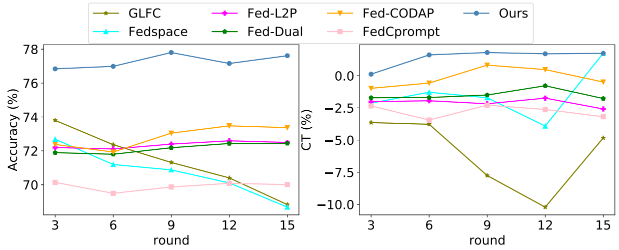

Figure 3. Changes in accuracy and transfer during the FCL process, on ImageNet-R with $C = 5$ clients and $r = 15$ rounds.

based methods have difficulty in transferring knowledge between tasks when the encoder is frozen, therefore, we combine these methods with basic prompt-tuning and CODAPrompt for a fair comparison. Please refer to Appendix D for their results with prompt-tuning and CODAPrompt. 2) We combine state-of-the-art prompt-based CL methods (Wang et al., 2022b;a; Smith et al., 2023) with FedAvg (McMahan et al., 2016), including: Fed-L2P, Fed-Dual, Fed-CODAP. 3) The latest ViT-based method: FedCPrompt (Bagwe et al., 2023). Please refer to Appendix B for more implementation details.

5.2. Performance Comparison

Table 2 demonstrates the performance of all methods during the FCL process, measured by the first five metrics in Sec.5.1, with 5 clients in the FCL system and 15 rounds in total. Powder not only achieves a significantly leading Average Incremental Accuracy (AIA) across both datasets but also realizes the only positive Combined Transfer(CT) among all methods. Not prompt-based methods, despite utilizing a memory buffer, still exhibit relatively high catastrophic forgetting. Prompt-based methods experience less forgetting, but Forward Transfer (FT) remains negative, indicating that the negative knowledge transfer between different tasks' prompts outweighs the positive transfer, highlighting the necessity for selective transfer.

Furthermore, we compare the communication overhead and the storage overhead of Powder with prompt-based methods in Table 3. We achieve a performance improvement of 9% to 12% while reducing the communication overhead by 20% to 40%. Besides, we have the minimum storage overhead during task training and inference, namely the minimum additional parameters. These advantages alleviates the problem of limited client resources, while achieving less training and inference time under the same device conditions as shown in Table 4. Not prompt-based methods, which require a memory buffer to store raw data or prototypes and are designed based on transferring all model parameters, are not included in the comparison. Additionally, we analyze the changes in average accuracy and average combined transfer during the FCL process. See Fig.3, where not prompt-based methods show a gradual decrease in accuracy over the CL process, while prompt-based methods can maintain or slightly increase accuracy. This indicates that in FCL based on vision foundation models, prompt-tuning not only achieves rehearsal-free learning but also results in lighter catastrophic forgetting. Meanwhile, the combined transfer of baselines generally fluctuates around a negative value, with only Powder consistently maintaining a stable positive value.

5.3. Ablation Study

As shown in Table 5, we evaluate the effects of each module through ablation studies. OURS in Table 5 represents the proposed Powder. FED-CODAP++ in Table 5 adapts the prompt-tuning method we use to CODAPrompt, proving that our advantage is not solely derived from the prompt-tuning method. AGG represents the first-step aggregation introduced in Section 4.1. SEL represents the selection of prompts from top-k related tasks during S2C communication. For a fair comparison, when removing SEL, we replace it with the randomly selected prompts of the same size. OURS-AGG and OURS-SEL in Table 5 demonstrate the necessity of filtering more relevant knowledge for transfer with AGG and SEL, and also prove that our proposed two-step aggregation achieves more effective dual knowledge transfer under communication-efficient conditions. DUAL represents the proposed dual distillation loss $\mathcal{L}_{dual}$, with DIS representing the classic knowledge distillation (Hinton et al., 2015), $\mathcal{L}_{dis}(\hat{y}_{cu}, \hat{y}_{tr}) = -\hat{y}_{tr}\log\frac{\hat{y}_{cu}}{\hat{y}_{tr}}$. We remove $\mathcal{L}_{dual}$ directly in OURS-DUAL in Table 5 and replace $\mathcal{L}_{dual}$ with the classic knowledge distillation (Hinton et al., 2015) in OURS-DIS in Table 5. They prove the significant role of $\mathcal{L}_{dual}$ in avoiding the erasure of transferred knowledge, while avoiding negative effect on learning new tasks. Additionally, as can be seen from Table 5, only when the various modules of Powder work in concert can the overall positive knowledge transfer be ultimately achieved.

To validate that Powder can maintain its leading performance in larger-scale real-world applications, we expand our experimental setting to 20 clients and 60 rounds, where each client sequentially learns up to 20 tasks. We selected Fedspace (Shenaj et al., 2023) and Fed-CODAP as representatives of not prompt-based and prompt-based methods, respectively. As can be seen from Fig.6, while existing non prompt-based and prompt-based methods suffer from negative knowledge transfer, Powder consistently maintains a higher accuracy and knowledge transfer capability in larger-scale FCL systems.

5.4. Effect of Top-k in Prompt Selection

Due to the critical role of top-k prompt selection in the second-step aggregation (Sec.4.1) for controlling commu-

Table 2. Performance measured by metrics in Sec.5.1, on ImageNet-R and DomainNet with with $C = 5$ clients and $r = 15$ rounds.

			ImageNet-R			
	Method	AIA(%)	FM(%)	FT(%)	BT(%)	CT(%)
No-prompt	FedWEIT-ViT	$29.26_{(0.82)}$	$18.12_{(0.63)}$	$6.74_{(1.27)}$	$-24.51_{(2.23)}$	$-12.87_{(1.00)}$
	CFeD-ViT	$59.79_{(1.17)}$	$3.81_{(0.39)}$	$-17.67_{(8.19)}$	$-14.92_{(2.65)}$	$-29.60_{(6.28)}$
	GLFC-ViT	$75.21_{(1.05)}$	$1.10_{(0.16)}$	$-3.87_{(0.70)}$	$-1.55_{(0.76)}$	$-5.11_{(0.99)}$
	FedSpace-ViT	$73.36_{(0.82)}$	$2.01_{(0.13)}$	$-2.60_{(0.59)}$	$-4.19_{(0.51)}$	$-5.95_{(0.48)}$
Prompt	Fed-L2P	$75.03_{(0.88)}$	$0.41_{(0.06)}$	$-2.79_{(0.13)}$	$-0.17_{(0.26)}$	$-2.92_{(0.23)}$
	Fed-Dual	$74.91_{(0.87)}$	$0.49_{(0.08)}$	$-3.12_{(0.17)}$	$0.22_{(0.32)}$	$-2.95_{(0.43)}$
	Fed-CODAP	$75.14_{(1.06)}$	$\mathbf{-0.68}_{(0.55)}$	$-2.53_{(2.87)}$	$1.69_{(1.63)}$	$-1.18_{(1.62)}$
	FedCPrompt	$72.59_{(0.44)}$	$0.63_{(0.06)}$	$-3.16_{(0.94)}$	$0.00_{(0.00)}$	$-3.16_{(0.94)}$
	Powder	$\mathbf{84.08}_{(0.56)}$	$-0.54_{(0.07)}$	$\mathbf{4.48}_{(0.13)}$	$\mathbf{1.95}_{(0.60)}$	$\mathbf{6.04}_{(0.50)}$
			DomainNet			
	Method	AIA(%)	FM(%)	FT(%)	BT(%)	CT(%)
No-prompt	FedWEIT-ViT	$28.58_{(0.82)}$	$17.12_{(1.10)}$	$7.38_{(0.74)}$	$-22.13_{(5.14)}$	$-10.32_{(3.37)}$
	CFeD-ViT	$60.19_{(0.23)}$	$1.65_{(0.55)}$	$-4.98_{(0.25)}$	$-13.32_{(0.51)}$	$-15.64_{(0.46)}$
	GLFC-ViT	$70.34_{(0.00)}$	$1.23_{(0.02)}$	$-4.08_{(0.42)}$	$-2.46_{(0.10)}$	$-6.04_{(0.50)}$
	FedSpace-ViT	$70.71_{(0.19)}$	$1.80_{(0.12)}$	$1.87_{(0.23)}$	$-4.16_{(0.22)}$	$-1.45_{(0.06)}$
Prompt	Fed-L2P	$72.36_{(0.44)}$	$0.16_{(0.03)}$	$-2.18_{(0.21)}$	$0.10_{(0.04)}$	$-2.09_{(0.24)}$
	Fed-Dual	$72.15_{(0.22)}$	$0.16_{(0.02)}$	$-1.82_{(0.11)}$	$0.41_{(0.03)}$	$-1.49_{(0.08)}$
	Fed-CODAP	$72.84_{(0.40)}$	$\mathbf{0.01}_{(0.04)}$	$-0.82_{(0.37)}$	$\mathbf{0.83}_{(0.28)}$	$-0.15_{(0.29)}$
	FedCPrompt	$69.92_{(0.56)}$	$0.19_{(0.09)}$	$-2.78_{(0.36)}$	$0.00_{(0.00)}$	$-2.78_{(0.36)}$
	Powder	$\mathbf{77.28}_{(0.18)}$	$0.10_{(0.06)}$	$\mathbf{1.28}_{(0.04)}$	$0.14_{(0.20)}$	$\mathbf{1.40}_{(0.19)}$

Table 3. Communication overhead and storage overhead compared with prompt-based methods. Communication overhead is measured on ImageNet-R with $C = 5$ clients and $r = 15$ rounds. Storage overhead is measured by additional parameters needed for the training and inference of a task on ImageNet-R.

Method	Communication	Storage
Fed-L2P	686.69MB	3.96MB
Fed-Dual	621.78MB	4.73MB
Fed-CODAP	815.63MB	11.43MB
Fed-CPrompt	815.63MB	11.43MB
Powder	**493.08MB**	**2.64MB**

Table 4. Average training and inference time between tasks compared with prompt-based methods on ImageNet-R with $C = 5$ clients and $r = 15$ rounds.

Method	Training	Inference
Fed-L2P	297.63s	5.69s
Fed-Dual	294.06s	6.56s
Fed-CODAP	316.03s	6.52s
Fed-CPrompt	458.89s	6.47s
Powder	**262.65s**	**3.73s**

nication overhead, we aim to avoid significant impact on performance caused by the selection of k. In the Fig.4, we test the AIA metric and CT metric for k ranging from 1 to 5, and the experimental results show that the AIA fluctuates around 0.2%, while the CT fluctuates around 0.5%. This demonstrates that our method provides users with ample room to balance model performance and communication overhead. In the experiments of this paper, we choose $k = 3$ with the highest AIA.

5.5. Effect of Task Frequency

In real-world applications, the task frequency can be variable, namely the minimal task duration can be variable. Therefore, we analyzed the performance of powder under different task frequencies. As can be seen in Fig.5, with the minimal task duration increases, the AIA first increases and then fluctuates around 84%. The CT first increases and then decreases, eventually stabilizing at around 4.84%. Powder still achieves positive dual knowledge transfer. When the minimum task duration is small, it is difficult to train a task adequately, and there is few interactions between tasks through the server. This leads to a lower AIA and CT. As the minimum task duration increases, the training and knowledge transfer for a task become more sufficient, and the performance of a task gradually reaches its upper bound, reflected by an increase in AIA. As for CT, knowledge transfer has the effect of accelerating training, enabling the model to improve its performance on a task more quickly. However, when calculating CT following evaluation metrics in Section 5.1, the performance improvement of a model trained with single task training is slower, and it takes a

Table 5. Ablation study on the proposed Powder, on ImageNet-R with $C = 5$ clients and $r = 15$ rounds.

					ImageNet-R				
Method	FUS	SEL	DUAL	DIS	AIA(%)	FM(%)	FT(%)	BT(%)	CT(%)
Fed-CODAP++	✗	✗	✗	✗	$74.96_{(1.47)}$	$\mathbf{-0.73}_{(0.76)}$	$-5.13_{(1.69)}$	$3.42_{(1.00)}$	$-2.39_{(1.87)}$
OURS-FUS	✗	✓	✓	✗	$79.88_{(0.83)}$	$0.56_{(0.36)}$	$-0.10_{(0.42)}$	$-0.04_{(0.86)}$	$-0.12_{(0.90)}$
OURS-SEL	✓	✗	✓	✗	$79.23_{(0.87)}$	$0.41_{(0.31)}$	$-0.76_{(0.39)}$	$0.12_{(0.49)}$	$-0.66_{(0.78)}$
OURS-DUAL	✓	✓	✗	✗	$76.38_{(0.89)}$	$1.48_{(0.22)}$	$-4.13_{(0.81)}$	$-0.30_{(0.59)}$	$-4.37_{(0.92)}$
OURS-DIS	✓	✓	✗	✓	$81.35_{(0.94)}$	$0.01_{(0.06)}$	$0.97_{(0.49)}$	$0.79_{(0.19)}$	$1.60_{(0.48)}$
OURS	✓	✓	✓	✗	$\mathbf{84.08}_{(0.56)}$	$-0.54_{(0.07)}$	$\mathbf{4.48}_{(0.13)}$	$\mathbf{1.95}_{(0.60)}$	$\mathbf{6.04}_{(0.50)}$

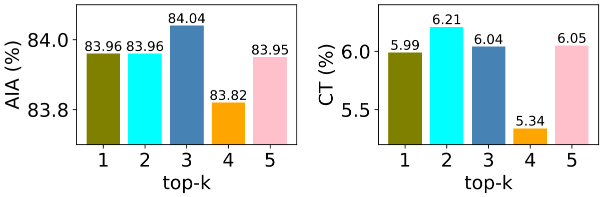

Figure 4. Effect of Top-k in prompt selection on the performance on ImageNet-R, with k ranging from 1 to 5.

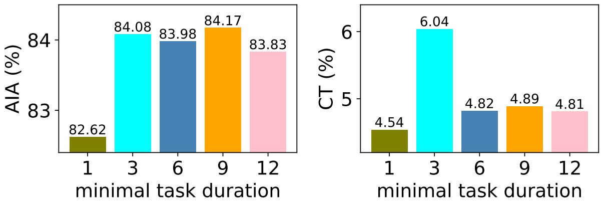

Figure 5. Effect of task frequency on the performance on ImageNet-R, with minimal task duration from 1-5.

longer task duration to reach the upper bound, which results in a decrease in CT as the task duration increases. This experimental result confirms the advantage of Powder in utilizing dual knowledge transfer to accelerate training. In summary, Powder is capable of adapting to asynchronous tasks with different duration in real-world applications.

6. Conclusion

In this paper, we propose Powder, to tackle the dual knowledge transfer in federated continual learning. Specifically, Powder selectively transfer the most related knowledge via task correlation estimation and two-step aggregation, which also takes the communication overhead and privacy protection into consideration. To retain the knowledge transferred from different tasks without limiting learning plasticity, we proposed a dual distillation loss based on class correlation estimation. Moreover, we construct a scalable benchmark with controllable task correlation for thorough evaluation. Comprehensive experiments verify that Powder is the first FCL method that achieves positive dual knowledge transfer. For the future work, we will continue to explore new prompt-tuning methods to achieve more efficient dual knowledge transfer in FCL.

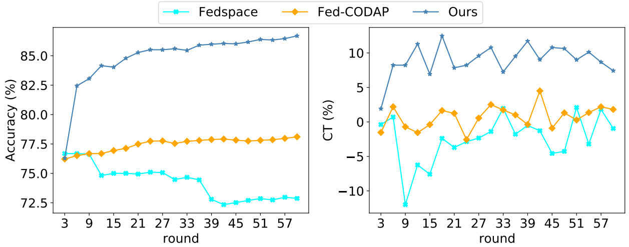

Figure 6. Accuracy and transfer in large-scale experiment, on ImageNet-R with $C = 20$ clients and $r = 60$ rounds.

Acknowledgements

We thank all the anonymous reviewers for their constructive suggestions on improving this paper. This work was supported by the Innovation and Technology Fund (No. MHP/034/22) funded by the Innovation and Technology Commission, the Government of the Hong Kong Special Administrative Region.

Impact Statement

This paper presents work whose goal is to advance the field of federated continual learning. Federated continual learning is a cutting-edge concept that holds immense significance in a variety of critical and rapidly evolving fields, including but not limited to medical imaging analysis, autonomous driving technology, and robotics. In these scenarios, different models face private, biased and changing environments, thus they need to transfer knowledge to each other without leaking private information to better adapt to new tasks and improve previous learned tasks. Our research has taken a systematic step towards addressing federated continual learning, leading to safer, more efficient, and more ethical applications across a broad spectrum of industries.

References

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Bagwe, G., Yuan, X., Pan, M., and Zhang, L. Fed-cprompt: Contrastive prompt for rehearsal-free federated continual learning. *arXiv preprint arXiv:2307.04869*, 2023.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018.

Chaudhry, A., Khan, N., Dokania, P., and Torr, P. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33:9900–9911, 2020.

Chen, H.-Y., Zhong, J., Zhang, M., Jia, X., Qi, H., Gong, B., Chao, W.-L., and Zhang, L. Federated learning of shareable bases for personalization-friendly image classification. *arXiv preprint arXiv:2304.07882*, 2023.

Chen, Y., Sun, X., and Jin, Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE transactions on neural networks and learning systems*, 31(10): 4229–4238, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., and Zhu, Q. Federated class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10164–10173, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pp. 86–102. Springer, 2020.

Feng, C.-M., Li, B., Xu, X., Liu, Y., Fu, H., and Zuo, W. Learning federated visual prompt in null space for mri reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8064–8073, 2023.

Gupta, G., Yadav, K., and Paull, L. Look-ahead meta learning for continual learning. *Advances in Neural Information Processing Systems*, 33:11588–11598, 2020.

Halbe, S., Smith, J. S., Tian, J., and Kira, Z. Hepco: Data-free heterogeneous prompt consolidation for continual federated learning. *arXiv preprint arXiv:2306.09970*, 2023.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Henning, C., Cervera, M., D'Angelo, F., Von Oswald, J., Traber, R., Ehret, B., Kobayashi, S., Grewe, B. F., and Sacramento, J. Posterior meta-replay for continual learning. *Advances in neural information processing systems*, 34:14135–14149, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Javed, K. and White, M. Meta-learning representations for continual learning. *Advances in neural information processing systems*, 32, 2019.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

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.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference* *on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lee, G., Jeong, M., Shin, Y., Bae, S., and Yun, S.-Y. Preservation of the global knowledge by not-true distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Lin, S., Yang, L., Fan, D., and Zhang, J. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177, 2022.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Ma, X., Zhang, J., Guo, S., and Xu, W. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10092–10101, 2022a.

Ma, Y., Xie, Z., Wang, J., Chen, K., and Shou, L. Continual federated learning based on knowledge distillation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, volume 3, 2022b.

McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2:2, 2016.

Mirzadeh, S. I., Farajtabar, M., Gorur, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*, 2020a.

Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020b.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Qi, D., Zhao, H., and Li, S. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*, 2023.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Shenaj, D., Toldo, M., Rigon, A., and Zanuttigh, P. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5054–5062, 2023.

Shoham, N., Avidor, T., Keren, A., Israel, N., Benditkis, D., Mor-Yosef, L., and Zeitak, I. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., and Kira, Z. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11909–11919, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Q., Wang, R., Wu, Y., Jia, X., and Meng, D. Cba: Improving online continual learning via continual bias adaptor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19082–19092, 2023.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648. Springer, 2022a.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022b.

Wu, Y., Huang, L.-K., Wang, R., Meng, D., and Wei, Y. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth International Conference on Learning Representations*, 2024.

Yang, F.-E., Wang, C.-Y., and Wang, Y.-C. F. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19159–19168, 2023.

Ye, F. and Bors, A. G. Dynamic self-supervised teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5731–5748, 2022.

Yoon, J., Jeong, W., Lee, G., Yang, E., and Hwang, S. J. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pp. 12073–12086. PMLR, 2021.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. PMLR, 2017.

Zhang, J., Chen, C., Zhuang, W., and Lv, L. Addressing catastrophic forgetting in federated class-continual learning. *arXiv preprint arXiv:2303.06937*, 2023.

A. Transferability of Prompting Method

Prompt-tuning In this paper, we focus on prompt-based knowledge transfer among tasks in FCL. Therefore, we empirically explore the transferability of the knowledge learned by different prompt methods. It is noteworthy that the transferability here refers to the transfer between downstream tasks, rather than the transfer from a pre-trained foundation model to downstream tasks.. In the CL, FL, and FCL communities, the two mainstream prompt methods are prompt-tuning and prefix-tuning. Both methods operate on the multi-head self-attention layers (MHA) (Vaswani et al., 2017):

$$\begin{aligned} \mathrm{MHA}(\mathbf{h}_Q, \mathbf{h}_K, \mathbf{h}_V) &= \mathrm{Concat}(\mathbf{h}_1, \ldots, \mathbf{h}_m)\mathbf{W}^O \\ \text{where} \quad \mathbf{h}_i &= \mathrm{Attention}(\mathbf{h}_Q\mathbf{W}_i^Q, \mathbf{h}_K\mathbf{W}_i^K, \mathbf{h}_V\mathbf{W}_i^V), \end{aligned} \tag{9}$$

where m is the number of heads and $\mathbf{W}^O$, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ are projection matrices. $\mathbf{h}_Q$, $\mathbf{h}_K$, $\mathbf{h}_V$ are the same in ViT (Dosovitskiy et al., 2020). **Prompt-tuning** concatenates prompts to input tokens or input of MHA layers, which is equivalent to concatenate the same prompt parameter $\mathbf{p}$ to $\mathbf{h}_Q$, $\mathbf{h}_K$ and $\mathbf{h}_V$, namely $\mathrm{MHA}([\mathbf{p}; \mathbf{h}_Q], [\mathbf{p}; \mathbf{h}_K], [\mathbf{p}; \mathbf{h}_V])$. **Prefix-tuning** concatenates two prompt parameters $\mathbf{p}_K$, $\mathbf{p}_V$ to $\mathbf{h}_K$ and $\mathbf{h}_V$ respectively, thus keep the input and output sequence lengths the same, namely $\mathrm{MHA}(\mathbf{h}_Q, [\mathbf{p}_K; \mathbf{h}_K], [\mathbf{p}_V; \mathbf{h}_V])$. We empirically choose prompt-tuning in 4,5,6 layers by evaluating the transferability of prompts in AppendixA

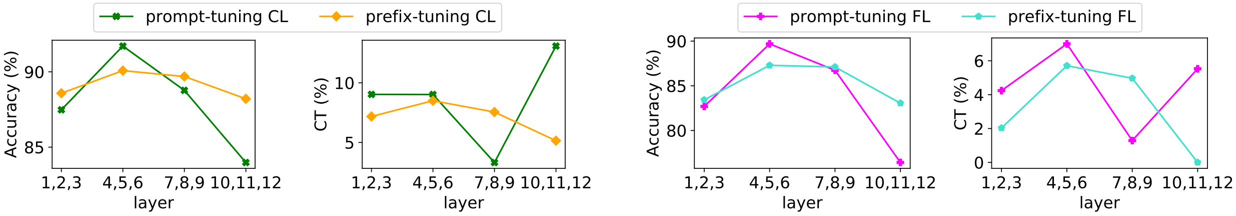

Figure 7. Transferability of different prompting method in CL and FL

DualPrompt (Wang et al., 2022a) empirically demonstrates that prefix-tuning has better performance in class-incremental learning and also explores which MHA layers to insert prompts in yields better results. However, in the class-incremental scenario, better performance may not necessarily come from task-to-task transfer but from:

- Prompts are able to learn task-specific knowledge more thoroughly.
- Prompts are able to distinguish between tasks to avoid mutual interference in the class-incremental scenario.

To our best knowledge, no method has yet been developed to study the transfer capability of the knowledge learned by prompts. We believe that in prompt-based FCL, prompt-based knowledge transfer between tasks is crucial:

- Allow previous tasks and current tasks on each client to fully leverage each other's knowledge to achieve better performance.
- Treating prompts as part of the pre-trained foundation model, we can help enhance the pre-trained foundation model continuously through a federated way, which has significant application value in fields such as CV and NLP that heavily rely on pre-trained foundation models.

We designed two sets of experiments from both the Federated and Continual dimensions. In the Federated experiments, two clients have the same task, each holding 20% of the task's data. In the continual side experiments, the two tasks are the same, with the first task occupying 80% of the task's data and the second task occupying 20%. Following DualPrompt, we evaluate prefix-tuning or prompt-tuning at different MHA layers of ViT. The experimental results, as shown in the figure, indicate that prompt-tuning at the middle layers (layers 4, 5, and 6 in ViT) performs better and learns more transferable knowledge, both in the FL side and the CL side.

B. Implementation Details

We implement our method and all baselines in PyTorch (Paszke et al., 2019) using the ViT-B/16 (Dosovitskiy et al., 2020) backbone pre-trained on ImageNet-21K (Deng et al., 2009). All results are averaged over three runs and are obtained on 46GB NVIDIA RTX A6000 GPU. Each task lasts for at least 3 rounds. Local epochs for each round are set to 10 for ImageNet-R and 4 for DomainNet for local convergence. To achieve asynchrony, after every 3 rounds, we randomly select 40% clients and switch their tasks. The learning rate is set to 0.005. The hyperparameters λ and p are set as 1 and 30 respectively. We select three hyperparameters above by cross-validation on small validation sets. For the size of prompt pool, we set it for Fed-L2P and Fed-Dual following (Wang et al., 2022b;a) with the best performance; $M = 10$, $L = 8$ and $D = 768$ for Fed-CODAP, Fed-CPrompt and Powder for a fair comparison. For the prompts initialization, the proposed Powder, Fed-CODAP and FedCPROMPT, following CODAPrompt (Smith et al., 2023), employ orthogonal initialization where the initialized prompts are orthogonal to each other. This is different from Fed-L2P and Fed-Dual, where the prompts are uniformly initialized following (Wang et al., 2022b;a)

C. Application Scenarios

Consider several hospitals, each equipped with a disease diagnosis model that continuously learns to diagonse new groups of diseases (i.e., new tasks). In a specific task at a particular hospital, the scarcity of disease samples often makes it challenging to effectively learn its local model (e.g., fine-tuning the hospital's pre-trained model with limited examples). In such situations, the proposed Powder significantly enhances the effectiveness of each hospital's disease diagnosis tasks by enabling dual knowledge transfer between tasks without the need to transmit private data between different clients.

D. Additional Experimental Results

Here, we additionally provide experimental results for combining FedWEIT, CFeD, GLFC, Fedspace with basic prompt-tuning (Basic PT) and with CODAPrompt (CODAP), as shown in Table 6. It can be seen that basic prompt-tuning and CODAPrompt have a certain enhancing effect on these four methods, and even a few methods achieve a combined transfer slightly greater than zero (e.g., GLFC-VIT+Basic PT). This phenomenon implies that parameter-efficient transfer learning methods such as prompt-tuning can complement classic federated learning, continual learning and federated continual learning methods that learn from scratch to achieve better performance. Our proposed Powder still maintains its leading performance.

Table 6. Performance measured by metrics in Sec.5.1, on ImageNet-R and DomainNet with with $C = 5$ clients and $r = 15$ rounds.

ImageNet-R						
	Method	AIA(%)	FM(%)	FT(%)	BT(%)	CT(%)
No-prompt	FedWEIT-ViT	$29.26_{(0.82)}$	$18.12_{(0.63)}$	$6.74_{(1.27)}$	$-24.51_{(2.23)}$	$-12.87_{(1.00)}$
	CFeD-ViT	$59.79_{(1.17)}$	$3.81_{(0.39)}$	$-17.67_{(8.19)}$	$-14.92_{(2.65)}$	$-29.60_{(6.28)}$
	GLFC-ViT	$75.21_{(1.05)}$	$1.10_{(0.16)}$	$-3.87_{(0.70)}$	$-1.55_{(0.76)}$	$-5.11_{(0.99)}$
	FedSpace-ViT	$73.36_{(0.82)}$	$2.01_{(0.13)}$	$-2.60_{(0.59)}$	$-4.19_{(0.51)}$	$-5.95_{(0.48)}$
Basic PT	FedWEIT-ViT	$73.91_{(0.62)}$	$0.80_{(0.46)}$	$-1.48_{(0.68)}$	$-0.98_{(0.55)}$	$-2.27_{(1.07)}$
	CFeD-ViT	$57.60_{(2.58)}$	$7.97_{(2.21)}$	$-4.44_{(1.89)}$	$-26.48_{(0.93)}$	$-25.63_{(2.40)}$
	GLFC-ViT	$76.86_{(0.78)}$	$0.91_{(0.15)}$	$0.48_{(0.89)}$	$-0.59_{(0.20)}$	$0.01_{(1.00)}$
	FedSpace-ViT	$75.17_{(0.79)}$	$1.13_{(0.38)}$	$-3.49_{(0.92)}$	$-1.43_{(0.79)}$	$-4.64_{(1.53)}$
CODAP	FedWEIT-ViT	$74.13_{(0.43)}$	$0.59_{(0.02)}$	$-1.63_{(0.63)}$	$-0.60_{(0.36)}$	$-2.11_{(0.63)}$
	CFeD-ViT	$55.71_{(2.84)}$	$8.98_{(2.52)}$	$-4.92_{(1.88)}$	$-28.32_{(1.03)}$	$-27.57_{(2.46)}$
	GLFC-ViT	$75.52_{(0.85)}$	$0.83_{(0.08)}$	$0.51_{(0.82)}$	$-0.59_{(0.19)}$	$0.04_{(0.95)}$
	FedSpace-ViT	$73.92_{(0.96)}$	$1.44_{(0.43)}$	$-3.19_{(0.31)}$	$-1.92_{(0.82)}$	$-4.73_{(0.39)}$
Prompt	Fed-L2P	$75.03_{(0.88)}$	$0.41_{(0.06)}$	$-2.79_{(0.13)}$	$-0.17_{(0.26)}$	$-2.92_{(0.23)}$
	Fed-Dual	$74.91_{(0.87)}$	$0.49_{(0.08)}$	$-3.12_{(0.17)}$	$0.22_{(0.32)}$	$-2.95_{(0.43)}$
	Fed-CODAP	$75.14_{(1.06)}$	$\mathbf{-0.68}_{(0.55)}$	$-2.53_{(2.87)}$	$1.69_{(1.63)}$	$-1.18_{(1.62)}$
	FedCPrompt	$72.59_{(0.44)}$	$0.63_{(0.06)}$	$-3.16_{(0.94)}$	$0.00_{(0.00)}$	$-3.16_{(0.94)}$
	Powder	$\mathbf{84.08}_{(0.56)}$	$-0.54_{(0.07)}$	$\mathbf{4.48}_{(0.13)}$	$\mathbf{1.95}_{(0.60)}$	$\mathbf{6.04}_{(0.50)}$

DomainNet						
	Method	AIA(%)	FM(%)	FT(%)	BT(%)	CT(%)
No-prompt	FedWEIT-ViT	$28.58_{(0.82)}$	$17.12_{(1.10)}$	$7.38_{(0.74)}$	$-22.13_{(5.14)}$	$-10.32_{(3.37)}$
	CFeD-ViT	$60.19_{(0.23)}$	$1.65_{(0.55)}$	$-4.98_{(0.25)}$	$-13.32_{(0.51)}$	$-15.64_{(0.46)}$
	GLFC-ViT	$70.34_{(0.00)}$	$1.23_{(0.02)}$	$-4.08_{(0.42)}$	$-2.46_{(0.10)}$	$-6.04_{(0.50)}$
	FedSpace-ViT	$70.71_{(0.19)}$	$1.80_{(0.12)}$	$1.87_{(0.23)}$	$-4.16_{(0.22)}$	$-1.45_{(0.06)}$
Basic PT	FedWEIT-ViT	$71.78_{(0.36)}$	$0.42_{(0.22)}$	$0.64_{(0.64)}$	$-0.97_{(0.34)}$	$-0.13_{(0.91)}$
	CFeD-ViT	$57.14_{(3.68)}$	$5.11_{(3.06)}$	$-9.27_{(3.18)}$	$-19.35_{(8.01)}$	$-24.75_{(8.09)}$
	GLFC-ViT	$73.44_{(0.44)}$	$0.92_{(0.15)}$	$-2.60_{(0.58)}$	$-1.49_{(0.35)}$	$-3.79_{(0.83)}$
	FedSpace-ViT	$72.53_{(0.35)}$	$0.94_{(0.07)}$	$-1.75_{(0.35)}$	$-1.73_{(0.05)}$	$-3.13_{(0.38)}$
CODAP	FedWEIT-ViT	$71.77_{(0.47)}$	$0.37_{(0.17)}$	$0.98_{(0.40)}$	$-0.72_{(0.20)}$	$0.40_{(0.56)}$
	CFeD-ViT	$55.28_{(3.85)}$	$5.72_{(3.12)}$	$-9.59_{(2.99)}$	$-20.49_{(7.62)}$	$-25.98_{(7.59)}$
	GLFC-ViT	$72.35_{(0.55)}$	$0.66_{(0.11)}$	$-3.15_{(0.58)}$	$-0.99_{(0.20)}$	$-3.94_{(0.64)}$
	FedSpace-ViT	$71.70_{(0.32)}$	$1.05_{(0.20)}$	$-1.89_{(0.88)}$	$-2.05_{(0.49)}$	$-3.53_{(1.14)}$
Prompt	Fed-L2P	$72.36_{(0.44)}$	$0.16_{(0.03)}$	$-2.18_{(0.21)}$	$0.10_{(0.04)}$	$-2.09_{(0.24)}$
	Fed-Dual	$72.15_{(0.22)}$	$0.16_{(0.02)}$	$-1.82_{(0.11)}$	$0.41_{(0.03)}$	$-1.49_{(0.08)}$
	Fed-CODAP	$72.84_{(0.40)}$	$\mathbf{0.01}_{(0.04)}$	$-0.82_{(0.37)}$	$\mathbf{0.83}_{(0.28)}$	$-0.15_{(0.29)}$
	FedCPrompt	$69.92_{(0.56)}$	$0.19_{(0.09)}$	$-2.78_{(0.36)}$	$0.00_{(0.00)}$	$-2.78_{(0.36)}$
	Powder	$\mathbf{77.28}_{(0.18)}$	$0.10_{(0.06)}$	$\mathbf{1.28}_{(0.04)}$	$0.14_{(0.20)}$	$\mathbf{1.40}_{(0.19)}$