

异构边缘环境下自适应分层联邦学习协同优化方法

冯奕铭 钱 珍 李光辉 代成龙

(江南大学人工智能与计算机学院 江苏无锡 214122)

(richardvon@stu.jiangnan.edu.cn)

Synergistic Optimization Method for Adaptive Hierarchical Federated Learning in Heterogeneous Edge Environments

Feng Yiming, Qian Zhen, Li Guanghui, and Dai Chenglong

(School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu 214122)

Abstract Traditional hierarchical federated learning (HFL) encounters significant challenges in real world due to device heterogeneity, data heterogeneity (e.g., variations in data volume and feature distribution), and communication resource constraints. Device heterogeneity results in inefficient cross-device collaboration during model training, whereas data heterogeneity induces accuracy degradation and diminished generalization capabilities in the global model. To address these limitations while maximizing the utilization of computation, communication, and data resources in the heterogeneous edge networks, we propose an adaptive synergistic method for hierarchical federated learning. This method synergistically integrates model partitioning and client selection under hardware resource constraints, communication bottlenecks, and non-independent and identically distributed (Non-IID) data conditions to accelerate federated learning training while enhancing model accuracy and adaptability across heterogeneous environments. To quantify the influence of local datasets on global model convergence, a data contribution metric is introduced to evaluate the consistency of client contributions. Furthermore, by integrating deep reinforcement learning (DRL) with real-time resource monitoring and data contribution quantification, the DRL agent dynamically optimizes client selection and edge-cloud collaborative model partitioning strategies prior to each training iteration. This adaptive mechanism leverages system resource availability (e.g., bandwidth, device status) and local data contribution scores to derive optimal policies, thereby accelerating training convergence and enhancing global model accuracy. Simulation results demonstrate that the proposed method achieves significant improvements in model accuracy and training efficiency compared with baseline methods, while exhibiting robust adaptability across diverse heterogeneous environment configurations.

Key words hierarchical federated learning; heterogeneous edge computing; model partitioning; client selection; deep reinforcement learning

摘要 传统联邦学习在应用中面临设备异构、数据异构、通信资源约束等挑战。终端设备异构导致训练过程中过低的协作效率，而数据异构所包括的数据量和数据特征分布异构则导致全局模型精度损失以及模型缺少泛化性。为了有效利用终端的计算、通信以及数据资源，提出了一种自适应优化的分层联邦学习方法。该方法在考虑设备硬件资源约束、通信资源约束以及数据非独立同分布（Non-IID）特性下，结合模

收稿日期：2025-03-01；修回日期：2025-04-10

基金项目：国家自然科学基金项目（62372214）

This work was supported by the National Natural Science Foundation of China (62372214).

通信作者：李光辉 (ghli@jiangnan.edu.cn)

型分割和客户端选择技术加速联邦学习训练,提高模型准确率以及其在不同异构环境下的适应性。为了反映各客户端数据对全局模型的一致性影响,引入数据贡献度以度量本地模型对全局模型的影响。通过深度强化学习方法,在每一轮训练前智能体根据系统的资源分布以及本地数据贡献度来学习如何选择合理的训练客户端集合及相应边端协同模型划分方案,以加速本地训练及全局模型收敛。仿真结果表明,与基线方法相比,所提算法在模型准确率与训练效率2个方面均表现出显著优势,且在不同异构环境配置下显示出良好的鲁棒性及适应性。

关键词 分层联邦学习; 异构边缘计算; 模型分割; 客户端选择; 深度强化学习

中图法分类号 TP301; TP393

DOI: [10.7544/issn1000-1239.202550146](https://doi.org/10.7544/issn1000-1239.202550146) **CSTR:** [32373.14.issn1000-1239.202550146](https://cstr.cnki.net/32373.14.issn1000-1239.202550146)

随着物联网(IoT)和人工智能(AI)的快速发展,智能设备的数量和种类不断增加,这些设备产生了海量的数据。这些数据不仅包含丰富的信息,还涉及用户的隐私及安全。传统的集中式数据处理方法面临着数据传输成本高、隐私泄露风险大等问题^[1-2]。因此,联邦学习(federated learning, FL)作为一种新兴的分布式机器学习范式,近年来在边缘计算和深度学习等领域得到了广泛关注和应用。联邦学习允许多个设备或客户端在不共享原始数据的情况下,在本地设备上进行模型训练并将模型更新上传至参数服务器进行聚合,避免了数据集中存储和传输,从而在保护用户隐私的同时实现数据的高效利用^[3-6]。然而,尽管联邦学习在隐私保护和数据安全方面具有显著优势,其在实际应用中仍面临诸多挑战,主要包括设备异构性、数据异构性以及通信资源约束等问题^[7]。

首先,设备异构性是联邦学习中的一个主要瓶颈^[8-9]。终端设备(如智能手机、嵌入式设备等)在计算能力、内存和通信资源方面存在显著差异,导致训练过程中的协作效率低下。特别是计算能力较弱的设备会显著拖慢整体训练进度,影响系统的整体性能,这被称为“straggler问题”^[4]。设备异构性不仅导致训练时间的不平衡,还影响全局模型的收敛速度以及最终性能。其次,由于不同设备上的数据分布通常满足非独立同分布(Non-IID),客户端数据集差异导致本地模型的训练质量差异较大,进而影响全局模型的精度和泛化能力^[10-13]。例如,由于不同客户端的数据分布不同,每次全局模型更新时,不同客户端的参数更新方向不一致,甚至相互冲突。这会导致模型在某些方向上过度调整,而在其他方向上调整不足,影响全局模型的收敛和性能。此外,通信资源约束(如带宽限制)也会对联邦学习的训练效率产生负面影响,尤其是在大规模设备参与的情况下,通信开销成为系统的主要瓶颈^[14-16]。在边缘计算环境中,设备与

服务器之间的网络带宽是有限的,频繁的模型更新和传输可能导致网络拥塞,进一步延长聚合等待时间并增加通信开销。特别是在深度神经网络(DNN)等复杂模型的训练过程中,模型参数量庞大,通信开销尤为显著^[13,17]。

为了应对这些挑战,研究者们提出了多种解决方案。分层联邦学习(hierarchical federated learning, HFL)通过在终端和云端之间引入边缘层来减少终端与参数服务器之间的通信开销^[5]。在云边端三层联邦学习架构中,终端设备负责收集数据并进行本地模型训练,边缘服务器负责聚合邻近终端的训练模型,云端服务器负责全局模型的聚合和更新。在此基础上,文献[18]引入客户端选择方法来解决数据和设备异构问题。通过选择数据分布更均匀的客户端,可以减少模型偏差,提高全局模型的泛化能力;而选择计算和通信能力较强的客户端有助于缓解 straggler 问题。然而,客户端的数据质量与硬件资源并不呈正相关^[19-22]。例如,一些物联网设备虽然拥有高质量的数据,但由于资源受限,无法承担复杂模型的训练,这导致了全局模型的精度损失。此外,由于局部聚合和全局聚合的分离,模型的收敛速度和精度也会受到较大的影响,特别在数据分布差异较大的情况下。另外一些研究将分割学习(split learning, SL)与联邦学习相结合^[23]。分割学习通过将深度神经网络分割成多个部分,并在不同的设备上分别训练这些部分,以此解决资源受限设备上模型训练问题。通过引入分割学习,系统可以根据设备的计算能力,动态调整每个设备的训练任务。这种动态调整不仅能够提高训练效率,还能有效减少 straggler 问题的影响。然而,如何确定模型分割策略是一个关键挑战。模型不同的层具有不同的输出大小和计算开销,分割策略不仅需要考虑本地设备的计算资源和带宽资源,还需要考虑到邻近终端对边缘服务器的占用。现有的一些

工作聚焦于动态调整模型分割策略,根据每个设备和服务器的负载情况动态分配任务^[24-27].例如,根据实时的网络条件和设备性能,为每台设备选择不同的模型分割点.虽然高效的动态模型分割策略可以有效减少本地计算以及通信开销,提高系统性能,但并没有解决不同客户端上较大的数据分布差异对全局模型的精度及收敛速度的影响.此外,在大规模分布式环境中,频繁的中间层激活传输和梯度回传会增加通信延迟,导致训练过程中的同步时延过程等问题,影响模型的收敛.

因此,面对设备异构性、数据异构性和通信资源约束等挑战,本文提出了一种联合动态模型分割以及客户端选择的自适应分层联邦学习方法ASSHFL (adaptive split and select hierarchical federated learning).与以往的分层联邦学习方法相比,ASSHFL并不限制每一轮参与训练的终端数量.为了衡量各终端对全局模型的影响,引入客户端贡献度作为终端被选中参与训练的概率,以优化每轮训练中的局部模型更新方向.此外,为了获得全局模型的精度最优解以及最小化模型训练更新时延,提出了基于多线程PPO (proximal policy optimization)的模型划分和客户端选择的自适应优化算法.每轮训练前,智能体根据设备的数据贡献度、当前计算资源和网络状况来决定参与本轮训练的终端集合以及每台设备上的最佳分割模型.具体而言,本文的主要贡献有3点:

1)客户端贡献度.为量化客户端对全局模型的综合价值,构建了计算资源-数据双模态评估方案.通过梯度相似性分析以及资源实时消耗状态选择高贡献的客户端,从而优化全局模型的收敛速度和精度.

2)深度强化学习驱动的自适应优化.提出了一种基于深度强化学习的自适应优化框架.在每一轮训练前,强化学习智能体根据系统的实时资源分布和本地数据贡献度,动态选择参与训练的客户端集合以及相应的本地训练模型层数.

3)实验验证.在多个经典数据集以及不同异构环境下进行了广泛的实验,验证了所提方法的有效性.实验结果表明,与现有的模型分割方法相比,本文所提方法在训练时间、模型精度以及泛化能力方面均取得了显著提升.

1 相关工作

1.1 分层联邦学习

针对传统联邦学习在复杂现实场景中面临数据

异构性、通信效率低等问题,HierFAvg^[5]通过分层聚合平衡全局与本地优化目标,减少跨层级数据分布差异的影响;并且,通过减少客户端与参数服务器之间的通信频次和数据量以降低通信开销.在此基础上,文献[6]根据客户端数据分布与边缘端无线通信带宽约束,优化训练任务分配与资源分配,显著降低通信轮次并提升分类准确率.文献[7]提出的FedCH机制,通过动态分簇与分层聚合,将客户端按计算和通信能力分组,减少Non-IID数据影响.HybridFL^[8]结合边缘层与云层的两级聚合,引入区域松弛因子增强客户端容错能力,并采用异步更新机制实现动态负载均衡,但其对网络带宽波动的鲁棒性仍依赖于历史数据预测模型,实时适应性有待提升.此外,以上研究多假设客户端资源同构,而针对边缘网络设备资源时空异构、通信受限及非独立同分布数据等核心挑战,文献[13]通过动态调整边缘服务器聚合频率和自适应异步加权策略,提升参数聚合效率;并根据模型精度与数据分布特征动态选取客户端,缓解资源异构对训练的影响.文献[14]通过插入早期退出分支实现模型异构,以适配不同客户端的异构资源;并提出半监督学习方法,利用服务器预训练模型为无标签数据生成伪标签,提升数据利用率.然而,文献[13-14]方法均依赖于静态资源假设,在实际场景中,客户端的资源呈现动态变化特性.现有方案多采用固定参数配置或周期性策略调整,在真实分布式场景中难以维持长期稳定的训练性能,需要引入更精细的在线感知与自适应优化机制.

1.2 客户端选择

为了有效解决数据分布不均、通信效率及设备资源约束等挑战,一些研究提出了客户端选择机制,以最大化拥有优质资源客户端参与训练的数量.COCS^[20]提出了一种基于上下文组合多臂强盗的在线客户端选择策略.通过观察客户端的本地计算和传输的辅助信息(上下文)做出客户端选择决策,提高系统训练性能以及减少通信开销.文献[21]提出了基于Shapley值的联邦平均算法ShapleyFL.该算法利用Shapley值来衡量每个客户端对全局模型的贡献,高概率挑选出高贡献的客户端集合.然而,以上研究仅面向数据异构或设备异构,未能同时考虑系统设备异构性和数据异构性.为此,文献[18]利用深度强化学习选择最佳客户端子集.其设计的奖励函数主要包括模型准确率提升、通信延迟以及系统单轮训练时间.策略网络通过最大化累积奖励来学习最优策略.文献[19]根据通信与计算资源进行客户端群组划分,并通过

感知模型精度与数据分布特征,估计客户端群组的组间与组内选取概率,动态选取参与模型训练的客户端,提高训练收敛速度与模型精度.然而,以上研究都没有充分利用所有客户端的计算和数据资源,导致某些资源受限的设备闲置.此外,忽略了在设备数量较多的情况下多层通信带来的额外通信开销.

1.3 分割学习

在资源受限的环境中,完整的模型训练和部署是不切实际的,并且需要快速的模型训练来不断更新数据集和定期更新全局模型. SplitFed^[23]首次结合了联邦学习与分割学习的优势,通过将DNN模型分割并在客户端设备和服务器之间进行分布式训练,实现了计算资源的有效利用和训练效率的提升. PSL^[24]采用了与SplitFed相同的训练过程,但PSL能够自适应地选择每个设备的迷你批量大小.这种自适应选择机制使得PSL能够更好地适应不同设备的计算能力和数据分布,从而提高训练效率和模型性能.与SplitFed和PSL不同,FSL^[25]采用层次化架构,通过为每个设备分配一个专门的边缘服务器,避免所有设备向同一个中央服务器卸载训练任务而导致的通信和计算瓶颈.然而,这些工作仅采取了简单且相同的分割策略,例如将卷积层部署在设备端,而将全连接层分配给边缘服务器,并没有对异构设备进行个性化适配. CoopFL^[26]通过考虑设备和边缘服务器的异构性,提出了一种基于列表调度的算法来动态确定DNN的分割策略,使得每台设备都能分配到适当的部分模型来进行训练. FedAdapt^[27]通过DNN层卸载来加速计算受限设备上的本地训练,并采用基于强化学习的优化和聚类技术,自适应地确定每台设备卸载到服务器的DNN层,以应对设备异构性和网络带宽变化.然而,现有联邦学习中的模型分割研究仅考虑了系统设备异构性,而忽略了数据异构性.在边缘计算场景中,不同设备上的训练数据质量存在差异,这会导致局部模型训练结果偏离全局最优,进而影响全局模型的收敛.此外,现有的研究忽略了边缘服务器和多台客户端之间的协作冲突.例如,上述文献[26–27]采用最简单的先到先服务方式处理设备卸载的训练任务,这不可避免地会导致排队延迟.

综上所述,利用多层聚合以及客户端选择可以较好地平衡数据异构性和设备异构性,但不可避免地忽略某些资源受限的客户端;而模型分割可以加速本地训练,确保每台设备都能参与到联邦学习中,但增加了频繁的中间结果传输开销,导致通信延迟

增加.为此,本文旨在资源受限的异构边缘计算环境中,充分利用每台设备的算力和本地数据集,加速联邦学习训练,提高模型精度和泛化性.

2 问题的提出与建模

2.1 动机实验

随着深度学习的快速发展,神经网络模型规模从最初的较小尺寸(如AlexNet^[28]的 61×10^6 的参数量)扩展到庞大的规模(如GPT-4o的 200×10^9 的参数量^[29]).然而,边缘终端设备(例如树莓派4B)仅配备4GB的存储容量,无法支持大型模型的部署.为了探究神经网络模型在资源受限的终端设备上部署的真实运行情况,我们在2台异构设备上部署DNN模型进行训练并对比训练效果.对于常见的深度学习模型(如ResNet50),树莓派4B的算力仅为9.69 GFLOPS^[30],运行ResNet50的单批次训练时延约为13 s.相比之下,支持DNN加速的终端设备(如Nvidia Jetson Xavier NX)仅GPU部分就拥有845 GFLOPS的算力^[31],运行ResNet50的单轮训练时延仅276 ms,计算效能提升58倍.在云边端分层训练框架下,每轮模型聚合的同步等待时延接近本地训练时延.以ResNet50为例进行量化分析表明,其最大输出特征图(尺寸3.06 MB)在100 Mbps网络带宽下的传输时延为244 ms,对应部分模型的参数量大小仅16.5 KB,传输延迟仅10 ms.通过将高计算密度层卸载至边缘服务器,终端设备内存占用降低72%.尽管协同训练引入额外通信成本,但整体训练效率仍提升1.8倍.

已有的客户端选择技术通过量化客户端数据对全局模型的贡献度,优先选择高数据贡献度客户端,以此缓解数据异构问题,提升模型收敛速度.然而,传统的均匀采样或基于纯贡献度的选择方式都采用固定的参与数量^[10–13].我们在分层联邦学习框架下测试了不同客户端参与比例的影响.图1和图2分别展示了不同客户端参与率($frac$)下全局模型收敛速度以及每轮聚合的系统训练时延(实验设置详见第4节).实验揭示了客户端参与率对训练效果的非线性影响.当参与率超过60%时,模型的收敛速度基本不再提升,但训练耗时明显增加.例如,客户端参与率从60%提到80%时,每轮训练耗时增加近3 s.这表明当选择超过一定比例的客户端后,全局模型的准确率并不会提升,反而显著增加了计算和通信负担,性价比反而下降.因此,最优的客户端参与数量应当平衡模型精度与系统开销.

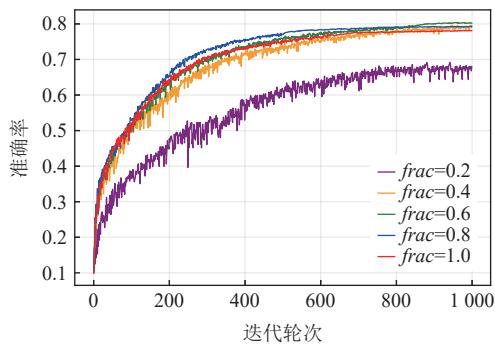


Fig. 1 Change of model convergence rates under different participation rates

图 1 不同参与率下的模型收敛速率变化

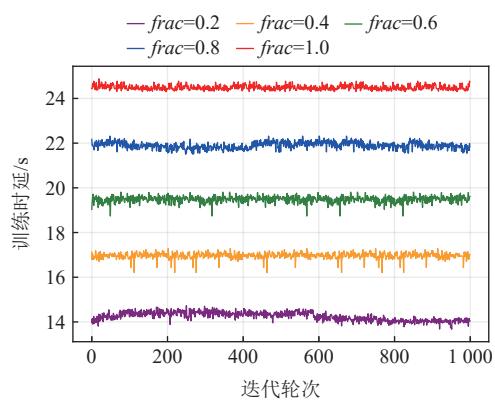


Fig. 2 Single-round cloud aggregation delay under different participation rates

图 2 不同参与率下的单轮云端聚合时延

2.2 系统模型

本文考虑一个支持模型分割和客户端选择的分层联邦学习架构。图 3 展示了分层联邦学习的系统架

构以及支持模型分割和客户端选择的分层联邦学习工作流程。与以往的分层联邦学习架构相比，本文在训练过程中对实时系统资源和用户数据进行评估，以便选择更合适的子模型分别在边缘端和终端进行协同训练。

1) 网络架构。整个分层联邦学习架构由异构终端设备、挂载边缘服务器的基站以及云端服务器组成。云端服务器作为全局参数服务器，负责全局模型的初始化、训练策略制定以及最终的全局参数聚合更新。其通过广域网(WAN)与分布式部署的边缘服务器通信，减少直接与终端设备交互的开销，有效缓解中心节点的带宽压力。每台边缘服务器 $m \in \{1, 2, \dots, M\}$ 通过基站与覆盖范围内的终端设备进行通信，形成局部的协同训练集群。边缘服务器 m 所连接的用户集合表示为 \mathcal{K}_m ，其中每轮训练参与设备子集表示为 $\hat{\mathcal{K}}_m$ 。边缘服务器负责局部模型分发、本地训练协同以及边缘聚合，其计算能力与内存容量分别表示为 \hat{C}_m 和 \hat{S}_m 。异构终端设备 $n \in \{1, 2, \dots, N\}$ 存储本地数据集 D_n ，其异构资源包括计算能力 C_n 、内存容量 S_n 和传输功率 P_n 。终端设备通过无线局域网(WLAN)与邻近边缘服务器协作来训练本地数据集。终端设备仅需要下载和上传部分模型，大大减少了模型分发和参数上传过程中的传输时延，并且允许一部分资源受限的终端设备可以参与到整体训练中。

2) 模型分割。终端设备与边缘服务器根据云中心决定的协同策略训练 DNN 模型。完整的神经网络通常由多个层次构成，主要包括输入层、隐藏层和输

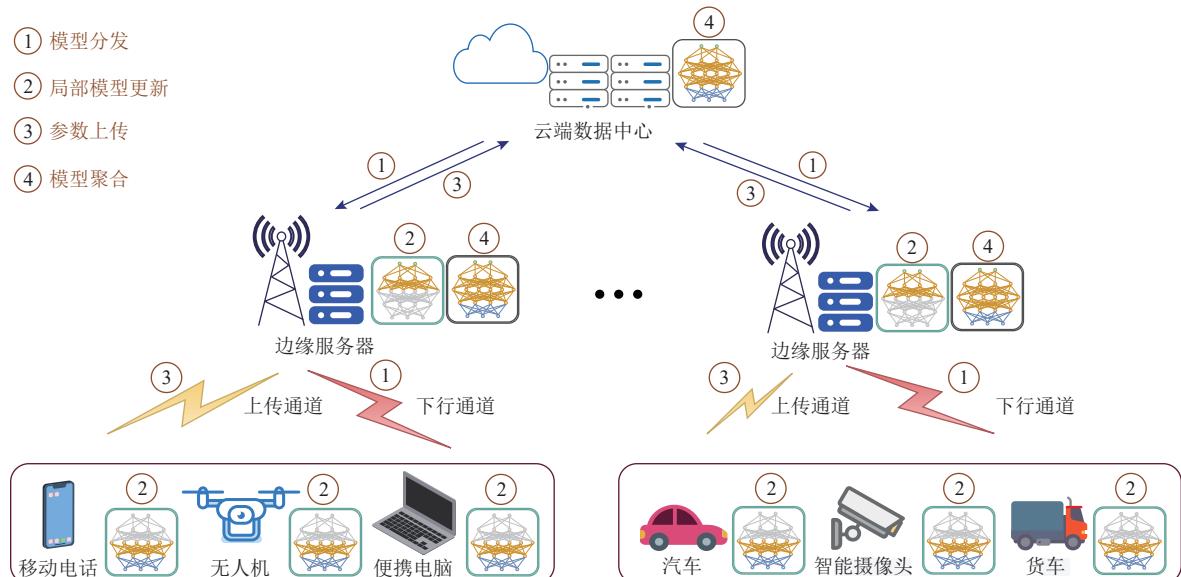


Fig. 3 Architecture of cloud-edge-end collaborative hierarchical federated learning

图 3 云边端协同分层联邦学习架构

出层。这里我们根据卷积神经网络的特性将其按层分割。假设 DNN 模型包含 L 层，其中第 0 层为输入层（第 0 层和第 1 层在终端设备上执行，不传输原始数据以保护用户隐私安全），第 $L-1$ 层为输出层，隐藏层数为 $L-2$ 。若输出层在边缘服务器上，则将中间结果和标签数据一起上传到边缘端。在训练过程中，神经网络通过前向传播计算输出，然后通过反向传播算法计算误差梯度，并根据梯度更新权重。对于终端设备 n ，其分配到的局部模型层数记为 $l_n \in [0, L]$ ，表示该设备在本地执行前 l_n 层的前向计算与反向传播，并将剩余模型卸载到边缘服务器上进行计算。如果在本轮训练中终端没被选中参与训练，则 $l_n = 0$ 。分割后的终端和边缘端的模型计算量（包括前向计算和反向传播）分别表示为 c_n 和 \hat{c}_n ，终端局部模型的前向输出结果和边缘端局部模型的反向传播梯度大小分别表示为 O_n^l 和 g_n^l 。在第 t 轮训练中完整的模型参数记为 ω^t ，分割后终端和边缘端的局部模型参数表示为 Ω_n^t 和 $\hat{\Omega}_n^t$ 。如果在本地执行完整的模型训练，则 $\Omega_n^t = \omega_n^t$ 。详细符号描述参见表 1。

Table 1 Annotations of Main Symbols

表 1 主要符号说明

符号	符号说明
M, N	边缘服务器、终端设备的数量
$\mathcal{K}_m, \hat{\mathcal{K}}_m$	与边缘服务器 m 相连的客户端集合，以及被选中参与训练的客户端集合
$D_n, D_n $	本地数据集及其大小
$\chi, \chi $	每批次训练的样本以及样本量
L, l_n	神经网络层数以及本地局部模型层数
\hat{C}_m, \hat{S}_m	边缘服务器的计算能力以及内存容量
C_n, S_n	终端设备的计算能力以及内存容量
P_m, P_n	边缘服务器以及终端设备的传输功率
$B_{m,n}$	边缘服务器 m 与终端设备 n 之间的带宽
$r_{n,m}, r_{m,n}$	上行通道与下行通道之间的传输速率
$\Omega_n^t, \hat{\Omega}_n^t$	终端以及边缘端局部模型参数
c_n, \hat{c}_n	终端和边缘端的计算量
O_n^l, g_n^l	第 l 层的前向输出及反向梯度大小
$\omega_n^t, \omega_m^t, \omega^t$	终端、边缘端、云端模型的参数

3) 训练流程

① 初始化。云端服务器收集终端设备与边缘服务器的资源信息 $(C_n, S_n, P_n, \hat{C}_m, \hat{S}_m, P_m)$ 及模型特征（层数 L 、各层的计算量以及各层输入输出大小）来制定分割策略。根据分割策略以及评估各终端设备的贡献度选择参与本轮训练的客户端集合。

② 模型分发。云端参数服务器将全局模型参数通过组播分发到各个边缘服务器中，而不是直接分发到终端设备上，以减少云端带宽的开销。边缘服务器收到全局模型参数后根据云端同步下发的协同策略将局部模型分发到参与本轮训练的客户端。

③ 边端协同训练。本轮参与训练的设备在本地数据集 D_n 上执行前 l_n 层的前向计算，并将中间结果 O_n^l 传输至边缘服务器继续计算。边缘服务器完成剩余层的前向计算后进行反向传播计算，并将梯度 g_n^l 回传到终端直至设备完成所有层的梯度更新。假定在本地训练的轮次为 τ ，总的训练轮次为 T 。在本地数据集上训练的损失函数定义为

$$F(\omega_n^t) = \frac{1}{|D_n|} \sum_{\chi \in D_n} f(\omega_n^t; \chi), \quad (1)$$

其中， χ 表示在 t 轮训练中从本地数据集 D_n 中选取的一批次数据样本， $|D_n|$ 表示本地数据集的大小。终端模型参数和边缘端聚合参数分别为 ω_n^t, ω_m^t 。本地训练采用随机梯度下降（stochastic gradient descent, SGD）进行参数更新，其更新公式为

$$\omega_n^{t+1} = \omega_n^t - \eta \nabla f(\omega_n^t; \chi), \quad \chi \in D_n, \quad (2)$$

其中， η 表示梯度更新时的学习率。

④ 模型聚合。为避免大规模的云端参数聚合导致的等待时延过长且阻塞主干网络等问题，每一台边缘服务器先一步聚合本区域的本地模型参数，并将聚合后的边缘级模型参数上传到云端进一步聚合。边缘聚合模型参数和损失函数分别为

$$\omega_m^t = \frac{1}{|\hat{\mathcal{K}}_m^t|} \sum_{n \in \hat{\mathcal{K}}_m^t} \omega_n^t, \quad (3)$$

$$F(\omega_m^t) = \frac{1}{|\hat{\mathcal{K}}_m^t|} \sum_{n \in \hat{\mathcal{K}}_m^t} F(\omega_n^t), \quad (4)$$

其中， $|\hat{\mathcal{K}}_m^t|$ 表示本轮选中参与训练的客户端数量。云端服务器收集完成 M 台边缘服务器的边缘聚合模型后进行全局模型聚合更新，具体的全局模型参数和全局模型损失函数分别为

$$\omega^{t+1} = \frac{1}{M} \sum_{m=1}^M \omega_m^t, \quad (5)$$

$$F(\omega^{t+1}) = \frac{1}{M} \sum_{m=1}^M F(\omega_m^t). \quad (6)$$

2.3 问题分析与建模

系统整体的训练效果受多重因素影响。首先，终端设备的异构性显著制约训练效率。不同设备在计算能力、内存容量及传输功率方面存在显著差异，直

接制约本地模型迭代速度与参数回传效率。其次，数据分布的复杂性导致模型收敛困难，全局模型精度下降。终端设备采集的数据集不仅存在规模差异，而且普遍具有 Non-IID 特性。数据异构性会引发本地模型更新方向的发散，需通过边缘服务器进行特征对齐来优化全局模型收敛轨迹。因此，本节对分布式训练系统中双重瓶颈问题进行分析，并构建面向模型精度-训练效率的联合优化问题模型。

在本地协同训练阶段，客户端与边缘服务器通过采用正交频分多址(OFDMA)技术分别在上行通道和下行通道上传输前向中间结果以及反向梯度。上行通道传输速率 $r_{n,m}$ 为

$$r_{n,m} = B_{n,m} \ln \left(1 + \frac{P_n |h_{n,m}|^2}{N_0 \cdot B_{n,m} + \sum_{i \neq n} P_i |h_{i,m}|^2} \right), \quad (7)$$

其中， $B_{n,m}$ 表示终端设备 n 与边缘服务器 m 之间的带宽， P_n 为终端设备的传输功率。 $h_{n,m}$ 表示信道增益，包括路径损耗、阴影衰落以及小尺度衰落。 $\sum_{i \neq n} P_i |h_{i,m}|^2$ 为同频干扰功率， N_0 为噪声功率谱密度。下行通道的传输速率 $r_{m,n}$ 与式(7)相似。因此，本地协同训练时，终端设备将中间结果上传到边缘服务器以及边缘服务器将梯度回传的总传输时延为

$$d_{t,n} = \frac{O_n^l}{r_{n,m}} + \frac{g_n^l}{r_{m,n}}. \quad (8)$$

在异构计算架构下，由于设备间存在处理器微架构差异(如 CPU/GPU/NPU 指令集并行度及缓存层次设计的不同)，传统的基于 CPU 时钟周期数的计算能力评估方法无法准确表示神经网络任务的执行效率。本文基于 Roofline^[32] 性能模型理论，引入标准化算力单位(FLOPS/W·s)作为异构设备的统一计算能力度量基准。则终端设备的算力为

$$C_n = \frac{\rho_n \times \zeta_n}{P_{TDP,n}}, \quad (9)$$

其中， ρ_n 表示设备理论峰值浮点性能(单位为 FLOPS)， ζ_n 表示神经网络算子与硬件微架构的适配系数， $P_{TDP,n}$ 表示设备的功耗。因此，终端和边缘端的总计算时延为

$$d_{c,n} = \frac{\hat{c}_n}{\hat{C}_m} + \frac{c_n}{C_n}, \quad (10)$$

其中，云端根据神经网络的层数和每层的计算量得出 c_n 和 \hat{c}_n 。此外，边缘模型分发以及收集的传输时延为

$$d_n = \frac{|\Omega_n^l|}{r_{m,n}} + \frac{|\Omega_n^l|}{r_{n,m}}. \quad (11)$$

在边缘计算环境中，当多个终端设备同时发起协同训练请求时，若边缘服务器计算资源满载，新任务将进入队列等待，直至已占用资源释放后方可执行，这一排队等待过程会直接增加整体训练时延。因此，第 t 轮训练的总时延为

$$d_{\text{all}}^t = \max_{m \in M} \left\{ \frac{|\omega_m^t|}{r_{0,m}} + \max_{n \in \hat{\mathcal{K}}_m} \{ d_n^t + \tau(d_{t,n} + d_{c,n} + d_{\text{queue},n}) \} \right\}, \quad (12)$$

其中， $r_{0,m}$ 表示云端和各边缘服务器之间的传输速率，而 $|\omega_m^t|$ 表示边缘端聚合模型的参数量大小， τ 表示本地训练的更新轮次。

综上，面向模型精度-训练效率的联合优化问题可以建模为

$$\left\{ \begin{array}{l} \text{minimize } F(\omega^t) + \lambda \cdot \sum_{t=1}^T d_{\text{all}}^t \\ \text{s.t. } \begin{cases} |\Omega_n^t| \leq S_n, \\ \sum_{n \in \hat{\mathcal{K}}_m} |\hat{\Omega}_n^t| \leq \hat{S}_m, \\ \sum_{n \in \hat{\mathcal{K}}_m} |\omega_n^t| \leq \hat{S}_m, \\ |\hat{\mathcal{K}}_m| \leq |\mathcal{K}_m|, \\ |\chi| \leq |D_n|, \\ d_{\text{queue},n} \leq \sum_{i \neq n, i \in \hat{\mathcal{K}}_m} \frac{\hat{c}_n}{\hat{C}_m}, \end{cases} \end{array} \right. \quad (13)$$

其中， λ 为效率-精度权衡系数，通过调整 λ 可以灵活地平衡模型精度和训练效率的需求。例如，对实时性要求较高的场景中，可以增加训练效率的权重，以缩短训练时间；而在对模型性能要求严格的场景中，则可以增加模型精度的权重，以确保模型的准确性和性能。

3 自适应模型分割及客户端选择方案

传统异构设备上的分布式任务调度问题(如最小化全局完成时间)已被广泛证明属于 NP-hard 难题^[3]。而动态调整神经网络分割点，需在离散参数空间中优化，与客户端选择形成多维决策变量，进一步增加复杂性。本方案旨在通过深度强化学习实现分层联邦学习中模型分割与客户端选择的联合优化，以最大化全局模型精度并最小化训练时延。其中深度强化学习智能体通过收集每一轮参与训练的执行情况，并对客户端数据贡献以及计算资源贡献进行评估，以选择下一轮参与训练的客户端集合，并通过策略梯度迭代实现长期收益最大化。本节首先对客

户端贡献进行度量，然后将模型分割与客户端选择的联合优化问题转化为马尔可夫决策过程(MDP)，最后设计了基于多线程近端策略优化的自适应学习方法最优的协同训练策略。

3.1 客户端贡献度量

客户端贡献度由计算资源贡献度以及数据质量贡献度构成。计算资源贡献度通过对客户端的算力、存储容量以及网络带宽进行度量，衡量客户端在计算和通信资源上的可用性。具体而言，计算资源贡献度高的客户端能够有效减少本地模型训练的延迟，并提升系统的并行处理能力，从而显著提高整体训练效率。我们以训练一个完整的神经网络模型为基准，对每台终端设备评估各硬件资源在训练过程中对效率的影响后，得出客户端计算资源贡献度：

$$Co_n^{\text{res}} = \frac{|\omega_n^t|}{S_n} - \delta \left(\frac{c_n + \hat{c}_n}{C_n} + \frac{|\omega_n^t|}{r_{n,m}} \right), \quad (14)$$

其中， δ 为放缩系数，将客户端内存容量的影响放缩到与训练和传输时间同一量级。其中，当客户端内存容量不足以支撑整个模型训练时，此时必须与边缘端进行协作，否则无法参与到联邦学习训练。并且本地内存容量越小则越需要将绝大部分层卸载到边缘端进行训练，抢占了大量边缘端资源，影响其他终端与边缘端的协作。

在联邦学习中，Non-IID 数据特性显著影响模型性能^[3]，其数据质量特征主要体现在 3 个方面：

1) 标签分布。客户端数据可能仅覆盖部分类别，导致本地局部模型在特性类别上过拟合，而在其他类别的分类能力较弱。

2) 数据量规模。不同客户端采集数据的频率及存储的数据量不同，其数据规模可能存在悬殊差距。特别地，部分客户端样本量极少，导致本地模型训练参数噪声过大，其上传更新方向偏离全局优化方向。

3) 特征空间。不同传感器采集到的数据由于采集时环境或设备差异导致相同标签的数据在特征分布上存在差异(如光照、角度等)，进一步加剧模型更新偏差。

余弦相似度通过计算向量空间内两两实体向量的夹角余弦值来确定实体间相似性。本文采用余弦相似度来表示本地模型与全局模型的梯度相似性。在分层联邦学习训练过程中，全局模型的优化方向由各边缘区域本地数据集的联合分布决定。当客户端与全局模型的余弦相似度越大，代表本地训练更新方式与全局更新方式更为一致，其全局模型训练

所需轮次更少；反之，较低的梯度余弦相似度则会导致本地模型更新偏离全局最优方向，拖慢收敛。考虑到本地数据样本采样的随机性，为增加鲁棒性，引入时间衰减因子 $\zeta \in [0, 1)$ ，将当前梯度相似性与历史梯度相似性结合。本地客户端的数据贡献度为

$$Co_n^{\text{data}}(t) = \zeta Co_n^{\text{data}}(t-1) + (1-\zeta) Co_n^{\text{data}}, \quad (15)$$

$$Co_n^{\text{data}} = \frac{1}{2} \left(1 + \frac{\nabla F(\omega_n^t) \cdot \nabla F(\omega^t)}{\|\nabla F(\omega_n^t)\| \times \|\nabla F(\omega^t)\|} \right). \quad (16)$$

这里，通过线性变化将值域从 $[-1, 1]$ 映射到 $[0, 1]$ ，以避免偶然的极端负值对整体贡献度的干扰。

定理 1. 存在每轮聚合的最佳全局梯度 ω_*^t 以及客户端数据贡献度 Co_n^{data} 筛选下界 $\mu \in (0, 1)$ ，即对于任一客户端梯度满足条件： $\nabla F(\omega_n^t) \cdot \nabla F(\omega_*^t) / (\|\nabla F(\omega_n^t)\| \times \|\nabla F(\omega_*^t)\|) \geq 2\mu - 1 > -1$ 。此时，平均聚合梯度方向与最佳全局梯度具有一致性。

证明。在联邦学习过程中，对 N 个训练客户端梯度进行平均，其聚合梯度与最佳全局梯度满足：

$$\begin{aligned} \nabla F(\omega^t) \cdot \nabla F(\omega_*^t) &= \frac{1}{N} \sum_{n=1}^N \nabla F(\omega_n^t) \cdot \nabla F(\omega_*^t) \geq \\ &\frac{2\mu-1}{N} \|\nabla F(\omega_*^t)\| \times \sum_{n=1}^N \|\nabla F(\omega_n^t)\| \geq \frac{2\mu-1}{N} \|\nabla F(\omega_*^t)\| \times \\ &\left\| \sum_{n=1}^N \nabla F(\omega_n^t) \right\| = \frac{2\mu-1}{N} \|\nabla F(\omega_*^t)\| \times N \|\nabla F(\omega^t)\| = \\ &(2\mu-1) \|\nabla F(\omega_*^t)\| \times \|\nabla F(\omega^t)\|. \end{aligned} \quad (17)$$

因此，当存在每轮聚合的最佳全局梯度方向及对应客户端数据贡献度下界时，此时平均全局聚合梯度与最佳全局梯度方向具有一致性。证毕。

定理 2. 在随机梯度下降过程中，收敛速度与梯度估计的方差相关。对于满足 β -强凸且 L -平滑的目标损失函数，其收敛速率满足： $\mathbb{E}[F(\omega^t) - F(\omega_*^t)] \leq O\left(\frac{\sigma^2}{\beta t}\right)$ ，其中， σ^2 为梯度方差，用于反映客户端梯度的差异。通过筛选具有更高余弦相似度的客户端，可以降低聚合梯度的方差，从而加快收敛速率。

证明。在联邦学习过程中，全局聚合梯度与客户端本地梯度满足：

$$\begin{aligned} \mathbb{E}[\|\nabla F(\omega^t) - \nabla F(\omega_*^t)\|^2] &= \\ \frac{1}{N^2} \sum_{n=1}^N \mathbb{E}[\|\nabla F(\omega_n^t) - \nabla F(\omega_*^t)\|^2], \end{aligned} \quad (18)$$

$$\begin{aligned} \|\nabla F(\omega_n^t) - \nabla F(\omega_*^t)\|^2 &= \|\nabla F(\omega_n^t)\|^2 + \\ \|\nabla F(\omega_*^t)\|^2 - 2\nabla F(\omega_n^t) \cdot \nabla F(\omega_*^t), \end{aligned} \quad (19)$$

因此，此时被筛选后的客户端方差为：

$$\mathbb{E} \left[\|\nabla F(\omega^t) - \nabla F(\omega_*^t)\|^2 \right] \leq \frac{1}{|\hat{\mathcal{K}}|} \left(\sigma^2 + 2(1-\mu) \left(\mathbb{E} \left[\|\nabla F(\omega_n^t)\|^2 \right] + \|\nabla F(\omega_*^t)\|^2 \right) \right). \quad (20)$$

此时, 随着 μ 的增大, 方差项随之线性下降。优化过程的收敛速率可以改写为:

$$\mathbb{E} [F(\omega^t) - F(\omega_*^t)] \leq O \left(\frac{\sigma_{\text{sel}}^2}{\beta t} \right), \quad (21)$$

其中,

$$\sigma_{\text{sel}}^2 = \frac{1}{|\hat{\mathcal{K}}|} \left(\sigma^2 + 2(1-\mu) \left(\mathbb{E} \left[\|\nabla F(\omega_n^t)\|^2 \right] + \|\nabla F(\omega_*^t)\|^2 \right) \right).$$

因此, 通过选择高数据贡献度客户端, 可以减少梯度方差, 并提升收敛速率。此外, 与基于 Shapley 值的贡献评估(需多次模型重训练)或影响函数(需计算海森矩阵逆)相比^[13], 梯度相似性仅需要计算向量内积和归一化操作, 适合大规模客户端的分层联邦学习场景。
证毕。

3.2 MDP 建模与奖励函数设计

联邦学习训练是多轮迭代过程, 每一轮需动态选择客户端并确定模型分割策略, 这符合 MDP 的“状态→动作→奖励→状态转移”的序列化决策范式, 并且下一轮迭代的结果仅依赖当前迭代的结果和协同策略。因此, 本文将自适应模型分割与客户端选择问题转化为 MDP 进行求解。

1) 状态空间。系统观测每一轮训练中的环境状态, 包括模型信息、设备信息、网络状态、模型信息以及上一轮训练的协同方案及训练成果。环境状态 S_t 可以表示为一个五元组 $S_t = (\Psi_n, E_m, \mathcal{H}_n, \Theta, \Gamma)$, 其中 Ψ_n 表示每台客户端的设备信息与边缘服务器之间的带宽、实时算力以及内存容量, E_m 表示边缘服务器的算力、内存容量以及上一轮边缘端维持的排队队列情况, \mathcal{H}_n 表示每台客户端上数据集的大小以及数据贡献度, Θ 为模型信息向量(模型层数、每层的输入输出大小以及参数量), Γ 为上一轮训练的结果向量(被选中的客户端贡献得分、最终的云端聚合时延以及全局模型精度)。

2) 动作空间。在每个联邦学习训练回合前, 云端需要根据上一轮回传的系统状态信息来决定本轮训练的协同策略, 包括选定参与本轮训练的客户端集合以及为这些客户端选择适当的局部模型进行协同训练。具体的节点选择和模型划分方案可以表示为向量 $A_t = \{I_1^t, I_2^t, I_3^t, \dots, I_n^t\}$, $I_n^t \in [0, L]$, 若 $I_n^t = 0$ 则表示该客户端本轮为被选中参与训练, 若 $I_n^t = L$ 则表示该客户端执行完整的本地训练, 无需边缘服务器辅助。

3) 奖励函数。当云端制定了本轮训练的协同训练策略后, 训练完成时系统环境状态随之更新并根据奖励机制得到相应策略的得分。为平衡全局模型精度以及训练效率, 本文设计的奖励机制为

$$R_{S_t, A_t} = - \left(\alpha_1 d_{\text{all}}^t + \alpha_2 \Delta F + \alpha_3 \left(\frac{\sum_{m=1}^M |\mathcal{K}_m|}{N} \right) \right), \quad (22)$$

其中 $\alpha_1, \alpha_2, \alpha_3$ 为权重系数且均大于 0, ΔF 表示与上轮的损失函数差值。通过第 2 节动机实验, 可以看出训练客户端数量对最终收敛效果的影响。因此, 在奖励机制设计中, 鼓励智能体尽量选择少的客户端来达到更优的训练效率以及最佳的模型精度。

3.3 自适应协同算法

在复杂的云边端网络下进行分层联邦学习训练需要充分考虑环境信息的动态变化, 并依据实时状态进行自适应调整。本文设计的基于多线程 PPO 的自适应协同算法如图 4 所示, 整个算法包含 4 个核心组件: 边缘端代理集群、云端全局 Actor-Critic(AC)网络、全局环境、经验池。边缘端代理监测终端设备的计算能力、内存占用率和网络带宽波动, 采集本地模型梯度以及数据分布等信息。全局 AC 网络负责收集各代理的信息并指导各边缘端进行协同训练。其中, Actor 网络制定具体的协同策略; Critic 网络根据执行的协同策略效果计算奖励, 并根据状态价值函数进

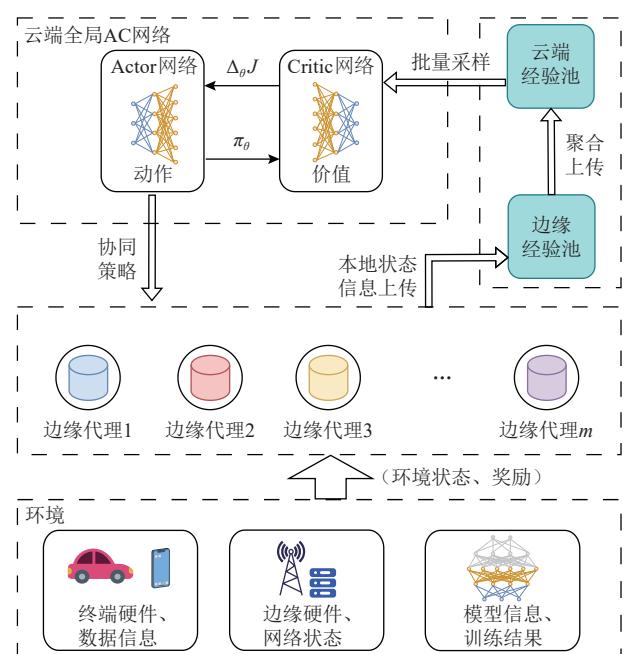


Fig. 4 AC network training framework based on multi-threading

图 4 基于多线程的 AC 网络训练框架

行评估。其输出通过策略梯度优化进行网络参数更新。此外，全局 AC 网络的参数与联邦学习全局模型参数采用异步更新的方式。经验池采用两级存储架构，边缘端本地缓存保留最近 β 轮本地训练数据作为经验池副本，并且边缘代理将整合后的本地以及边缘端的状态信息上传至云端服务器。云端全局存储通过设置优先经验回放缓冲区利用历史经验。PPO 算法通过策略梯度优化实现协同策略迭代更新。详细的云端智能体训练流程如算法 1 所示。

算法 1. 基于多线程 PPO 的自适应协同算法。

输入：初始联邦学习全局模型信息，边缘端设备信息，边端网络状态，终端设备信息，两级经验池（包含 T_1 轮所有设备参与训练信息）；

输出：参与训练客户端集合及对应模型分割方案。

- ① 初始化云端全局 Actor-Critic 网络，随机生成策略参数 θ 和价值评估参数 ϕ ；
- ② for $t \in \{1, 2, \dots, T\}$ do
- ③ 边缘代理根据云端下发策略 π_θ 选择本区域的训练客户端以及模型分割方案；
- ④ for $j \in \{1, 2, \dots, \tau\}$ do
- ⑤ 执行本地协同训练，并收集每台设备本地训练时算力，内存使用情况，网络带宽，本地梯度，数据量以及每轮训练的时延；
- ⑥ end for
- ⑦ 各边缘代理将本地经验池中的历史环境信息同步到云端经验池；
- ⑧ 云端整合各边缘的环境信息，并根据式(22)计算 R_{S_t, A_t} ，将完整经验元组 $(S_t, A_t, S_{t+1}, R_{S_t, A_t})$ 加入云端经验池；
- ⑨ Actor-Critic 网络从云端经验池批量采样经验，并根据式(25)(26)更新 Actor(θ) 和 Critic(ϕ)；
- ⑩ if $t \bmod \kappa = 0$ then
- ⑪ 清空所有经验池；
- ⑫ end if
- ⑬ end for

在 Critic 网络中用于评估策略以及动作价值的函数分别定义为

$$V_\pi(S_t) = \mathbb{E}_\pi [r_t + V_\pi(S_{t+1})], \quad (23)$$

$$Q_\pi(S_t, A_t) = R_{S_t, A_t} + \gamma V_\pi(S_{t+1}), \quad (24)$$

其中， R_{S_t, A_t} 表示采取当前策略后的即时奖励， $V_\pi(S_t)$ 表示依照当前策略可以获得的长期价值期望。此外，本文使用优势函数以减少方差，优势函数代表在状

态 S_t 下选择动作 A_t 获得的相对优势，可以表示为

$$ADV_\pi(S_t, A_t) = Q_\pi(S_t, A_t) - V_\pi(S_t). \quad (25)$$

在强化学习的策略优化方法中，策略函数 $\pi_\theta(A_t | S_t)$ 定义了智能体在状态 S_t 下选择动作 A_t 的概率分布，其中 θ 为策略网络的参数。为了确保策略更新的稳定性，PPO 算法引入 KL 散度作为约束条件。Actor 网络通过最小化含 KL 散度约束的目标函数来更新策略：

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t [\min (\rho_\theta ADV_t, \text{clip}(\rho_\theta, 1 - \varepsilon, 1 + \varepsilon) ADV_t)], \quad (26)$$

$$\rho_\theta = \frac{\pi_\theta(A_t | S_t)}{\pi_{\theta_{\text{old}}}(A_t | S_t)}, \quad (27)$$

其中 ρ_θ 为新旧策略之间的比率， ε 代表策略更新幅度的超参数。通过约束 KL 散度可以防止策略参数 θ 在一次更新中变化过大，从而导致策略性能的剧烈波动或崩溃。

其中，经验池通过定期清理，可以收集到更新更优策略的经验，避免重复采样到执行效果差的经验数据。

算法 2. 自适应分层联邦学习算法。

输入：终端设备集合，边缘服务器集合，训练轮次 T ，收集经验轮次 T_1 ；

输出：全局模型 ω^t 。

- ① 初始化全局模型参数 ω^0 , $t = 0$, 参与训练设备集合为 N ；
- ② while $t \leq T$ do
- ③ $t = t + 1$;
- ④ if $t \leq T_1$ then
- ⑤ 将模型参数 ω^{t-1} 下发到边缘端；
- ⑥ if $S_n \leq |\omega^{t-1}|$ then
- ⑦ 仅在设备 n 上执行第一层；
- ⑧ 剩余层在边缘端执行；
- ⑨ else
- ⑩ 完整执行本地训练；
- ⑪ end if
- ⑫ 将完整的执行过程信息及采集到的边缘及网络状况上传到边缘历史经验池；
- ⑬ 边缘模型聚合 ω_m^t （使用联邦平均）；
- ⑭ 云端模型聚合 ω^t （使用联邦平均）；
- ⑮ else
- ⑯ 根据云端 Actor 网络得出的策略 π_θ 选定参与设备的集合以及对应的局部模型层；
- ⑰ 本地联合训练；
- ⑱ 将完整的执行过程信息及采集到的边缘及网络状况上传到边缘历史经验池；

- ⑯ 边缘模型聚合 ω_m' (使用联邦平均);
- ⑰ 云端模型聚合 ω' (使用联邦平均);
- ⑱ Actor 及 Critic 网络在云端后台根据采集到的状态信息进行异步更新;
- ⑲ end if
- ⑳ end while

算法 2 中, 通过 T_1 轮的全部设备参与训练, 可以充分采集到每一台设备的硬件信息以及本地数据集的梯度信息, 以此得到本地客户端的数据贡献以及计算资源贡献。在算法 1 中根据全部设备的训练经验信息初始化原始的经验池进行采样更新, 可以增加经验池的采样范围。特别地, 若当前策略在终端无法执行, 则该设备上的执行时间、梯度损失值等返回正无穷。

4 实验结果与分析

4.1 实验设置

本文构建了一个涵盖云端、边缘层及终端设备的 3 层异构计算架构, 全面模拟真实边缘计算场景下的计算资源分布与协同训练过程。云端采用部署一台配备 NVIDIA 4090 GPU(24 GB 显存)与 Intel i9-13900K CPU 的高性能服务器模拟, 负责全局模型管理与聚合任务。边缘层由 2 类异构服务器模拟: 一类为搭载 NVIDIA 2080Ti GPU(11 GB 显存)与 Intel Xeon Silver 4310 CPU 的高算力节点, 另一类为基于 Jetson Xavier NX(384 核 Volta GPU, 8 GB 显存)的嵌入式设备, 分别模拟算力充足与资源受限的边缘服务场景。终端设备则采用树莓派 4B(4 核 ARM CPU, 4 GB 内存)与 Jetson Nano(128 核 Maxwell GPU, 4 GB 内存)混合配置, 覆盖从纯 CPU 到轻量 GPU 加速的多样化设备类型。为精确量化硬件异构性对算法的影响, 首先对各设备进行完整的模型训练(Cifar-10 数据集, 批量大小为 32), 记录完整模型训练周期内的资源占用与端到端训练耗时。此外, 对参与训练的神经网络(如 ResNet50)的网络结构、每层的参数量以及输入输出大小进行分析, 用于仿真的设备资源信息如表 2 所示。

实验的通信环境采用分层网络模型, 云端与边缘服务器之间通过广域网连接, 带宽固定为 100 Mbps。本文聚焦边缘层与终端设备之间的局域无线网络, 边缘层与终端设备通过 5 GHz WiFi 频段(IEEE 802.11ac 协议, 信道数为 10)进行通信。部署边缘服务器的基站可以覆盖的最大通信距离为 1 000 m。每台终端设

Table 2 Performance Parameters of Device Hardware

表 2 设备硬件性能参数

设备	算力/GFLOPS	内存或显存/GB
高性能服务器(配备 4090)	82 600	24
高性能服务器(配备 2080Ti)	13 400	11
Jetson Xavier NX	845	8
Jetson Nano	250	4
树莓派 4B	9.7	4

备的传输功率设定为 0.1 W, 基站的传输功率为 10 W, 背景噪声统一为 -100 dBm 以模拟典型室内干扰场景。详细的通信参数如表 3 所示。

Table 3 Communication Parameters

表 3 通信参数

参数	设置
终端传输功率/W	0.1
基站传输功率/W	10
信道数量	10
背景噪声功率/dbm	-100
基站通信覆盖范围/m	1~1 000
路径损耗指数	3.7

实验采用公开基准数据集 CIFAR-10^[33] 与 Fashion-MNIST^[34] 构建边缘计算场景下的分布式学习任务。为模拟真实边缘环境中数据的空间异质性, 实验通过标签偏斜与数量偏斜联合策略生成 Non-IID 数据划分^[19]。具体而言, 客户端的标签分布通过 Dirichlet 分布(浓度参数 $\alpha = 0.5$)进行差异化配置。同时, 客户端拥有的样本总数服从幂律分布(长尾系数 $\beta = 2.0$), 模拟实际场景中部分设备因存储或采集能力限制仅能贡献少量数据。实验选取 3 种经典 DNN 模型, 包括 ResNet 系列^[35](ResNet18, ResNet34, ResNet50, ResNet101, ResNet152) 以及模型 VGG16^[36] 和 AlexNet^[28], 以验证本文设计的算法在不同模型复杂度下的适应性。其中, ResNet 变体通过调整残差块数量与通道维度实现模型层数深化。与文献[26]中神经网络训练超参数设置相似, 实验训练参数统一采用 SGD 优化器, 基础学习率设为 0.01, 动量项为 0.9, 权重衰减系数为 1E-4。每轮本地训练执行 10 次迭代, 批量大小按模型及数据集特性动态配置。本文中的数据划分、分层联邦学习以及强化学习代码均通过 Pytorch 框架实现。

为全面评估所提方法的泛化性及性能, 本文选取了 5 种联邦学习架构作为对比基线, 深入分析不同算法在异构环境下的性能差异。

1) TFL^[4]. 采用典型的参数服务器架构, 设备独立训练完整模型后直接与中心服务器通信进行全局聚合.

2) HFL^[5]. 引入边缘服务器作为中间层, 形成云边端三级架构. 设备将模型上传至边缘服务器进行局部聚合, 再由边缘服务器与云端同步全局模型. 这一设计虽减少了设备与云端的直接通信, 但设备仍需独立训练完整模型.

3) SplitFed^[23]. 首次将联邦学习与分割学习结合, 通过模型分割和差分隐私的客户端模型更新来提供模型隐私, 并通过并行处理客户端来加快训练速度.

4) CoopFL^[26]. 通过在设备和边缘服务器之间进行DNN的分割和卸载, 实现边端之间协同训练. 其中, 模型分割根据最小化训练时延以及充分使用资源为目标动态调整.

5) FedAdapt^[27]. 通过自适应地将DNN的部分层从IoT设备卸载到服务器来加速联邦学习训练. 该框架利用强化学习优化卸载策略, 以应对设备的硬件资源异构性和网络带宽变化.

为公平比较, 除TFL与SplitFed方法因算法不兼容云边端3层架构(直接在终端与云端间传输完整模型参数)外, 其余基线算法均在相同的云边端异构仿真环境下进行验证. TFL与SplitFed方法则部署于移除边缘层的云端-终端两层异构仿真环境. 其中, 云端以及终端的硬件配置与云边端3层异构环境配置相同, 通信采用相同带宽以及信道数的终端-云端直连链路. 在基准异构环境的配置中, 我们设定边缘服务器的数量为5, 其中1台使用资源相对较弱的Jetson Xavier NX; 终端设备的数量为100, 按照1:1的比例随机部署树莓派4B以及Jetson Nano. 数据的异构划分以及网络设置均按照4.1节中的方式进行基础配置.

4.2 实验结果分析

实验分别从准确率、训练时延以及系统可扩展性角度对5种方法进行了对比分析.

1) 不同模型在不同数据集上的准确率比较

本文在CIFAR-10与Fashion-MNIST这2个差异化数据集上进行实验, 分别采用ResNet50, AlexNet和VGG16这3种不同深度和架构的DNN模型, 验证ASSHFL算法在模型准确率上的泛化能力. 如表4所示, ASSHFL在2类数据集及3种模型上均表现出显著优势. 在CIFAR-10数据集上, ASSHFL的测试准确率相较于最差基线方法TFL平均提升9个百分点, 其中ResNet50, AlexNet与VGG16的准确率分别提升10.9个百分点, 7.3个百分点和9.4个百分点; 与次优算法相比, 平均提升幅度仍达5个百分点, 尤其在

ResNet50等深层模型中优势更为显著, 其提升幅度较AlexNet高出42个百分点, 表明算法的效果随模型深度增加而增强. 在Fashion-MNIST数据集上, 由于任务复杂度较低, 所有基线算法的平均准确率均超过83个百分点, 但ASSHFL准确率仍保持稳定提升: 相较于最差基线方法, 其在ResNet50, AlexNet与VGG16上的准确率分别提升10.26个百分点, 4.56个百分点与6.5个百分点; 与次优算法相比, 提升幅度收敛至1.39~3.16个百分点, 印证了算法在简单特征空间中的性能增益存在上限. 值得注意的是, 当模型深度增加时, ASSHFL的优势在CIFAR-10场景中更为突出. 值得注意的是, 当采用传统模型分割方法, 模型准确率有所下降. 这种现象源于设备异构性引发的计算偏差累积: 在联邦学习过程中, 不同终端设备的硬件计算单元对同一模型层的正向传播输出存在细微差异, 此类偏差在反向传播时通过链式法则逐层放大. 同样采用模型分割的方法, ASSHFL通过数据贡献感知的设备动态筛选机制, 有效缓解了异构数据分布对模型准确率的负面影响.

Table 4 Training Accuracy Performance Under Different Models and Different Data Sets

表4 不同模型及不同数据集下的训练准确率表现

数据集	模型	算法					
		TFL	HFL	SplitFed	CoopFL	FedAdapt	ASSHFL (本文)
ResNet50	69.27	72.36	70.49	73.83	74.29	80.17	
CIFAR-10	AlexNet	61.52	64.26	59.65	64.70	64.14	68.83
	VGG16	63.14	67.31	62.27	65.92	66.85	72.52
	ResNet50	91.34	90.71	83.26	92.13	91.72	93.52
Fashion-MNIST	AlexNet	87.19	88.30	85.61	88.36	87.29	90.17
	VGG16	89.42	87.62	86.74	89.77	90.08	93.24

注: 黑体数值以及下划线数值分别表示最优结果和次优结果.

2) 不同模型在不同边缘端配置训练效果比较

为了探究本文所提算法在不同边缘服务器配置下的性能表现以及在环境变化时的适应性, 实验分别测试了边缘端服务器数量在2, 5, 10, 15, 20时的表现, 其中, 高性能与资源受限的边缘服务器按照1:1配置, 其他环境参数按照基准设置. 如图5所示, 在3种模型ResNet50、AlexNet、VGG16以及不同的边缘服务器数量配置下, ASSHFL均保持了较高的准确率, 并在不同的异构边缘端配置下表现出良好的稳定性. 对于模型层数多、参数量大的模型, 如ResNet50, VGG16, ASSHFL的准确率提升尤为显著, 提升幅度接近10个百分点. 以ResNet50为例, 从图5(a)(c)可

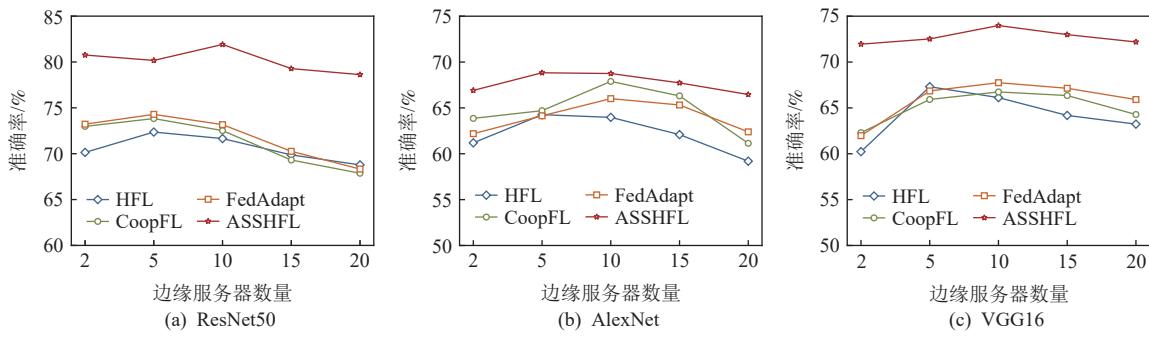


Fig. 5 Accuracy changes of different models under different edge configurations

图 5 不同边缘配置下各模型准确率变化

以看出, ASSHFL 在不同边缘服务器数量下的准确率均显著高于其他方法, 尤其在边缘服务器数量较少的情况下, 准确率提升更为明显。这是由于 ASSHFL 通过数据贡献度挑选合适的客户端参与训练, 减少因边缘服务器聚合而导致的聚合梯度分散问题。从图 5(b)可以看出, ASSHFL 在不同边缘服务器数量下训练 AlexNet 获得的准确率均高于其他方法, 但提升幅度相对较小, 不如 ResNet50 和 VGG16 上明显, 但仍然比次优方法提升了约 3 个百分点。这是因为 AlexNet 模型相对简单, 其可选择的层以及卷积层的参数量相较其他模型偏小, 导致模型分割方案的优势难以充分发挥。值得注意的是, ASSHFL 所获得模型准确率在不同服务器配置下的波动幅度始终维持在较小范围, 充分验证了其对异构边缘环境的强适应能力。

图 6 展示了不同边缘服务器配置下训练 3 种网络所需的整体训练时间。对于传统的 HFL 训练模型, 使用模型分割的方案, 如 CoopFL、FedAdapt、ASSHFL 均取得了较大的训练效率提升。以 ResNet50 为例, 从图 6(a)可以看出, ASSHFL 的训练时间显著低于 HFL, 且在边缘服务器数量较多时, 训练时间进一步减少。对于 AlexNet, 从图 6(b)可以看出, ASSHFL 的训练时

间也显著低于 HFL, 但提升幅度相对较小。对于 VGG16, 由图 6(c)所示, ASSHFL 的训练时间显著低于 HFL, 且在边缘服务器数量较多时, 训练时间进一步减少。这是由于模型分割可以有效减少传输本地模型的传输时延, 以及缓解整体训练受拖尾设备的影响。当边缘服务器数量较少时, 由于需要与边缘端协作, 会加剧边缘服务器的资源消耗, 导致边缘服务器协作等待队列延长, 增加本地训练中等待协作时间。当边缘端计算资源充裕时, 排队时延减少; 边端信道资源充足时, 协同训练过程中的传输时间因此减少。对比 VGG16 这类参数密集但层级较浅的神经网络架构, 模型分割的优化效果在层级复杂的模型中更为显著。这主要源于其层间参数分布特性, 当单层参数体量过大时, 分割策略难以像处理深层 DNN 那样有效分解计算负载。虽然模型分割通过降低传输时延与均衡计算资源来提升训练效率, 但对于此类层级精简但参数冗余的模型结构, 其传输优化与负载调节的空间均受限, 导致最终效率提升有限。

3) 不同模型深度的训练效果比较

模型分割算法的设计基于神经网络拓扑结构, 其效果随模型深度变化呈显著差异。为探究模型深度对分割效果的影响, 本文研究选取 ResNet 系列模型

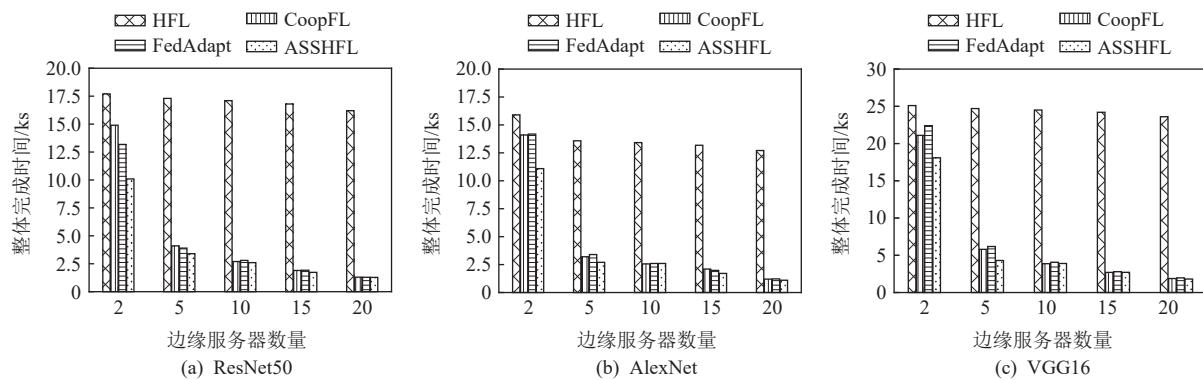


Fig. 6 Overall training delay of different models under different edge configurations

图 6 不同边缘配置下各模型全局训练时延

进行对比实验。图 7 表明，在不同模型深度下，ASSHFL 相较于其他方法均表现出显著优势。且随着模型深度的增加，模型的准确率相应提升。然而，当模型卷积层深度超过 50 时，所有算法的准确率均出现一定程度的下降。这是由于随着模型深度的增加，特征的表达能力逐渐达到瓶颈，甚至引入冗余信息，导致模型难以进一步优化。同时，模型复杂度的提升虽拓展了分割方案的寻优空间，但也加剧了最优分割策略的搜索难度。图 8 展示了训练不同深度 ResNet 模型的单轮云端聚合时延。单轮云端聚合时延是指模型从云端下发到云端聚合完成过程中新模型之间的时延。由于随着模型深度的变化，基于模型分割的算法与不采用模型分割方法，如 TFL、HFL 之间在性能上出现了数量级的差异，因此在图 8 中未展示 TFL 以及 HFL 的单轮云端聚合时延。例如，ResNet 152 模型的单轮本地训练时间可达 2000 s，而 HFL 还需叠加边缘端聚合与边缘端传输时延，整体耗时更为显著。从图 8 中可以看出，ASSHFL 与其他分割方法的效率差距随模型深度的增加持续扩大，这得益于其独特的客户端选择机制：通过动态筛选高资源贡献度客户端参与训练，在不限制参与规模的前提下有效规避拖尾设备的影响，从而在深度模型训练中展现出更显著的训练效率优势。

4) 不同模型在不同数据分布下的准确率比较

本文在不同数据分布下进行实验，探究在数据分布差异大的异构环境下，各方法的模型准确率表现。实验中分别采用了独立同分布以及 Non-IID 的本地数据集划分。Non-IID 的数据划分按照标签偏斜与数量偏斜的联合策略设置，具体的设置方案如 4.1 节所述。

由图 9 可以看出，在独立同分布(IID)的本地数据集上的训练相比于在 Non-IID 本地数据集上的训

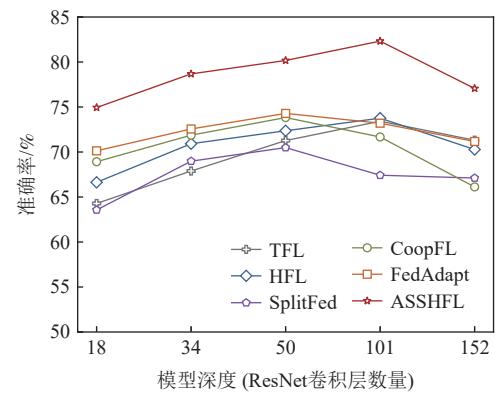


Fig. 7 Accuracy for different depth of ResNet models
图 7 不同 ResNet 模型深度下的准确率

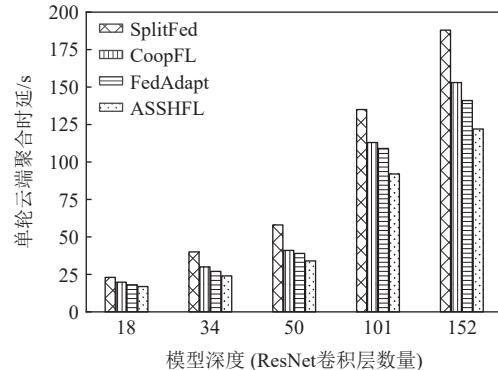


Fig. 8 Single-round cloud aggregation delay for different depth of ResNet models
图 8 不同 ResNet 模型深度下单轮云端聚合时延

练，模型准确率最高可获得接近 10 个百分点的提升。此外，ASSHFL 在 ResNet50 上相较于其他基线算法获得了更显著的准确率提升，尤其在 Non-IID 的数据异构情况下，这种优势更为明显。由于 ASSHFL 相较于其他方法引入了数据贡献度筛选机制，通过挑选与全局梯度更新方向更相似的本地客户端参与训练，有效减少了因数据异构导致的模型准确率下降。此外，图 7 及图 8 均显示基线方法在 ResNet50、VGG16

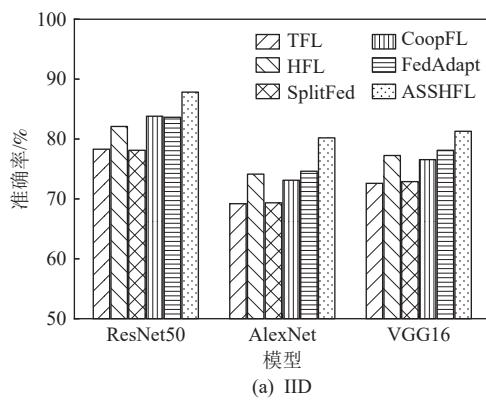
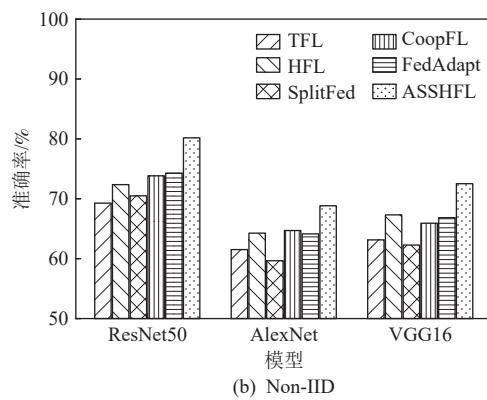


Fig. 9 Accuracy of models under different data distribution

图 9 不同数据分布下的模型准确率



上的表现不如在 AlexNet 上的表现。这是由于随着模型复杂度增加, 模型分割方案的寻优空间变大, 难以找到最优的分割方案; 而 ASSHFL 通过不断学习模型信息以及与环境进行交互获得奖励反馈, 并根据奖励反馈不断迭代生成更优的分割策略。

5) 不同终端配置下训练效果比较

本文针对联邦学习系统中普遍存在的设备异构性问题, 通过仿真 50~250 台终端设备规模的分布式计算环境, 系统评估 ASSHFL 方法与基线方法在不同终端数量配置下的模型准确率及训练效率表现。图 10 以及图 11 分别展现了 ASSHFL 方法在模型准确率与训练效率 2 个方面对比基线算法的优势, 并且验证了其对异构联邦场景的强适应性。

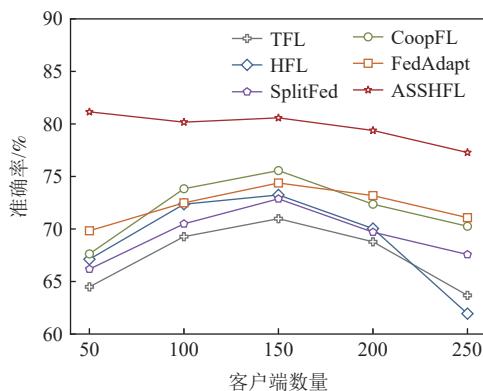


Fig. 10 Accuracy of different number of clients

图 10 不同客户端数量下的准确率

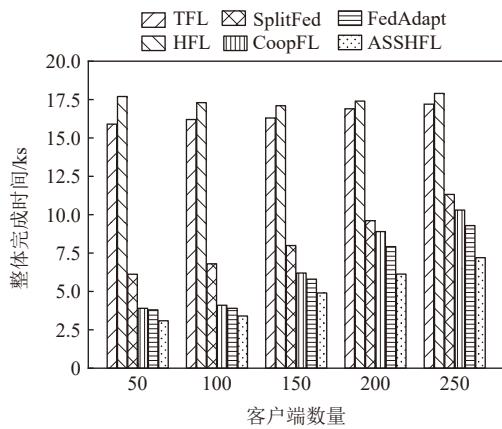


Fig. 11 Overall completion time under different number of clients

图 11 不同客户端数量下的整体完成时间

在模型准确率方面, ASSHFL 展现出显著的优势。具体而言, 当客户端数量为 50 时, ASSHFL 的模型准确率达到 80% 以上, 相比基线方法(64%~70%)提升了 12~17 个百分点。即使在客户端规模扩大至

250 时, ASSHFL 仍能保持 75% 以上的模型准确率。这得益于 ASSHFL 的动态优化机制: 通过多个边缘代理实时收集本地设备信息及数据分布特征, 并基于这些信息持续学习如何调整模型分割以及选择参与训练的客户端集合策略。这种动态调整有效缓解了因客户端数量增加而导致的数据分布变化对模型准确率的负面影响, 确保了方法在不同终端规模下的稳定。

训练效率方面, 如图 11 所示, ASSHFL 在 50 个客户端场景下的训练耗时仅为 3 ks, 较最差方法 HFL (17.7 ks) 提升 4.9 倍; 较次优算法 FedAdapt(3.8 ks) 提升 21 个百分点。并且随着客户端数量增加到 250, ASSHFL 较次优算法 FedAdapt 训练效率提升 25%。尽管同样基于强化学习框架, 但是 ASSHFL 通过联合数据贡献以及计算资源贡献, 选择尽量少的设备参与到训练中, 有效降低了由于大量贡献度不高的设备参与而影响到整体训练的效率。实验验证了 ASSHFL 在大规模终端设备参与训练的情况下, 可以有效筛选出高效的设备并通过模型分割的方式协同训练, 以加速整体的训练效率。

6) 不同边缘端带宽下训练时间比较

在分层联邦学习的计算架构中, 边缘端网络性能对系统效率具有决定性影响, 尤其在基于模型分割与训练任务卸载的场景下。由于边缘服务器需与终端设备持续交换中间特征张量和梯度参数, 网络带宽不足将导致通信延迟急剧上升。例如, 当边缘端带宽降至 10 Mbps 时, 单次梯度传输耗时可达 12.7 s, 甚至接近低配置终端设备(如树莓派 4B)的本地训练耗时(约 13 s), 此时系统整体效率受限于通信瓶颈。为了验证 ASSHFL 及基线方法在不同边缘端网络状态下的执行效果, 我们在边缘端网络带宽设置为 10 Mbps, 50 Mbps, 100 Mbps, 150 Mbps, 200 Mbps 下验证 ASSHFL 与基线方法的训练效率。

图 12 展示了在不同边缘端带宽下的整体训练完成时间。当边缘端带宽降低到 50 Mbps 以下时, 基于分层联邦学习的算法性能急剧下降。这是由于分层联邦学习架构对比传统的参数服务器架构需要频繁的边缘端模型分发以及上传的传输开销。对比其他 2 种基于模型分割的分层联邦学习方法, ASSHFL 分别获得了 7%~8% 的效率提升。这表明在带宽受限的分层联邦学习场景中, ASSHFL 通过动态感知底层信道与屏蔽数据分布选择其数据分布接近的客户端参与训练, 能够有效减少边缘端不必要的模型传输次数, 更好地利用有限的带宽资源提高训练效率。当带

宽增加到 100 Mbps 以上时, 基于模型分割的分层联邦学习算法性能趋于稳定, 并且模型训练效率并没有显著提升。这表明, 当边缘端带宽足够大时, 网络传输不再是训练效率的主要瓶颈, 此时制约训练效率的主要因素是本地训练的计算能力。因此, 在通信资源丰富的场景下, 将算力薄弱设备上的全部训练任务卸载到边缘端, 就可以获得极大的训练效率提升。

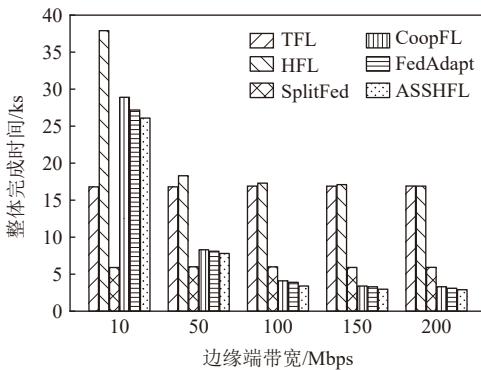


Fig. 12 Overall completion time under different edge bandwidth

图 12 不同边缘端带宽下的整体完成时间

5 结 论

本文设计并实现了一种基于模型分割与客户端选择的自适应分层联邦学习方法(ASSHFL), 旨在有效提升联邦学习的训练效率并增强全局模型的性能表现。通过感知终端设备的硬件资源异构性与数据分布特性, ASSHFL 基于多线程 PPO 方法, 动态优化边端协同训练策略。具体而言, 方法通过评估终端硬件资源、网络状况及分析客户端梯度特征, 智能筛选出数据质量高且与全局模型相似度高的本地数据集进行训练, 并通过模型分割与边端协同机制, 显著提升资源受限设备的参与度, 从而充分利用优质数据源以优化训练过程。实验结果表明, 在不同异构边缘计算环境下, ASSHFL 与基于分层联邦学习的基线方法(HFL)以及基于模型分割的方法(SplitFed, CoopFL, FedAdapt)进行对比, 均在各项指标上展现出显著优势, 验证了 ASSHFL 在复杂异构环境下的良好适应性与鲁棒性。

作者贡献声明: 冯奕铭负责方案整体设计并撰写论文; 钱珍、李光辉负责部分算法思路和实验方法; 李光辉、代成龙提出指导意见并修改论文。

参 考 文 献

- [1] Sun Bing, Liu Yan, Wang Tian, et al. Survey on efficiency optimization of federated learning in mobile edge networks[J]. *Journal of Computer Research and Development*, 2022, 59(7): 1439–1469 (in Chinese)
(孙兵, 刘艳, 王田, 等. 移动边缘网络中联邦学习效率优化综述[J]. *计算机研究与发展*, 2022, 59(7): 1439–1469)
- [2] Zhang Xueqing, Liu Yanwei, Liu Jinxia, et al. Survey on federated learning for edge intelligence[J]. *Journal of Computer Research and Development*, 2023, 60(6): 1276–1295 (in Chinese)
(张雪晴, 刘延伟, 刘金霞, 等. 面向边缘智能的联邦学习综述[J]. *计算机研究与发展*, 2023, 60(6): 1276–1295)
- [3] Ye Mang, Fang Xiwen, Du Bo, et al. Heterogeneous federated learning: State-of-the-art and research challenges[J]. *ACM Computing Surveys*, 2024, 56(3): 1–44
- [4] Verma D C, Julier S. J, Cirincione G. Federated AI for building AI solutions across multiple agencies[J]. arXiv preprint, arXiv: 1809.10036, 2018
- [5] Liu Lumin, Zhang Jun, Song S H, et al. Client-edge-cloud hierarchical federated learning[C]//Proc of the 2020 IEEE Int Conf on Communications (ICC). Piscataway, NJ: IEEE, 2020: 1–6
- [6] Abdellatif A A, Mhaisen N, Mohamed A, et al. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data[J]. *Future Generation Computer Systems*, 2022, 128: 406–419
- [7] Wu Wentai, He Ligang, Lin Weiwei, et al. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 32(7): 1539–1551
- [8] Wang Lun, Xu Yang, Xu Hongli, et al. Accelerating decentralized federated learning in heterogeneous edge computing[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(9): 5001–5016
- [9] Huang Wenke, Ye Mang, Du Bo. Adaptive heterogeneous federated learning[J]. *Journal of Image and Graphics*, 2024, 29(7): 1849–1860 (in Chinese)
(黄文柯, 叶茫, 杜博. 自适应异构联邦学习[J]. *中国图象图形学报*, 2024, 29(7): 1849–1860)
- [10] Tang Xiaolan, Liang Yuting, Chen Wenlong. Multi-stage federated learning mechanism for non-IID data in Internet of vehicles[J]. *Journal of Computer Research and Development*, 2024, 61(9): 2170–2184 (in Chinese)
(唐晓岚, 梁煜婷, 陈文龙. 面向非独立同分布数据的车联网多阶段联邦学习机制[J]. *计算机研究与发展*, 2024, 61(9): 2170–2184)
- [11] Diao Enmao, Ding Jie, Tarokh V. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients[J]. arXiv preprint, arXiv: 2010.01264, 2021
- [12] Liu Yan, Wang Tian, Peng Shaoliang, et al. Federated learning model cleansing and device clustering method based on edge computing[J]. *Chinese Journal of Computers*, 2021, 44(12): 2515–2528 (in Chinese)
(刘艳, 王田, 彭绍亮, 等. 基于边缘的联邦学习模型清洗和设备聚

- 类方法[J]. *计算机学报*, 2021, 44(12): 2515–2528)
- [13] Wang Ruyan, Chen Wei, Zhang Puning, et al. Resource-efficient hierarchical collaborative federated learning for heterogeneous IoT[J]. *Journal of Electronics & Information Technology*, 2023, 45(8): 2847–2855 (in Chinese)
(王汝彦, 陈伟, 张普宁, 等. 异构物联网下资源高效的分层协同联邦学习方法[J]. *电子与信息学报*, 2023, 45(8): 2847–2855)
- [14] Zhong Zhengyi, Bao Weidong, Wang Ji, et al. A hierarchical heterogeneous federated learning method for cloud-edge-end systems[J]. *Journal of Computer Research and Development*, 2022, 59(11): 2408–2422 (in Chinese)
(钟正仪, 包卫东, 王吉, 等. 一种面向云边端系统的分层异构联邦学习方法[J]. *计算机研究与发展*, 2022, 59(11): 2408–2422)
- [15] Gao Yujia, Wang Pengfei, Liu Liang, et al. Personalized federated learning method based on attention-enhanced meta-learning network[J]. *Journal of Computer Research and Development*, 2024, 61(1): 196–208 (in Chinese)
(高雨佳, 王鹏飞, 刘亮, 等. 基于注意力增强元学习网络的个性化联邦学习方法[J]. *计算机研究与发展*, 2024, 61(1): 196–208)
- [16] Mishra R, Gupta H P. A model personalization-based federated learning approach for heterogeneous participants with variability in the dataset[J]. *ACM Transactions on Sensor Networks*, 2024, 20(1): 1–28
- [17] Zhang Yingqi, Xia Hui, Xu Shou, et al. AdaptFL: Adaptive federated learning framework for heterogeneous devices[J]. *Future Generation Computer Systems*, 2025, 165: 107610
- [18] He Wencheng, Guo Shaoyong, Qiu Xuesong, et al. DRL-based node selection method for federated learning[J]. *Journal on Communications*, 2021, 42(6): 62–71 (in Chinese)
(贺文晨, 郭少勇, 邱雪松, 等. 基于 DRL 的联邦学习节点选择方法[J]. *通信学报*, 2021, 42(6): 62–71)
- [19] Du Tian, Chen Xingyan, Kou Gang, et al. Clustered federated learning for cloud-edge personalized model decoupling[J]. *Chinese Journal of Computers*, 2025, 48(2): 407–432 (in Chinese)
(杜甜, 陈星延, 寇纲, 等. 面向云边个性化模型解耦的聚类联邦学习方法[J]. *计算机学报*, 2025, 48(2): 407–432)
- [20] Qu Zhe, Duan Rui, Chen Lixing, et al. Context-aware online client selection for hierarchical federated learning[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2022, 33(12): 4353–4367
- [21] Sun Qiheng, Li Xiang, Zhang Jiayao, et al. ShapleyFL: Robust federated learning based on Shapley value[C]//Proc of the 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2023: 2096–2108
- [22] Mo Zijia, Gao Zhipeng, Yang Yang, et al. Efficient distributed model sharing strategy for privacy-preserving in IoV data[J]. *Journal on Communications*, 2022, 43(4): 83–94 (in Chinese)
(莫梓嘉, 高志鹏, 杨杨, 等. 面向车联网数据隐私保护的高效分布式模型共享策略[J]. *通信学报*, 2022, 43(4): 83–94)
- [23] Thapa C, Arachchige P C M, Camtepe S, et al. SplitFed: When federated learning meets split learning[C]//Proc of the AAAI Conf on Artificial Intelligence. New York: ACM, 2022: 8485–8493
- [24] Jeon J, Kim J. Privacy-sensitive parallel split learning[C]//Proc of the 2020 Int Conf on Information Networking (ICOIN). Piscataway, NJ: IEEE, 2020: 7–9
- [25] Turina V, Zhang Zongshun, Esposito F, et al. Federated or split? A performance and privacy analysis of hybrid split and federated learning architectures[C]//Proc of the 2021 IEEE 14th Int Conf on Cloud Computing (CLOUD). Piscataway, NJ: IEEE, 2021: 250–260
- [26] Wang Zhiyuan, Xu Hongli, Xu Yang, et al. CoopFL: Accelerating federated learning with DNN partitioning and offloading in heterogeneous edge computing[J]. *Computer Networks*, 2023, 220: 109490
- [27] Wu Di, Ullah R, Harvey P, et al. FedAdapt: Adaptive offloading for IoT devices in federated learning[J]. *IEEE Internet of Things Journal*, 2022, 9(21): 20889–20901
- [28] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. *Communications of the ACM*, 2017, 60(6): 84–90
- [29] Shahriar S, Lund B, Mannuru N R, et al. Putting GPT-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency[J]. arXiv preprint, arXiv: 2407.09519, 2024
- [30] Zubov D, Kupin A. Performance Evaluation of Raspberry Pi 4B Microcomputer: Case Studies on MPICH Cluster, VMware ESXi ARM fling, and Windows 11 ARM OS[M]. Berlin: Springer, 2022
- [31] Czaja J, Gallus M, Wozna J, et al. Applying the Roofline model for deep learning performance optimizations[J]. arXiv preprint, arXiv: 2009.11224, 2020
- [32] Deng Yongheng, Ren Ju, Tang Cheng, et al. A hierarchical knowledge transfer framework for heterogeneous federated learning[C]//Proc of the 2023 IEEE Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2023: 1–10
- [33] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[EB]. *Handbook of Systemic Autoimmune Diseases*. Toronto: University of Toronto, 2009
- [34] Xiao Han, Rasul K, Vollgraf R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms[J]. arXiv preprint, arXiv: 1708.07747, 2017
- [35] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[J]. arXiv preprint, arXiv: 1512.03385, 2015
- [36] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint, arXiv: 1409.1556, 2014



Feng Yiming, born in 1996. PhD candidate. His main research interests include edge computing and AIoT.

冯奕铭, 1996 年生. 博士研究生. 主要研究方向为边缘计算、智能物联网.



Qian Zhen, born in 1996. PhD candidate. Her main research interests include edge caching and federated learning.

钱珍, 1996 年生. 博士研究生. 主要研究方向为边缘缓存、联邦学习.



Li Guanghui, born in 1970. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include wireless sensor network, edge computing, and intelligent nondestructive detection technology.

李光辉, 1970年. 博士, 教授, 博士生导师.
CCF高级会员. 主要研究方向为无线传感网
络、边缘计算、智能无损检测技术.



Dai Chenglong, born in 1992. PhD, associate professor, master supervisor. His main research interests include electroencephalogram processing and electroencephalogram analyzing.

代成龙, 1992年. 博士, 副教授, 硕士生导师.
主要研究方向为脑电图处理、脑电图分析.