

# 面向动态算力任务流的增量元蒸馏联邦持续学习方法

郑泽章<sup>1</sup>, 陈宁江<sup>1,2,3</sup>, 杨玉斌<sup>1</sup>, 陈娟<sup>1</sup>

<sup>1</sup>(广西大学 计算机与电子信息学院, 南宁 530004)

<sup>2</sup>(广西高校并行分布与智能计算重点实验室, 南宁 530004)

<sup>3</sup>(广西智能数字服务技术创新中心, 南宁 530004)

E-mail: chnj@gxu.edu.cn

**摘要:** 联邦学习凭借隐私保护特性成为算力网络关键技术, 但实际应用中面临灾难性遗忘与系统动态适应性差的双重挑战. 现有方法聚焦历史知识蒸馏效率, 却忽视节点间知识共享程度与动态环境下模型性能. 本文提出一种增量元蒸馏联邦持续学习方法, 内循环通过历史知识蒸馏与客户端知识共享实现本地知识固化, 引入多视图对比蒸馏策略, 利用特征增强与注意力加权机制实现细粒度知识筛选. 外循环设计动态元更新规则, 采用梯度差异补偿实现新旧节点渐进知识整合. 实验基于 CIFAR100 和 FMNIST 扩展的 4 个数据集验证, 本文方法较基线平均准确率提升 4.7% ~ 9.7%, 遗忘率降低 3.1% ~ 13.6%, 新增节点时系统收敛速度提升约 30%, 证明其在算力网络系统中的有效性与鲁棒性.

**关键词:** 联邦学习; 算力网络; 知识蒸馏; 持续学习; 元学习

中图分类号: TP391

文献标识码: A

文章编号: 1000-1220(2026)04-0868-09

## Incremental Meta-distillation Approach for Federated Continual Learning in Dynamic Computing Task Streams

ZHENG Zezhang<sup>1</sup>, CHEN Ningjiang<sup>1,2,3</sup>, YANG Yubin<sup>1</sup>, CHEN Juan<sup>1</sup>

<sup>1</sup>(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)

<sup>2</sup>(Key Laboratory of Parallel, Distributed and Intelligent Computing (Guangxi University), Education Department of Guangxi Zhuang Autonomous Region, Nanning 530004, China)

<sup>3</sup>(Guangxi Center of Technology Innovation for Intelligent Digital Services, Nanning 530004, China)

**Abstract:** Federated Learning has become a critical technology in computing power network due to its privacy-preserving attributes, but it confronts dual challenges of catastrophic forgetting and dynamic adaptability. While existing methods focus on enhancing historical knowledge distillation efficiency, they neglect the degree of inter-node knowledge sharing and model performance in dynamic environments. This paper proposes an incremental meta-distillation federated continual learning method. The inner loop achieves local knowledge stabilization through historical knowledge distillation and client-side knowledge sharing, incorporating a multi-view contrastive distillation strategy that leverages feature augmentation and attention-weighted mechanisms for fine-grained knowledge selection. The outer loop implements dynamic meta-update rules and employs gradient discrepancy compensation to realize progressive knowledge integration between legacy and new nodes. Experimental validation on four datasets extended from CIFAR100 and FMNIST demonstrates that the proposed method outperforms baselines with an average accuracy improvement of 4.7% ~ 9.7%, a forgetting rate reduction of 3.1% ~ 13.6%, and a 30% acceleration in system convergence speed when incorporating new nodes, proving its efficiency and robustness in computational power network systems.

**Keywords:** federated learning; computing power network; knowledge distillation; continual learning; meta learning

## 0 引言

算力网络的发展显著提升了计算资源的共享与利用效率,但在实际部署中仍面临隐私保护、数据异构、通信开销及系统扩展等挑战. 尽管已有针对算力建模、信息感知、资源分配平台的广泛研究,但如何在超级计算集群的持续任务流场景中构建安全可靠、通信高效的算力网络仍亟待深入探索<sup>[1]</sup>. 如跨域算力协作中,模型需持续适应新任务以优化本

地模型并保留历史记忆,但现有算力网络联邦学习系统难以保障大规模 AI 训练与长期任务的高效稳定运行.

传统的联邦学习方法多在静态场景中建模,数据类别固定<sup>[2]</sup>. 而在动态任务输入的算力网络中,现有联邦学习方法依赖全量历史数据存储训练全局模型,导致存储与算力成本飙升. 联邦持续学习整合分布式算力与动态学习能力,通过持续学习机制改造联邦学习静态框架,实现动态数据流下免全模型重训的持续更新<sup>[3]</sup>. 尽管联邦持续学习旨在适应实际算

收稿日期:2025-03-14 收修改稿日期:2025-05-12 基金项目:国家自然科学基金项目(62162003)资助;中央引导地方科技发展资金项目(桂科 ZY24212059)资助. 作者简介:郑泽章,男,1997年生,硕士,CCF 学生会员,研究方向为云计算、机器学习等;陈宁江(通信作者),男,1975年生,博士,教授,博士生导师,CCF 杰出会员,研究方向为软件工程、网络分布计算、云计算等;杨玉斌,男,1999年生,硕士,研究方向为云计算等;陈娟,女,2000年生,硕士,研究方向为云计算等.

力网络的动态任务流,但简单融合联邦学习与持续学习仍面临双重挑战.一方面,本地更新中模型仅优化当前私有数据易引发灾难性遗忘<sup>[4]</sup>(如存储受限节点因无法保留历史数据致模型过度拟合新任务而遗忘旧知识).另一方面,动态增删节点时新节点引入会拖慢收敛速度并触发系统振荡,需选择性复用跨客户端知识以加速系统收敛与抑制干扰.

本文针对算力网络中联邦持续学习的灾难性遗忘与动态适应性难题,突破传统单层知识蒸馏框架,提出基于增量元蒸馏的联邦持续学习系统 FedIMD (Federated Incremental Meta-Distillation),通过内外循环协同实现知识高效积累与节点动态扰动抑制.具体贡献如下:

1) 提出一种面向动态任务流环境的联邦增量元蒸馏框架.该框架采用双层优化架构,内循环将历史模型输出分布约束与多视图特征对比损失共同注入本地训练目标,形成双重正则化机制,缓解客户端在持续任务输入下的灾难性遗忘问题.外循环通过将动态元更新规则嵌入全局更新过程,采用指数衰减调控策略与元梯度驱动的权重分配机制,提升客户端动态环境下模型的收敛速度.

2) 为解决异构环境下客户端知识共享的精度-干扰权衡难题,引入多视图对比蒸馏策略.通过多视图数据增强与注意力驱动的特征筛选提取判别性知识,设计基于余弦相似度的对比损失函数,建立跨客户端正负样本对关系图谱,结合温度缩放策略实现知识的选择性迁移.相较传统蒸馏方式,多视图机制加速新客户端的收敛并提升模型精度,为本文联邦增量元蒸馏框架提供更灵活、精细的知识共享机制.

## 1 相关工作

联邦学习通过分布式客户端协同训练全局模型,在保护数据隐私的同时破解“数据孤岛”难题<sup>[5]</sup>.然而,实际应用中存在通信效率限制、数据异构性、灾难性遗忘三大挑战.针对上述问题,学界发展出联邦蒸馏、联邦元学习及联邦持续学习等主要技术路线<sup>[6,7]</sup>,以下将综述相关研究进展.

### 1.1 联邦蒸馏策略

联邦蒸馏策略通过引入知识蒸馏机制,在多个任务和设备之间共享知识,提升模型的泛化能力和计算效率<sup>[8]</sup>.为实现高效蒸馏,Xie 等人<sup>[9]</sup>提出了一种新颖的联邦学习范式

MH-pFLID,设计信使模型和收发模块实现高效信息传递.在提升联邦蒸馏系统的适应性方面,Han 等人<sup>[10]</sup>构建了一种无监督的联邦学习框架 FedX,采用双侧知识蒸馏处理异构无监督场景.而为了进一步推动异构客户端模型的灵活聚合,Lin 等人<sup>[11]</sup>设计了用于模型融合的集成蒸馏方法,即通过无标签数据对来自客户端模型的输出进行训练,融合客户端模型输出.

相比上述方法,本文所提出的框架同样基于联邦蒸馏知识共享的理念,但更聚焦于通过细粒度选择性蒸馏,优化客户端在不同任务间的知识转移过程,确保在新任务学习的同时,有效保持旧任务的知识,减少客户端间相互干扰.

### 1.2 联邦元学习

在联邦学习研究中,联邦元学习和联邦蒸馏是两个重要且具有广泛应用前景的发展方向<sup>[12]</sup>.联邦元学习通过在多个客户端上引入元学习框架,以解决数据异构性问题并增强模型泛化能力<sup>[13]</sup>.如 Fallah 等人<sup>[14]</sup>提出了 Per-FedAvg 方法,在全局模型更新过程中同时优化元目标,促进了模型对新任务的快速适应性.后续的研究则进一步深入探讨了在异构联邦环境下,元学习驱动的全局更新策略.然而,现有方法通常假设客户端数据分布相似,忽略实际场景中的分布偏移问题.Yang 等人<sup>[15]</sup>提出了一种基于分组的联邦元学习框架 G-FML,该框架基于数据分布相似性分组进行元学习,从而使得按组优化的元模型能够实现更为个性化的学习效果.为了进一步降低联合学习中的通信成本,Liu 等人<sup>[16]</sup>设计了一种元知识表示与动态权重分配机制,通过编码分布式客户端的元知识特征,结合样本贡献度自适应的权重更新策略,显著缓解异构性问题,为后续研究提供重要范式参考.

本文融合元学习与联邦蒸馏技术优势,主要针对算力网络中节点动态变化带来的挑战,设计动态元更新规则,提升客户端动态环境下模型的整体性能和收敛速度.

### 1.3 联邦持续学习

当前联邦学习与持续学习在各自方向已有显著进展,但两者交叉研究仍较为有限.Casado 等人<sup>[17]</sup>的工作探索了数据分布变化的联邦学习,但局限于单任务场景,难以应对多任务动态场景的适应性挑战.而 Yoon 等人<sup>[18]</sup>则从持续学习的角度提出了联邦持续学习的设定,提出了一种在联邦学习框架

表 1 FedIMD 与当前方法对比

Table 1 Comparison of FedIMD with state-of-the-art methods

类别	方法	快速收敛	灾难性遗忘	知识或参数共享
联邦蒸馏	Xie <sup>[9]</sup>	自适应优化	无	参数相似性约束
	Han <sup>[10]</sup>	双边知识蒸馏	无	知识蒸馏
	Lin <sup>[11]</sup>	早停机制	无	知识蒸馏
联邦元学习	Per-FedAvg <sup>[14]</sup>	元参数共享	元更新规则	跨任务元知识迁移
	Yang <sup>[15]</sup>	累积梯度更新	无	微调蒸馏
	Liu <sup>[16]</sup>	不确定性加权集成	无	贝叶斯集成
联邦持续学习	Casado <sup>[17]</sup>	梯度剪裁	历史经验重现	无
	Yoon <sup>[18]</sup>	动态学习率调整	历史知识蒸馏	权重共享
	GLFC <sup>[21]</sup>	无	类增量学习	知识蒸馏
	FedCIL <sup>[25]</sup>	无	新类生成	知识蒸馏
	FedIMD(本文方法)	动态元更新	历史知识蒸馏	多视图对比蒸馏

下进行持续学习的模型,但其仅关注单客户端性能且缺乏全

局模型维护机制.此外,Wuerkaixi 等人<sup>[19]</sup>、Chung 等人<sup>[20]</sup>以

及 Dong 等人<sup>[21]</sup>等人的研究同样与本文相关,但这些研究大多依赖固定的任务设定,在面临算力网络环境下的持续任务流以及动态变化时,仍然缺乏好的适应性策略。

现有方法对关键挑战的应对策略如表 1 所示,多数方案因未充分考量系统动态性与复杂性,在动态算力网络等复杂场景中存在全面性问题,致使其实际效果受限。为此,本文提出了一种增量元蒸馏为核心的双层蒸馏优化框架,通过内外循环协同作用,实现知识的高效积累与有序扩展,提升模型动态适应性。

## 2 问题定义

在动态任务流的算力网络场景下,系统中的持续学习模型需要在分布式算力节点上依次学习一系列任务序列  $\{\varpi^{(1)}, \varpi^{(2)}, \dots, \varpi^{(T)}\}$ 。其中,第  $t$  个任务  $\varpi^{(t)}$  对应一个标注数据集  $D^{(t)} = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N_t}$ ,由  $N_t$  个实例  $x_i^{(t)}$  及其标签  $y_i^{(t)}$  组成。图 1 展示了一个基于联邦持续学习的动态任务流算力网络框架,支持动态任务流的高效处理和持续优化。其核心特点在于:算力资源的动态性、任务输入的持续性以及隐私保护下的知识共享,以适应不断变化的计算需求。在现实条件下,任务序列往往以未知但有序的方式到达,模型在处理任务  $\varpi^{(t)}$  的期间能够访问其数据,但该访问权限在训练完成后即不可再现。因此,如何在任务不断变化和数据不可回溯的情况下,保持模型对所有任务的长期适应能力,成为系统核心挑战。

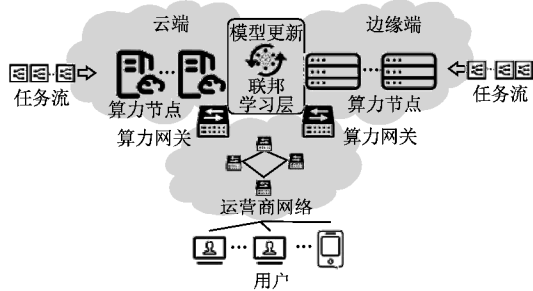


图 1 基于联邦持续学习的动态任务流算力网络框架  
Fig. 1 Dynamic task-stream computing power network framework based on federated continual learning

本文将传统的持续学习扩展到联邦学习设置中,所有算力节点同步参与每轮通信。假设系统有  $C$  个节点,其中每个节点  $c \in \{c_1, \dots, c_C\}$  在一个私有可访问的任务序列  $\{\varpi_c^{(1)}, \varpi_c^{(2)}, \dots, \varpi_c^{(T)}\} \subseteq \varpi$  上训练一个模型。当前的目标是在各算力节点的私有任务流上高效训练  $C$  个持续学习模型,并最大化算力节点的本地模型对当前以及历史  $T$  个任务的分类准确率。因此给定  $\varpi^{(t)}$  和到目前为止学到的模型,节点  $c$  在当前任务  $t$  时的学习目标如下:

$$\min_{\theta_c^{(t)}} (\mathcal{L}_{task}^{(c,t)}(\theta_c^{(t)} + \alpha \cdot \mathcal{L}_{forget}^{(c,t)}(\theta_c^{(t)}, \theta_c^{(t-1)}) + \gamma \cdot \mathcal{L}_{contrast}^{(c,t)}(\theta_c^{(t)})) \quad \forall c \in \{1, \dots, C\} \quad (1)$$

其中,  $\theta_c^{(t)}$  是算力节点(即客户端)  $c$  在任务  $t$  时的模型参数集,  $\mathcal{L}_{task}^{(c,t)}$  表示当前任务损失,  $\mathcal{L}_{forget}^{(c,t)}$  表示历史遗忘损失,  $\mathcal{L}_{contrast}^{(c,t)}$  表示多视图对比损失,  $\alpha$  和  $\gamma$  分别是控制遗忘抑制和对比知识迁移的强度的超参数。

## 3 方法

### 3.1 联邦元蒸馏框架

本文提出的基于双层优化架构的联邦持续学习框架 FedIMD,其核心由内循环本地优化层与外循环元知识融合层协同构成。如图 2 所示,该框架通过内循环与外循环的交替优化,实现动态任务流场景下的高效持续学习与跨节点知识共享。以下分别阐述框架的核心模块及其运行机理:

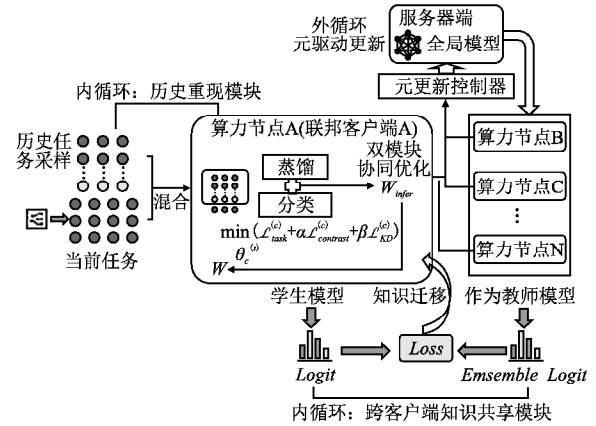


图 2 增量元蒸馏内外循环框架

Fig. 2 Incremental meta-distillation framework with inner-outer loop architecture

1) 内循环本地优化层:内循环本地优化层包含两个并行运作的核心模块,其中历史知识重现模块通过 KL 散度约束与历史模型输出对齐,固化关键记忆。然后以多视图对比学习筛选相似样本,吸收全局知识增强鲁棒性。

2) 外循环全局融合层:全局模型基于动态元聚合策略,以客户验证损失计算权重,衰减系数控制新节点影响力,生成跨任务泛化参数,实现新客户快速适配。

3) 内外循环协同:算力网络动态环境中,联邦持续学习需同步解决灾难性遗忘、知识迁移低效以及节点变化对模型稳定性的冲击。内外循环中历史知识重现通过 KL 散度约束确保了知识保留的底线,而多视图对比蒸馏则在此底线之上通过特征空间的对比学习提升知识质量,动态元更新进一步将这些优化结果在全局层面进行动态平衡。三者协同通过构建多层次的优化框架,将原本相互冲突的目标转化为互补关系。

### 3.2 内循环

#### 3.2.1 历史知识重现模块

在联邦持续学习场景下,客户端  $c$  在学习第  $t$  个任务时,其可用训练数据由两部分构成:1) 当前任务的数据集  $D_t$ ,包含  $K$  个类别的标注数据;2) 历史记忆库  $\epsilon_{1:t-1}$ ,存储从先前任务  $\{\varpi_c^{(1)}, \varpi_c^{(2)}, \dots, \varpi_c^{(t-1)}\}$  中采样的少量代表性示例,以在存储空间内保留历史知识。

完成当前任务训练后,需对本地模型  $\theta_c^{(t)}$  在所有已学习任务  $\{\varpi_c^{(1)}, \varpi_c^{(2)}, \dots, \varpi_c^{(t)}\}$  的类别上进行性能评估。受本地存储资源限制,历史记忆库的容量随任务数量增加而动态压缩,导致每个历史类别的样本数量逐步减少。为平衡新旧知识学习,内循环采用交叉熵损失与蒸馏损失联合优化的策略,具体

设计如下:

对于当前任务  $\mathcal{D}_c^{(t)}$  的输入样本  $(x, y) \sim \mathcal{D}_c^{(t)}$ , 交叉熵损失定义为:

$$\mathcal{L}_{task}^{(c,t)} = -\frac{1}{|B|} \sum_{x,y \in B} \sum_{k=1}^K \mathbb{I}(y=k) \cdot \log p_{\theta_c^{(t)}}(y=k|x) \quad (2)$$

式中  $B$  表示小批量数据,  $|B|$  为批量大小,  $\mathbb{I}(y=k)$  表示真实标签的独热编码,  $p_{\theta_c^{(t)}}(y=k|x)$  为模型对类别  $k$  的预测概率。

在联邦持续学习框架中, 仅依赖交叉熵损失无法解决任务间知识冲突引发的灾难性遗忘问题, 为此本文引入知识蒸馏损失作为核心优化项, 通过对比当前模型(学生模型, 参数为  $\theta_c^{(t)}$ ) 与历史模型(教师模型, 参数为  $\theta_c^{(t-1)}$ ) 的输出分布差异, 实现无原始数据依赖的历史知识迁移。即对教师模型与学生模型的 logits 输出  $f_{\theta_c^{(t-1)}}(x)$  和  $f_{\theta_c^{(t)}}(x)$  进行温度缩放(温度系数  $\tau$ ), 生成平滑概率分布以缓解类别不平衡问题, 计算公式为:

$$p^{(t)}(x) = \frac{\exp(f_{\theta_c^{(t)},k}(x)/\tau)}{\sum_{k=1}^K \exp(f_{\theta_c^{(t)},k}(x)/\tau)} \quad (3)$$

其中,  $f_{\theta_c^{(t)},k}(x)$  表示模型对类别  $k$  的 logits 输出。通过最小化教师与学生分布间的 KL 散度, 定义蒸馏损失函数为:

$$\mathcal{L}_{forget}^{(c,t)} = -\frac{1}{|B|} \sum_{x \in B} \sum_{k=1}^K \hat{p}_k^{(t-1)}(x) \log p_k^{(t)}(x) \quad (4)$$

公式中  $p_k^{(t-1)}(x)$  和  $p_k^{(t)}(x)$  分别为教师模型与学生模型对类别  $k$  的软化概率。该损失函数显式约束当前模型参数  $\theta_c^{(t)}$  的更新方向, 使其在优化当前任务交叉熵损失  $\mathcal{L}_{task}^{(c,t)}$  的同时, 继承历史任务  $\{\mathcal{D}_c^{(1)}, \mathcal{D}_c^{(2)}, \dots, \mathcal{D}_c^{(t-1)}\}$  的知识表征, 从而有效缓解灾难性遗忘问题。

### 3.2.2 跨客户端知识共享模块

在有效缓解灾难性遗忘后, 本文进一步探讨客户端间知识的精细化选择性共享与安全传递。为此, 本框架引入多视图对比蒸馏策略, 以确保从多重特征视角对客户端模型进行评估与知识提炼。在该策略中, 每个客户端模型分别通过多种数据增强和特征变换生成多视图表示, 这些视图间的特征分布差异可为模型提供更加丰富而稳定的判别信息。

对于联邦学习场景中第  $c$  个客户端, 设其本地数据集为  $\mathcal{D}_c = \{x_i^{(c)}, y_i^{(c)}\}_{i=1}^{N_c}$ 。为获得多视图表示, 对每个样本  $x_i^{(c)}$  进行数据增强, 产生  $M$  个视图:

$$\{x_{i,m}^{(c)}\}_{m=1}^M = \text{AugmentTransform}(x_i^{(c)}), x_{i,m}^{(c)} \in \mathcal{D}_c \quad (5)$$

每个增强样本  $x_{i,m}^{(c)}$  经本地模型  $f_c(\cdot; \theta_c)$  编码为特征表示:

$$z_{i,m}^{(c)} = f_c(x_{i,m}^{(c)}; \theta_c^{(t)}), m=1, \dots, M \quad (6)$$

其中,  $M$  表示每个样本生成的视图数量,  $z_{i,m}^{(c)} \in \mathbb{R}^d$  表示第  $m$  个视图的  $d$  维特征向量, 用于捕捉同一数据实例在不同变换下的鲁棒表示, 为跨客户端对比学习提供基础。

数据增强包括随机裁剪(尺寸  $32 \times 32$ )、旋转( $\pm 30^\circ$ )、颜色抖动(亮度/对比度调整)、水平翻转(50% 概率)及高斯噪声注入, 每样本生成 4 个多样化视图(随机组合 2~3 种增强), 可以在避免失真的同时, 通过多视图对比增强方向鲁棒性、抗干扰能力及图像语义捕捉精度。

在全局阶段系统聚合客户端多视图特征表示, 构建对比学习空间中的正负样本对。正样本为同源多视图特征(同客

户端数据实例), 负样本为异源特征(跨客户端或不同实例)。通过汇聚所有客户端的多视图特征, 构建出对比学习空间:

$$\mathcal{Z} = \bigcup_{c=1}^C \bigcup_{i=1}^{N_c} \bigcup_{m=1}^M \{z_{i,m}^{(c)}\} \quad (7)$$

而对比损失函数采用余弦相似度与温度缩放:

$$\mathcal{L}_{contrast} = -\sum_{z_{i,m}^{(c)}} \log \frac{\exp(s(z_{i,m}^{(c)}, z_{i,m}^{(c)}/\tau))}{\sum_{z' \in \{z_{i,m}^{(c)}, z_{i,m}^{(c)'}\}} \exp(s(z_{i,m}^{(c)}, z')/\tau)} \quad (8)$$

公式中参数说明如下:  $z_{i,m}^{(c)}$  是与  $z_{i,m}^{(c)}$  配对的正样本,  $s(\cdot, \cdot)$  为余弦相似度函数,  $\tau$  为温度参数, 用于调节分布平滑度。通过最小化对比损失  $\mathcal{L}_{contrast}$ , 模型强化正样本间的相似性, 抑制负样本干扰, 从而提炼出高判别性特征。

在完成对比学习后, 客户端  $c$  基于学习到的特征分布, 对特征子空间进行筛选, 本文采用注意力机制对特征维度加权, 滤除噪声:

$$\mathcal{Z}_{teacher} = \mathcal{Z}_{select}(\{z_{i,m}^{(c)}\}) = \sum_{d=1}^D w_d \cdot z_{i,m,d}^{(c)} \quad (9)$$

其中  $w_d \propto \text{Attention}(z_{i,m,d}^{(c)})$ , 权重  $w_d$  反映特征维度  $d$  的重要性。通过特征筛选, 客户端保留关键特征子集以促进学生知识吸收, 优化特征表达能力, 使客户端  $c$  本地模型能吸收其他客户端(如  $c'$ ) 的同任务判别特征并过滤跨任务无关信息, 实现高质量跨客户端知识选择。

为了将筛选后的跨客户端知识迁移至目标客户端模型, 需要从其他客户端选择高相似度教师模型集合  $\{\theta_{c'}^{(t-1)}\}_{c' \in S}$ , 并进一步对教师模型的输出 logits 进行概率重采样与对数概率集合操作, 形成集合对数概率:

$$l_{ensemble}(x_i^{(c)}) = \frac{1}{|S|} \sum_{c' \in S} f_{c'}(x_i^{(c)}; \theta_{c'}^{(t-1)}) \quad (10)$$

其中,  $S$  表示基于任务相似度筛选的客户端集合。集合对数概率通过多教师信息融合, 进一步增强鲁棒性和泛化性。

对于学生模型(即待更新的客户端  $c$  模型)参数  $\theta_c^{(t)}$ , 通过 KL 散度损失, 驱动本地学生模型  $\theta_c^{(t)}$  拟合集成知识:

$$\mathcal{L}_{KD}^{(c)} = \sum_{x_i^{(c)} \in \mathcal{D}_c} \text{KL}(\text{softmax}(l_{ensemble}(x_i^{(c)})/\tau) \parallel \text{softmax}(f_c(x_i^{(c)}; \theta_c^{(t)})/\tau)) \quad (11)$$

其中,  $\text{softmax}(\cdot/\tau)$  表示温度软化后的概率分布,  $\text{KL}(p \parallel q)$  表示 KL 散度, 用于衡量分布  $p$  与  $q$  的差异。此时模型已通过多视图对比蒸馏策略从其余客户端中提炼出稳定、有判别力的知识。

内循环中学生模型通过最小化优化目标  $\mathcal{L}_{local}^{(c)} = \mathcal{L}_{task}^{(c)} + \alpha \mathcal{L}_{contrast}^{(c)} + \beta \mathcal{L}_{KD}^{(c)}$  来实现任务适配、防止灾难性遗忘及跨客户端知识筛选, 使其在学习新任务时同步重现历史任务, 选择性共享和吸收不同客户端知识。通过对比知识分布筛选有用信息, 避免简单聚合客户端知识产生的噪声, 从而提升学习效率与性能。

### 3.3 外循环

在动态任务流算力网络场景中, 模型需持续适应新客户端变化, 同时保留已有客户端知识。为此, 外循环采用元更新策略, 选择性地融合新旧客户端参数, 确保全局模型既适应新特性又保留旧知识。

#### 3.3.1 动态梯度方向调整原理

动态元学习 DML (Dynamic Meta-Learning) 外循环参数

迭代过程如图3所示:新客户端  $C_i$  和旧客户端  $C_j$  在特征空间中分别对应两个最优解. 通过外循环迭代, DML 将特征参数逐步移至两者间的最优邻点, 平滑传递新旧客户端知识, 避免模型因客户端交替出现性能振荡, 同时维持对新旧任务的知识掌握能力, 实现动态客户端加入或离线时全局模型的持续优化与平稳适应.

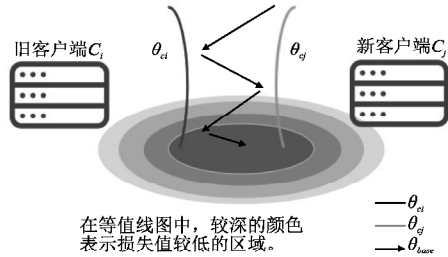


图3 外循环元更新规则

Fig. 3 Outer-loop meta-update rules

外循环以全局视角平衡知识融合, 初始全局参数为  $\theta_{base}$ , 学习新任务  $T_i$  之后客户端  $c$  生成本地参数  $\theta_c^{(i)}$ . 外循环将参数更新  $\theta_{base} - \theta_c^{(i)}$  视为外循环中的梯度下降方向, 使得模型从  $\theta_{base}$  向所有客户端特定参数的平均方向移动, 从而在外循环中获得更优的梯度更新.

### 3.3.2 联邦动态元学习流程

以上述原理为基础, 为适应算力网络的增量学习场景, 在前期训练中, 模型应快速适应新任务, 收敛至优良参数集; 在后期训练阶段, 模型不宜再进行过于频繁的大幅度调整, 以避免性能振荡. 为此, 本文引入基于指数衰减调控的策略  $\gamma(t) = \exp(-\alpha \cdot \frac{t}{T})$ ,  $\alpha \geq 0$  (式中  $\alpha$  表示衰减系数,  $t$  为当前任务索引,  $T$  表示总任务数). 它可以根据任务的总数进行调整, 默认值为  $\alpha = 1$ . 控制器确保模型从一开始的快速学习逐渐过渡到后期的慢速更新. 因此, 全局参数更新公式写为:

$$\theta_{global}^{(t)} = (1 - \gamma(t)) \cdot \theta_{base} + \gamma(t) \cdot \frac{\sum_{c=1}^C w_c(t) \cdot \theta_c^{(t)}}{\sum_{c=1}^C w_c(t)} \quad (12)$$

其中,  $\theta_{base}$  表示全局基础参数, 作为元学习的起点,  $\theta_c^{(t)}$  表示客户端  $c$  在第  $t$  轮训练后的本地参数,  $C$  表示参与更新的客户端数量,  $w_c(t)$  表示客户端  $c$  的动态权重, 基于其元损失  $\mathcal{L}_{meta}^{(c)}$  计算. 元损失  $\mathcal{L}_{meta}^{(c)}$  用于反映客户端在当前任务上的泛化性能, 性能越优 (损失越低) 的客户端权重越高, 元损失  $\mathcal{L}_{meta}^{(c)}$  将在下文流程中阐述.

首先, 客户端  $c$  基于本地验证集  $D_c^{val}$  计算元损失, 其中仅包含任务损失:

$$\mathcal{L}_{meta}^{(c)} = \frac{1}{|D_c^{val}|} \sum_{(x,y) \in D_c^{val}} -\log_{p_{\theta_c^{(t)}}(y|x)} \quad (13)$$

公式中  $D_c^{val}$  表示客户端  $c$  的本地验证数据集,  $p_{\theta_c^{(t)}}(y|x)$  表示本地模型对样本  $x$  的预测概率.

然后, 服务器端为低元损失客户端赋予更高权重, 根据客户端元损失计算其动态权重, 并叠加衰减系数:

$$w_c(t) = \gamma(t) \cdot \frac{\exp(-\mathcal{L}_{meta}^{(c)}/\tau)}{\sum_{c'=1}^C \exp(-\mathcal{L}_{meta}^{(c')}/\tau)} \quad (14)$$

其中,  $\tau$  表示温度系数,  $\gamma(t)$  为上文的动态衰减系数, 随时间降低新客户端的初始影响力.

最后, 服务端按权重聚合本地参数, 通过公式 (12) 更新全局模型, 并将  $\theta_{global}^{(t)}$  下发至所有客户端, 作为下一轮本地训练的初始参数.

本文在外循环中引入动态元学习思想, 其核心优势包括:

1) 全局稳定性: 通过整体视角优化参数, 平衡新旧客户端性能, 提升泛化能力; 2) 灵活同步: 动态调整同步策略, 降低新客户加入对旧知识的冲击, 避免模型失衡.

### 3.4 算法流程与描述

设计联邦增量元蒸馏算法如下所示:

#### 算法1. 联邦增量元蒸馏算法 (FedIMD)

输入: 算力节点 (客户端) 集合  $C = \{c_1, c_2, \dots, c_n\}$ , 任务序列  $T_c = \{T_c^{(1)}, T_c^{(2)}, \dots, T_c^{(T)}\}$ , 初始全局参数  $\theta_{global}^{(0)}$ , 客户端本地数据  $D_c$ ,  $K$  为全局通信轮数;

输出: 优化后全局模型参数  $\theta_{global}^{(K)}$ , 客户端本地优化后模型 ( $\theta_{c_1}^{(T,K)}, \theta_{c_2}^{(T,K)}, \dots, \theta_{c_n}^{(T,K)}$ );

1. 初始化: 向各客户端分发全局模型参数  $\theta_{global}^{(0)}$ ;
2. While  $k \leq K$  do;
3. For  $c$  in  $C$  do;
4. For  $t$  in  $T_c$  do;
5. 内循环:
6. 客户端  $c$  加载当前任务数据  $D_c^{(t)}$  及历史记忆缓存  $\epsilon_{1:t-1}$ ;
7. 对样本  $x_i^{(c)}$  进行数据增强, 生成多视角特征  $\{x_{i,m}^{(c)}\}_{m=1}^M$ ;
8. 计算交叉熵损失  $\mathcal{L}_{task}^{(c,t)}$  和条件蒸馏损失  $\mathcal{L}_{forget}^{(c,t)}$ ;
9. 提取多视角特征  $z_{i,m}^{(c)}$  计算对比损失  $\mathcal{L}_{contrast}^{(c,t)}$ ;
10. 更新本地参数;
11. End for;
12. End for;
13. 外循环:
14. 各客户端上传本地参数并计算元损失  $\mathcal{L}_{meta}^{(c)}$ ;
15. 按公式 (14) 计算客户端权重;
16. 聚合各客户端参数按公式 (12) 更新  $\theta_{global}^{(n)}$ ;
17. End;
18. 返回全局模型参数  $\theta_{global}^{(K)}$ , 优化后各客户端模型参数.

以上算法时间复杂度为  $O(K \times [C \times T \times (M + D) + P])$ , 其中,  $M$  和  $D$  分别表示数据增强倍数以及蒸馏损失计算量,  $P$  表示参数传输量. 相较于传统联邦持续学习算法  $O(K \times [C \times T \times D + P])$  的时间复杂度, 本方法引入的多视角数据增强模块和历史蒸馏模块可并行计算优化, 实际计算时间增量控制在较低范围内. 同时算法设计通过仅修改本地损失函数, 能够保持模型聚合协议基本与 FedAvg 一致, 使得系统能够轻易集成到主流联邦学习框架中.

## 4 实验

本文实验旨在验证联邦持续学习框架 FedIMD 在算力网络动态任务流场景下的核心能力:

- 1) 抑制灾难性遗忘: 通过历史知识蒸馏与对比学习机制, 保留旧任务知识;
- 2) 动态适应性: 应对客户端动态加入/退出, 维持全局模型的稳定性与泛化性.

## 4.1 实验配置

### 4.1.1 数据集

本文选用两个常用数据集 CIFAR100 与 FMNIST<sup>[22]</sup> 作为基准,构建如下 4 个不同设置的联邦学习场景验证 FedIMD:

1) Overlapping CIFAR100: 将 50,000 样本划分为 10 个逐步增加类别重叠的任务,客户端数据按 Dirichlet = 0.5 (控制数据分布的异构程度,值越小客户端数据越集中在极少数类别)分配,测试旧任务记忆能力;

2) CIFAR100-shuffle: 基本设置与 Overlapping CIFAR100 相同,但每个客户端的任务顺序独立随机生成(如 Client1: Task3 → Task1 → Task5 → ⋯, Client2: Task5 → Task2 → Task4 → ⋯). 每个客户端的任务顺序独立随机生成,用于验证模型对任务顺序变化的鲁棒性;

3) FMNIST-CPTO (Client Partial Task Overlap): 划分为 6 个任务,每个客户端动态分配 1~3 个 Dirichlet = 0.5 的重叠任务与 1~3 个独立任务(阶梯式均匀分布),验证动态知识迁移与任务冲突鲁棒性;

4) FMNIST-NonIID: 采用 Dirichlet = 0.1 生成高度异构数据(部分客户端仅含 2~3 个类别),测试极端分布适应性.

### 4.1.2 模型架构及实验环境

本文实验对 FMNIST 和 CIFAR100 数据集分别选用改进版 LeNet 和 ResNet-18 网络架构:前者通过双卷积层(20/50 通道,5×5 核)与局部响应归一化层提取图像特征,后接 800/500 单元的全连接层;后者调整残差模块适配 CIFAR100 任务,使用 SGD 优化器(动量 0.9,学习率 0.01)微调网络,最终通过 Softmax 分类器实现多类别判别. 实验环境部署于 Ubuntu 18.04.6 系统,硬件采用 Intel Xeon Gold 5220R 处理器(64GB 内存)和 RTX 3090 显卡,基于 Python3 与 PyTorch 框架构建联邦学习系统,设置 10 个客户端执行 20 轮全局通信,每轮随机选取客户端进行 3 次本地迭代训练.

本文实验对比的基线方法为:1) FedAvg<sup>[5]</sup>: 联邦学习的经典算法 FedAvg; 2) FedProx<sup>[24]</sup>: 是一种在联邦学习中引入正则化项的算法,用于解决数据异质性问题; 3) AF-FCL<sup>[19]</sup>: 基于对比学习的自适应联邦学习算法,优化了客户端间的表示学习,增强了模型泛化能力; 4) FedProx-EWC<sup>[18]</sup>: 结合了 FedProx 算法与弹性权重固定的方法,用于处理联邦学习中的长期学习和灾难性遗忘问题; 5) FedWeIT<sup>[18]</sup>: 通过在模型权重上进行分布式的增量更新,优化了联邦学习中的通信和计算效率; 6) GLFC<sup>[21]</sup>: 联邦类增量算法; 7) FedMGP<sup>[23]</sup>: 多任务联邦学习算法,支持多个任务并行学习,通过共享梯度信息提高多任务的学习性能; 8) FedCIL<sup>[25]</sup>: 扩展了 ACGAN- Replay 方法到联邦场景中; 9) 本文算法 FedIMD.

## 4.2 实验结果分析

本文首先在 CIFAR100 任务序列上验证了所提出模型的有效性,并与多种基线方法进行了对比,包括多个先进的联邦持续学习方法、传统联邦学习方法(FedAvg、FedProx)以及基于 FedProx 的简单联邦持续学习方法. 表 2 展示了在 Overlapping CIFAR100 以及 CIFAR100-shuffle 数据集上完成联邦持续学习后的平均任务性能. 完成第  $t$  个任务后所有历史任务的平均测试准确率公式为:

$$A_t = \frac{1}{t} \sum_{i=1}^t a_{t,i} \quad (15)$$

其中,  $a_{t,i}$  表示在完成第  $t$  个任务后,任务  $i$  的测试准确率.

表 2 两种结构 CIFAR100 数据集上的准确率及遗忘率对比  
Table 2 Accuracy and forgetting rate comparison of two architectures on CIFAR100 dataset

方法	Overlapping CIFAR100		CIFAR100-shuffle	
	准确率	遗忘率	准确率	遗忘率
FedAvg	25.5%	20.3%	24.8%	21.1%
FedProx	26.7%	18.9%	26.1%	19.5%
FedProx-EWC	32.4%	13.6%	31.8%	16.2%
FedMGP	37.1%	13.5%	33.5%	16.0%
FedWeIT	34.9%	12.1%	32.2%	14.8%
FedCIL	36.3%	10.5%	35.7%	13.2%
AF-FCL	38.5%	11.3%	37.9%	12.9%
GLFC	35.8%	9.8%	34.1%	12.2%
Ours w/o HKR	40.1%	14.8%	39.4%	17.9%
Ours w/o MVCD	38.7%	12.5%	37.0%	13.1%
Ours w/o FDML	40.9%	7.2%	39.2%	7.5%
FedIMD	<b>43.2%</b>	<b>6.7%</b>	<b>42.1%</b>	<b>7.1%</b>

在计算平均遗忘率时,本文使用加权平均的方法,平均遗忘率指标用于评估连续学习过程中向后迁移的程度,其通过每个任务的最高精度与最终精度之间的差异来量化,公式如下:

$$AF = \frac{1}{\sum_{k=1}^N \sum_{i=1}^{T-1} n_k^i} \sum_{k=1}^N \sum_{i=1}^{T-1} \max_{t \in \{1, \dots, T-1\}} (a_k^{t,i} - a_k^{T,i} \times n_k^i) \quad (16)$$

公式中,  $N$  表示客户端或设备的总数,  $T$  表示总任务数,  $a_k^{t,i}$  表示客户端  $k$  在完成第  $t$  个任务后,对任务  $i$  的测试准确率,  $n_k^i$  表示客户端  $k$  的样本数量,作为加权因子.

### 4.2.1 模型遗忘率及准确率对比

表 2 结果表明,使用了联邦持续学习的方法在重叠任务中性能明显低于未采用联邦持续学习的相应方法,这主要是因为客户端参数的聚合过程,导致了对各个任务学习的严重干扰,从而引发了灾难性遗忘现象. 尽管 FedCIL 和 AF-FCL 等方法通过引入参数间的曲率信息或选择性生成回放方法在一定程度上缓解了任务间的差异,但仍未能有效最小化任务间的干扰. 相比之下, FedIMD 在两个数据集上的表现均优于其他 FCL 基线方法,在 Overlapping CIFAR100 上的平均遗忘率仅为 6.7%,显著低于 FedAvg (20.3%), 低于 FedCIL (10.5%)、AF-FCL (11.3%) 以及 GLFC (9.8%), 同时准确率较排名第 2 的 AF-FCL 提高了 4.7%.

图 4 对比了 80 轮全局通信下从 Task1 与 Task8 开始训练的各任务准确率曲线(每个任务 20 轮),可以看出, FedIMD 显著抑制遗忘,而其它联邦学习基线均在任务切换时产生较严重灾难性遗忘. 这一性能提升源于 FedIMD 通过内循环的双重机制筛选跨客户端知识,实现旧任务知识的长效保持与迁移,其对比学习约束抑制了模型更新的灾难性遗忘,从而提升最终性能.

而在新旧任务交替出现的复杂场景下,如 CIFAR100-shuffle 数据集下, FedIMD 依旧表现出了很好的稳定性,遗忘

率提高了0.4%, 优于 FedCIL(2.7%)、AF-FCL(1.6%) 以及 GLFC(2.4%), 可见 FedIMD 能够平衡新任务的高效适应与旧任务的知识保留, 进一步增强了模型在分布式环境下的适应能力、泛化性能以及任务切换的稳定性. 此外, 尽管 AF-FCL 等方法在一定程度上避免了灾难性遗忘, 但由于其知识转移机制的局限性, 整体性能仍不及 FedIMD. 这表明, FedIMD 在保持知识连续性的同时, 能够更好地适应多任务环境, 显著提升了模型的稳定性和记忆保持能力.

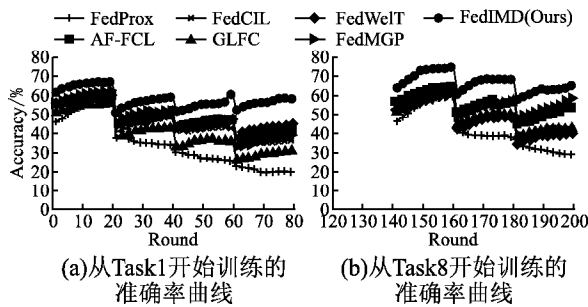


图4 在 Overlapping CIFAR100 数据集上从 Task1、Task8 开始训练的遗忘率对比

Fig. 4 Forgetting rate comparison between training from Task1 and Task8 on Overlapping CIFAR100

为验证 FedIMD 在客户端间知识转移中的有效性, 本文使用 FMNIST-CPTO 数据集来将部分重叠任务划分到不同的客户端进而模拟训练过程中需要共享端间知识的情况. 利用该数据集, 可以通过观察模型在一个客户端上学习到的知识在其他客户端上的表现, 评估算法在客户端任务间知识迁移的效率. 如表3所示, FedIMD 在 FMNIST-CPTO 数据集上准确率达到56.2%, 优于同样拥有端间知识迁移机制的 FedWeIT(52.3%) 以及其它基线.

表3 两种结构 FMNIST 数据集上的准确率及遗忘率对比

Table 3 Accuracy and forgetting rate comparison of two architectures on FMNIST dataset

方法	Overlapping CIFAR100		CIFAR100-shuffle	
	准确率	遗忘率	准确率	遗忘率
FedAvg	43.5%	17.3%	25.1%	19.8%
FedProx	45.1%	15.9%	27.7%	18.2%
FedProx-EWC	48.4%	12.7%	35.6%	15.3%
FedMGP	49.8%	14.2%	36.9%	16.8%
FedWeIT	52.3%	11.9%	36.5%	14.5%
FedCIL	50.4%	9.4%	38.2%	13.1%
AF-FCL	52.1%	12.5%	42.8%	15.6%
GLFC	50.8%	9.7%	42.3%	13.9%
Ours w/o HKR	54.5%	15.8%	51.2%	17.3%
Ours w/o MVCD	50.1%	11.2%	45.7%	13.7%
Ours w/o FDML	52.9%	6.5%	49.4%	8.3%
FedIMD	<b>56.2%</b>	<b>6.2%</b>	<b>52.5%</b>	<b>8.1%</b>

这是由于多视图对比蒸馏通过利用多个视角进行学习, 能够增强模型的代表能力和泛化能力, 从而提高其鲁棒性和适应性. 相比普通蒸馏, 它更有效地减少数据偏差的影响, 并加速知识迁移, 使得模型在面对复杂和多样的数据时表现更好. 同时在表3关于 FMNIST-NonIID 数据集的准确率对比实验中, FedIMD 的表现更佳, 相较于排名第2的 AF-FCL 方法提升了9.7%. 这是由于更精确的知识迁移还同时提高了 FedIMD 在异构数据环境下的适应能力, 进一步验证了 FedIMD 在实际联邦持续学习场景中的实用性和优势.

#### 4.2.2 客户端动态变化下模型性能对比

本小节评估了新增客户端对模型性能的影响, 通过模拟客户端退出替换和客户端新增两种情况以验证 FedIMD 在动态环境中的适应能力. 通过分析实验结果, 如图5所示, 在20

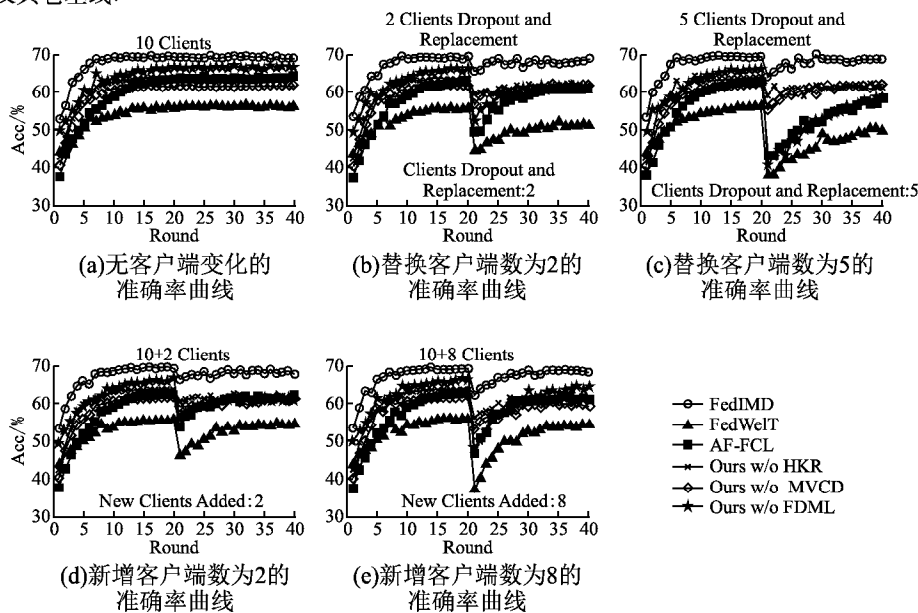


图5 20轮时替换新增客户端对 Task5 的训练性能影响

Fig. 5 Impact of client replacement and addition at round 20 on Task5 training performance

轮全局通信时, FedIMD 在动态增加2个和8个客户端的情况下, 对 Task5 的训练性能展现出显著优势. 新增5个客户端

后, FedIMD 在 Task5 上的准确率仅下降6.1%, 显著低于 FedWeIT(13.6%) 和 AF-FCL(14.2%), 同时收敛速度相较

两个基线提高约 30%。且随着新增客户端数量, FedIMD 动态适应性优势更显著。

而对于动态客户端替换对模型训练的影响, 本文通过设置不同规模的客户端替换实验(替换比例 20% 及 50%) 验证算法鲁棒性。如图 5 所示, 当系统执行小规模替换(替换客户端数  $N=2$ ) 时, FedIMD 的收敛轮次为 5 轮, 显著低于 FedWeIT(13 轮) 和 AF-FCL(10 轮)。而在极端替换场景( $N=5$ ) 下, FedIMD 仍能在 10 轮内完成收敛, 而 FedWeIT 和 AF-FCL 在第 40 轮时仍表现出明显的收敛障碍。这说明通过多视图知识蒸馏和动态知识选择机制, FedIMD 能够高效地从新增客户端中提取有价值的知识, 同时保持对已有客户端知识的稳定整合。此外, FedIMD 引入的元更新优化策略进一步提升了模型的全局适应能力, 使得全局模型在应对动态客户端加入时能够更快速地调整学习方向, 从而显著缩短收敛时间并提高任务精度。与其他基线方法相比, FedIMD 在 Task5 上的训练性能均表现出更强的鲁棒性、快速适应性和更高的精度, 充分证明了其在知识转移效率和优化策略上的优越性。

#### 4.2.3 消融实验

为验证增量元蒸馏框架各模块的有效性, 本文在相同实验环境下设计了 3 组消融实验, 分别移除历史知识重现模块(HKR)、多视图对比蒸馏模块(MVCD)以及联邦动态元学习策略(FDML)。如表 2、表 3 所示, 完整框架 FedIMD 在各项指标上均达到最优性能。其中, 当移除 HKR 模块后, 模型在 Overlapping CIFAR100 数据集上的平均遗忘率从 6.7% 增至 14.8% (增幅达 8.1%), 这表明 HKR 模块通过知识蒸馏机制有效约束了参数更新方向, 其历史知识重现策略能显著缓解灾难性遗忘。而移除 MVCD 模块后导致模型在各个数据集上准确率平均降低 5.4%, 对准确率影响最为明显, 同时在 FMNIST-NonIID 数据集上的遗忘率增加 5.6% (从 8.1% 升至 13.7%)。这证实了多视图对比蒸馏模块的关键作用, 即通过多维度特征空间的对比学习, 模型能够提取更具判别性的跨客户端知识。特别地, 在数据异构最严重的 FMNIST-NonIID 场景中, 移除 MVCD 使模型性能降低 13%, 降幅最大, 说明多视图机制能有效增强异构数据下的特征鲁棒性。而移除 FDML 模块后, 新增 5 个客户端时系统收敛所需轮数从 20 轮延长至 30 轮(增幅 50%), 且 Task5 准确率波动幅度扩大 37%。这体现了 FDML 的双重优势: 1) 基于元损失的动态权重分配优化了知识融合过程, 使全局模型快速适应新节点特性; 2) 指数衰减调控策略平衡了新旧客户端影响力, 在各数据集上将显著降低系统扰动, 如图 5 所示, 完整框架在系统动态扩展场景下展现出更平滑的收敛曲线。下面对模块间的协同互补效果进行分析。

MVCD 对 HKR 的补强作用: 在移除 FDML 与移除 MVCD 的对比中, 前者在 4 种数据集上的平均遗忘率降低 5.25% (如 CIFAR100-shuffle 从 13.1% 降至 7.5%)。这表明, MVCD 通过多视图特征对比筛选跨客户端共性知识, 弥补了 HKR 仅依赖输出分布约束的局限性, 从特征空间层面强化知识保留能力, 从而减少任务冲突导致的遗忘。

动态调度下的协同适应性: 在客户端动态新增/替换场景中, 移除 HKR 相比移除 MVCD 的收敛速度提升 20%。这一提升表明了 MVCD 与 FDML 的深度协同, MVCD 通过高质

量特征迁移降低异构客户端间的知识差异, 而 FDML 基于动态元更新快速适配新增节点特性, 二者共同优化了动态环境下的模型稳定性与收敛效率。

以上实验表明, FedIMD 联邦学习框架在处理客户端动态变化时具备高度的鲁棒性和适应性, 即便在客户端数量大幅变动的情况下, 依然能够维持有效的学习效率和稳定的性能表现。通过多视图知识蒸馏和动态知识选择机制, FedIMD 在客户端间实现了高效的知识共享, 使新增客户端能够快速融入全局学习过程, 同时避免对已有客户端任务造成干扰。此外, FedIMD 通过元更新优化策略和知识保留机制, 有效缓解了灾难性遗忘问题, 即便在动态客户端增删或异构数据分布的场景下, 依然能够保持对旧任务知识的长期记忆。这不仅验证了 FedIMD 在分布式环境下的优越性, 也为其在实际算力网络应用中的大规模部署提供了坚实的理论和实验支持。

## 5 结束语

本文针对算力网络中联邦持续学习面临的灾难性遗忘、知识共享效率低以及节点动态管理带来的系统扰动等挑战, 提出了一种基于增量元蒸馏的联邦持续学习框架(FedIMD)。通过引入双层蒸馏优化策略, 在本地与全局层面实现了知识的有序累积与共享, 有效缓解了任务间的干扰与遗忘问题。实验结果表明, 与现有的联邦持续学习方法相比, FedIMD 在多个非独立同分布任务环境中显著提升了任务适应性与稳定性, 能够更好的克服持续任务输入下的灾难性遗忘问题。特别是在客户端动态管理的情况下, FedIMD 仍能有效缓解因新增客户端带来的系统扰动。本文方法在验证阶段聚焦于图像分类任务, 尚未拓展至文本、时序数据等场景。针对非图像模态需重构特征生成机制, 同时客户端动态性实验假设局限于随机增删模式, 未涵盖实际算力网络中可能存在的分层联邦或区域化集群动态, 跨集群知识迁移稳定性仍需复杂拓扑场景下深入验证。在大规模算力网络的可扩展性方面, 本文方法通过知识蒸馏驱动模型瘦身技术有效适配低算力节点, 维持算力网络兼容性并降低计算开销。同时针对超大规模客户端管理, 设计元更新策略, 新增节点可基于跨任务泛化参数快速适配, 依据算力类型与带宽资源动态调整本地训练轮次, 从而实现异构资源的弹性适配, 支撑大规模算力网络部署。未来的研究可以进一步探索 FedIMD 在更复杂场景中的扩展性与适应性, 以提升动态任务调度的计算效率和鲁棒性, 实现异构网络中的实时任务处理。此外, 可进一步研究高效的参数传输和梯度压缩方法, 以减少通信开销。

## References:

- [1] Sun Y, Lei B, Liu J, et al. Computing power network: a survey[J]. China Communications, 2024, 21(1): 109-145.
- [2] Li L, Fan Y, Tse M, et al. A review of applications in federated learning[J]. Computers & Industrial Engineering, 2020, 149(1): 106854, doi:10.1016/j.cie.2020.106854.
- [3] Wen J, Zhang Z, Lan Y, et al. A survey on federated learning: challenges and applications[J]. International Journal of Machine Learning and Cybernetics, 2023, 14(2): 513-535.
- [4] Zhu H, Xu J, Liu S, et al. Federated learning on non-IID data: a survey[J]. Neurocomputing, 2021, 465(1): 371-390.

- [ 5 ] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data [ C ] // Artificial Intelligence and Statistics (PMLR), 2017:1273-1282.
- [ 6 ] Li Q, Diao Y, Chen Q, et al. Federated learning on non-iid data silos: an experimental study [ C ] // IEEE 38th International Conference on Data Engineering (ICDE), 2022:965-978.
- [ 7 ] Liu Y, Kang Y, Zou T, et al. Vertical federated learning: concepts, advances, and challenges [ J ]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7):3615-3634.
- [ 8 ] Ye M, Fang X, Du B, et al. Heterogeneous federated learning: state-of-the-art and research challenges [ J ]. ACM Computing Surveys, 2023, 56(3):1-44.
- [ 9 ] Xie L, Lin M, Luan T, et al. MH-pFLID: model heterogeneous personalized federated learning via injection and distillation for medical data analysis [ J ]. arXiv preprint arXiv:2405.06822, 2024.
- [ 10 ] Han S, Park S, Wu F, et al. Fedx: unsupervised federated learning with cross knowledge distillation [ C ] // European Conference on Computer Vision (ECCV), 2022:691-707.
- [ 11 ] Lin T, Kong L, Stich S U, et al. Ensemble distillation for robust model fusion in federated learning [ J ]. Advances in Neural Information Processing Systems, 2020, 33(1):2351-2363.
- [ 12 ] Ma X, Zhu J, Lin Z, et al. A state-of-the-art survey on solving non-IID data in federated learning [ J ]. Future Generation Computer Systems, 2022, 135(1):244-258.
- [ 13 ] Charles Z, Konečný J. Convergence and accuracy trade-offs in federated learning and meta-learning [ C ] // International Conference on Artificial Intelligence and Statistics (PMLR), 2021:2575-2583.
- [ 14 ] Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach [ J ]. Advances in Neural Information Processing Systems, 2020, 33(1):3557-3568.
- [ 15 ] Yang L, Huang J, Lin W, et al. Personalized federated learning on non-IID data via group-based meta-learning [ J ]. ACM Transactions on Knowledge Discovery from Data, 2023, 17(4):1-20.
- [ 16 ] Liu P, Yu X, Zhou J T. Meta knowledge condensation for federated learning [ J ]. arXiv preprint arXiv:2209.14851, 2022.
- [ 17 ] Criado M F, Casado F E, Iglesias R, et al. Non-iid data and continual learning processes in federated learning: a long road ahead [ J ]. Information Fusion, 2022, 88(1):263-280.
- [ 18 ] Yoon J, Jeong W, Lee G, et al. Federated continual learning with weighted inter-client transfer [ C ] // International Conference on Machine Learning (PMLR), 2021:12073-12086.
- [ 19 ] Wuerkaixi A, Cui S, Zhang J, et al. Accurate forgetting for heterogeneous federated continual learning [ C ] // 12th International Conference on Learning Representations (ICLR), 2024:174-180.
- [ 20 ] Chung S, Al Kontar R. Federated multi-output Gaussian processes [ J ]. Technometrics, 2024, 66(1):90-103.
- [ 21 ] Dong J, Wang L, Fang Z, et al. Federated class-incremental learning [ C ] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022:10164-10173.
- [ 22 ] Xiao H. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms [ J ]. arXiv preprint arXiv:1708.07747, 2017.
- [ 23 ] Yu H, Yang X, Gao X, et al. Personalized federated continual learning via multi-granularity prompt [ C ] // Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024:4023-4034.
- [ 24 ] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks [ J ]. Proceedings of Machine Learning and Systems, 2020, 2(1):429-450.
- [ 25 ] Li Y, Xu W, Wang H, et al. Personalized federated domain-incremental learning based on adaptive knowledge matching [ C ] // European Conference on Computer Vision (ECCV), 2024:127-144.