Federated Continual Learning via Orchestrating Multi-Scale Expertise

Xiaoyang Yi[1,3,4] **Yang Liu**[2] **Binhan Yang**[2] **Jian Zhang**[1,2,3,4,*]
[1] College of Cryptology and Cyber Science, Nankai University, China
[2] College of Computer Science, Nankai University, China
[3] Tianjin Key Laboratory of Network and Data Security Technology, Tianjin, China
[4] Key Laboratory of Data and Intelligent System Security, Ministry of Education, Tianjin, China
{xiaoyangyi, liuyang99. yangbinhan}@mail.nankai.edu.cn
zhang.jian@nankai.edu.cn

Abstract

Federated continual learning (FCL) aims to maintain the model's performance on old tasks (i.e., stability) while enhancing its ability to acquire knowledge from current tasks (i.e., plasticity). With the development of pre-trained models (PTMs), fine-tuning PTMs on clients has become a promising approach to leveraging their extensive knowledge in FCL. In this paper, we propose MultiFCL, a novel FCL framework that fine-tunes PTMs to adapt to FCL while preserving their strong generalization capabilities. Specifically, to ensure the stability, MultiFCL introduces lightweight adapters for task adaption, which are subsequently frozen to prevent catastrophic forgetting. Moreover, by utilizing the semantic features of old tasks, MultiFCL performs multi-modal initialization of new task class prototypes. To enhance the plasticity, MultiFCL employs a multi-expert training mechanism that integrates multi-scale feature learning with multi-teacher dynamic self-distillation. Through intra-client and inter-client expert communication, MultiFCL facilitates cross-task and cross-client knowledge fusion. Experimental results demonstrate that MultiFCL achieves state-of-the-art performance across multiple datasets and settings, showcasing its effectiveness in FCL scenarios.

1 Introduction

Federated learning (FL) is a distributed machine learning paradigm that allows clients to collaboratively train models without sharing sensitive data (McMahan et al., 2017). In recent years, FL has gained significant attention across various domains, such as healthcare (Yang et al., 2021; Ogier du Terrail et al., 2022; Chen et al., 2024b; Xie et al., 2024) and the internet of things (IoT) (Nguyen et al., 2023; Cui et al., 2022; Yu et al., 2024a; Jia et al., 2024). Traditional FL approaches typically assume that client data distributions do not change over time. However, this assumption is unrealistic in practical applications, as client data usually evolves continuously (Ma et al., 2022; Lu et al., 2024). For example, healthcare models must adapt to emerging diseases and their variants (Babakniya et al., 2023), while IoT models need to keep pace with changes in user behaviors (Zhang et al., 2024). When there are such shifts in data distributions, directly retraining existing models on new data can lead to catastrophic forgetting (Smith et al., 2023b; Li et al., 2024; Yu et al., 2024b), which significantly impairs the model's performance on previously learned tasks.

To address this, federated continual learning (FCL) integrates continual learning (CL) techniques into FL (Yoon et al., 2021), aiming to seek a balance between stability (i.e., maintaining performance on

[*] Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

old tasks) and plasticity (i.e., acquiring knowledge from new tasks). Unlike traditional CL methods that rely on extensive data retention to perform experience replay, many FCL approaches instead opt for generative replay (Qi et al., 2023; Wuerkaixi et al., 2024; Tran et al., 2024) or model decomposition (Bakman et al., 2024), due to privacy concerns and resource limitations. These approaches train models from scratch, trying to continuously absorb knowledge from new tasks while reviewing knowledge from previous ones (Zhou et al., 2024b). However, the heterogeneity among client data presents challenges in maintaining both stability and plasticity during the process of training a model from scratch. Fortunately, the emergence of pre-trained models (PTMs) has offered a promising alternative, owing to their remarkable generalization capabilities (Dosovitskiy et al., 2021; Radford et al., 2021). Trained on large-scale datasets, PTMs are better equipped to acquire optimal knowledge compared to incrementally trained model, thereby significantly alleviating the learning burden (Zhou et al., 2024c).

Although PTMs exhibit strong generalization capabilities, how to fine-tune them for downstream tasks is still a dilemma: fully fine-tuning risks diminishing their generalizable features, while freezing the backbone can prevent the integration of downstream information (Zhou et al., 2024a). To this end, some FCL approaches incorporate trainable prompt parameters into PTMs, training prompts locally on clients and updating the global prompt pool on the server by aggregating local prompt pools (Halbe et al., 2024; Yu et al., 2024c). A query mechanism is then employed to dynamically select a subset of prompts from the global prompt pool, guiding PTMs in adapting to different data distributions (Wang et al., 2022; Smith et al., 2023a). By keeping the pre-trained weights frozen during this process, these prompt-based approaches effectively preserve the generalizability of PTMs while allowing task-specific adaptions. However, prompt selection often focuses on specific subsets, limiting the adaptability to broader data distributions (Moon et al., 2023). Moreover, as new data accumulates, the prompt pool size expands, leading to increased communication overhead, whereas restricting the prompt pool size constrains its representational capacity (Piao et al., 2024).

Inspired by these insights, we propose a novel FCL framework, MultiFCL, which effectively fine-tunes PTMs while preserving their strong generalization capabilities to adapt to FCL. Specifically, to maintain the stability of PTMs, MultiFCL employs lightweight adapters to quickly adapt PTMs to downstream image classification tasks, and then freezes them to prevent catastrophic forgetting. Furthermore, MultiFCL leverages semantic features from old tasks to perform multi-modal initialization of new class prototypes, thereby incorporating prior knowledge. To enhance the plasticity of PTMs, MultiFCL introduces a multi-expert training mechanism, which combines multi-scale feature learning and multi-teacher dynamic self-distillation to strengthen feature representation for current tasks and achieve effective knowledge integration. Through both intra-client and inter-client expert communication, MultiFCL facilitates cross-task and cross-client knowledge fusion, addressing the dynamic and heterogeneous data distribution challenges inherent in FCL.

Our contributions are as follows:

- We propose MultiFCL, a novel FCL framework that preserves the strong generalization capabilities of PTMs while fine-tuning them, enabling PTMs to maintain both stability and plasticity.
- To ensure stability, we employ lightweight adapters for task-specific fine-tuning and utilize multi-modal initialization of new task class prototypes based on semantic features from old tasks.
- To enhance plasticity, we design a multi-expert training method that combines multi-scale feature learning and multi-teacher dynamic self-distillation, achieving effective knowledge fusion.
- Experimental results demonstrate that MultiFCL achieves state-of-the-art performance across multiple datasets and configurations, highlighting its effectiveness and the robustness of PTMs.

2 Related Work

FCL can be broadly categorized into task-incremental learning (TIL) and class-incremental learning (CIL). The former focuses on learning new tasks sequentially, while the latter involves learning classes in each task incrementally. Specifically, Federated TIL was first introduced in FedWeIT (Yoon et al., 2021). To prevent forgetting old tasks while learning new ones, FedWeIT decomposes network weights into global parameters and sparse task-specific parameters, each client selectively transfers knowledge from others by their task-specific parameters. CFeD (Ma et al., 2022) employs continual distillation using an unlabeled proxy dataset on both clients and the server, assigning different learning

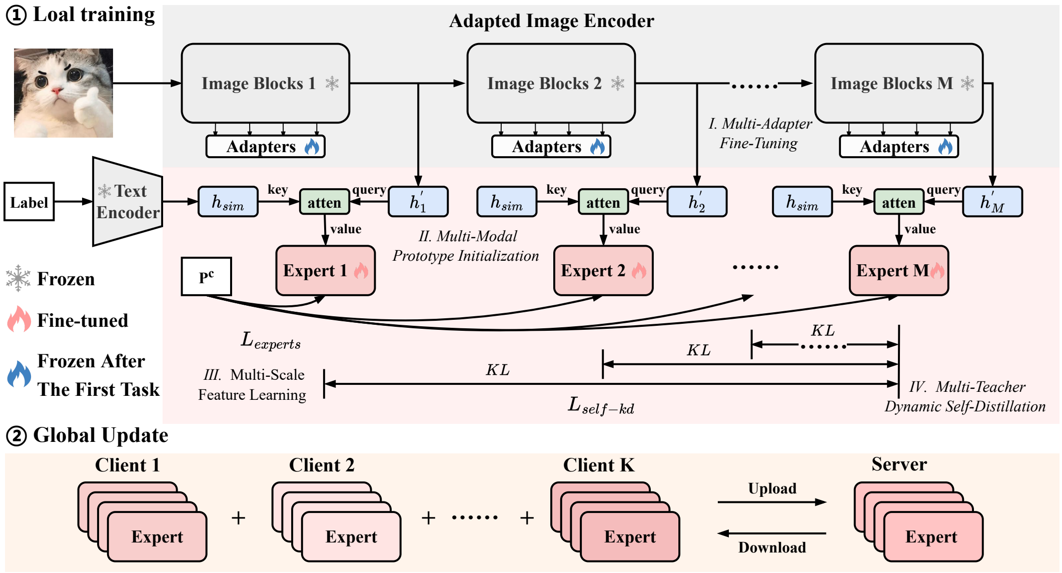


Figure 1: The overall framework of MultiFCL. MultiFCL employs adapters to fine-tune the PTM and leverages the semantic features of old tasks to initialize new class prototypes. Then, MultiFCL establishes multiple experts, employing feature learning loss $\mathcal{L}_{\text{experts}}$ and the multi-teacher dynamic self-distillation $\mathcal{L}_{\text{self-kd}}$ to transfer knowledge to the final expert.

objectives to clients for learning new tasks and revisiting old ones. AF-FCL (Wuerkaixi et al., 2024) selectively leverages prior knowledge in the federated network and uses a probabilistic framework based on normalized flow models to quantify the reliability of this prior knowledge. FOT (Bakman et al., 2024) extracts global input subspaces for old tasks at each layer and modifies the aggregated updates for new tasks to ensure orthogonality with these global subspaces. Powder (Piao et al., 2024) introduces a novel prompt generation and aggregation method to effectively facilitate knowledge transfer encapsulated in prompts across sequential tasks and clients. FedMGP (Yu et al., 2024c) employs coarse-grained global prompts for efficient shared knowledge transfer and fine-grained local prompts for personalized learning, mitigating spatial and temporal forgetting.

Class-incremental FCL, also known as federated class-incremental learning (FCIL), was first introduced by GLFC (Dong et al., 2022). To mitigate forgetting of old classes while learning new ones, GLFC proposes a class-aware gradient compensation loss and a class-semantic relation distillation loss to balance old-class retention. FedCIL (Qi et al., 2023) employs ACGAN (Odena et al., 2017) to synthesize data from old distributions for replay, alleviating catastrophic forgetting. Similarly, MFCL (Babakniya et al., 2023) uses a generative model to synthesize samples from old data distributions and trains on the server using a data-free method. FedET (Liu et al., 2023) introduces a small Enhancer module to absorb and communicate new knowledge and proposes an enhancer distillation method to address the imbalance between old and new knowledge. LANDER (Tran et al., 2024) uses label text embeddings as anchors to constrain the feature embeddings of corresponding training samples, generating meaningful samples for replay. FedCBC (Yu et al., 2024b) constructs a class-specific binary classifier for each class, extracting prior knowledge from the global model into multiple local models for selective knowledge fusion. LGA (Dong et al., 2024) introduces a class-balanced gradient adaptive compensation loss and a class-gradient-induced semantic distillation loss to balance the heterogeneous forgetting rates of old classes.

3 Method

The MultiFCL framework is shown in Figure 1. Specifically, to adapt the powerful PTM to downstream image classification tasks, MultiFCL introduces adapters into the PTM for task-specific fine-tuning (Section 3.1). These adapters remain frozen during the training process of subsequent tasks, effectively mitigating catastrophic forgetting. By integrating semantic features from old tasks with those from new tasks, MultiFCL performs multi-modal initialization of new task prototypes (Section 3.2), which enhances the representation of new class features while preserving old knowledge. During the training process, multi-scale features and new task prototypes are used for multi-scale feature learning to ensure the plasticity (Section 3.3). Additionally, MultiFCL employs multi-teacher self-distillation to transfer knowledge to the final expert, and dynamically adjusts the distillation loss using uncertainty-weighted perception in the knowledge transfer process (Section 3.4).

3.1 Multi-Adapter Fine-Tuning

In FCL, each client k ($1 \leq k \leq K$) trains a local model using its local dataset $D_k^{\mathcal{T}} = \{(x_i, y_i)\}_{i=1}^{n_k^{\mathcal{T}}}$ to sequentially deal with tasks $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_{\mathcal{T}}\}$, where $n_k^{\mathcal{T}}$ is the dataset size for task $\mathcal{T}$ on client k. For the same task $\mathcal{T}$, data distributions across clients are non-independent and identically distributed (Non-IID). Meanwhile, within the same client k, the class sets across different tasks are disjoint. The overall optimization objective for clients is two fold: (1) minimizing the loss function $L(\cdot)$ on the current task's dataset to achieve high performance (i.e., plasticity), (2) and retaining knowledge from previously learned tasks' datasets to mitigate catastrophic forgetting (i.e., stability).

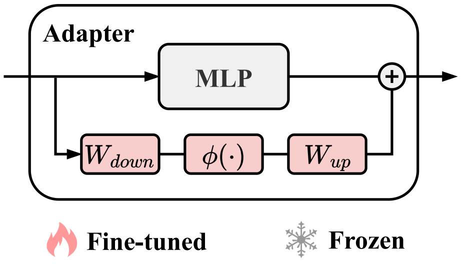


Figure 2: The structure of a adapter in the PTM.

To bridge the domain knowledge gap between the PTM and downstream image classification tasks, we introduce trainable adapter parameters w_k into the frozen PTM, enabling adaptation to FCL tasks for each client k. Adapter introduces small neural network modules independently into various layers of the model, separate from its main structure. Only these adapter parameters are updated during fine-tuning. In contrast, LoRA (Hu et al., 2022) introduces low-rank matrices into the model's weight matrices. These low-rank matrices act as modifications applied to the original weight matrices, adjusting them during computation. Specifically, we employ a transformer-based PTM (e.g., CLIP (Radford et al., 2021)) as the backbone model, which consists of multiple repeated blocks with several layers. Adapters are inserted into each layer of the image encoder to facilitate fine-tuning. As illustrated in Figure 2, each adapter is a bottleneck module, which is composed of a down-sampling layer W_{down} for reducing feature dimensions, a nonlinear activation function $\phi(\cdot)$ (e.g., ReLU (Glorot et al., 2011)), and an up-sampling layer W_{up} for restoring the original dimensions (Chen et al., 2024a). This process can be formulated as:

$$h^{'} = h + \phi(hW_{down})W_{up} \tag{1}$$

where h and $h^{'}$ are the input and output feature representations through an adapter, respectively.

Since the local dataset of client k constantly evolves with changing tasks, training adapters for each task can easily lead to catastrophic forgetting. To fully preserve the generalization ability of the PTM, we fine-tune adapters only on the first task and then freeze them to prevent excessive updates, as follows:

$$f'(x) = L(f(x), D_k^1, w_k) \tag{2}$$

Here, $f(\cdot)$ represents the original PTM, $f'(\cdot)$ is the PTM with adapters, and D_k^1 is the dataset for the first task of client k. By fine-tuning with adapters on the first task, the PTM can rapidly adapt to downstream image classification tasks.

During each communication round t of the first task training, each client uploads its locally trained adapters to the server for aggregation, enabling the model to acquire more comprehensive knowledge as follows:

$$w^t = \sum_{k=1}^{K} \frac{|X_k^1|}{\sum_{k=1}^{K} |X_k^1|} w_k^t \tag{3}$$

where w^t denotes the global adapters, and $|X_k^1|$ is the total number of samples owned by client k for the first task. Once the first task is completed, all adapters will be frozen and no longer updated during model training, preventing performance from degrading due to excessive fine-tuning.

3.2 Multi-Modal Prototype Initialization

Adapter fine-tuning enables the PTM to quickly adapt to downstream image classification tasks, while for classification tasks, a classifier is also needed to ensure its generalizability. Since the adapter is only trained in the first task, it focuses more on the knowledge related to that task and may overlook some irrelevant high-level features (Zhou et al., 2024b). Therefore, instead of using a conventional linear classifier for classification, we use prototypes as weights of the classifier, performing classification by measuring the distance between class prototypes and sample features. Specifically, we continuously update the optimal class prototype $\mathbf{p}^c$ ($c \in \mathcal{T}$) through training, as follows:

$$\mathbf{p}^c_{(0)} = \mathbf{p}^c_{\text{init}} \tag{4}$$

$$\mathbf{p}^c_{(j)} = \frac{\mathbf{p}^c_{(j-1)} + f'(\mathbf{x}^c_j)}{2}, \quad \forall j \in \{1, 2, \ldots, n^{\mathcal{T}}_c\} \tag{5}$$

where $\mathbf{p}^c_{\text{init}}$ is the initialization prototype of class c, $\mathbf{p}^c_{(j)}$ is the current prototype from class c to sample $\mathbf{x}^c_j$, and $n^{\mathcal{T}}_c$ is the number of class c in the task $\mathcal{T}$.

A common initialization method is random initialization, which makes the training of the classifier independent learning around the task. For subsequent tasks, it may lead to the classifier of the new task lacking knowledge from old tasks, resulting in lower discriminability (Kurniawan et al., 2024). To address this, the bias-adjustment heuristic (Kahneman et al., 1982) is proposed, which uses available related knowledge to estimate unknowns. Therefore, we introduce relevant semantic features from old tasks as available knowledge to estimate the new class prototypes, and use multi-modal fusion to form their initialized prototypes.

In order to fuse semantic features with sample features of the new class for initializing the new class prototype, we first calculate the similarity between the feature of the first sample of the new class and the semantic features $C^{\mathcal{T}_{\text{old}}}$ of the old task labels to establish the basic relationship assumption between the old and new classes, as follows:

$$\text{sim}\left(f'(x^c_1), f_{text}(C^{\mathcal{T}_{\text{old}}})\right) = \frac{f'(x^c_1) \cdot f_{\text{text}}(C^{\mathcal{T}_{\text{old}}})}{\|f'(x^c_1)\| \|f_{text}(C^{\mathcal{T}_{\text{old}}})\|} \tag{6}$$

where $\text{sim}(\cdot)$ is a similarity calculation function, and $f_{text}(\cdot)$ is the text encoder of the PTM.

Subsequently, the most similar label semantic features are selected and merged with the sample features of the new class through a cross-attention mechanism to initialize the new class prototype, thereby incorporating prior implicit bias as follows:

$$\mathbf{p}^c_{\text{init}} = \text{Attention}\left(f'(x^c_1), f_{text}(c^{\mathcal{T}_{\text{old}}}_{\text{sim}}), \mathbf{p}^c_{\text{old}}\right) \tag{7}$$

where $c^{\mathcal{T}_{\text{old}}}_{\text{sim}}$ and $\mathbf{p}^c_{\text{old}}$ is the semantic feature and the class prototype of the old task that is most similar to the current class, respectively. $\text{Attention}(\cdot)$ is the cross-attention. The initialized prototypes obtained in this way can enhance the feature representation of the new class while preventing the forgetting of old class knowledge, thereby ensuring the stability of the model.

3.3 Multi-Scale Feature Learning

While ensuring the stability of the model, it is also crucial to maintain its plasticity, which refers to the model's ability to acquire knowledge from the current task. Research (Jung et al., 2023) has shown that multi-scale features can help the model distinguish between different samples by providing multi-level information, ranging from shallow to deep knowledge. Therefore, we divide the PTM into *M* modules, each consisting of multiple blocks, which can be treated as experts with different levels of knowledge.

Specifically, the feature output from the last layer of each module serves as the input for the next module. Additionally, the features output by these modules will be used to initialize and update the new task class prototypes, as described in Section 3.2. By utilizing these prototypes, the output features at different scales from each module are continuously trained to move closer to their corresponding class centers, allowing the experts to effectively consolidate the class knowledge of each new task. Subsequently, the experts optimize the acquired knowledge through mutual communication, with the optimization objective as:

$$\mathcal{L}_{\text{experts}} = \sum_{m=1}^{M} \left(- \sum_{x_i \in \mathcal{T}} y_i \log p_{x_i,m} \right) \tag{8}$$

where M is the number of experts, and $p_{x_i,m}$ represents the prediction probability that sample x_i belongs to label y_i in expert m. Through multi-scale feature learning, the multi-level experts in clients are able to make decisions from different perspectives on the same sample, thus capturing the diversity and complexity of the sample more comprehensively. As the data distribution continues to evolve, clients become more flexible in adapting to new tasks.

In addition to the collaboration of experts within each client to jointly classify samples, coordination and communication between clients are also necessary. To achieve this, during each communication round t, each client uploads the class prototypes obtained after local training to the server for aggregation as follows:

$$\mathbf{p}^{c,t} = \sum_{k=1}^{K} \frac{\left|X_k^{c,t}\right|}{\sum_{k=1}^{K} \left|X_k^{c,t}\right|} \sum_{m}^{M} \mathbf{p}_{k,m}^{c,t} \tag{9}$$

where $\mathbf{p}^{c,t}$ represents the global class prototype, $|X_k^{c,t}|$ denotes the total samples of class c in client k, and $\mathbf{p}_{k,m}^{c,t}$ is the class prototype in expert m of client k. As a result, the server collects the multi-scale prototypes from different clients to form global class prototypes, which are then sent back to the clients as classifier weights for each module in the next round of training. By continuously sharing and updating class prototypes, clients are able to better learn representative multi-scale knowledge during the federated learning process. This provides more precise task guidance for subsequent feature learning.

3.4 Multi-Teacher Dynamic Self-Distillation

Multi-scale feature learning enables better predictions by allowing different experts to make judgments on the samples from different levels of knowledge (Aljundi et al., 2017). However, using multiple experts to generate predictions introduces additional computational overhead, which becomes particularly redundant during testing (Yan et al., 2024). Therefore, we propose a novel self-distillation method that transfers knowledge from multi-level experts to the final high-level expert. Among them, the former experts act as teachers, while the latter serves as the learner to integrate the knowledge from all the experts, as follows:

$$\mathcal{L}_{\text{self-kd}} = \sum_{m=1}^{M-1} \mathrm{KL}\left(\sigma(x_{i,m}) \parallel \sigma(x_{i,\text{final}})\right) \tag{10}$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler divergence function, and $\sigma(\cdot)$ represents the softmax function. Moreover, $x_{i,m}$ and $x_{i,\text{final}}$ denote the output of expert m and the final expert, respectively.

It calculates the distance between the output of each expert and the output of the final expert, thereby consolidating the knowledge from multi-level experts into the final expert and reducing the number of experts that need to be computed during testing. During training, the distillation loss encourages the final expert to learn from the knowledge of all the experts, including the fine-grained features from lower-level experts and the abstract decisions from higher-level experts. In this way, the final expert is able to independently perform the task while avoiding excessive reliance on the features from the final layer, thereby obtaining more comprehensive decision-making capabilities.

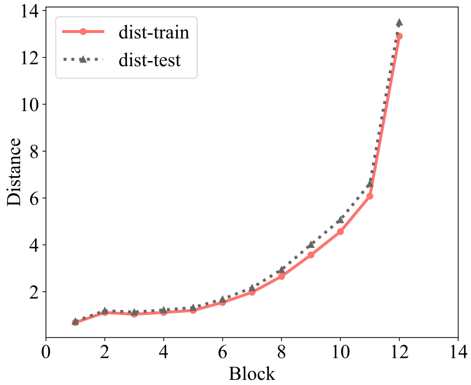


Figure 3: The distance between feature outputs at each block.

During self-distillation, the output quality of different experts may vary. Figure 3 shows the distance between the feature outputs at each block of the PTM, with a significant difference between deep and shallow features. Simply adding the distance between each expert's output and that of the final expert does not fully reflect each expert's contribution to the task, as it tends to emphasize shallow knowledge. Therefore, we propose an uncertainty-weighting method to dynamically adjust the importance of each expert in the self-distillation process.

Specifically, the output quality of each expert is measured not only by the distance between its output and the final expert's output but also by considering the uncertainty of its output features. To quantify uncertainty, we use the entropy of the output from each expert. A higher entropy indicates that the expert's output is more ambiguous or uncertain, and its contribution should be correspondingly reduced. Conversely, an expert with lower entropy represents more certain outputs, which can provide more stable knowledge, and thus should be assigned a higher weight, as follows:

$$\lambda_{\text{expert}} = \frac{1}{\mathrm{KL}\left(\sigma(x_{i,\text{expert}}) \parallel \sigma(x_{i,\text{final}})\right) + \epsilon} \tag{11}$$

where ϵ is a small positive constant used to prevent the denominator from becoming zero and to avoid the weights from growing infinitely large.

To ensure fairness among the weights of different experts, we normalize these weights as follows:

$$\lambda'_{\text{expert}} = \frac{\lambda_{\text{expert}}}{\sum_{m=1}^{M} \lambda_m} \tag{12}$$

By suppressing the weights of shallow experts with high uncertainty and enhancing the weights of deep experts with low uncertainty, the knowledge from experts at various layers can be more effectively integrated. It not only prevents excessive interference from shallow knowledge during the self-distillation process, but also ensures that deep knowledge is fully utilized during learning, thereby enhancing the plasticity of the model.

Ultimately, the overall loss with multi-scale feature learning and multi-teacher dynamic self-distillation is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{experts}} + \lambda'_{\text{expert}} \cdot \mathcal{L}_{\text{self-kd}} \tag{13}$$

Through cross-task and cross-client multi-modal knowledge fusion, MultiFCL effectively handles changes in data distribution in the FCL setting, preventing catastrophic forgetting while maintaining the powerful capabilities of the PTM.

4 Experiments

4.1 Experiment Setting

We evaluate CIL and TIL experiments using CLIP (Radford et al., 2021) as our base model, with the image encoder being ViT (Dosovitskiy et al., 2021). In addition, we use CIFAR100 (Krizhevsky & Hinton, 2009), TinyImageNet (mnmoustafa & Ali, 2017), ImageNet-R (Hendrycks et al., 2021), and CUB-200-2011 (Wah et al., 2011), which are not learned by CLIP during pre-training. We divide all datasets into 5 and 10 tasks for incremental learning, using accuracy in the last task (Last), average accuracy in all tasks (Avg.), and the forgetting rate $\mathcal{F}$ as the metric for evaluation, where $\mathcal{F}$ is defined as the difference in accuracy between the first task and the final task. We present partial results for CIL on CIFAR100, with the remaining experiments in Appendix.A. Following the setting of existing FCL methods, we employ two different data partition strategies, including distribution-based partitioning and quality-based partitioning, where the former fixes the number of samples per class and specifies the number of classes owned by the client through α, while the latter uses β to specify the degree of data heterogeneity based on the Dirichlet distribution. Each partition is further subdivided into three scenarios, with specific settings provided in Appendix.B.

To demonstrate the capabilities of PTMs, we compare ViT and CLIP with FL implementations of several FCL methods such as MFCL (Babakniya et al., 2023), LANDER (Tran et al., 2024) and FedCBC (Yu et al., 2024b). Furthermore, to validate the effectiveness of MultiFCL, we compare it with PTM-based FCL methods such as FedMGP (Yu et al., 2024c) and PiLoRA (Guo et al., 2024), which also use ViT. We also replace the base models of MFCL and PiLoRA with CLIP to maintain

Table 1: Main results of performance comparison on CIFAR100 with 10 tasks in distribution-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	65.0	68.2	22.9	63.3	66.7	23.7	60.3	64.4	24.2
CLIP	ICML 2021	65.0	68.2	25.6	63.3	66.7	27.8	60.3	64.4	28.5
MFCL	NeurIPS 2023	32.9	44.1	25.9	31.4	43.2	26.4	30.7	42.8	28.8
MFCL-△	NeurIPS 2023	52.8	61.3	31.9	52.1	60.4	32.8	51.9	59.1	33.2
FedCBC	MM 2024	17.6	22.9	34.7	17.4	23.1	35.3	15.9	21.0	36.6
LANDER	CVPR 2024	34.8	48.5	27.6	33.4	45.2	28.5	32.1	43.5	30.5
FedMGP	KDD 2024	23.4	33.1	34.9	21.8	32.7	35.6	20.9	32.4	36.9
PiLoRA	ECCV 2024	69.6	72.1	26.7	63.7	65.4	27.8	63.0	68.0	29.4
PiLoRA-△	ECCV 2024	68.7	70.3	26.6	63.8	66.2	28.0	63.0	67.6	29.8
MultiFCL	This Paper	**70.7**	**76.1**	**16.4**	**68.9**	**74.6**	**18.5**	**67.9**	**72.8**	**17.5**

Table 2: Main results of performance comparison on CIFAR100 with 10 tasks in quality-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\alpha = 6$			$\alpha = 4$			$\alpha = 2$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	64.6	67.8	20.9	63.5	66.4	22.0	63.0	66.5	21.6
CLIP	ICML 2021	64.3	67.7	21.4	63.6	66.7	21.6	62.7	64.4	22.5
MFCL	NeurIPS 2023	30.4	39.8	33.6	24.6	33.1	34.5	15.9	21.2	35.9
MFCL-△	NeurIPS 2023	49.1	57.9	29.4	48.7	56.2	29.9	47.1	55.3	31.2
FedCBC	MM 2024	17.5	21.8	27.4	15.5	20.6	27.9	15.1	19.5	28.5
LANDER	CVPR 2024	20.3	26.7	28.9	19.8	24.8	28.6	18.7	24.4	29.3
FedMGP	KDD 2024	19.8	23.5	30.3	13.6	17.3	31.6	12.6	16.5	32.5
PiLoRA	ECCV 2024	70.5	74.8	22.8	63.8	68.2	24.4	63.6	69.5	25.7
PiLoRA-△	ECCV 2024	69.8	73.2	23.4	64.6	69.2	24.6	64.5	70.1	25.3
MultiFCL	This Paper	**71.0**	**76.1**	**15.9**	**70.3**	**75.3**	**15.9**	**69.7**	**75.2**	**17.1**

consistency with our base model for comparison. Specifically, we set up 10 clients, with each client owning 4 experts and performing 5 epochs of local training. For non-PTM methods, we use a learning rate of 1e-2 and conduct 100 communication rounds per task. For PTM methods, we use a learning rate of 1e-5 and perform 5 communication rounds per task.

4.2 Main Results

We compare MultiFCL with other baselines, the results are shown in Table 1 and Table 2. For FCL methods trained from scratch, MFCL and LANDER use generators deployed on the server to synthesize data and apply knowledge distillation to transfer knowledge to the global model, yet without guidance from real samples the generators struggle to produce high-quality data. FedCBC builds a binary classifier for each class and performs selective knowledge fusion between local and global models in an attempt to balance new and old knowledge, yet these classifiers focus only on single-class decision boundaries and cannot capture the complex semantic relationships among multiple classes. In addition, their accuracy is much lower than that of pre-trained ViT and CLIP, which demonstrates the powerful generalization ability of PTMs.

For PTM-based FCL methods, FedMGP has lower accuracy under quality-based data partitioning due to difficulties in finding task similarities. Its poor performance stems from the limitations of its prompt pool mechanism in FCL, especially when tasks differ greatly. Furthermore, its data split differs substantially from ours because the first task contains more classes giving the model an opportunity to overfit early categories, so it performs poorly in our experiments. PiLoRA learns independent low-rank LoRA parameters for each task and adds regularization to the loss to mitigate forgetting, but it lacks a multi-scale cross-task knowledge fusion and dynamic integration mechanism, its forgetting rate reaches 29.4% under distribution heterogeneity ($\beta = 0.05$) and its accuracy drops to only 63.6% under quality heterogeneity ($\alpha = 2$).

Meanwhile, we also replace the base model in MFCL and PiLoRA with CLIP to create MFCL-CLIP and PiLoRA-CLIP. CLIP's strong alignment of vision languages improves the accuracy of MFCL,it still relies on the generator to synthesize old samples of limited quality.And PiLoRA-CLIP still lacks cross-task knowledge integration. In contrast, direct fine-tuning of ViT and CLIP does not include any anti-forgetting design, whereas MultiFCL achieve sustained and robust performance gains across all settings and experimental results confirm its effectiveness and expressiveness.

4.3 Ablation Study

We perform the ablation study of MultiFCL on CIFAR100 with 10 Tasks, the results are shown in Table 3. The remaining experiments can be found in the appendix. Specifically, when no modules are added, the results correspond to the PTM itself, which establishes a clear baseline for comparison. When multiple adapters are added for fine-tuning on the first task, the average accuracy increases, demonstrating that task-specific adaptation allows the PTM to quickly adapt to downstream image classification tasks and exhibit its strong generalization ability.

When multi-modal features are added to initialize prototypes, MultiFCL supplements old class knowledge, thereby mitigating catastrophic forgetting in the current task. This addition yields a further gain in accuracy, confirming the benefit of richer prototype representations. Furthermore, when multi-experts are incorporated for training using $\mathcal{L}_{\text{experts}}$, each expert specializes in a different scale of features and together provide multi-scale knowledge, helping the model better distinguish differences between samples, resulting in higher accuracy. After incorporating multi-teacher self-distillation $\mathcal{L}_{\text{self-kd}}$, MultiFCL transfers knowledge from all experts to the final expert, reducing computational overhead during testing while consolidating expert knowledge, ultimately improving classification accuracy. The cumulative results of these experiments validate the design of MultiFCL and illustrate how each component contributes to stable and effective continual learning under federated and heterogeneous conditions.

Table 3: Ablation results of performance comparison on CIFAR100 with 10 tasks. The best results are shown in bold.

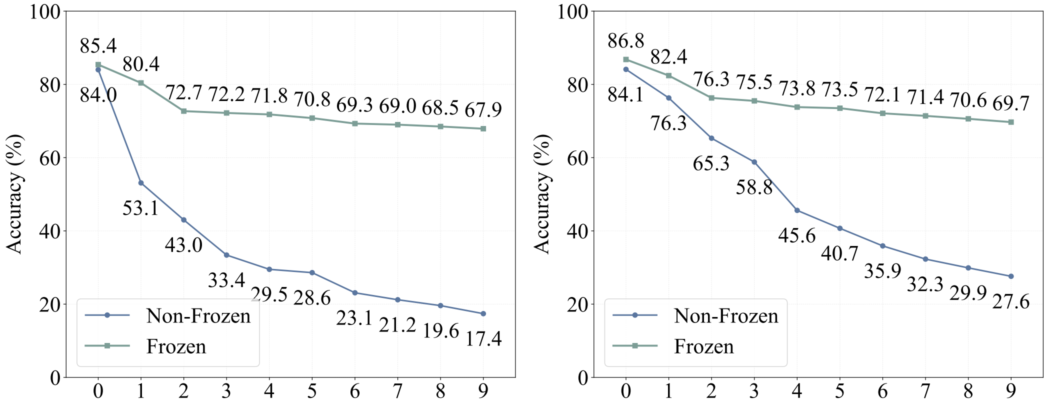
Adapter	Init.	$\mathcal{L}_{\text{experts}}$	$\mathcal{L}_{\text{self-kd}}$	$\alpha = 6$	$\alpha = 4$	$\alpha = 2$	$\beta = 0.5$	$\beta = 0.1$	$\beta = 0.05$
				64.3	63.5	63.0	65.0	63.3	60.3
✓				67.6	64.9	65.2	66.0	64.5	63.1
✓	✓			68.5	67.0	66.9	67.1	65.3	64.7
✓	✓	✓		70.1	68.7	68.6	68.9	67.8	66.3
✓	✓	✓	✓	**71.0**	**70.3**	**69.7**	**70.0**	**68.9**	**67.9**

4.4 Adapter Frozen Study

Figure 4 shows the experimental results for freezing adapters versus keeping them trainable on CIFAR-100 with $\beta = 0.05$ and $\alpha = 2$. When adapters continue to be trained in later tasks, the model's performance drops rapidly because prolonged training leads to catastrophic forgetting, which undermines generalization and compromises plasticity. Furthermore, due to client data heterogeneity, the adapters tend to drift during training, resulting in a severe loss of plasticity. Under these conditions of extreme heterogeneity and limited class samples, the model experiences pronounced forgetting as adapter updates quickly diverge from the task's optimal direction, causing the PTM's strong generalization ability to be lost. Therefore, freezing the adapters after the first task not only reduces computational cost and communication overhead but also effectively preserves the model's generalization capacity, facilitating the transfer of previous knowledge to new tasks.

4.5 Expert Study

We use a uniform hierarchical strategy to divide the 12 Transformer layers into 4 modules, each containing 3 layers that correspond to shallow texture features, mid-level semantic features, and



(a) 10 tasks with $\alpha = 2$

(b) 10 tasks with $\beta = 0.05$

Figure 4: Adapter frozen results in the last task on CIFAR100, the x-axis represents which task it is.

Table 4: Expert number results of performance comparison on ImageNet-R with 10 Tasks. The best results are shown in bold.

	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
12 Experts	72.5	78.3	14.7	72.5	78.2	15.3	72.2	78.1	15.8
6 Experts	73.2	78.7	14.3	73.2	78.7	14.6	72.7	78.3	15.6
4 Experts	**74.0**	**79.5**	**13.9**	**73.5**	**79.2**	**14.3**	**72.9**	**79.1**	**15.4**
2 Experts	72.6	78.4	14.9	72.3	78.4	15.5	72.2	78.1	16.1

deep abstract features, and we insert an expert into each module. It is important to note that this expert allocation strategy is not unique. We investigate how the number of experts affects model performance on distribution-based splits of ImageNet-R, and the results are shown in Table 4. With 12 experts, there is one per layer, with 6 experts, there is one every 1 layer. And with 2 experts, there is one every 5 layers. The results show that having too many experts prevents the final expert from fully learning the relevant knowledge, while having too few experts limits the amount of knowledge that can be transferred. Therefore, it is essential to choose an appropriate number of experts to ensure that sufficient and important knowledge is passed on to the final expert.

5 Conclusion

In this paper, we introduce MultiFCL, a novel FCL framework to address the stability-plasticity trade-off in FCL while leveraging the strengths of PTMs. Specifically, to maintain the stability of PTMs, MultiFCL incorporates lightweight adapters for efficient task adaption, and prevents catastrophic forgetting by freezing them during fine-tuning. To further preserve knowledge retention, it employs multi-modal prototype initialization, incorporating semantic features and sample features to enhance class representations. On the other hand, to strengthen the plasticity of PTMs, MultiFCL employs a multi-expert training mechanism, which combines multi-scale feature learning with multi-teacher self-distillation to improve knowledge transfer and fusion. Experimental results demonstrate that our proposed MultiFCL achieves state-of-the-art performance across multiple datasets and configurations, highlighting its effectiveness in FCL.

Acknowledgment

This work is supported by the National Key Research and Development Program (No. 2022YFB3103202) of China.

References

Aljundi, R., Chakravarty, P., and Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7120–7129. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.753.

Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M., and Avestimehr, S. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Bakman, Y. F., Yaldiz, D. N., Ezzeldin, Y. H., and Avestimehr, S. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=nAs4LdaP9Y`.

Chen, H., Zhang, Y., Krompass, D., Gu, J., and Tresp, V. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 11285–11293. AAAI Press, 2024a. doi: 10.1609/AAAI.V38I10.29007.

Chen, J., Ma, B., Cui, H., and Xia, Y. Think twice before selection: Federated evidential active learning for medical image analysis with domain shifts. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11439–11449, 2024b. doi: 10.1109/CVPR52733.2024.01087.

Cui, Y., Cao, K., Cao, G., Qiu, M., and Wei, T. Client scheduling and resource management for efficient training in heterogeneous iot-edge federated learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8):2407–2420, 2022. doi: 10.1109/TCAD.2021.3110743.

Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., and Zhu, Q. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10154–10163. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00992.

Dong, J., Li, H., Cong, Y., Sun, G., Zhang, Y., and Gool, L. V. No one left behind: Real-world federated class-incremental learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(4):2054–2070, 2024. doi: 10.1109/TPAMI.2023.3334213.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 315–323. JMLR.org, 2011.

Guo, H., Zhu, F., Liu, W., Zhang, X., and Liu, C. Pilora: Prototype guided incremental lora for federated class-incremental learning. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXV*, volume 15123 of *Lecture Notes in Computer Science*, pp. 141–159. Springer, 2024. doi: 10.1007/978-3-031-73650-6_9.

Halbe, S., Smith, J. S., Tian, J., and Kira, Z. Continual adaptation of vision transformers for federated learning. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=vsZ5A3Zxyr`.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 8320–8329. IEEE, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Jia, Z., Zhou, T., Yan, Z., Hu, J., and Shi, Y. Personalized meta-federated learning for iot-enabled health monitoring. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(10):3157–3170, 2024. doi: 10.1109/TCAD.2024.3388908.

Jung, D., Lee, D., Hong, S., Jang, H., Bae, H., and Yoon, S. New insights for the stability-plasticity dilemma in online continual learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=fxC7kJYwA_a`.

Kahneman, D., Slovic, P., and Tversky, A. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, 1982.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Kurniawan, M. R., Song, X., Ma, Z., He, Y., Gong, Y., Qi, Y., and Wei, X. Evolving parameterized prompt memory for continual learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 13301–13309. AAAI Press, 2024. doi: 10.1609/AAAI.V38I12.29231.

Li, Y., Xu, W., Qi, Y., Wang, H., Li, R., and Guo, S. SR-FDIL: synergistic replay for federated domain-incremental learning. *IEEE Trans. Parallel Distributed Syst.*, 35(11):1879–1890, 2024. doi: 10.1109/TPDS.2024.3436874.

Liu, C., Qu, X., Wang, J., and Xiao, J. Fedet: A communication-efficient federated class-incremental learning framework based on enhanced transformer. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3984–3992. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/443.

Lu, Y., Yang, L., Chen, H., Cao, J., Lin, W., and Long, S. Federated class-incremental learning with dynamic feature extractor fusion. *IEEE Trans. Mob. Comput.*, 23(12):12969–12982, 2024. doi: 10.1109/TMC.2024.3419096.

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, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 2182–2188. ijcai.org, 2022. doi: 10.24963/IJCAI.2022/303.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

mnmoustafa and Ali, M. Tiny imagenet. `https://kaggle.com/competitions/tiny-imagenet`, 2017. Kaggle.

Moon, J., Park, K., Kim, J. U., and Park, G. Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11697–11707. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01077.

Nguyen, M.-D., Lee, S.-M., Pham, Q.-V., Hoang, D. T., Nguyen, D. N., and Hwang, W.-J. Hcfl: A high compression approach for communication-efficient federated learning in very large scale iot networks. *IEEE Transactions on Mobile Computing*, 22(11):6495–6507, 2023. doi: 10.1109/TMC.2022.3190510.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2642–2651. PMLR, 2017.

Ogier du Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., Muzellec, B., Philippenko, C., Silva, S., Teleńczuk, M., Albarqouni, S., Avestimehr, S., Bellet, A., Dieuleveut, A., Jaggi, M., Karimireddy, S. P., Lorenzi, M., Neglia, G., Tommasi, M., and Andreux, M. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5315–5334. Curran Associates, Inc., 2022.

Piao, H., Wu, Y., Wu, D., and Wei, Y. Federated continual learning via prompt-based dual knowledge transfer. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=Kqa5JakTjB`.

Qi, D., Zhao, H., and Li, S. Better generative replay for continual federated learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=cRxYWKiTan`.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. FLAVA: A foundational language and vision alignment model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15617–15629. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01519. URL `https://doi.org/10.1109/CVPR52688.2022.01519`.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11909–11919, 2023a. doi: 10.1109/CVPR52729.2023.01146.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 11909–11919. IEEE, 2023b. doi: 10.1109/CVPR52729.2023.01146.

Tran, M., Le, T., Le, X., Harandi, M., and Phung, D. Text-enhanced data-free approach for federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 23870–23880. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02253.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. J. The caltech-ucsd birds-200-2011 dataset. 2011. URL `https://api.semanticscholar.org/CorpusID:16119123`.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C., Ren, X., Su, G., Perot, V., Dy, J. G., and Pfister, T. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, volume 13686 of *Lecture Notes in Computer Science*, pp. 631–648. Springer, 2022. doi: 10.1007/978-3-031-19809-0_36.

Wu, K., Peng, H., Zhou, Z., Xiao, B., Liu, M., Yuan, L., Xuan, H., Valenzuela, M., Chen, X. S., Wang, X., Chao, H., and Hu, H. Tinyclip: CLIP distillation via affinity mimicking and weight inheritance. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 21913–21923. IEEE, 2023. doi: 10.1109/ICCV51070.2023.02008. URL `https://doi.org/10.1109/ICCV51070.2023.02008`.

Wuerkaixi, A., Cui, S., Zhang, J., Yan, K., Han, B., Niu, G., Fang, L., Zhang, C., and Sugiyama, M. Accurate forgetting for heterogeneous federated continual learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=ShQrnAsbPI`.

Xie, L., Lin, M., Luan, T., Li, C., Fang, Y., Shen, Q., and Wu, Z. MH-pFLID: Model heterogeneous personalized federated learning via injection and distillation for medical data analysis. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=Jvh8HM9YEJ`.

Yan, H., Wang, L., Ma, K., and Zhong, Y. Orchestrate latent expertise: Advancing online continual learning with multi-level supervision and reverse self-distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 23670–23680. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02234.

Yang, Q., Zhang, J., Hao, W., Spell, G. P., and Carin, L. FLOP: federated learning on medical datasets using partial networks. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 3845–3853. ACM, 2021. doi: 10.1145/3447548.3467185.

Yoon, J., Jeong, W., Lee, G., Yang, E., and Hwang, S. J. Federated continual learning with weighted inter-client transfer. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12073–12086. PMLR, 2021.

Yu, H., Xu, R., Zhang, H., Yang, Z., and Liu, H. Ev-fl: Efficient verifiable federated learning with weighted aggregation for industrial iot networks. *IEEE/ACM Transactions on Networking*, 32(2): 1723–1737, 2024a. doi: 10.1109/TNET.2023.3328635.

Yu, H., Yang, X., Gao, X., Feng, Y., Wang, H., Kang, Y., and Li, T. Overcoming spatial-temporal catastrophic forgetting for federated class-incremental learning. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pp. 5280–5288. ACM, 2024b. doi: 10.1145/3664647.3681384.

Yu, H., Yang, X., Gao, X., Kang, Y., Wang, H., Zhang, J., and Li, T. Personalized federated continual learning via multi-granularity prompt. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 4023–4034. ACM, 2024c. doi: 10.1145/3637528.3671948.

Zhang, Z., Guo, B., Sun, W., Liu, Y., and Yu, Z. Cross-fcl: Toward a cross-edge federated continual learning framework in mobile edge computing systems. *IEEE Trans. Mob. Comput.*, 23(1):313–326, 2024. doi: 10.1109/TMC.2022.3223944.

Zhou, D., Sun, H., Ning, J., Ye, H., and Zhan, D. Continual learning with pre-trained models: A survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 8363–8371. ijcai.org, 2024a.

Zhou, D.-W., Cai, Z.-W., Ye, H.-J., Zhan, D.-C., and Liu, Z. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 2024b.

Zhou, D.-W., Wang, Q.-W., Qi, Z.-H., Ye, H.-J., Zhan, D.-C., and Liu, Z. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024c.

A Supplemental Experiments

A.1 Main Results Supplement

Table 5 and Table 6 presents the results on CIFAR100 with 5 tasks. It can be observed that MultiFCL consistently achieves the best performance across all data partitioning settings, demonstrating its superior ability to adapt to new tasks while maintaining strong generalization.

Specifically, as α decreases, MultiFCL exhibits the smallest performance drop, indicating that it effectively leverages PTM pre-training characteristics along with multi-scale experts and self-distillation to mitigate catastrophic forgetting and enhance model stability.

Moreover, as β decreases, the data distribution becomes increasingly heterogeneous, leading to an overall decline in model accuracy. However, MultiFCL still demonstrates the highest robustness. For instance, under the most extreme partitioning condition ($\beta = 0.05$), MultiFCL achieves 68.1% (Last) and 72.4% (Avg.), significantly outperforming PiLoRA (63.5% and 68.3%) and FedViT (60.3% and 64.4%). This result highlights the strong learning capability of MultiFCL in low-data and highly heterogeneous federated environments, effectively preventing severe knowledge forgetting during the FCL process.

Table 5: Main results of performance comparison on CIFAR100 with 5 tasks in distribution-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	65.0	68.2	25.1	63.3	66.7	25.7	60.3	64.4	26.0
CLIP	ICML 2021	64.5	67.6	25.8	63.6	67.3	25.9	60.5	64.1	26.3
MFCL	NeurIPS 2023	36.9	47.2	25.7	32.5	44.8	26.4	30.7	41.9	27.2
FedCBC	MM 2024	23.9	23.0	30.5	20.4	24.2	31.1	17.8	22.9	32.5
LANDER	CVPR 2024	25.8	26.6	28.7	24.6	26.3	29.4	24.7	26.0	30.7
FedMGP	KDD 2024	24.4	35.2	31.2	24.4	33.5	32.1	24.7	31.4	32.6
PiLoRA	ECCV 2024	64.2	68.6	26.9	66.8	69.5	27.3	63.5	68.3	27.4
PiLoRA-△	ECCV 2024	66.3	68.2	26.8	66.4	68.8	27.1	63.4	68.0	27.6
MultiFCL	This Paper	**72.4**	**75.5**	**12.6**	**70.0**	**73.1**	**12.7**	**68.1**	**72.4**	**12.6**

Table 6: Main results of performance comparison on CIFAR100 with 5 tasks in quality-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\alpha = 6$			$\alpha = 4$			$\alpha = 2$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	63.4	66.9	27.4	62.8	66.4	28.1	62.3	65.3	28.7
CLIP	ICML 2021	63.4	67.2	27.7	63.1	66.7	28.0	62.6	65.8	28.5
MFCL	NeurIPS 2023	35.4	43.7	30.6	31.3	35.8	31.4	21.9	26.4	32.8
FedCBC	MM 2024	20.3	25.7	28.9	18.5	22.4	29.5	17.1	22.5	31.0
LANDER	CVPR 2024	36.8	45.5	27.8	32.6	37.8	28.0	27.6	31.2	29.1
FedMGP	KDD 2024	19.8	24.3	30.6	16.6	19.1	31.2	14.7	18.9	32.0
PiLoRA	ECCV 2024	66.4	72.9	28.4	64.4	68.9	29.1	64.2	69.3	29.6
PiLoRA-△	ECCV 2024	65.5	72.1	28.9	64.1	68.8	29.3	63.9	68.7	29.8
MultiFCL	This Paper	**69.2**	**74.1**	**10.4**	**69.9**	**74.6**	**11.0**	**68.5**	**73.5**	**12.3**

A.2 Ablation Results Supplement

To further analyze the contributions of key components in MultiFCL, we conduct ablation studies on CIFAR100. Table 7 illustrate the impact of each module on the final model performance, including adapter, prototype initialization, $\mathcal{L}_{\text{experts}}$, and $\mathcal{L}_{\text{self-kd}}$.

Table 7: Ablation results of performance comparison on CIFAR100 with 5 Tasks. The best results are shown in bold.

Adapter	Init.	$\mathcal{L}_{\text{experts}}$	$\mathcal{L}_{\text{self-kd}}$	$\alpha = 6$	$\alpha = 4$	$\alpha = 2$	$\beta = 0.5$	$\beta = 0.1$	$\beta = 0.05$
				63.4	62.8	62.3	65.5	63.8	61.2
✓				65.7	64.2	62.8	67.4	65.0	63.2
✓	✓			66.5	65.7	63.8	68.2	66.9	64.6
✓	✓	✓		68.1	68.6	67.3	70.9	68.7	66.7
✓	✓	✓	✓	**69.2**	**69.9**	**68.5**	**72.4**	**70.0**	**68.1**

Table 8: Few-shot results of performance comparison on CUB-200-2011 with 10 tasks in distribution-based partitioning, where -$\triangle$ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	73.5	75.6	29.7	71.2	74.9	30.3	70.9	73.2	31.7
CLIP	ICML 2021	73.0	74.9	30.0	71.4	75.0	30.8	70.9	73.3	32.0
MFCL	NeurIPS 2023	22.8	27.6	33.6	21.1	25.7	34.1	20.6	25.9	35.3
FedCBC	MM 2024	20.2	29.5	30.5	19.9	27.4	31.7	17.1	26.3	32.8
PiLoRA	ECCV 2024	63.0	73.5	29.6	53.9	68.0	30.7	57.3	69.3	31.5
PiLoRA-$\triangle$	ECCV 2024	62.6	72.2	30.2	53.6	67.3	31.6	57.5	69.4	31.6
MultiFCL	This Paper	**78.5**	**84.5**	**16.7**	**77.7**	**84.9**	**18.2**	**77.2**	**84.9**	**18.0**

Introducing adapters improves performance across all settings, with 5 tasks improving by 2.3%. And with the addition of prototype initialization, it shows further accuracy gains. After incorporating $\mathcal{L}_{\text{experts}}$, accuracy for 5 tasks improves by 1.6% - 3.5%, confirming that it facilitates knowledge sharing across tasks and enhances the model's ability to handle imbalanced data distributions.

Finally, with $\mathcal{L}_{\text{self-kd}}$, the model achieves the highest accuracy. Overall, these results validate that MultiFCL, by integrating multiple critical modules, significantly boosts model performance in FCL scenarios across various data partitioning settings.

A.3 Few-Shot Study

In FL, resource-constrained clients may have only a limited number of samples for training, and this data scarcity exacerbates the challenge of adapting to new tasks over the long-term learning process. To address this, we conduct the few-shot study, exploring the performance of existing methods under few-shot conditions. Specifically, we experiment on the few-shot dataset CUB-200-2011. Under the distribution-based data partitioning, each client has a few dozen samples, while under the quality-based partitioning, each client only has three samples per class. The results are shown in Table 8 and Table 9.

It can be observed that MultiFCL demonstrates the most significant advantage in the few-shot setting. Since PTMs are pre-trained on large-scale datasets, they can achieve expressive results with only a small number of samples. In contrast, FCL methods that train models from scratch tend to over-fit in the few-shot setting, leading to a significant performance decline. Meanwhile, MultiFCL enhances the model's ability to adapt to new tasks through multi-teacher self-distillation and multi-scale feature learning, preventing excessive reliance on the limited sample information. Experimental results show that MultiFCL effectively alleviates adaptation challenges in few-shot settings, maintaining its performance even in data-scarce scenarios.

What's more, Table 10 and Table 11 presents the results on CUB-200-2011 with 5 tasks. MultiFCL consistently achieves the best performance across all data partitioning settings. For instance, when $\alpha = 6$, MultiFCL attains 66.6% (Last) and 75.9% (Avg.) accuracy, outperforming PiLoRA by 27.5% and 24.8%, respectively, and surpassing FedViT by 5.6% and 6.8%. Even under extremely low-data conditions, MultiFCL effectively leverages pre-trained knowledge while enhancing model

Table 9: Few-shot results of performance comparison on CUB-200-2011 with 10 tasks in quality-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\alpha = 6$			$\alpha = 4$			$\alpha = 2$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	61.0	69.1	33.9	57.9	64.3	34.5	53.1	56.5	35.2
CLIP	ICML 2021	61.1	69.3	34.0	57.6	64.5	34.4	53.2	56.6	35.5
MFCL	NeurIPS 2023	18.8	21.5	34.4	13.6	19.9	35.0	10.8	17.3	35.7
FedCBC	MM 2024	17.4	21.4	34.3	15.9	21.0	35.2	15.2	20.5	35.6
PiLoRA	ECCV 2024	39.1	51.1	35.7	43.9	57.0	36.4	53.2	65.9	37.2
PiLoRA-△	ECCV 2024	40.3	51.6	36.1	44.0	57.2	37.3	53.2	65.8	37.0
MultiFCL	This Paper	**66.6**	**75.9**	**24.0**	**64.4**	**73.5**	**25.2**	**57.0**	**69.0**	**26.0**

Table 10: Few-shot results of performance comparison on CUB-200-2011 with 5 tasks in distribution-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	73.2	75.0	29.9	72.0	74.9	30.5	70.4	73.6	31.1
CLIP	ICML 2021	73.0	75.2	30.4	72.1	75.0	30.7	71.3	73.9	32.0
MFCL	NeurIPS 2023	23.9	28.6	30.3	21.2	26.7	32.6	20.5	26.0	32.1
FedCBC	MM 2024	21.5	30.7	30.2	19.5	27.3	30.8	18.3	27.0	31.9
PiLoRA	ECCV 2024	61.0	70.8	27.6	55.6	66.0	28.3	58.4	68.3	29.0
PiLoRA-△	ECCV 2024	60.7	70.2	28.0	55.7	66.4	28.3	58.3	68.0	29.3
MultiFCL	This Paper	**77.4**	**83.7**	**15.0**	**77.8**	**83.8**	**14.8**	**77.8**	**84.1**	**15.6**

generalization through multi-scale learning and self-distillation mechanisms. As a result, it maintains stable and competitive performance even under highly heterogeneous data distributions.

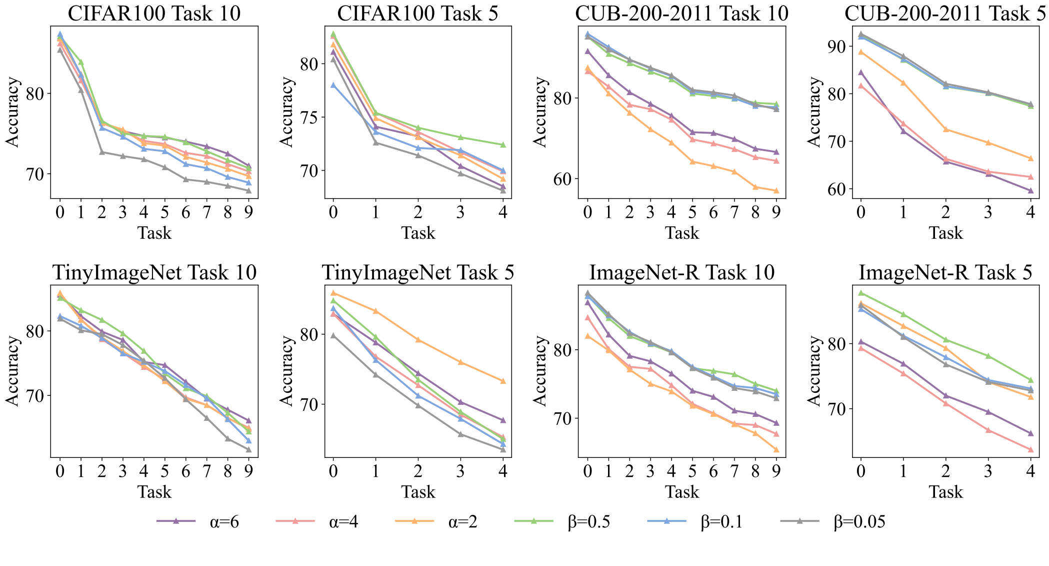


Figure 5: The performance of MultiFCL per task in the 5 and 10 tasks on all datasets.

A.4 Incremental Trend Study

Figure 5 presents the FCIL results of MultiFCL across all datasets and data splits, where fine-tuning the adapters on the first task enables the PTM to achieve highly expressive performance. As

Table 11: Few-shot results of performance comparison on CUB-200-2011 with 5 tasks in quality-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\alpha = 6$			$\alpha = 4$			$\alpha = 2$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	60.3	68.9	30.2	58.4	65.5	31.0	54.6	58.7	31.7
CLIP	ICML 2021	60.4	68.2	30.4	58.6	65.7	31.1	55.0	59.2	32.0
MFCL	NeurIPS 2023	20.4	22.5	33.3	18.7	20.3	33.8	15.2	18.6	35.1
FedCBC	MM 2024	19.5	21.6	32.3	16.9	22.1	33.9	15.2	21.2	34.5
PiLoRA	ECCV 2024	49.8	59.3	30.9	50.9	61.9	31.6	62.5	72.9	32.7
PiLoRA-△	ECCV 2024	50.0	59.9	30.1	51.0	66.3	31.7	62.4	67.6	32.4
MultiFCL	This Paper	**66.4**	**75.9**	**19.2**	**62.5**	**69.6**	**22.4**	**59.6**	**69.0**	**24.9**

the number of tasks increases and the model must recognize all previously seen classes, accuracy gradually declines. However, from the second through fourth tasks, by using multiple experts to help the model quickly understand the knowledge of new and old tasks, PTM is able to distinguish old classes from new ones, maintaining a balance between stability and plasticity.

A.5 Extreme Heterogeneity Study

In our experiments, distribution-based partitioning corresponds to Non-IID splitting, and the value of β determines the degree of heterogeneity, with smaller values producing more heterogeneous data. We evaluate MultiFCL's performance on the final ImageNet-R task under extreme client distribution heterogeneity, as shown in Figure 6. The results show that MultiFCL maintains expressive performance even at $\beta = 0.05$. In addition, we test the model at β values of 0.01, 0.005, and 0.001 and found that MultiFCL handles all of these scenarios effectively. Even in the most extreme cases the model's accuracy fluctuates by only 1.4%. This robustness comes from clients learning diverse perspectives through experts and sharing knowledge internally, while FL enables external knowledge exchange from multiple angles, allowing the model to quickly adapt to new tasks and remain highly resilient.

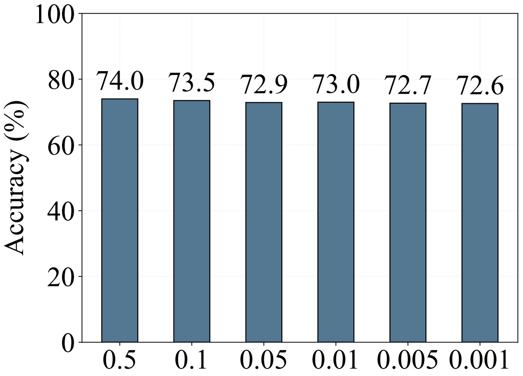


Figure 6: Extreme heterogeneity results on ImageNet-R with 10 tasks, the x-axis represents the value of β.

A.6 Loss Weight Study

We explore the relationship between weights of two loss functions during client local training, with the CIFAR100 results shown in Table 12, and the CUB-200-2011 results are shown in Table 13. Specifically, we conduct experiments by selecting values for λ'_{expert} from {0.1, 0.3, 0.5, 0.7, 0.9}. As the value decreases, the importance of multi-teacher self-distillation $\mathcal{L}_{\text{self-kd}}$ becomes smaller. From the results, it is evident that the performance of MultiFCL on CIFAR100 experiences only minor fluctuations, and different loss function weights do not significantly impact the overall performance of the model. This indicates that the coordination between multi-teacher self-distillation and multi-scale feature learning losses is quite robust during training, allowing for flexible adjustments within a certain range without affecting the model's performance.

Additionally, when the value of λ'_{expert} is around 0.5, MultiFCL achieves optimal performance. This suggests that learning knowledge from different experts helps the final expert better distinguish different samples, but the knowledge learned from former experts should not be too important, as this could cause the final expert to neglect the features it has extracted. By appropriately adjusting the

Table 12: Loss weight results of performance comparison on CIFAR100 with 10 tasks. The best results are shown in bold.

	10 Tasks					5 Tasks				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
$\alpha = 2$	68.6	69.2	**69.7**	69.6	69.5	67.9	**68.2**	68.1	68.0	67.7
$\alpha = 4$	69.7	70.0	**70.3**	**70.3**	70.2	68.8	69.6	**69.9**	69.5	69.1
$\alpha = 6$	69.0	69.3	**71.0**	69.6	68.5	68.3	68.9	**69.2**	68.9	68.7
$\beta = 0.5$	69.1	**70.7**	70.6	69.9	70.1	71.9	**72.4**	72.0	71.8	71.5
$\beta = 0.1$	68.3	68.7	**68.9**	68.8	68.7	69.3	69.7	**70.0**	69.9	69.5
$\beta = 0.05$	66.5	66.9	67.2	**67.9**	67.8	67.7	68.0	**68.1**	**68.1**	**68.1**

Table 13: Loss weight results of performance comparison on CUB-200-2011 with 10 tasks. The best results are shown in bold.

	10 Tasks					5 Tasks				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
$\alpha = 2$	56.4	56.8	**57.0**	56.7	56.5	69.0	59.1	58.4	**59.6**	59.4
$\alpha = 4$	64.1	64.2	**64.4**	64.3	64.0	61.9	62.1	62.4	**62.5**	62.1
$\alpha = 6$	66.1	66.3	66.4	**66.6**	66.4	65.8	66.0	**66.4**	66.3	66.4
$\beta = 0.5$	78.0	78.3	78.3	**78.5**	78.1	77.3	**77.4**	77.0	76.9	76.6
$\beta = 0.1$	77.3	77.6	**77.7**	77.6	77.4	77.3	77.6	**77.8**	77.6	77.5
$\beta = 0.05$	76.5	76.9	**77.2**	77.0	76.8	77.4	77.5	**77.8**	**77.8**	77.7

value, MultiFCL can be flexibly configured across different tasks and datasets, further enhancing the adaptability of the model.

A.7 Long Tasks Study

We evaluate MultiFCL's performance on long task sequences by extending ImageNet-R to 20, 30, 40, and 50 tasks using distribution-based splits with $\beta = 0.05$, and the results are shown in Figure 7. It can be seen that the model remains stable across long sequences, even though each task contributes less new knowledge. The client experts effectively retain previous knowledge while assimilating new information, allowing the model to enhance plasticity without sacrificing stability. As a result, MultiFCL exhibits only a 1.3% fluctuation in accuracy over these extended task sequences, demonstrating its effectiveness and robustness.

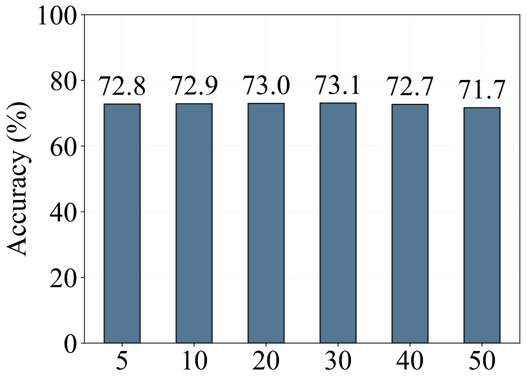


Figure 7: Long task results on ImageNet-R with $\beta = 0.05$, the x-axis is the number of tasks.

A.8 TIL Study

We analyze the TIL results of MultiFCL on CIFAR-100 under both 10-task and 5-task settings, as shown in Tables 14 and 15. MultiFCL consistently achieves the best performance. The tables show that TIL tasks are easier than CIL tasks since they do not require evaluating the model's performance on previous tasks and focus only on the current task. MultiFCL is designed to enhance plasticity while preserving stability, and its performance on the final task far surpasses that of other PTM-based methods, demonstrating its superior plasticity and validating its effectiveness.

Table 14: TIL results of performance comparison on CIFAR100 with 10 tasks in distribution-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	70.6	75.7	25.4	69.2	72.9	25.9	68.8	72.7	26.4
CLIP	ICML 2021	70.7	76.0	25.9	69.8	72.4	26.1	68.2	72.4	25.9
PiLoRA	ECCV 2024	74.4	75.4	24.3	67.3	72.9	24.1	74.8	72.8	25.7
PiLoRA-△	ECCV 2024	74 6	75.5	24.1	67.0	73.3	25.0	74.7	64.3	26.6
MultiFCL	This Paper	**75.3**	**79.5**	**10.6**	**74.0**	**77.8**	**12.8**	**72.5**	**76.5**	**11.9**

Table 15: TIL results of performance comparison on CIFAR-100 with 5 tasks in distribution-based partitioning, where -△ means using CLIP as base model. The best results are shown in bold.

	Methods	$\beta = 0.5$			$\beta = 0.1$			$\beta = 0.05$		
		Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$	Last	Avg.	$\mathcal{F}$
ViT	ICLR 2021	71.2	74.7	15.5	70.8	72.6	15.0	70.0	71.8	16.8
CLIP	ICML 2021	72.0	75.0	15.3	71.1	72.7	74.9	69.8	71.9	17.5
PiLoRA	ECCV 2024	72.0	72.9	13.6	71.7	72.7	14.0	71.9	74.7	14.5
PiLoRA-△	ECCV 2024	72.6	73.0	13.0	71.7	72.7	14.2	72.1	74.3	13.9
MultiFCL	This Paper	**77.8**	**78.8**	**1.0**	**75.5**	**74.9**	**1.5**	**76.0**	**76.9**	**2.4**

A.9 PTM Study

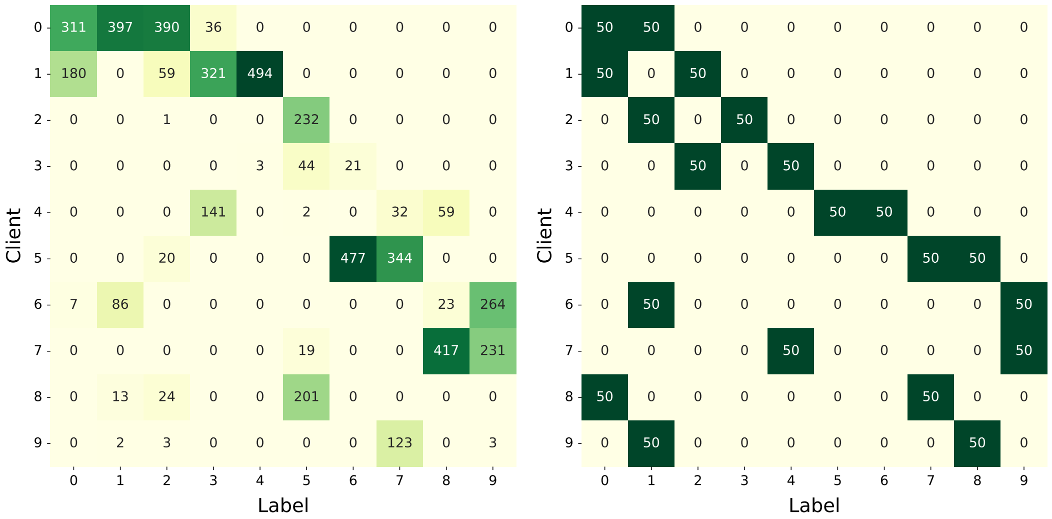
MultiFCL relies solely on CLIP's image-text alignment capability and can be extended to any PTM with such alignment. For example, we supplement Flava (Singh et al., 2022) and TinyCLIP ViT-39M/16 Text-19M (Wu et al., 2023) as our backbone model and report its last accuracy results in Table 16. MultiFCL maintains relatively stable performance across different PTMs. Beyond this, image-text alignment can also be achieved by adding projection layers, offering a pathway to extend compatibility with more PTMs. It can be seen that using TinyCLIP does not degrade MultiFCL's performance. Conversely, it remains highly stable, which demonstrates the potential for balancing resource constraints and performance requirements in resource-limited scenarios (e.g., edge devices).

Table 16: Comparison results of last accuracy on CIFAR100 with 10 tasks in both quality-based partitioning and distribution-based partitioning.

Model	$\alpha = 6$	$\alpha = 4$	$\alpha = 2$	$\beta = 0.5$	$\beta = 0.1$	$\beta = 0.05$
CLIP-based	71.0	70.3	69.7	70.7	68.9	67.9
Flava-based	70.4	69.9	69.7	69.5	68.2	67.5
TinyCLIP ViT-39M-based	70.8	70.2	69.5	70.9	70.0	69.2

B Data Partition Details

Referring to the data partitioning strategies described in PiLoRA (Guo et al., 2024), greater numbers of classes possessed by a client indicate higher data quality. Thus, in quality-based partitioning, each client is assigned data from α randomly selected classes. Specifically, the number of samples for each class on a client is determined by dividing the total number of samples for that class by the number of clients, ensuring that data across clients do not overlap. Therefore, a larger α corresponds to higher data quality for the clients. Additionally, In the distribution-based data partitioning, each client's data distribution is determined by the Dirichlet distribution, with smaller values of β indicating higher data heterogeneity. Figure 8 shows the data distribution under different datasets and partitioning strategies in the last task, which the ID of clients and labels are only used to distinguish between different clients and labels. The code is available at `https://github.com/yang12318/MultiFCL`.



(a) 10 tasks with $\beta = 0.05$

	0	1	2	3	4	5	6	7	8	9
0	311	397	390	36	0	0	0	0	0	0
1	180	0	59	321	494	0	0	0	0	0
2	0	0	1	0	0	232	0	0	0	0
3	0	0	0	0	3	44	21	0	0	0
4	0	0	0	141	0	2	0	32	59	0
5	0	0	20	0	0	0	477	344	0	0
6	7	86	0	0	0	0	0	0	23	264
7	0	0	0	0	0	19	0	0	417	231
8	0	13	24	0	0	201	0	0	0	0
9	0	2	3	0	0	0	0	123	0	3

(b) 10 tasks with $\alpha = 2$

	0	1	2	3	4	5	6	7	8	9
0	50	50	0	0	0	0	0	0	0	0
1	50	0	50	0	0	0	0	0	0	0
2	0	50	0	50	0	0	0	0	0	0
3	0	0	50	0	50	0	0	0	0	0
4	0	0	0	0	0	50	50	0	0	0
5	0	0	0	0	0	0	0	50	50	0
6	0	50	0	0	0	0	0	0	0	50
7	0	0	0	0	50	0	0	0	0	50
8	50	0	0	0	0	0	0	50	0	0
9	0	50	0	0	0	0	0	0	50	0

Figure 8: The Data Partition in the Last Task on CIFAR100.

C Method Analysis

Communication Cost. The model parameters consist of the PTM, multiple experts, and adapters. Notably, the PTM does not require communication, and adapters only need to be exchanged during the first task. For instance, in a scenario with 10 tasks, the total number of expert parameters is 384,000, while the adapters account for 156,768 parameters. Approximately 0.54 million parameters are communicated during the first task, and only about 0.38 million in subsequent tasks. The following comparison of parameters in Table 17 with others demonstrates that MultiFCL is highly communication-efficient.

Table 17: Parameter Comparison.

Model	Parameters
ResNet-18	11M
ResNet-50	23M
ResNet-101	43M
ViT-B/16-IN1K	86M
ViT-B/16-CLIP	86M
MultiFCL	0.54/ 0.38M

Computational Efficiency. In clients, for forward propagation, the multi-expert feature learning requires extracting features $\{h_m\}_{m=1}^{M}$ from M modules. For a PTM with L total layers divided into M modules, the computational complexity remains: $\mathcal{C}_{\text{forward}} = O\left(\frac{L}{M} \cdot (d^2 \cdot S)\right) \times M = O(Ld^2S)$, where S represents the sequence length, d is the dimension of a prototype. For backward propagation, it is equivalent to forward propagation, which is $O(Ld^2S)$. In the server, for the first task, the server needs to aggregate experts and adapters uploaded by clients, with a computational complexity of $O(K \times M \times d) + O(K \times N \times D)$, where D is the dimension of an adapter, and N is the total number of adapters in a client. For the subsequent tasks, the server only needs to aggregate the prototype experts and adapters are frozen, with a computational complexity of $O(K \times M \times d)$.

We compare communication parameters and wall-clock per round for MultiFCL, PiLoRA, MFCL, and LANDER on CIFAR100. Each round includes 5 local epochs across 10 clients plus one server update, as shown in Table 18. In MFCL, the server must distribute three components to clients, including the current-task global model, previous-task global model, and generator. Although the global model uses ResNet-18 (11.2M), this still incurs substantial communication overhead. Meanwhile, despite its wall-clock per round remains low, it requires 100 communication rounds per task to achieve competitive results, whereas MultiFCL and PiLoRA attain expressive outcomes within merely 5 rounds. LANDER transmits the ResNet-18 as global model each round. Its integration of pre-trained CLIP and generator yields wall-clock per round comparable to PTM-based methods while requiring 100 rounds per task. PiLoRA communicates LoRA parameters and prototypes per round.

Although parameter volume stays low, continuous LoRA tuning prevents effective knowledge retention and increases computation time, resulting in suboptimal performance versus MultiFCL. MultiFCL freezes adapters after initial tasks, reducing communication parameters (0.54M to 0.38M) and wall-clock per round (228.29s to 150.24s) while achieving competitive performance within 5 communication rounds per task. Collectively, MultiFCL maintains computational efficiency while reducing communication costs versus existing methods.

Table 18: Comparison results of communication parameters and wall-clock per round on CIFAR100 with 10 tasks in distribution-based partitioning.

Model	Parameters	Wall-clock
MFCL	24.27M	35.20s
LANDER	11.20M	169.65s
PiLoRA	0.0584M	207.10s
MultiFCL	0.54/ 0.38M	228.29/ 150.24s

D Algorithm

Algorithm 1 displays the detailed process in MultiFCL. It employs adapters to fine-tune the PTM and leverages the semantic features of old tasks to initialize new class prototypes. Then, it establishes multiple experts, employing feature learning loss and the multi-teacher dynamic self-distillation to transfer knowledge to the final expert.

Algorithm 1 MultiFCL Framework

```
Require: Pre-trained model $f$, clients $K$, communication rounds $T$, tasks $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_\tau\}$, local
    epochs $E$
for each task $\mathcal{T}_t \in \{\mathcal{T}_1, \dots, \mathcal{T}_\tau\}$ do
    if $t = 1$ then
        for round $r = 1$ to $T$ do
            for each client $k = 1$ to $K$ in parallel do
                Fine-tune adapters on $D_k^1$
                Upload $w_k^r$ and sample count $|X_k^1|$ to server
            end for
            Server aggregates: $w^r \leftarrow \sum_{k=1}^{K} \frac{|X_k^1|}{\sum |X_k^1|} w_k^r$
            Server broadcasts global adapter $w^r$ to clients
        end for
        Freeze global adapters $w^* \leftarrow w^T$
    else
        for each new class $c \in \mathcal{T}_t$ do
            Compute semantic similarity: sim $\leftarrow \frac{f'(x_1^c) \cdot f_{\text{text}}(C^{\mathcal{T}_{\text{old}}})}{\|f'(x_1^c)\| \|f_{\text{text}}(C^{\mathcal{T}_{\text{old}}})\|}$
            Initialize prototype: $\mathbf{p}_{\text{init}}^c \leftarrow \text{Attention}(f'(x_1^c), f_{\text{text}}(c_{\text{sim}}^{\mathcal{T}_{\text{old}}}), \mathbf{p}_{\text{old}}^c)$
        end for
    end if
    for round $r = 1$ to $T$ do
        for each client $k = 1$ to $K$ in parallel do
            for local epoch $e = 1$ to $E$ do
                Extract multi-scale features: $\{h_m\}_{m=1}^{M} \leftarrow f'(x; w^*)$
                Compute expert loss: $\mathcal{L} = \mathcal{L}_{\text{experts}} + \lambda'_{\mathrm{expert}} \cdot \mathcal{L}_{\text{self-kd}}$
                Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta(\mathcal{L})$
                Update prototypes: $\mathbf{p}_{(j)}^c \leftarrow \frac{\mathbf{p}_{(j-1)}^c + f'(x_j^c)}{2}$
            end for
            Upload prototypes $\{\mathbf{p}_{k,m}^c\}_{m=1}^{M}$ and sample counts $|X_k^{c,t}|$ to server
        end for
        Server aggregates prototypes: $\mathbf{p}^{c,r} \leftarrow \sum_{k=1}^{K} \frac{|X_k^{c,t}|}{\sum |X_k^{c,t}|} \sum_m \mathbf{p}_{k,m}^{c,r}$
        Server broadcasts global prototypes $\mathbf{p}^r$ to clients
    end for
end for
```

E Limitations

Although the proposed MultiFCL framework delivers strong performance across a variety of heterogeneous data partitions and CL scenarios, it still has several limitations that call for further study. MultiFCL depends on large scale PTMs and their multi-scale feature modules, which may impose considerable computational and storage demands on devices with limited resources. The mechanisms for prototype initialization and semantic feature matching need refinement in extreme class imbalance or few sample scenarios to ensure that both new and existing classes are represented effectively. The current evaluation focuses mainly on image classification tasks and it remains necessary to investigate how MultiFCL can be applied to other modalities such as text and speech or extended to more complex tasks.

F Code of Ethics

This paper is guided by a commitment to user privacy, performs all model updates and feature extraction entirely in clients without collecting or storing any raw data. We provide a detailed description of each algorithmic component including multi-adapter fine-tuning, prototype initialization, multi-scale feature learning and multi-teacher adaptive self-distillation along with network architectures and hyperparameter settings to ensure that others can reproduce and validate our findings without accessing private data. During evaluation, we use publicly available datasets to emulate realistic distributions and prevent exposure of any individual-level information. All original data remains on local devices under the full control of each participant throughout the entire process.

G Societal Impacts

This paper presents work that aims to advance the field of deep learning. Our research explores how to leverage pre-trained models to enhance performance in FCL. By introducing mechanisms such as multi-adapter fine-tuning, multi-modal prototype initialization, multi-scale feature learning, and multi-teacher dynamic self-distillation, we improve the model's adaptability and generalization, ensuring performance is maintained while preventing catastrophic forgetting.

Since this work provides a novel perspective for deep learning from a technical standpoint, we recognize that with the broader application of such technologies, there may be some impacts to consider. For instance, FCL has the potential to play a critical role in fields like healthcare and the internet of things. This requires us to design methods that avoid bias or misleading decisions, and we are actively working towards that goal.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explicitly state in the abstract and introduction that this paper is a research on FCL. Please refer to Section 1 for contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix.E.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Section 4, Appendix.A and Appendix.B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please refer to Section 4, Appendix.B and our code in supplemental materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Section 4.1 and Appendix.B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the priority given to the repeatability of the dataset, these results are obtained with fixed hyperparameters and fixed sequence test segmentation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Please refer to Appendix.F.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Appendix.G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[Yes]

    Justification: We referenced the open-source datasets and models we used in Section 4.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.