HiFlash：采用自适应陈旧度控制与异质性感知的客户端-边缘节点关联的高通信效率分层联邦学习

Qiong Wu，Xu Chen，*IEEE* 高级会员，Tao Ouyang，Zhi Zhou，*IEEE* 会员，Xiaoxi Zhang，*IEEE* 会员，Shusen Yang，*IEEE* 高级会员，以及 Junshan Zhang，*IEEE* 会士

arXiv:2301.06447v1 [cs.NI] 16 Jan 2023

摘要—联邦学习（FL）是一种颇具前景的范式，能够在保护训练数据留在本地的同时，让大量客户端协同训练共享模型。然而，对于许多现有的联邦学习系统，客户端需要通过广域网（WAN）频繁地与远程云服务器直接交换数据量庞大的模型参数，从而造成显著的通信开销和较长的传输时间。为缓解通信瓶颈，我们采用 HiFL 分层联邦学习范式，充分发挥移动边缘计算的优势，并结合同步客户端-边缘模型聚合与异步边缘-云模型聚合，大幅减少广域网传输的流量。具体而言，我们首先从理论上分析 HiFL 的收敛界，并确定提升模型性能的关键可控因素。随后，我们提出增强版 HiFlash 设计，创新性地结合基于深度强化学习的自适应陈旧度控制与异构性感知的客户端-边缘关联策略，从而提升系统效率并减轻陈旧度效应，同时不牺牲模型准确率。大量实验验证了 HiFlash 在模型准确率、通信量减少和系统效率方面的卓越性能。

索引词—联邦学习、分层机制、陈旧度控制、客户端-边缘节点关联。

1 引言

NOWADAYS，联邦学习（FL）因能够以分布式方式协同训练全局机器学习（ML）模型，同时不泄露私有客户端的 [1], [2]数据，近年来受到越来越多的关注。在 FL 的训练过程中，客户端与云服务器之间会反复交换（本地/全局）模型更新，直至获得精度理想的模型。因此，通过将训练数据保留在本地客户端，FL 实现了隐私保护学习。在这一框架下，涌现出许多热门的 AI 应用，例如计算机视觉 [3], 、语言处理 [4] 和人体活动识别 [5] 。

对于许多现有的 FL 系统，无论采用同步更新（例如 FedAvg [1]及其变体 [6], [7]）还是异步更新（例如 FedAsync [8]），都需要在多轮更新迭代中交换大量模型参数。然而，地理位置分散在网络边缘节点上的客户端通常通过广域网（WAN）和长距离传输连接到远程云服务器，这会产生高昂的通信成本并导致严重的网络拥塞。这种通信效率低下会严重降低大规模分布式训练的系统性能，并进一步阻碍 FL 系统在实际中的广泛部署。因此，提升 FL 通信效率这一研究问题近年来受到了广泛关注 [9]。

分层架构是缓解云服务器巨大通信压力的一种有前景的方案，因为在较低层聚合本地模型后，传输到云端的模型更新数据量可减少一个数量级提前完成。鉴于移动边缘计算（MEC）在实践中的优势，可将边缘节点（例如，5G边缘服务器）设为本地模型聚合的中间节点 [10]。原因如下：1）由于局域网（LAN）中的路由路径更短、跳数距离更小，边缘层的网络时延更低，网络抖动也更小 [11]。此外，云端与客户端之间通信效率较低所导致的拖尾问题也能得到显著缓解；2）与传统联邦学习（FL）中使用广域网（WAN）的高昂货币成本相比，边缘节点丰富且成本更低的局域网资源有助于在现实中部署联邦学习 [12]; 3）联邦学习应用通常分布在大量设备上，这些设备自然聚集成许多边缘域（例如校园和医院） [13]。分层联邦学习能够很好地适应这种分布式模式。

受这些事实的启发，近期提出了一种客户端—边缘节点—云端分层联邦学习的新范式 [14], [15]，该范式包含两个层级的同步模型聚合：下层由边缘节点控制客户端模型聚合，上层由云服务器控制边缘模型聚合。该框架旨在利用同步更新的优势，在下层训练出准确率高、收敛速度快的全局模型，并受益于边缘节点较高的局域网带宽和充足的计算资源。然而，由于边缘异构性（例如，不同的广域网连接状况，以及因客户端规模不同而导致的边缘聚合时间差异）和边缘节点到云端之间的通信瓶颈，上层会出现严重的拖尾问题。具体而言，上层同步全局模型聚合中不可避免地会产生较长的等待时间。

为充分发挥分层联邦学习的优势，本文将同步和异步操作的优点融入分层联邦学习的不同聚合层中，并将其称为 HiFL，以区别于 HierFAVG,

Q. Wu、*X. Chen*、*T. Ouyang*、*Z. Zhou* 和 *X. Zhang* 就职于中国广州 *510006* 的中山大学计算机科学与工程学院。*Shusen Yang* 就职于中国西安 *710049* 的西安交通大学数学与统计学院。*Junshan Zhang* 就职于美国加州大学戴维斯分校电子与计算机工程系。

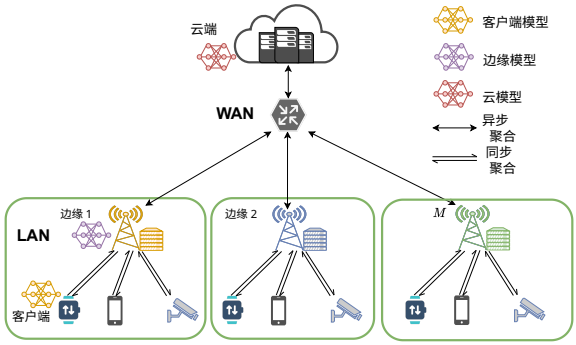


图 1. HiFL 方法概览。

后者是具有两级同步模型聚合的分层联邦学习版本。如图 1 所示，面对边缘节点之间以及边缘节点与云端之间显著的边缘异构性和复杂的广域网环境，我们采用异步更新进行边云模型聚合，通过免等待通信提高学习效率。在较低层，客户端与边缘节点之间的同步模型聚合可确保较高的准确率和快速收敛。此外，在通信高效的单跳接入边缘网络环境中，受益于局域网的高带宽以及边缘节点充足的计算资源，与高时延广域网上的异步边云聚合通信相比，同步客户端-边缘聚合期间的拖尾问题非常轻微，可以忽略。

然而，由于采用异步聚合和分层机制设计，HiFL 也带来了新的挑战。一方面，异步更新中产生的陈旧度效应会对模型准确率和收敛速度造成负面影响[16]。现有的陈旧容忍机制通常仅根据陈旧度 [8]，通过权衡收敛速率与方差缩减来减轻陈旧模型更新的影响。然而，这些机制较少考虑其对系统效率（例如训练时间、资源效率）的影响。例如，陈旧度较高的模型对全局模型的贡献可能微乎其微，导致异步联邦学习需要更多轮通信才能达到目标准确率。因此，控制模型陈旧度对于实现通信高效的模型学习至关重要。另一方面，分层机制会导致边缘节点之间的数据异质性，而分层的客户端—边缘—云端模型聚合可能进一步放大这种异质性，导致模型性能下降 [16]。此外，边缘关联客户端之间的资源异质性会加剧拖尾效应，可能延长客户端-边缘模型聚合中的等待时间。

为应对上述挑战，本文首先研究 HiFL，以深入了解其性能界和系统效率方面的理论特性，并确定影响学习性能的关键可控参数。基于理论研究结果，我们设计了一种用于边缘-云端层的自适应陈旧度控制策略，以及一种用于客户端-边缘层的异质性感知关联机制，以提升 HiFL 的整体效率。在陈旧度控制方面，现有方法通常假设为参与客户端预先设定一个固定阈值，这难以适应现实中的动态环境。此外，由于联邦学习环境复杂（例如客户端的数据异质性和资源异质性，以及联邦学习模型当前所处的运行阶段），确定阈值并非易事。与之不同，我们采用深度强化学习（DRL）方法，并基于深度Q网络（DQN）设计了一个深度强化学习智能体 [17]，使其能够针对动态且复杂的联邦学习环境，明智地自适应确定陈旧度阈值。我们通过双重深度Q网络训练该深度强化学习智能体，以提升效率和鲁棒性。在客户端-边缘节点关联方面，我们设计了一种高效的加权启发式方法，以求得近似最优解，同时最小化边缘节点间的数据异质性以及边缘关联客户端的资源异质性。

总而言之，本文做出了以下贡献：

- 为实现通信高效且准确的模型学习，我们采用 HiFL，这是一种分层联邦学习方法，执行同步客户端-边缘模型聚合和异步边缘-云端模型聚合。我们对 HiFL 的收敛性进行了严格的理论分析，涵盖凸和非凸学习目标。

- 受理论收敛性分析的启发，我们进一步提出了增强版 HiFlash 设计，该设计基于 HiFL，引入了自适应陈旧度控制和考虑异质性的客户端-边缘节点关联。HiFlash 方法可支持大规模部署，同时提升模型性能和系统效率。

- 我们设计了一个基于深度Q网络（DQN）的深度强化学习智能体，通过精心设计学习奖励来实现自适应陈旧度控制，从而在不牺牲模型准确率的前提下提升系统效率。为缓解数据和资源异质性导致的准确率下降和拖尾效应，我们设计了一种复杂度低且高效的客户端-边缘节点关联加权启发式方法，能够良好地权衡模型准确率与系统效率。

- 我们使用三个广泛采用的图像分类数据集进行了大量实验，以评估 HiFlash 的有效性。结果表明，在不牺牲模型准确率的前提下，HiFlash 在通信效率方面显著优于其他基于联邦学习的方法。例如，即使客户端之间的数据分布高度偏斜，HiFlash 仍能实现较高的模型准确率，同时大幅降低通信开销；相较于 HierFAVG 和 FedAvg 基准，其通信开销分别降低了 42% 和 89%。

本文其余部分的组织结构如下：第2节介绍联邦学习和深度强化学习的预备知识。第3节介绍一种名为 HiFL 的分层联邦学习方法。第4节对 HiFL 进行理论分析，并在第5节进一步设计 HiFlash，这是一种增强型 HiFL，具有自适应陈旧度控制和考虑异质性的客户端-边缘节点关联。第6节进行大量实验。第7节回顾相关工作，第8节总结全文。

2 预备知识

2.1 联邦学习

联邦学习 [1], 最早由谷歌于2016年提出，旨在以隐私保护方式在大量客户端之间训练全局共享模型，其中中央服务器充当聚合器，协调客户端模型学习。通常，联邦学习系统由一台云服务器和 N 分散的客户端组成。每个客户端 k 都拥有一个本地数据集 $\mathcal{D}_k = \{\mathbf{x}j,\ y_j\}_{j=1}^{|\mathcal{D}_k|}$，其中 $\mathbf{x}j$ 是

训练样本的特征 j，而 y_j 是其真实标签。为了协同训练全局机器学习模型 $\omega \in \mathbb{R}^d$，与数据样本 $(\mathbf{x}j, y_j)$ 相关联的损失函数记作$f(\mathbf{x}j, y_j, \omega)$，其中 d是可训练参数的总数。为便于说明，我们用 $f_j(\omega)$ 替代 $f(\mathbf{x}j, y_j, \omega)$这一记法。因此，联邦学习的学习目标是最小化 N 客户端训练数据集合上的损失函数，即，

$$\min_{\omega \in \mathbb{R}^d} F(\omega) = \sum_{k=1}^{N} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} F_k(\omega), \tag{1}$$

其中，$F_k(\omega) = \frac{1}{|\mathcal{D}_k|} \sum_{j \in \mathcal{D}_k} f_j(\omega)$ 是客户端 k 上数据样本的损失经验目标函数，其由任务指定，例如，图像分类任务的学习目标可以是交叉熵损失。假设 $\mathcal{D}_k \cap \mathcal{D}_{k'} = \emptyset$ for $k \neq k'$，我们定义 $\mathcal{D} = \cup_{k=1}^{N} \mathcal{D}_k$，并用 $|\cdot|$ 表示集合的大小。

为求解方程 (1) 中的优化问题，应用最广泛的联邦学习框架 FedAvg 提出在抽样选取的客户端子集上并行执行本地随机梯度下降（SGD），并偶尔通过中央服务器进行同步模型聚合 [1]。该过程会持续进行，直到模型达到预期的准确率。由于云服务器与地理位置分散的客户端之间的网络连接速度较慢且成本高昂（例如，频繁回传以及通信距离较远）[10],，FedAvg 会先执行多个本地学习步骤，再将模型更新上传至云服务器，从而大幅减少通信轮次，进一步减轻网络负担。

2.2 深度强化学习

在强化学习（RL）中，强化学习智能体会在离散时隙中与环境交互，以最大化长期奖励。在每个时隙 i，强化学习智能体都会观测状态 $\mathbf{s}^i$，执行动作 a^i 并从环境中获得奖励 r^i。随后，环境状态 $\mathbf{s}^i$ 转移至$\mathbf{s}^{i+1}$，以便在下一个时隙进行动作决策。整个过程遵循马尔可夫决策过程（MDP）[18]，可表示为元组 $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$，其中 $\mathcal{S}$是状态空间，$\mathcal{A}$是动作空间，$\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ 是概率转移函数。$\mathcal{R}$ 是即时奖励函数，$\gamma \in [0, 1]$ 是用于折扣未来奖励的因子。强化学习智能体的目标是学习一种策略 π^*，即状态与动作之间的映射，使累积折扣奖励$R = \sum_{i=1}^{I} \gamma^{i-1} r^i$最大化，其中 I 是总运行时隙数。

为估计期望累积折扣奖励，开始-
从状态$\mathbf{s}^i$出发，基于价值的强化学习方法采用动作价值函数

$$\begin{aligned} Q_\pi(\mathbf{s}^i, a^i) &= \mathbb{E}_\pi[\sum_{\hat{i}=1}^{\infty} \gamma^{\hat{i}-1} r^{i+\hat{i}-1} | \mathbf{s}^i, a^i] \\ &= \mathbb{E}_\pi[r^i + \gamma Q_\pi(\mathbf{s}^{i+1}, a^{i+1}) | \mathbf{s}^i, a^i], \end{aligned} \tag{2}$$

其中，π 是状态-动作映射策略。最优动作价值函数 $Q^*(\mathbf{s}^i, a^i)$ 定义为累积折扣奖励的最大期望：

$$Q^*(\mathbf{s}^i, a^i) = \mathbb{E}_\pi[r_i + \gamma \max_a Q^*(\mathbf{s}^{i+1}, a) | \mathbf{s}^i, a^i]. \tag{3}$$

因此，我们可以应用函数逼近技术来学习动作价值函数 $Q_\pi(\mathbf{s}^i, a^i, \theta_i)$，以逼近最优函数 Q^*。

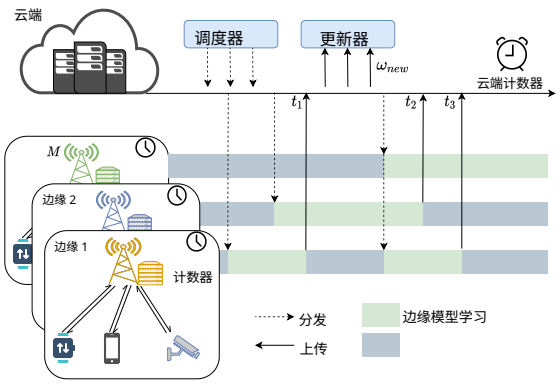


图 2. HiFL 的模型训练流程。

然而，对于许多现实世界的问题，状态空间会变得过于庞大，无法跟踪所有 Q 值。为缓解这一问题，深度强化学习（DRL）提出采用深度神经网络（DNN）作为动作价值函数 Q_π 的逼近器，利用 DNN 强大的泛化能力。例如，深度Q网络（DQN）[17] 使用 DNN 估计状态和动作的 Q 值，其目标是最小化目标 $r^i + \gamma \max_{a^{i+1}} Q_\pi(\mathbf{s}^{i+1}, a^{i+1}, \theta_i)$ 与如下所述逼近器之间的均方误差（MSE）损失：

$$\arg\min_{\theta_i} \mathbb{L}(\theta_i) = (r^i + \gamma \max_{a^{i+1}} Q_\pi(\mathbf{s}^{i+1}, a^{i+1}, \theta_i) - Q_\pi(\mathbf{s}^i, a^i, \theta_i))^2. \tag{4}$$

为便于收敛，深度Q网络将深度强化学习转化为监督学习的形式，并引入经验回放 [19],，其中包含丰富的转移样本，以降低样本间的相关性。

3 HIFL 设计

3.1 问题定义与 HiFL 概述

我们首先基于传统联邦学习给出 HiFL 的问题定义。如图 2 所示，客户端模型更新不会直接发送到云端，而是发送到边缘节点。具体而言，N参与客户端根据其特征（例如地理位置）划分为 M 个互不重叠的组，每组对应一个边缘节点。通常，边缘节点的数量远少于客户端，即 $M \ll N$。我们将 $\mathcal{C}^m$ 表示为边缘节点 m的客户端集合，所有参与客户端可定义为 $\mathcal{N} = \cup_{m=1}^{M} \mathcal{C}^m$。因此，基于这种分层联邦学习架构，方程 (1) 中的学习目标可扩展为：

$$\min_{\omega \in \mathbb{R}^d} F(\omega) = \sum_{m=1}^{M} \frac{|\mathcal{D}^m|}{|\mathcal{D}|} F^m(\omega), \tag{5}$$

其中，$F^m(\omega) = \sum_{k \in \mathcal{C}^m} \frac{|\mathcal{D}_k|}{|\mathcal{D}^m|} F_k(\omega)$ 表示边缘节点 m 上的目标函数，它是 $\mathcal{C}^m$ 中各客户端经验目标函数的线性组合。$|\mathcal{D}^m| = \sum_{k \in \mathcal{C}^m} |\mathcal{D}_k|$ 表示与边缘节点 m 关联的所有客户端的样本总数据量。表1列出了本文中的关键符号说明。

基于云端和边缘节点的不同行为，HiFL 采用同步客户端-边缘模型聚合和异步边缘-云模型聚合，以经济高效的方式协同训练高质量的机器学习模型。在边缘侧，较高的局域网带宽和较低的网络抖动大幅缩短了梯度交换的通信延迟

表1 关键符号说明。

符号	描述
一般联邦学习设置	
N	客户端数量
M	边缘节点数量
$\mathcal{D}_k$	客户端 k的数据集
$\mathcal{C}^m$	与边缘节点 m关联的客户端集合
t_e, t_c	边缘计数器和云端计数器
η	联邦学习的学习率
$\omega(t_c)$	第 t_c次全局迭代时的云模型
$\omega^m(t_c,\ t_e)$	基于 t_e计算的第 $\omega(t_c)$次迭代时的边缘模型
$\omega_k(t_c,\ t_e)$	基于 t_e计算的第 $\omega(t_c)$次迭代时的客户端模型
在客户端-边缘模型聚合中	
H	客户端-边缘模型聚合次数
c	客户端的学习轮次 与边缘节点同步之前
在边缘-云模型聚合中	
τ	边缘模型的陈旧度
α, υ	边缘模型的初始模型权重和惩罚项 用于计算混合超参数的系数 α_τ 在云端模型聚合中
在自适应陈旧度控制中	
$\mathbf{s}^i, a^i$	时隙 i中深度强化学习智能体的状态和动作
r^i	时隙 i的系统总成本
σ_1, σ_2	计算和通信成本的惩罚因子
在异质性感知关联中	
λ	数据和资源异质性的加权参数
$L^{m,k}$	与边缘节点 m关联的客户端 k 的响应延迟

或模型下载。在同一局域网环境中，边缘节点与客户端之间更适合采用同步模型聚合，因为这种方式具有较高的模型训练精度和较快的收敛速度。然而，在云端，模型训练会受到复杂广域网环境下通信瓶颈的影响（例如，长距离传输时间波动较大，而且不同边缘节点连接的客户端数量不同，导致边缘模型聚合时间各异），进而引发严重的拖尾问题。因此，采用异步聚合，通过减少中央服务器与边缘节点之间模型更新的等待时间来缓解拖尾效应。

如图 2 所示，我们设计了两个核心组件，即调度器和更新器，它们在云服务器上异步并行运行，以实现无等待目标。前者负责分发最新模型（可与后文指定的深度强化学习智能体集成控制功能），后者则负责全局模型聚合。具体而言，一旦某个空闲边缘节点开始执行其感兴趣的模型训练，就会主动通知云服务器下载最新版本的全局模型。随后，云服务器会检查其更新器，并立即将结果发送给相应的边缘节点。收到全局模型后，边缘节点会迅速将其广播给关联的客户端，并利用它们的本地数据集，通过高效的客户端—边缘通信，以同步方式协同训练共享边缘模型。如果云服务器收到来自某个边缘节点的训练模型，更新器就会立即更新全局模型，而无需等待其他边缘节点。为控制异步聚合造成的模型陈旧度，更新器会对收到的模型更新施加权重惩罚后进行模型聚合，具体内容将在第3节第3.2节中详细阐述。

请注意，为了提高 HiFL 的吞吐量，多个边缘模型训练过程可以并行执行，从而产生多个针对全局模型使用读写锁的更新器线程。这种异步模型聚合策略缓解了云端的网络拥塞，并实现了无等待的全局模型学习，进一步降低通信开销并加快模型训练过程。具体而言，由于客户端在没有云服务器协调的情况下更新边缘模型，因此在这种分层设置中设置了不同的计数器来记录模型更新时间。因此，我们分别设计了用于异步云端模型聚合和同步边缘模型聚合的云端计数器t_c 和边缘计数器 t_e 。

3.2 HiFL训练过程

HiFL的学习过程包含两个主要步骤：1）客户端-边缘模型聚合；2）边缘-云模型聚合，具体如下所述。

1）同步客户端-边缘模型聚合。当边缘节点 m 从云服务器收到当前全局模型 $\omega(t_c)$ 后，边缘模型将初始化为 $\omega^m(t_c,\ t_e)\ =\omega(t_c)$ ，其中 $t_e=0$；这里， t_c 表示在云模型上执行的全局聚合次数， t_e 表示收到云模型 $\omega(t_c)$ [1]后，为边缘模型聚合而执行的本地模型更新次数。随后，边缘模型会发送给关联的客户端，以更新客户端模型。

我们采用广泛使用的 FedAvg算法 [1] 进行协同训练，以获得令人满意的边缘模型。具体而言，在第 t_e 次迭代中，客户端 $k\in\mathcal{C}^m$ 使用其本地数据执行模型更新。为降低通信开销，经典的 FedAvg 方法会聚合与边缘节点 m 关联的所有客户端模型，并在每个客户端执行 c 步本地更新后，与边缘模型 ω^m 同步。设 $\omega_k^m(t_c,\ t_e)$ 为客户端 $k\in\mathcal{C}^m$的模型参数，则 $\omega_k^m(t_c,t_e)$ 按如下方式演化：

$$\omega_k^m(t_c,t_e)=\begin{cases}\omega_k^m(t_c,t_e-1)-\eta\nabla F_k(\omega_k^m(t_c,t_e-1)), & t_e \bmod c\neq 0\\ \omega^m(t_c,t_e), & t_e \bmod c=0\end{cases} \tag{6}$$

其中

$$w^m(t_c,t_e)=\sum_{k\in\mathcal{C}^m}\frac{|\mathcal{D}_k|[\omega_k^m(t_c,t_e-1)-\eta\nabla F_k(\omega_k^m(t_c,t_e-1))]}{|\mathcal{D}^m|}. \tag{7}$$

不失一般性，我们假设边缘节点在每个全局训练轮次中执行若干次 H 模型聚合（例如， $t_e\leq Hc$），这意味着在客户端-边缘模型聚合过程中，某个客户端已执行 Hc 次客户端模型更新。完成此次协同模型训练后，边缘模型更新为 $w^m(t_c,\ Hc)$ ，并将与云服务器异步更新全局模型。

2）异步边缘-云模型聚合。边缘-云模型聚合中的异步机制带来了陈旧度问题，因为多个边缘节点可以自由地在任意时间执行模型训练和上传。例如，在全局计数器 t_c处，云服务器收到一个陈旧模型 $\omega^m(t_c-\tau,\ Hc)$，该模型由边缘节点 m 基于全局模型 $\omega(t_c-\tau)$训练得到，其中 τ 表示边缘模型的陈旧度。由于边缘模型是基于过时的云端模型版本训练的，该陈旧模型会给云端模型训练过程带来噪声，减缓甚至阻碍训练收敛 [20]。

为控制异步性造成的误差，HiFL 引入混合超参数 α_τ ，如 [8],所示，并使用过时的边缘模型更新全局模型。

$$\omega(t_c)=(1-\alpha_\tau)\omega(t_c-1)+\alpha_\tau\omega^m(t_c-\tau,Hc), \tag{8}$$

1. t_c 的最大值可由云服务器根据模型的收敛情况确定，而 t_e 的最大值可由各边缘节点自行设定，因此在不同节点间可能有所不同。

其中， α_τ 是陈旧度为 τ 的边缘模型 $\omega^m(t_c-\tau,\ Hc)$对全局模型的贡献权重。较小的 α_τ会导致更多的联邦学习训练轮次，而较大的 α_τ则可能引起较大的准确率波动。通过调整α_τ的值，我们可以在模型学习过程中自适应地控制收敛速度与方差缩减之间的权衡。本文使用以下指数函数来确定 α_τ的值，

$$\alpha_\tau = \alpha \cdot \upsilon^\tau, \tag{9}$$

其中 $\alpha \in (0,1)$ 是边缘模型的初始模型权重。我们可以减小 α，以通过惩罚系数 $\upsilon \in (0,1)$来缓解大陈旧度τ 造成的误差。

HiFL中的云服务器和边缘节点会异步执行模型更新，直至云模型收敛。不同客户端组上的同步客户端-边缘聚合可以并行执行，而异步边缘-云模型聚合则避免了长时间等待；二者共同促进了快速模型学习和无等待通信。HiFL算法的详细内容见算法1。

算法1 HiFL训练过程

输入： 来自 N个分布式客户端 $\{\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_N\}$的数据集，本地更新次数 c和学习率 η。

1: 根据某些预定义标准进行客户端聚类（例如，地理位置）或采用异构性感知的客户端-边缘关联策略（算法3）
2: **云服务器执行**:
3: 初始化云模型 ω, $t_c \leftarrow 0$
4: **调度器:**
5: 定期分发全局模型 ω（t_c）和全局计数器 t_c 发送至一个边缘节点以更新边缘模型
6: **更新器:**
7: **对于** $t_c = 1,\ 2,\ .....,\ T_c$ **执行**
8: 从一个边缘节点接收一对 $(\omega^m(\tilde{t}_c,\ Hc)\ \tilde{t}_c)$
9: 计算模型陈旧度 $\tau = t_c -$ $\tilde{t}_c$
10: 使用方程（8）更新云模型
11: **结束循环**
12: **边缘节点执行**:
13: **对于** $m \in 1、2、.....、M$ 并行执行**执行**
14: **如果** 收到一对云模型和云端计数器 $(\omega(t_c)\ t_c)$ 来自调度器**则**
15: Set $t_c = t_c$ 和 $\tilde{\omega}^m(\ \tilde{t}_c,0) = \omega(t_c)$
16: 初始化 $\omega_k^m(\ \tilde{t}_c,\ 0) = \omega^m(\ \tilde{t}_c,\ 0)$ 对于 $k \in \mathcal{C}^m$
17: **对于** $t_e = 1,\ .....,\ Hc$ **执行**
18: /*客户端执行 */
19: **对于** $k \in \mathcal{C}^m$ **执行**
20: 根据方程 (6) 计算当前客户端模型
21: **结束循环**
22: **结束循环**
23: 发送 $(\omega^m(\tilde{t}_c, Hc),\tilde{t}_c)$ 至云服务器。
24: **结束条件判断**
25: **结束循环**

输出： 云模型 $\omega(T_c)$。

4 收敛性分析

4.1 定义与假设

为进行分析，我们对损失函数引入以下定义和假设。

假设 ***1.*** *(光滑性).* 函数 $F_k(\omega)$ 是 β-光滑的，若$\forall\omega,\omega'$，

$$||\nabla F_k(\omega) - \nabla F_k(\omega')|| \le \beta||\omega - \omega'||, \tag{10}$$

其中 $\beta > 0$。

假设 ***2.*** *(强凸性)*。函数 $F_k(\omega)$ 是 μ- 强凸的，当 $\forall\omega,\omega'$，

$$\langle \nabla F_k(\omega'), \omega - \omega' \rangle + \frac{\mu}{2}||\omega - \omega'||^2 \le F_k(\omega) - F_k(\omega'), \tag{11}$$

其中 $\mu \ge 0$。注意，如果 $\mu = 0$, $F_k(\omega)$ 是凸的。

假设***3***。（弱凸性）。函数 $F_k(\omega)$ 是 μ-弱凸的，当且仅当函数 $G_k(\omega) = F_k(\omega) + \frac{\mu}{2}||\omega||^2$ 是凸的，其中 $\mu \ge 0$。具体而言，当 $\mu = 0$ 时， $F_k(\omega)$ 是凸的；当 $\mu > 0$时，它可能是非凸的。

假设***4***。（利普希茨）。函数 $F_k(\omega)$ 是 ρ-利普希茨的，如果$\forall\omega$, ω'，

$$||F_k(\omega) - F_k(\omega')|| \le \rho||\omega - \omega'||. \tag{12}$$

在这些假设下，引理1适用于边缘模型和云端模型的损失函数。

引理***1***。 $F^m(\omega)$ 和 $F(\omega)$ 是 μ-强凸、 β-光滑且 ρ-利普希茨的。

证明： 由上述假设、 $F^m(\omega)$和 $F(\omega)$ 的定义以及三角不等式，可直接得出。 □

在 HiFL 的云模型训练过程中，有两个层级的模型聚合并行进行：客户端-边缘模型聚合和边缘-云模型聚合。接下来[6], 我们在定义 1 中引入虚拟集群模型学习的概念，以求得通过同步客户端-边缘模型聚合训练的边缘模型与虚拟集群模型之间的损失偏差；在虚拟集群模型中，假设训练数据存在于虚拟中央存储库中。随后，我们在假设 5 中形式化定义**基于集群的梯度偏差**，以描述客户端与边缘节点之间数据分布的差异对 HiFL 的影响。

定义 ***1.*** *(虚拟集群模型学习)*。对于客户端-边缘模型聚合，我们使用简写符号 $[h] = [(h-1)c, hc)$ 表示两次连续边缘模型聚合之间的时间间隔。给定与边缘节点 m 关联的某个客户端簇 $\mathcal{C}^m$，以及初始化的边缘模型$\omega^m(t_c,0) = \omega(t_c)$，对于任意间隔 $[h]$, $h = 1,2,...,H$，虚拟集群模型 $v_{[h]}^m(t_c,t_e)$ 通过对 $\mathcal{C}^m$ 拥有的集中式数据样本 $\mathcal{D}^m$ 执行梯度下降来更新，并在每个间隔开始时与联邦边缘模型 ω^m同步，如方程 (13) 所示，

$$v_{[h]}^m(t_c,t_e) = \begin{cases} v_{[h]}^m(t_c,t_e-1) - \eta\nabla F^m(v_{[h]}^m(t_c,t_e-1)), & t_e \bmod c \ne 0 \\ \omega^m(t_c,t_e), & t_e \bmod c = 0. \end{cases} \tag{13}$$

假设 ***5***。（基于集群的梯度偏差）。对于任意客户端 $k \in \mathcal{C}^m$，假设 δ_k^m 是客户端 k 的局部损失函数与边缘节点 m 的边缘节点损失函数之间梯度差异的上界，可表示如下:

$$||\nabla F_k(\omega) - \nabla F^m(\omega)|| \le \delta_k^m, \tag{14}$$

然后，对于与边缘节点 m 关联的客户端簇 $\delta^m \triangleq \frac{\sum_{k\in\mathcal{C}^m}|D_k|\delta_k^m}{\sum_{k\in\mathcal{C}^m}|D_k|}$，我们将 δ_{max} 设为跨越 $\{\delta^m\}_{m=1}^M$ 的最大梯度差异。

我们假设 Δ 是任意边缘节点 m 的损失函数与全局损失函数之间梯度差异的上界，即，

$$||\nabla F^m(\omega) - \nabla F(\omega)||^2 \leq \Delta. \tag{15}$$

我们将 δ_{max} 称为客户端-边缘节点发散度，将 Δ 称为边缘节点-云端发散度。此外，任意客户端 k 上随机梯度的期望平方范数均假设为一致有界，即，

$$\mathbb{E}||\nabla F_k(\omega)||^2 \leq V. \tag{16}$$

对于 μ-弱凸损失函数 F_k（当 $\mu > 0$ 时，该函数可以是非凸的），我们定义 $G_{\tilde{\omega}} = F(\omega) + \frac{\tilde{\mu}}{2}||\omega - \tilde{\omega}||^2$，其中 $\tilde{\mu} > \mu$。与凸情形类似，我们假设 $G_{\tilde{\omega}}$ 是 β-光滑且 $\tilde{\rho}$-利普希茨的。$\forall \tilde{\omega} \in \mathbb{R}_d$，我们有 $||\nabla G_{k,\tilde{\omega}}(\omega) - \nabla G^m_{\tilde{\omega}}(\omega)|| \leq \tilde{\delta}^m_k$，$\tilde{\delta}^m \triangleq \frac{\sum_{k\in\mathcal{C}^m}|D_k|\delta^m_k}{\sum_{k\in\mathcal{C}^m}|D_k|}$，其中 $\tilde{}$ 表示与边缘节点 m 关联的客户端簇，而 δ_{max} 表示跨 $\{\tilde{\delta}^m\}_{m=1}^M$ 的最大梯度差异。此外，我们假设 $||\nabla G^m_{\tilde{\omega}}(\omega) - \nabla G_{\tilde{\omega}}(\omega)||^2 \leq \tilde{\Delta}$ 且 $||\nabla G_{k,\tilde{\omega}}(\omega)||^2 \leq \tilde{V}$。

4.2 HiFL 的收敛

根据上述假设和定义，我们得到以下收敛保证。

引理 ***2***。在客户端-边缘模型聚合期间，对于任意区间$[h]$ 和 $t_e \in [h]$，我们有

$$||\omega^m(t_c, t_e) - v^m_{[h]}(t_c, t_e)|| \leq g(t_e - (h-1)c), \tag{17}$$

其中

$$g(x) = \frac{\delta_{max}}{\beta}((\eta\beta + 1)^x - 1) - \eta\delta_{max}x, \tag{18}$$

对于任意 $x = 0,\ 1,\ 2,\ \cdots$。

此外，由于 $F^m(\cdot)$ 是 ρ-利普希茨的，因此有 $F^m(\omega^m(t_c,\ t_e)) - F^m(v^m_{[h]}(t_c,\ t_e)) \leq \rho g(t_e - (h-1)c)$。

证明：详细内容请参阅单独的补充文件附录 A。 □

因此，当客户端-边缘模型聚合完成时，例如 $t_e = Hc$，由联邦学习训练的边缘模型与虚拟集群模型之间的损失偏差为 $F^m(\omega^m(t_c,\ Hc)) - F^m(v^m_{[h]}(t_c,\ Hc)) \leq \rho g(c)$。借助权重偏差上界，我们现在可以证明 HiFL 在凸和非凸损失函数下的收敛性。

定理 ***1***。假设损失函数 F_k 是 μ-强凸的，并且每个边缘节点在将边缘模型推送至云服务器之前执行 $H \in [H_{min}, H_{max}]$ 次客户端-边缘聚合。令 $\eta < \frac{1}{\beta}$，则云服务器经过 T_c 次全局更新后，HiFL 的收敛上界可表示为：

$$\mathbb{E}[F(\omega(T_c) - F(\omega^*)] \leq \kappa[F(\omega(0)) - F(\omega^*)] + (1-\kappa)\frac{A_1V + A_2\delta_{max} + A_3\Delta}{B}, \tag{19}$$

其中 $\kappa = (1 - \alpha_\tau + \alpha_\tau(1-\eta\mu)^{cH_{min}})^{T_c}$、$A_1 = \frac{1}{2\mu}$、$A_2 = \rho H_{max}\frac{\frac{(\eta\beta+1)^c - 1}{\beta} - \eta c}{1 - (1-\eta\mu)^{cH_{min}}}$、$A_3 = \frac{cH_{max}\eta}{2}$ 以及 $B = (1-\eta\mu)$。

证明：详情请参阅单独补充文件中的附录 B。 □

定理 ***2***。假设损失函数 F_k 是 μ-弱凸的（当 $\mu > 0$时可以是非凸的），并且每个边缘节点在将边缘模型推送到云服务器之前执行 $H \in [H_{min}, H_{max}]$ 次客户端-边缘聚合。令 $\eta < \min(\frac{1}{\beta}, \frac{2}{\tilde{\mu}-\mu})$，则经过 T_c 次全局更新后，云服务器上的 HiFL 收敛上界可表示为：

$$\mathbb{E}[F(\omega(T_c) - F(\omega^*)] \leq \tilde{\kappa}[F(\omega(0)) - F(\omega^*)] + (1-\tilde{\kappa})\frac{\tilde{A}_1\tilde{V} + \tilde{A}_2\tilde{\delta}_{max} + \tilde{A}_3\tilde{\Delta}}{\tilde{B}}, \tag{20}$$

where $\tilde{\kappa} = (1 - \alpha_\tau + \alpha_\tau[(1 - \frac{\eta(\tilde{\mu}-\mu)}{2})^{cH_{min}}])^{T_c}$, $\tilde{A}_1 = \frac{12\rho^2H_{max}}{(\tilde{\mu}-\mu)^3} + \frac{5\tilde{\mu}-\mu}{2(\tilde{\mu}-\mu)^2}$, $\tilde{A}_2 = \frac{\tilde{\rho}H_{max}}{\tilde{\beta}}((\eta\tilde{\beta}+1)^c - 1 - \tilde{\beta}\eta c)$, $\tilde{A}_3 = \frac{2H_{max}}{\tilde{\mu}-\mu}$ and $\tilde{B} = 1 - (1 - \frac{\eta(\tilde{\mu}-\mu)}{2})^{cH_{min}}$.

Proof: We first give the 客户端模型与虚拟集群模型之间的收敛保证，并在单独的补充文件附录 C 中详细说明云模型的收敛性分析。 □

基于定理 1 和定理 2，我们得出以下两条关于 HiFL 收敛性的值得注意的注释。

注释 ***1***。（收敛速率。）超参数 α_τ 控制 HiFL 的收敛速率。由于 α_τ 随着 τ 的减小而增大，因此采用较小的 τ 时，随着全局聚合总次数 T_c 的增加，κ 会更快地趋近于 0，这意味着收敛速率更快。

注释 ***2***。（收敛界。）当 $T_c \to \infty$、$\kappa \to 0$ 时，对于强凸函数，收敛界可简化为 $\frac{A_1V + A_2\delta_{max} + A_3\Delta}{B}$，其主要受客户端 V 的随机梯度、客户端-边缘节点偏差 δ_{max} 以及边缘-云端偏差 Δ 的影响。这里，两个耦合项 δ_{max} 和 Δ 的取值由客户端-边缘关联策略决定。对于弱凸函数，也可以观察到类似的结论。

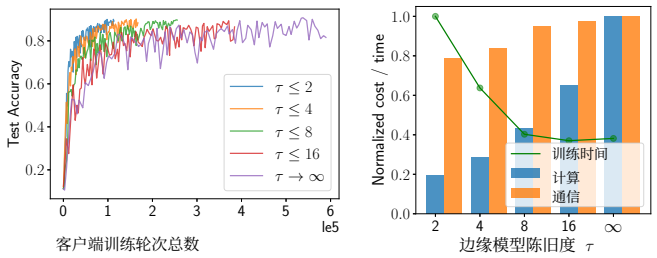
注释 ***3***。（τ 对收敛界的影响。）公式（19）的右侧可以重新表述为 $U = (C_1)^{T_c}(C_2 - C_3) + C_3$，其中 $C_1 = 1 - \alpha_\tau + \alpha_\tau($ $1 - \eta\mu)^{cH_{min}}$，$C_2 = F(\omega(0)) - F(\omega^*)$ $C_3 = \frac{A_1V + A_2\delta_{max} + A_3\Delta}{B} - C_2$（）以及 。由于 通常非常大，我们假设 $C_2 - C_3 > 0$。因此，U 随 C_1 单调增加。由于 C_1 随 τ增加，在实践中给定 T_c 的固定值时，较大的 τ 表示更大的上界 U。对于弱凸函数，也可以观察到类似现象。

5 HIFLASH：具有自适应陈旧度控制和异质性感知的客户端-边缘节点关联的 HiFL

在本节中，基于上述理论分析，我们首先对具有不同模型陈旧度值和客户端-边缘关联机制的 HiFL 性能进行初步评估。受实验结果中的经验性洞察启发，我们随后设计了增强版 HiFL，命名为 HiFlash，通过自适应陈旧度控制和异质性感知的客户端-边缘关联实现高效率。

5.1 HiFL 在部署中的性能

我们首先以 MNIST 数据集 [21]为例，从模型性能和系统成本两个角度，研究不同模型陈旧度下 HiFL 的模型训练性能。



(a) (b)

图 3. 在 MNIST 数据集上，边缘模型陈旧度阈值 τ 取不同值时，HiFL 在不同方面的模型性能。(a) 显示，τ 的值越大，准确率波动越剧烈。(b) 展示了 τ 取不同值时 HiFL 的客户端计算成本、边缘节点通信成本和训练时间。

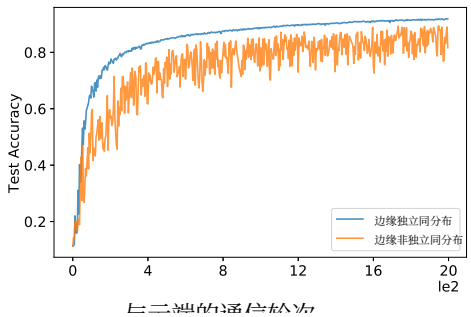


图4. 与边缘非独立同分布相比，边缘独立同分布关联策略能够减小边缘-云端模型差异，最终实现更快收敛和更高的模型准确率。

如图 3(a) 所示，$\tau \leq 8$ 表示所有边缘节点具有相同的最大陈旧度阈值，即 8。因此，只要 τ 小于 8，每个边缘节点都可以将训练好的、具有不同 τ 的边缘模型上传至云端。对于不进行陈旧度控制的 HiFL（例如，$\tau \to \infty$），无论边缘模型陈旧度如何，边缘节点上传的所有边缘模型都可用于更新云模型。可以看到，对于采用$\tau \to \infty$的 HiFL，由于较大的模型陈旧度会导致收敛速度变慢且准确率大幅波动，客户端需要进行 5.3×10^5 个训练轮次才能达到 0.9 的目标测试准确率。相比之下，采用陈旧度限制调整的 HiFL 仅允许使用陈旧度较低的边缘模型更新，从而确保令人满意的收敛速度。例如，当 $\tau \leq 2$时，云模型可以在客户端进行 9.8×10^4 个训练轮次内达到 0.9 的测试准确率，计算成本不到 $\tau \to \infty$情况下的 $1/5$。

此外，在现实的大规模联邦学习系统中，应更准确地量化并综合考虑系统总成本。因此，我们为 HiFL 引入了三项性能指标：云端的总训练时间、边缘节点的通信成本和客户端的计算成本。如图 3(b) 所示，长期来看，未进行陈旧度控制的 HiFL 会产生更高的通信成本和计算成本。不过，较小的 τ 表示模型并行度降低，即允许同时训练模型的边缘节点更少，这会显著延长联邦学习模型学习的训练时间。例如，在采用 $\tau = 0$的 HiFL 中，只有其他所有边缘节点都处于空闲状态时，某个边缘节点才能进行模型训练。因此，陈旧度控制对于 HiFL 实现快速且具成本效益的模型学习至关重要，必须精心设计，以便在训练时间和成本效益之间实现更好的权衡。

由于数据异质性 [22]可能是联邦学习中的一个关键问题，我们研究了通过多种不同的客户端-边缘关联策略，客户端-边缘端差异和边缘端-云端差异的变化对 HiFL 的影响。从结果中可以得到一个有益的发现，即边缘端-云端差异

Δ作为主导因素，会对云模型准确率产生负面影响。如图4所示，我们考虑一个包含云服务器、10个边缘节点和100个客户端的联邦学习系统。每个客户端在 MNIST 数据集中仅拥有一个类别的样本。边缘独立同分布表示将客户端分组到不同的边缘组中，且边缘节点上的数据分布为独立同分布（例如，10个类别的样本数量相同）。而在边缘非独立同分布的情况下，一个边缘节点所持有的样本来自5个类别。边缘独立同分布关联策略会将边缘-云端模型差异较小的客户端分到一起，最终实现快速收敛和较高的准确率。因此，对于具有非独立同分布数据分布的客户端，需要一种能够考虑异质性的客户端-边缘关联策略，使边缘节点上的数据分布接近全局独立同分布。

受上述观察的启发，我们设计了 HiFlash，这是一种增强型 HiFL 方法，在边缘-云端层采用自适应陈旧度控制，并在客户端-边缘层采用异质性感知关联，具体如下所述。

5.2 边缘-云端层的自适应陈旧度控制

固定陈旧度控制（例如，图3中的 $\tau <= 8$ ）要求为所有边缘节点预先设定陈旧度阈值，这在复杂的动态联邦学习环境中可能效果不佳（例如，边缘节点的通信能力高度动态变化、当前训练边缘节点的数量随时间变化），并进一步降低模型性能。因此，我们为愿意参与全局模型训练的边缘节点设计了自适应边缘陈旧度阈值，该阈值基于其控制域的状况（例如，客户端的计算资源、边缘节点的通信能力和训练时间）确定。具体而言，我们构建系统成本模型，并采用深度强化学习方法动态控制陈旧度阈值。

系统成本模型。为充分刻画环境动态，我们采用时隙结构进行陈旧度控制，将长期时间范围划分为一系列离散时隙。请注意，每个时隙的长度通常较短，因此我们假设在时隙 i 开始时，至多有一个边缘节点向云端发送签到请求。同样，在一个时隙结束时，至多有一个边缘节点完成边缘模型训练并向云服务器上传模型更新。我们将时隙 i 中边缘节点的运行/空闲模式定义为 $\mathbf{e}^i$，它由在前一时隙未完成边缘模型训练任务的边缘节点 $\tilde{\mathbf{e}}^i$ ，以及发送签到请求并获云服务器接受、参与联邦学习训练的边缘节点 $\acute{\mathbf{e}}^i$组成。图5展示了 $\mathbf{e}^i$、 $\tilde{\mathbf{e}}^i$ 和 $\acute{\mathbf{e}}^i$ 各定义之间关系的示例。

1）计算成本：在给定时隙 i，我们将边缘节点 m 的计算成本定义为其关联客户端的计算成本总和：

$$C_{comp}^{i,m} = \sum_{k \in \mathcal{C}_m} C_{comp}^{i,k}, \tag{21}$$

其中 $C^{i,}_{comp}{}^{k}$ 表示客户端 k的计算成本。与许多现有研究类似 [23], [24], [25]，根据实证测量研究 [14], ，我们假设 $C^{i,}_{comp}{}^{k} = cD_k\zeta_k$ $f_{i,k}$ ，其中 $f_{i,k}$ 表示客户端 k 在时隙 i的处理速度，且 ζ_k

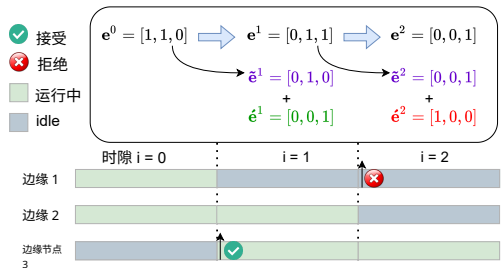


图5. $\mathbf{e}^i$、 $\tilde{\mathbf{e}}^i$ 和 $\acute{\mathbf{e}}^i$之间关系的示意图。

是客户端 k [2]的处理密度。 D_k 是客户端 k 每次本地迭代的训练数据总比特数， c是本地迭代次数。因此， c 与 D_k的乘积表示客户端 k的工作负载。时隙 i 所有客户端的计算成本表示为

$$C^i_{comp}=\sum_{m=1}^{M} e^i_m \cdot C^{i,m}_{comp}, \tag{22}$$

其中， $e^i_m \in \{0,1\}$ 表示时隙 i边缘节点m 的空闲/运行模式。

2) 通信成本：时隙i的通信成本表示为

$$C^i_{comm}=\sum_{m=1}^{M}(e^i_m-\tilde{e}^{i+1}_m)\cdot C^{i,m}_{comm}, \tag{23}$$

其中， $C^i_{comm}=\sum_{k\in\mathcal{C}_m} C^{i,mk}_{comm}$ ，是边缘节点 m 在时隙 i的通信成本。随后， [26], 边缘节点 m 与客户端 k 在时隙 i 之间的通信成本计算为 $C^{i,mk}_{comm}=$ $\mathbb{M}_\omega\times 32$ ，比特/参数$B_{i,m,k}\log_2($1+信噪比)，其中，$B_{i,m,k}$是边缘节点 m 在时隙 i为客户端 k 分配的带宽， $\mathbb{M}_\omega$是 ω 的参数数量，信噪比设为17dB。

用于自适应陈旧度控制的深度强化学习智能体。模型陈旧度控制的主要目标是在实现目标准确率（例如目标准确率Ω）的同时，最小化HiFL系统的总系统成本（包括云端的总训练时间、边缘节点的通信成本以及客户端的计算成本）。由于复杂的联邦学习学习环境，我们设计了一种基于深度强化学习的经验驱动算法，用于自适应陈旧度控制。我们首先将自适应陈旧度阈值优化问题表述为如下马尔可夫决策过程：

1) 状态：在每个时隙 t，系统状态由三类信息组成，用于描述当前的 HiFL 训练环境，具体如下：

- 边缘训练性能信息包括估算的计算成本 $\tilde{\mathbf{C}}^i_{comp}=[\tilde{C}^{i,1}_{comp},...,\tilde{C}^{i,M}_{comp}]$ 、估算的通信成本 $\tilde{\mathbf{C}}^i_{comm}=[\tilde{C}^{i,1}_{comm},...,\tilde{C}^{i,M}_{comm}]$，以及每个边缘节点完成边缘模型计算所需的估算时隙数 $\tilde{\mathbf{T}}^i_{train}=[\tilde{T}^{i,1}_{train},...,\tilde{T}^{i,M}_{train}]$。
- 当前运行中的边缘节点信息以当前边缘节点的剩余训练时间表示 $\tilde{\mathbf{e}}^i$，记为 $\tilde{\mathbf{T}}^i_{rem}=[\tilde{T}^{i,1}_{rem},...,\tilde{T}^{i,M}_{rem}]$，其中 $\tilde{T}^{i,m}_{rem}=0$若边缘节点 m处于空闲状态（例如 $\tilde{e}^i_m=0$）。
- 当前边缘节点签到请求的信息$\acute{\mathbf{e}}^i$ 表明将通知该边缘节点陈旧度阈值。请注意，每个时隙最多只会有一个来自空闲边缘节点的签到请求（例如， $|\acute{\mathbf{e}}^i|\in\{0,1\}$）。此外，发起签到请求的边缘节点 m 不属于当前运行中的边缘节点集合，这意味着 $\sum_{m=1}^{M}\acute{e}^i_m\cdot\tilde{T}^{i,m}_{rem}=0$。

总之，状态可表示为 $\mathbf{s}^i=[\tilde{\mathbf{C}}^i_{comp},\tilde{\mathbf{C}}^i_{comm},\tilde{\mathbf{T}}^i_{train},\tilde{\mathbf{T}}^i_{rem},\acute{\mathbf{e}}^i]$，$\mathbf{C}$，$\mathbf{C}$，$\mathbf{T}$，$\mathbf{T}$，$\mathbf{e}$。值得注意的是，状态中的估计成本和训练时间信息可由边缘节点与云服务器共同分析并收集，使云服务器能够获知发出签到请求的边缘节点的成本信息。

2) 动作：在每个时隙 i开始时，深度强化学习智能体需要根据当前状态 $\mathbf{s}^i$，为请求签到的空闲边缘节点决定可容忍的最大陈旧度 a^i （$\acute{e}^i_m=1$）。本文为陈旧度阈值 a^i 设置上限 τ_{max} 和下限 -1，其中下限表示云服务器拒绝签到请求。此处的拒绝操作可以调整运行中的边缘节点数量（控制模型并行度），从而减轻拖尾效应。因此，动作 a^i的决策空间为 $\{-1,0,1,...,\tau_{max}\}$。执行动作 a^i 后，边缘节点的空闲/运行模式将根据以下公式发生变化

$$\mathbf{e}^i=\begin{cases}\tilde{\mathbf{e}}^i+\acute{\mathbf{e}}^i, & a^i>=0,\\ \tilde{\mathbf{e}}^i, & a^i<0.\end{cases} \tag{24}$$

3) 奖励：我们将奖励函数 r^i 定义为时隙 i的系统总成本：

$$r^i=-\sigma_1 C^i_{comp}-\sigma_2 C^i_{comm}-1, \tag{25}$$

其中， σ_1 和 σ_2 分别是计算成本和通信成本的两个惩罚因子。

深度强化学习智能体旨在找到能够最小化长期系统成本的动作（陈旧度阈值），同时确保达到一定的学习质量（例如，达到目标准确率 Ω）。因此，该问题可表述为最大化从时隙 i 开始的累积折扣奖励期望，具体如下：

$$R^i=\sum_{\hat{i}=1}^{I}\gamma^{\hat{i}-1}r^{i+\hat{i}-1}=\sum_{\hat{i}=1}^{I}\gamma^{\hat{i}-1}(-\sigma_1 C^{i+\hat{i}-1}_{comp}-\sigma_2 C^{i+\hat{i}-1}_{comm}-1), \tag{26}$$

其中， $\gamma\in(0,1]$ 是用于折扣未来奖励的因子， I 是达到目标模型精度所需的总训练时隙数。

现在我们说明奖励设计的动机。奖励 r^i 定义为计算成本、通信成本和训练时间的加权和。前两项促使智能体选择能够降低通信成本和计算成本的动作。最后一项 -1则鼓励智能体在更少的时隙内完成训练（例如，减少总训练时隙数 I）。通过调整非负参数σ_1 和 σ_2，可以满足不同联邦学习任务对资源效率和学习时间的不同偏好。例如，我们可以提高资源成本的权重，以避免严重的陈旧度效应和资源浪费，从而倾向于选择较小的陈旧度阈值。而资源成本权重较低，则表明训练时间对某项联邦学习任务至关重要，因此会做出更多选择较大陈旧度阈值的决策，以促进并行模型训练。

深度强化学习智能体的训练过程。考虑到连续且高维的空间，以及可用

2. 对于具有图形处理器资源的设备，可以使用图形处理器训练本地模型，相应地，计算成本根据设备的图形处理器周期频率和图形处理器处理密度计算，这些参数可通过测量获得。

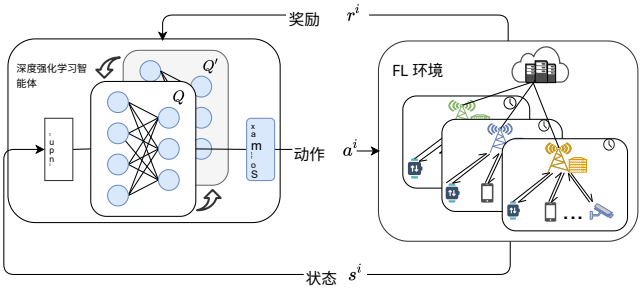


图6. 深度强化学习智能体与 FL 环境交互。

联邦学习任务中的轨迹有限，我们采用深度Q网络作为深度强化学习智能体，以高效学习最优陈旧度控制策略。由于联邦学习系统的动态变化复杂，深度Q网络探索不足时容易出现过估计问题。因此，为解决过估计问题，我们提出使用双重深度Q网络来学习近似器 $Q_\pi(\mathbf{s}^i, a^i)$ ，以近似最优动作价值函数 $Q^*(\mathbf{s}^i, a^i)$ [27]。双重深度Q网络引入了一个双重估计器 $Q'(\mathbf{s}^i, a^i, \theta_i')$，每隔 U 次更新冻结一次，以稳定动作价值函数的估计。

如图6所示，为训练深度强化学习智能体，当前状态信息 $\mathbf{s}^i$ 被输入动作价值函数 Q ，随后深度Q网络生成动作 a^i ，作为愿意参与联邦学习模型训练的边缘节点的陈旧度阈值。与 FL 环境交互若干轮后，深度强化学习智能体从经验记忆中采样若干状态-动作对，以求解方程 (4)：

$$\arg\min_{\theta_i} \mathbb{L}(\theta_i) = (Y_i' - Q_\pi(\mathbf{s}^i, a^i, \theta_i))^2, \tag{27}$$

其中，目标 Y_i'定义为

$$Y_i' = r^i + \gamma Q'(\mathbf{s}^{i+1}, \arg\max_{a^{i+1}} Q(\mathbf{s}^{i+1}, a^{i+1}, \theta_i), \theta_i'), \tag{28}$$

其中， θ_i 是每个时间步更新的在线参数， θ_i' 是双重估计器 Q'的参数。

通过梯度下降最小化$\mathbb{L}(\theta_i)$ 来更新动作价值函数 $Q(\mathbf{s}^i, a^i)$ ，即：

$$\theta_{i+1} = \theta_i + \alpha_Q(Y_i' - Q(\mathbf{s}^i, a^i, \theta_i)\nabla_{\theta_i} Q(\mathbf{s}^i, a^i, \theta_i)), \tag{29}$$

其中， α_Q 是标量步长。此外，在双重深度Q网络模型训练中采用经典的 ϵ-贪婪策略以助于探索[27]。基于深度强化学习的陈旧度控制流程详见算法2。

深度强化学习智能体部署在云服务器的调度器组件中，因此，调度器可以在向发送签到请求的边缘节点分发最新全局模型时，告知该边缘节点陈旧度阈值。

5.3 客户端-边缘层的异质性感知关联

受第4节的收敛性分析以及第5.1节讨论的启发，我们确定了可控的学习性能增强因素，并旨在设计一种能够最小化边缘-云端模型差异的客户端-边缘关联机制。由于采用同步机制，只有在关联的最慢客户端上传其最新更新的模型后，才能进行边缘模型聚合。因此，还应考虑客户端之间的资源异质性（例如响应延迟）。我们在边缘节点的数据异质性与客户端固有的资源异质性之间取得了良好平衡，从而构建

算法2用于自适应陈旧度控制的双重深度Q网络训练过程

输入： 折扣因子 γ、目标准确率 Acc_{target}、经验记忆最大容量 B_{max}、目标网络的更新频率 U。

初始化经验记忆$\mathbf{E}^{exp}$
初始化动作价值网络 Q ，使用初始权重 θ
将目标网络 Q' 初始化为 θ的副本
对每个回合 $epi = 1,\ 2,\ 3,$ .....**执行**
  对于 $i = 1, 2, ..., I$ **do**
    从联邦学习环境中获取当前状态$\mathbf{s}^i$
    根据 ϵ-贪心策略生成动作 a^i 并执行
    观察奖励 r^i 和下一状态 $\tilde{\mathbf{s}}^i$
    将元组 $(\mathbf{s}^i, a^i, r^i, \tilde{\mathbf{s}}^i)$ 存入 $\mathbf{E}^{exp}$
    如果 $|\mathbf{E}^{exp}| > B_{max}$ **则**
      移除最旧的元组
    结束如果
    从 $\mathbf{E}^{exp}$中采样一个由元组组成的小批量 $(\mathbf{s}\ \ a, r, \tilde{\mathbf{s}})$
    根据方程 (28) 获取目标值
    使用目标函数训练并更新动作价值网络
  方程 (27) 和方程 (29) 的
    将目标网络更新为动作价值网络权重的副本，
  每隔 U 步
    获取准确率 y Acc 云模型的
    如果 $Acc >= Acc_{target}$ **则**
      跳出当前回合
    结束如果
  结束循环
结束循环

一种异质性感知的客户端-边缘关联机制，以实现快速且准确的云模型学习。

对于联邦学习中的数据异质性，我们主要研究现实应用中普遍存在的标签分布偏斜[28]。由于边缘-云端模型差异源于边缘节点与云服务器之间的数据异质性，因此我们采用基于 Kullback-Leibler（KL）散度的 Jensen-Shannon（JS）散度 [29], 来计算两个数据集之间的差异。给定两个概率分布 P和 P'， P 与 P' 之间的 JS 散度定义为

$$\begin{aligned} JS(P||P') &= \frac{1}{2}KL(P||\frac{P+P'}{2}) + \frac{1}{2}KL(P'||\frac{P+P'}{2}), \\ KL(P_1||P_2) &= \sum_{x\in X} P_1(x) log\frac{P_1(x)}{P_2(x)}. \end{aligned} \tag{30}$$

JS散度具有对称性，且取值归一化于 $[0, 1]$, 之间的优良性质，这与无界的 KL 散度形成对比。 $JS(P||P') = 0$表示 P 和 P' 的分布相同，而 $JS(P||P') \to 1$ 则意味着这些分布被认为相距甚远。面对特征分布偏斜时，我们可以采用 FedBN [30] 作为聚合方法，在协同训练过程中帮助协调本地特征分布，这将是一个值得探索的未来研究方向。

针对客户端的资源异质性，我们首先定义与边缘节点 m 关联的客户端 k 的响应延迟：

$$L^{m,k} = l_{comp}^k + l_{comm}^{m,k}, \tag{31}$$

其中， l_{comp}^k 表示客户端 k 在该时间段内的平均计算延迟 $C^{i,\ k}_{\ comp}$， $l^{m,\ k}_{\ comm}$ 表示边缘节点 m 与客户端 k 在该时间段内的平均通信延迟 $C^{i,\ m\ k\ ,}_{\ \ \ \ \ comm}$ 。这里，l_{comp}^k 和 $l^{m,\ k}_{\ comm}$ 均可从历史记录中获取。因此，边缘模型聚合的时延可表示为

$$L^m = \max_{k\in\mathcal{C}^m} L^{m,k}, \tag{32}$$

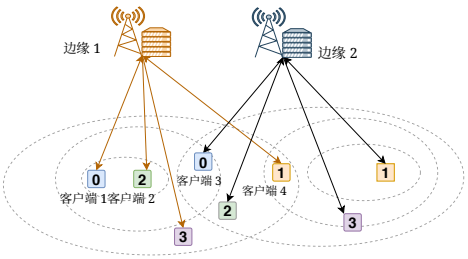


图 7. 与不同边缘节点通信的客户端响应延迟等高线图。每个客户端的样本仅来自一个类别，用方框中的数字表示。对于边缘节点 1，在选择客户端 1 和客户端 2 后，考虑到客户端响应延迟与数据分布之间的权衡，选择客户端 4 比选择客户端 3 更合适。

这表明，一个边缘节点应选择响应延迟较低的客户端来组建客户端簇，以减轻拖尾效应并加快模型训练进程。

在执行客户端-边缘节点关联之前，每个边缘节点都会先探测其通信范围内的客户端，以测量响应延迟并收集客户端的数据分布。客户端的数据分布记录了不同类别样本所占的比例。例如，对于一个包含 4 个不同标签的应用，若客户端数据集 $\mathcal{D}_k$ 包含一个标签为 0 的样本和两个标签为 2 的样本，则该客户端的标签分布可定义为 $LD_k(\mathcal{D}_k)=[\frac{1}{3},0,\frac{2}{3},0]$。请注意，客户端的标签信息仅以聚合形式公开（见方程 (33)），这通过安全多方计算（例如隐私保护 k 安全求和协议 [31]）实现；并且在添加噪声后，不会侵犯个人标签隐私。

获取通信范围内客户端的响应延迟和标签分布后，边缘节点 m 可以通过计算下文定义的成本，在测得的时延与边缘-云端模型差异之间取得平衡

$$COST_k^m = L^{m,k} + \lambda JS(\frac{|\mathcal{D}^m|\cdot LD^m + |\mathcal{D}_k|\cdot LD_k}{|\mathcal{D}^m|+|\mathcal{D}_k|}, LD_{IID}), \quad (33)$$

其中，λ 是用于权衡资源异质性与数据异质性的加权参数。LD^m 表示边缘节点 m上的标签分布，即当前客户端簇 $\mathcal{C}^m$中各标签分布的加权平均。LD_{IID} 表示云服务器持有的独立同分布标签分布。这里，我们假设全局数据分布为独立同分布，因为云服务器协调多个边缘节点上的模型学习，从而获得大量来自不同类别的样本。

因此，边缘节点可以通过以下双向选择方式进行异质性感知的客户端-边缘关联：

1）如果边缘节点 m 的客户端簇为空，则该边缘节点从其通信范围内尚未关联的客户端中选择响应延迟最低的客户端。否则，边缘节点将选择成本 $COST_k^m$较小的客户端 k 。2）如果某个客户端当前被多个边缘节点选中，该客户端将随机选择一个边缘节点。

双向选择过程将持续进行，直到所有客户端都与一个边缘节点关联。详细的异质性感知的客户端-边缘关联策略见算法3。

为了更直观地说明这一点，如图 7 所示，将客户端 1 和客户端 2 添加到边缘 1 的客户端簇后，为了使

算法3异质性感知的客户端-边缘节点关联

输入： 所有客户端的标签分布 $\{LD_k\}_{k=1}^N$，响应时延 $\{l_k^m\}_{k=1}^N$ 对应边缘节点 $m\in\{1,2,...,M\}$。

构建客户端池 客户端池 $=\{1,2,...,N\}$

为每个边缘节点构建空客户端簇：$\mathcal{C}^m=\emptyset$ 对于 $m\in\{1,2,...,M\}$，并为每个客户端构建空边缘节点集合 $S_k=\emptyset$ $k\in\{1,2,\cdots,N\}$

当客户端池 $\neq\emptyset$ **时**
  对于 $m\in\{1,2,...,M\}$，**执行**
    如果 $\mathcal{C}^m=\emptyset$，**则**
      选择客户端 k 响应延迟最低的，并设定 $\mathcal{C}^m=\{k\}$
      从客户端池中移除客户端 k
    else
      通过最小化方程 (33) 选择客户端 k
      $S_k=S_k\cup\{m\}$
    结束条件判断
  结束循环
  对于 $k\in\{1,2,...,N\}$，**执行**
    如果 $S_k\neq\emptyset$，**则**
      从 S_k中随机选择一个边缘节点 m
      $\mathcal{C}^m=\mathcal{C}^m\cup\{k\}$
      从客户端池中移除客户端 k 并清除 S_k
    结束 if
  结束 for

输出： $m\in\{1,2,...,M\}$的客户端集群 $\mathcal{C}^m$。

边缘 1 的数据分布接近独立同分布，选择客户端 4 比选择客户端 3 更合适。与许多现有研究类似，例如 [32], [33], ，我们假设在联邦学习训练过程中，客户端保持静止或位置变化缓慢。这主要是因为联邦学习需要大量计算和频繁通信，因此大多数客户端倾向于在条件稳定时（例如在家中或办公室为电池充电时）参与联邦学习 [2]。对于客户端位置快速变化的情况，我们可以采用自适应客户端-边缘关联策略，定期更新客户端所选的边缘节点。还可以进一步利用一些在线优化算法，提升动态客户端-边缘关联的性能。然而，这种情况下的理论分析要复杂得多，受篇幅限制，将留待未来研究。

在实际应用中，利用所有客户端的标签分布以及测得的客户端与边缘节点之间的时延，我们可以轻松地获得不同 λ 下的不同客户端-边缘关联策略。因此，我们可以计算不同 λ 下所有边缘节点的总 JS散度，然后选择 JS散度较小的客户端-边缘关联策略，以加快联邦学习模型训练。HiFlash 可以视为一种增强型 HiFL 方法，它具备自适应陈旧度控制和异质性感知的客户端-边缘关联，因此我们可以选择 HiFL 或 HiFlash 来高效地进行联邦学习模型训练。在 HiFlash 中，自适应陈旧度控制和异质性感知的客户端-边缘关联分别旨在缓解云服务器异步聚合中的陈旧问题以及边缘节点之间的数据异质性。因此，遵循本文提出的类似思路，可以将这两个部分分别与现有的异步联邦学习工作相结合，以提升性能。

6 实验

6.1 仿真设置

为了评估我们所提出算法的有效性，我们在模拟环境中进行了广泛的实验

表 2 数据集及其对应的模型。

数据集	模型	参数数量
MNIST	LeNet	21,840
CIFAR10	ResNet-18	3,504,554
FEMNIST	CNN	214,590

该环境包含 100 个客户端、10 个边缘节点和一台云服务器。我们将图像分类作为 FL 任务，并使用三个真实世界数据集评估 HiFL 和 HiFlash 的性能：MNIST[21], 、CIFAR10 [34] 和 FEMNIST [35]。由于 FEMNIST 是扩展 MNIST 数据集 [36] 的联邦版本，包含来自 3,550 名书写者的 805,263 个样本，因此我们随机选择 100 名书写者作为客户端，参与实验中的模型训练。对于包含 10 个类别的手写数字分类数据集 MNIST，我们使用 LeNet [37] 作为客户端训练的模型。对于 CIFAR10 数据集，我们采用标准的 ResNet-18 [38] 模型。对于包含 62 个类别的手写数字分类数据集 FEMNIST，我们设计了一个包含 214,590 个学习参数的卷积神经网络（CNN）作为学习模型。数据集及其对应模型总结于表 2 中。所有实验均在一块 Tesla P100 12GB 图形处理器上进行，算法使用 Pytorch 1.10.0 版本实现。

对于每个客户端的本地训练计算，我们采用小批量随机梯度下降（SGD），MNIST 和 FEMNIST 的批量大小为 60，CIFAR10 的批量大小为 50。与 [14], 一样，MNIST 和 FEMNIST 的初始学习率为 0.01，CIFAR10 的初始学习率为 0.1；这两者每 100 个 epoch 按 0.99 的速率指数衰减。每个客户端在一轮客户端-边缘端通信轮次中的本地更新次数 c 设为 3，而将边缘模型推送到云服务器之前的客户端-边缘聚合次数设为$H \in \{1,2,3\}$。[3] 系数 α_τ中的超参数 α 和 υ 在实际应用中可通过网格搜索确定。

对于深度强化学习训练，我们为三个数据集设置了不同的阈值范围（MNIST 和 FEMNIST 数据集为 τ_{max} =16 ，CIFAR10 数据集为 $\tau_{max} = 4$ ），因为对于复杂的 CIFAR10 数据集，较大的陈旧度阈值会导致训练时间大幅延长。因此，根据不同的动作大小，深度强化学习智能体中的双重深度Q网络（由两个双层多层感知机（MLP）网络构成）分别包含 4,607 个和 4,235 个可训练参数，对应 MNIST （和 FE MNIST）以及 CIFAR10 数据集。MLP 网络的输出经过 Softmax 层后，即为选择陈旧度阈值的概率。双重深度Q网络轻量化，每次训练迭代在图形处理器上只需数秒。

数据异质性。对于 MNIST 和 CIFAR10 数据集，为了模拟现实世界中客户端的数据异质性，我们为客户端生成了以下三种数据分布：

- **独立同分布：** 为每个客户端随机分配一个在 10 个类别上均匀分布的数据分布。
- **非独立同分布（1）：** 每个客户端仅拥有一个随机类别的图像。
- **非独立同分布（2）：** 将每个客户端中的样本分配到随机选出的两个类别中。

而 FEMNIST 数据集自然属于以下三种数据异质性情形：

- **标签分布偏斜：** 不同书写者之间的标签分布截然不同。
- **特征分布偏斜：** 由于书写者的字符特征（例如笔画宽度、倾斜度）各不相同，他们之间存在自然的特征分布偏斜。
- **数量偏斜：** 每个客户端中的样本数量范围为 $[4, 525]$。

边缘节点上的数据分布可以通过计算其关联客户端的数据分布的加权平均值来获得。我们可以使用 JS散度来衡量边缘节点的数据异质性。

资源异质性。客户端之间高度异质的硬件资源（CPU、网络连接）可以通过计算时延 $C^{i,k}_{comp}$ 和通信延迟 $C^{i,m,k}_{comm}$ 反映出来。对于每个客户端 k的计算能力，我们假设 $f_k \in [1,2]$ GHz为 CPU周期频率，$\zeta_k = 20$cycles/bit为执行一比特所需的 CPU周期数。对于通信能力，当客户端关联到不同的边缘节点 m 时，带宽$B_{k,m}$ 范围为1 MHz 至 10 MHz。

6.2 指标和基线方法

性能指标。我们将测试准确率、与云端的通信次数和系统总成本 作为评估 HiFlash 性能的三项指标。此外，我们还计算同步客户端-边缘聚合期间的响应延迟 L^m 和等待时间 $L^m_{waiting}$，以评估客户端-边缘关联策略的有效性。此处，等待时间 $L^m_{waiting}$ 以与边缘节点 m关联的客户端中拖后者的平均等待时间衡量，记为

$$L^m_{waiting} = \frac{\sum_{k\in\mathcal{C}^m}(L^{m,k} - \min_{k\in\mathcal{C}^m} L^{m,k})}{|\mathcal{C}^m|}. \tag{34}$$

基线方法。我们将所提出的算法与传统集中式方法和基于联邦学习的方案进行比较，以评估性能：

- **集中式学习：** 该方案将所有原始数据收集到云端进行训练，并为模型准确率提供上界。
- **FedAvg [1]:** 一种基于云端、采用同步聚合的联邦学习方案。每次随机选择 10 个客户端进行本地模型训练和全局模型聚合。
- **FedAsync [8]:** 一种基于云端的异步联邦学习算法，无需等待掉队客户端即可更新全局模型。
- **HierFAVG [14]:** 一种云-边-端分层联邦学习方案，在客户端-边缘节点聚合和边云聚合中均执行同步更新。为确保比较公平，每轮全局训练中随机选择 5 个客户端进行边缘模型聚合，并由 2 个边缘节点参与云模型更新。
- **FedAT [16]:** 一种结合了同步层内训练与异步跨层训练的分层联邦学习方案。FedAT 根据客户端的响应延迟进行客户端聚类，但不考虑客户端的数据异质性。

值得注意的是，为了与 HiFlash 进行公平比较，我们在异步联邦学习方案（例如 FedAsync、FedAT 和 HiFL）中采用具有固定陈旧度阈值的随机陈旧度控制机制。对于 MNIST 数据集，我们设置了一个较大的值

3. c和 H 的取值取决于实际设备的计算预算。由于本研究实验室的计算资源有限，我们为 c 和 H都设置了较小的取值，但这足以评估 HiFlash 的有效性。

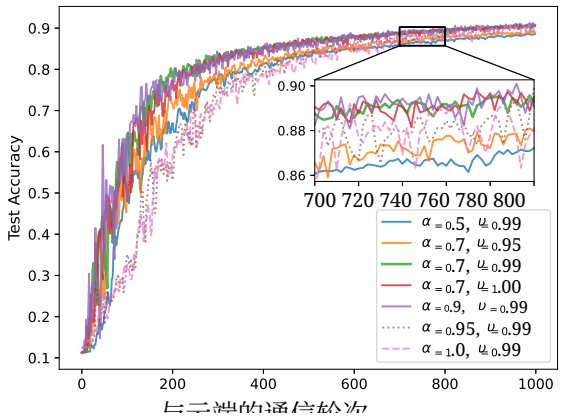


图 8. 不同混合超参数下非独立同分布（2）数据分布的 MNIST 数据集测试准确率。

($\tau_{max} = 16$)，以促进更多的并行模型训练并缩短总训练时间。而对于复杂的 CIFAR10 数据集，我们设置了一个较小的值（$\tau_{max} = 4$），以减少资源浪费，因为 CIFAR10 数据集的本地训练会产生较高的资源成本。面对新的数据集时，我们可以根据联邦学习任务的偏好（例如成本效益、训练时间）来确定陈旧度阈值。

6.3 实验结果

不同超参数设置下的性能评估。由于 HiFL 和 HiFlash 的全局模型更新由 α_τ 控制，而 α_τ 与边缘模型的初始权重 α 及惩罚系数 υ 有关，因此我们首先评估 HiFL 在不同 α 和 υ 设置下的测试准确率。如图 8 所示，在非独立同分布（2）数据分布下，对于不同的混合超参数设置，HiFL 均具有较强的鲁棒性，并能在 1,000 个通信轮次内收敛。然而，α 的值过大或过小都会导致收敛速度变慢。例如，当 α 过大时（如 $\alpha = 0.95$、$\alpha = 1.0$），当前全局模型无法保留上一轮全局模型的信息；而较小的 α（如 0.5）则会阻碍全局模型从新上传的边缘模型中学习。

因此，我们采用网格搜索为 α 和 υ 选择合适的取值。从图 8 可以看出，当 $\alpha = 0.9$ 和 $\alpha = 0.7$ 时，全局模型收敛较快；然而，由于边缘模型存在偏差，$\alpha = 0.9$ 会使测试准确率波动更大。此外，为解决模型陈旧度问题，适当的 υ（如 0.99）有助于实现快速且稳定的模型收敛。由于不同数据集在各种混合超参数设置下的收敛结果相似，因此我们在后续实验中将 $\alpha = 0.7$ 和 $\upsilon = 0.99$ 设为相应取值。

模型准确率与计算效率评估。考虑到分层联邦学习会在一个全局轮次中执行更多本地计算，以减少与云端之间成本高昂的通信，我们建议根据客户端训练轮次总数评估测试准确率。如图 9 和图 10 所示，我们分别针对 MNIST 和 CIFAR10 数据集，在三种数据异质性条件下研究不同训练方法的模型准确率和计算效率。由于集中式训练方法会将所有数据收集到云端进行模型学习，因此不会在设备上产生任何计算成本。因此，我们仅使用集中式学习的测试准确率，为其他比较方法的模型准确率提供上界。

通过结合同步与异步模型聚合的优点，并减轻模型陈旧度带来的负面影响，我们可以看到，HiFL 和 HiFlash 在独立同分布情形下能够实现与 FedAvg方法相当的训练性能。在非独立同分布情形下，若根据客户端训练轮次总数比较测试准确率，HiFL 和 HiFlash 的表现略逊于 FedAvg。这是因为 HiFL 和 HiFlash 中多轮客户端-边缘聚合可能会导致一定程度的梯度偏差，进而降低模型性能。此外，与 FedAvg 相比，HiFL 和 HiFlash 采用异步模型聚合，因此不可避免地会受到陈旧度效应的影响。

HiFL 和 HiFlash 的表现优于其他分层联邦学习方法（例如 HierFAVG）。分层联邦学习旨在以增加本地计算量为代价，减少昂贵的通信开销；如图 9 和图 10 所示，HierFAVG 是 FedAvg 在分层设置下的扩展，与 FedAvg 相比，计算效率较低。这是因为在本地训练轮次相同的情况下，HierFAVG 的边缘节点与云端聚合次数少于 FedAvg。借助 HiFL 和 HiFlash 中能够良好平衡全局模型与上传的边缘模型的异步更新机制，我们可以看到，与 HierFAVG 和 FedAvg 之间的性能差距相比，HiFL 和 FedAvg 之间的性能差距显著缩小，这表明 HiFL 和 HiFlash 中的异步聚合比其他分层联邦学习方案更具计算效率。

HiFL 和 HiFlash 的表现优于其他基于同步或异步的方法。例如，异步联邦学习方法（如 FedAsync 和 FedAT）的测试准确率低于 HiFL 和 HiFlash，因为它们忽视了偏置数据分布和模型陈旧度对云模型准确率的负面影响。FedAsync 算法曲线出现剧烈振荡有以下两个原因：（1）FedAsync 算法中，每当一个客户端上传其更新后的模型，全局模型便会随之更新，而无需等待拖后腿节点。这种异步聚合会使最终全局模型的性能存在很大的不确定性，尤其是在非独立同分布情形下。相比之下，其他算法会融合不同客户端的模型，以确保全局模型的泛化能力；（2）陈旧度效应会减缓 FedAsync 的收敛速度，并在面对较大的陈旧度时导致性能不稳定。

通信效率评估。我们将联邦学习过程中通信次数定义为边缘节点（或两层联邦学习框架中的客户端）与云服务器之间为交换模型而进行的通信总次数。与云端的通信次数越少，表示传输到云端的模型数据量越小。为评估 HiFlash 的通信效率，我们考察所有基于联邦学习的方法达到目标准确率所需的边缘节点与云端之间的通信次数。

如图11所示，对于 MNIST 数据集，随着目标准确率和客户端数据异质性的增加，不同方法所需的通信次数也随之增加。除 FedAsync 算法外，无论数据分布和目标准确率如何，我们提出的 HiFL 方案都比其他基于联邦学习的方法具有更高的通信效率。例如，在非独立同分布（2）数据分布且目标准确率为 90% 时，HiFL 在 MNIST数据集上相比 FedAT、HierFAVG 和 FedAvg，通信次数最多可分别减少 31.9%、28.2% 和 77.6% 。尽管 FedAsync 达到目标准确率所需的通信次数少于我们提出的 HiFL 方法

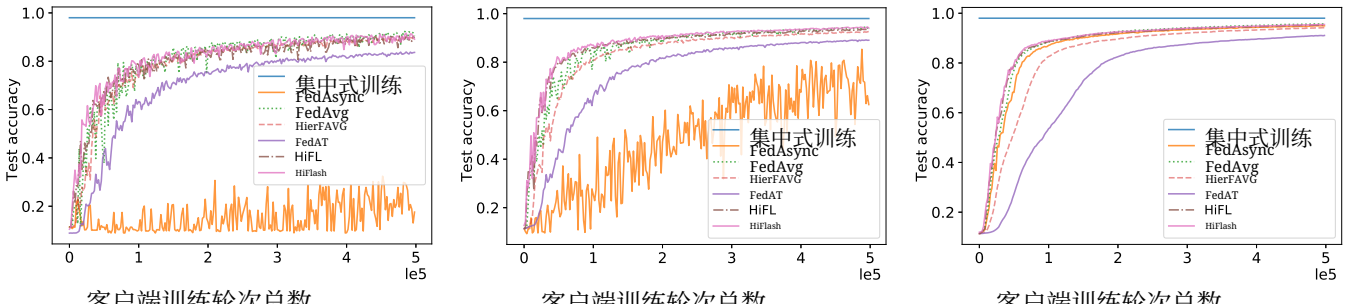


图 9. 不同数据分布下，MNIST 数据集的测试准确率与客户端训练轮次总数的关系。

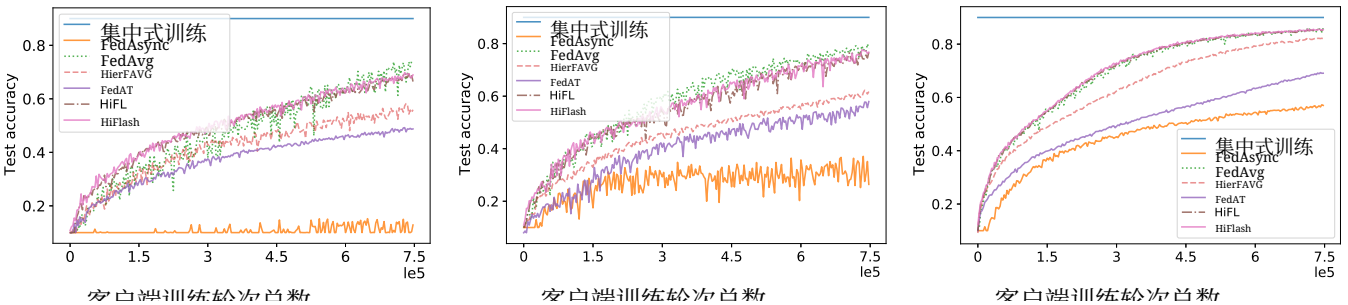
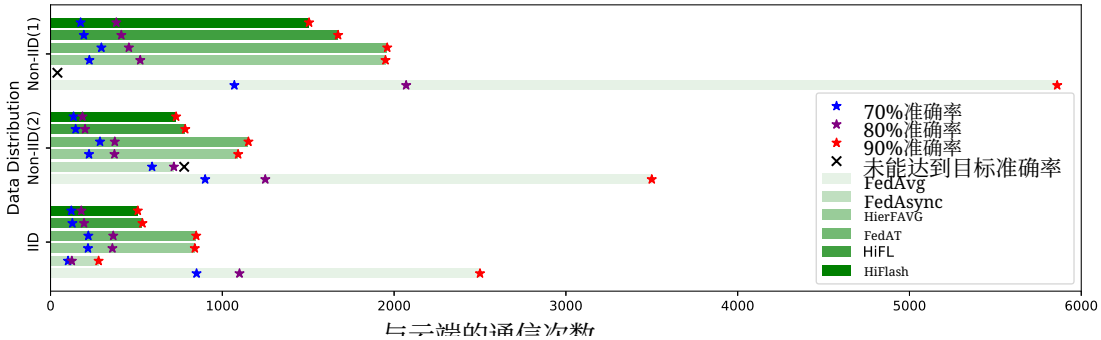


图 10. 测试准确率　　不同数据分布下 CIFAR10 数据集的测试准确率与客户端训练轮次总数的关系　　ts.



在 MNIST 数据集不同数据异质性程度下，HiFlash 与不同联邦学习方法达到目标准确率所需的边缘节点与云服务器之间的通信次数。

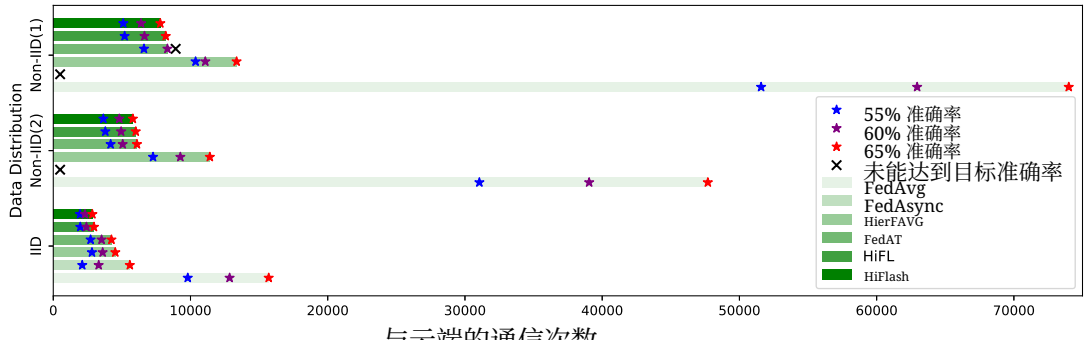
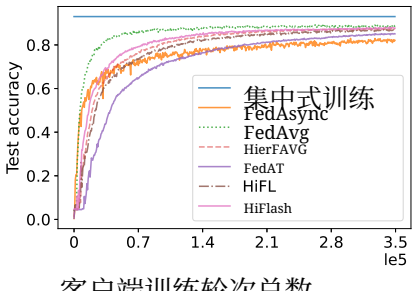


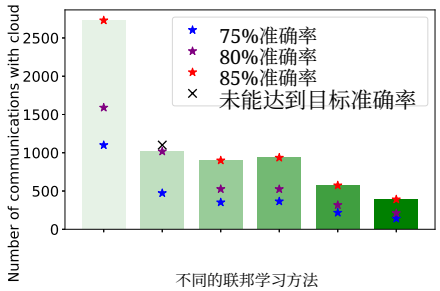
图12. 在 CIFAR10 数据集的不同数据异质性水平下，HiFlash 与不同联邦学习方法达到目标准确率所需的边缘节点与云服务器之间的通信次数。

在 MNIST 数据集的独立同分布设置下，但它无法处理参与客户端固有的数据异质性（例如非独立同分布（2）和非独立同分布（1）情况）。例如，在非独立同分布（2）数据分布下，FedAsync 无法达到 90% 的目标准确率；在非独立同分布（1）情况下，甚至无法达到 70% 的目标准确率，这表明 FedAsync 不适用于真实的联邦学习场景

其中数据以非独立同分布的方式分布。

分层联邦学习方法（例如 HiFL、HierFAVG）通过客户端-边缘聚合，大幅减少了与云端之间代价高昂的通信。此外，配备自适应陈旧度控制的增强版 HiFlash框架能够进一步加快模型训练过程，并减少与云端的通信轮次——





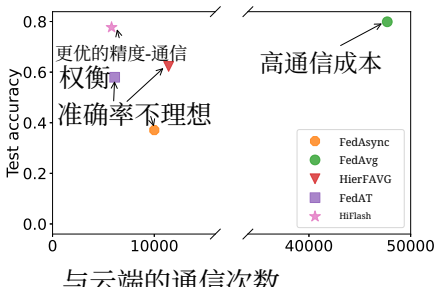
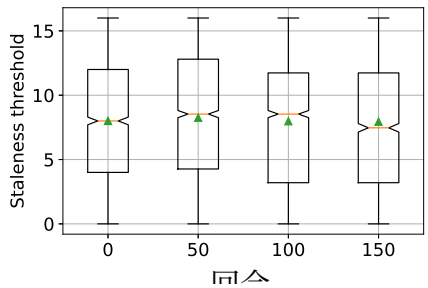
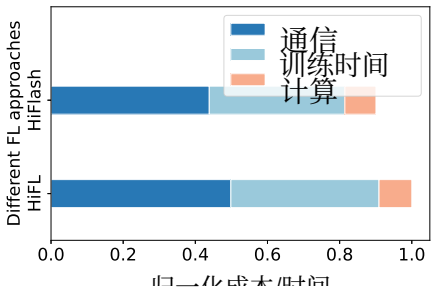


图 13. FEMNIST 数据集的测试准确率与客户端训练轮次总数的关系。图 14. 在 FEMNIST 数据集上达到目标准确率所需的边缘节点与云服务器之间的通信次数。

图 15. 与其他基于联邦学习的方法相比，HiFlash 实现了良好的准确率—通信权衡。





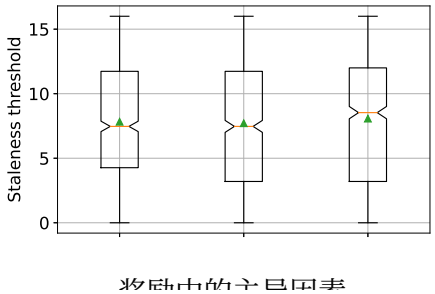


图16. 随着深度强化学习训练回合的增加，自适应陈旧度控制的策略改进过程。图17. HiFL 和 HiFlash 的系统总成本。图18. 面对不同奖励设计时的策略差异。

与 HiFL 相比（例如，在非独立同分布(1)数据分布情况下，通信轮次减少10%），这一结果与收敛性分析一致：通过将 τ 控制在较小值，模型将在更少的通信轮次内收敛T_c 。因此，客户端-边缘聚合和陈旧度控制有助于提升 HiFL 和 HiFlash 的通信效率。

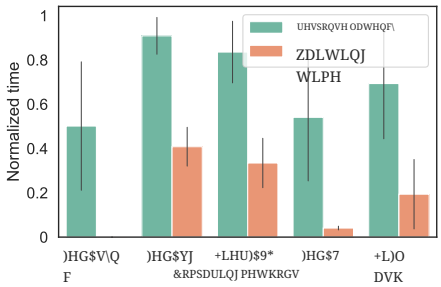
对于 FedAvg 和 HierFAVG 的比较，FedAvg 在每个训练轮次中有 10 个客户端与云端通信，而 HierFAVG 方法中有 5 个边缘节点与云端通信。此外，HierFAVG 在每轮中对客户端进行更多本地计算，以减少全局训练轮次，因此，在通信成本方面（例如，与云端的通信次数），HierFAVG 远优于 FedAvg。

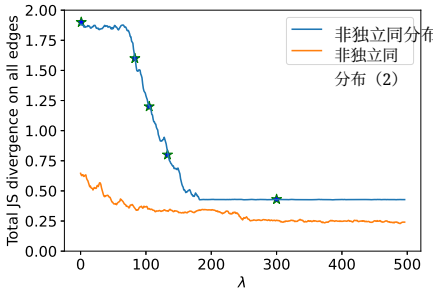
对于更复杂的数据集（即 CIFAR10），与所有基于联邦学习的方法相比，HiFlash 在所有数据分布情形下都能以最少的通信轮次达到不同的目标准确率。如图12所示，在非独立同分布(1)数据分布场景中，HiFlash 需要与云端通信6403 次才能达到60% 的目标准确率，这比 FedAT、HierFAVG 和 FedAvg 分别少23.07%、42.23% 和89.82% 。这表明，无论是基于异步机制的 FedAT，还是基于同步机制的 HierFAVG 方法，HiFlash 的表现都优于当前已有的分层联邦学习算法。

在 FEMNIST 数据集上的评估结果。我们还评估了 HiFlash 在 FEMNIST 数据集上的性能，该数据集中客户端的数据分布天然呈非独立同分布（在特征分布、标签分布和数量分布方面）。如图 13 所示，HiFlash 的性能仍优于其他分层联邦学习方法（例如 HierFAVG），这与在 MNIST 数据集上的实验结果相似。此外，与其他联邦学习方法相比，HiFlash 与云端的通信次数最少，如图 14 所示。评估结果表明，我们的 HiFlash 方法可以应用于标签分布和特征分布均存在偏度的真实世界数据集。

准确率与通信成本的权衡。 为了清晰展示 HiFlash 的优势，我们进一步研究了不同联邦学习方法在准确率与通信成本之间的权衡。以具有非独立同分布（2）分布的 CIFAR-10 数据集为例，我们在图15中绘制了最高模型准确率以及与云端通信的轮次。可以看出，流行的 FedAvg 方法通信成本较高，而基于异步的 FedAsync 方法和分层联邦学习方法（例如 HierFAVG 和 FedAT）则无法达到令人满意的准确率。相比之下，HiFlash 能够在模型准确率和通信效率之间实现良好的平衡。具体而言，HiFlash 大幅降低了通信成本（例如，比 FedAvg 低87%），同时模型准确率仅略有下降（例如2.1%）。与分层联邦学习方法（例如 FedAT 和 HierFAVG）相比，HiFlash 的通信成本可降低5%以上，模型准确率则提高16% 。此外，随着数据异质性增加，HiFlash 的优势会进一步扩大（见图11和图12）。

陈旧度阈值控制的效果。 HiFlash 实现了更快的模型收敛速度和更高的通信效率，这归功于设计完善的基于深度强化学习的自适应陈旧度控制，它能够根据过往经验和当前环境做出明智决策。如图 16 所示，自适应陈旧度控制的深度强化学习智能体在与 FL 环境交互并从深度强化学习训练回合中学习的过程中，不断改进陈旧度阈值决策策略。我们采用箱线图直观呈现不同训练回合中陈旧度阈值决策分布的五数概括，包括最小观测值、下四分位数、中位数、上四分位数和最大观测值。上四分位数、中位数和下四分位数共同构成一个带有分隔区域的箱体。各部分之间的间距反映了数据分布的方差和偏度，陈旧度阈值的均值则





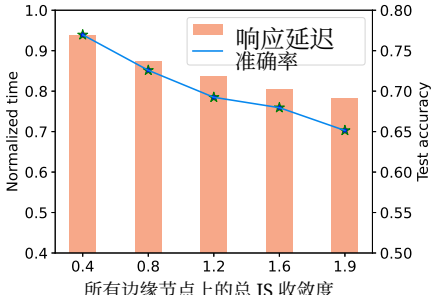


图 19. 不同联邦学习方法的平均响应延迟和等待时间。图 20. 不同数据异质性下，随着 λ 变化的 JS散度。 图 21. 面对不同 r奖励设计时的策略差异。

以绿色三角形表示。第 0 回合的决策策略是 HiFL 所采用的随机陈旧度控制策略，而第 150 回合改进后的策略则用于 HiFlash 方法。从偏斜的数据分布可以看出，与随机决策策略相比，HiFlash 中更多决策选择了较小的陈旧度阈值，从而实现通信高效的模型训练。

为了比较 HiFL 和 HiFlash 的系统成本（包括客户端的计算成本、边缘节点的通信成本以及云端的训练时间），我们通过简单地除以最大值，将系统成本归一化为$[0,1]$。采用这种归一化方法后，计算成本和通信成本也相应缩小。如图 17 所示，HiFL 的归一化成本为 1，这表明 HiFL 会带来较高的系统成本。而 HiFlash 较低的系统成本则归功于我们设计的自适应陈旧度控制策略的有效性。值得注意的是，尽管 HiFlash 中的 DQN训练会带来额外的计算开销，但可以离线方式在计算资源充足的云服务器上进行。

我们还通过为计算成本、通信成本和训练时间赋予不同权重（调整 σ_1和 σ_2的值），考察了不同奖励设计下的策略差异。如图 18 所示，当为资源成本（例如计算成本和通信成本）赋予较高权重时，负责陈旧度控制的深度强化学习智能体倾向于选择较小的阈值，以避免严重的陈旧度效应。而当训练时间的权重较高时，为促进模型并行，会做出更多选择较大陈旧度阈值的决策。

异质性感知的客户端-边缘节点关联的效果。此外，我们还评估了客户端-边缘聚合阶段中边缘关联客户端的响应延迟 L^m_k 和等待时间 $L^m_{waiting}$。对于 HiFlash，我们设置 $\lambda = 300$（将在下一段讨论），以便为每个边缘节点构建一个标签更均衡的数据集。因此，由于需要权衡数据异质性与时延降低，平均响应延迟可能会更长。如图19所示，FedAsync 的平均响应延迟和等待时间最低，因为它无需等待拖后者。FedAT 是一种与我们提出的 HiFL 类似的异步分层联邦学习方案，其响应延迟和等待时间也较短。然而，它仅关注客户端-边缘节点关联过程中的时延降低，而未考虑数据异质性缓解。因此，FedAsync 和 FedAT 都会导致准确率下降（如图 9-10 所示），并产生更多通信开销（如图 11-12 所示）。相比之下，我们提出的 HiFlash 能够在响应延迟和数据异质性之间实现权衡，从而获得令人满意的模型性能。

我们进一步研究了所有边缘节点上的总 JS 散度、由此产生的平均响应延迟，以及 HiFlash 的模型准确率，在不同 λ 下的表现。如图 20 所示，较大的 λ 表示相较于响应延迟，我们更关注边缘节点上的数据分布，因此在非独立同分布（1）和非独立同分布（2）两种情况下，总 JS 散度均随 λ 增大而降低。如图 21 所示，有偏差的边缘数据分布（JS 散度较大）会导致准确率显著下降。例如，当所有边缘节点上的总 JS 散度从 77% 降至 65%，总 JS 散度从 0.4 增至 1.9 时，可达到的准确率会从77% 降至 65%。通过控制参数 λ，我们提出的 HiFlash 能够灵活地帮助边缘节点在响应延迟与模型准确率之间取得平衡。

7 相关工作

7.1 高通信效率的联邦学习

经典的两层联邦学习框架（如 FedAvg [1]和 FedAsync [8]）在大规模分布式机器学习中不可避免地会遭遇过高的通信开销和网络拥塞，这是由于客户端与中央服务器之间需要进行大量的模型交换。为减少联邦学习中的梯度交换比特数，神经网络剪枝 [39]、权重量化 [40]、消息稀疏化 [41] 和知识蒸馏 [42]等技术着重于机器学习模型压缩，以减少传输信息量，同时保持较高的学习性能。

为进一步提高通信效率，研究者提出了分层联邦学习，引入边缘层并利用边缘节点作为中介，通过高效的客户端—边缘通信执行部分模型聚合，从而减轻云服务器的核心网络传输开销 [15]。例如，Liu等人提出了 HierFAVG，将传统 FedAvg算法扩展到分层场景，实现两级同步模型聚合 [14]。然而，由于同步模型聚合的特性，HierFAVG会面临严重的拖尾问题。Chai等人提出了 FedAT，这是一种新颖的联邦学习方法，通过协同结合层内同步训练与跨层异步训练，提高收敛速度并降低通信开销[16]。然而，FedAT使用所有最新边缘模型的加权和来更新全局模型，这与我们的异步聚合机制不同。此外，该方法忽略了陈旧度效应，而这一效应在异步聚合中不可避免。

7.2 模型陈旧度控制

陈旧度效应是异步模型聚合中常见的挑战。大多数现有的联邦学习解决方案会通过减弱在过时的全局模型版本上计算出的陈旧结果所带来的影响，来容忍陈旧度

。例如，Zhang 等人提出了一种陈旧度感知异步随机梯度下降算法，根据梯度陈旧性调整学习率 [43]。Xie 等人采用加权平均来更新全局模型，并引入混合超参数，以自适应地控制陈旧度导致的误差[8]。文献 [44]设计了一种梯度校正项，以补偿陈旧度带来的影响。然而，这些方法仅关注陈旧度对模型准确率和收敛速度的负面影响，对系统效率的考量较少，例如计算/通信成本和训练时间。

陈旧模型可能对全局模型略有贡献，但如果不及时终止模型训练并上传模型，就会造成大量资源浪费和时间消耗。为解决这一问题，FedSA [45], 这一具有陈旧度感知能力的异步联邦学习算法，根据各参与客户端的计算速度为其设置陈旧度阈值。然而，这种方法忽略了整个流程中的通信成本和训练时间。此外，FedSA 为每个客户端设置的陈旧度阈值是固定的，无法很好地适应现实中的动态环境。Zhang 等人提出了一种聚类半异步联邦学习（CSAFL）方法 [46], ，通过将学习目标不同的客户端划分为多个组并限制模型延迟，缓解模型陈旧度问题。然而，CSAFL 旨在学习个性化模型，即为不同的客户端组学习不同的组模型；而 HiFlash 方法的目标不同，旨在基于客户端生成的所有本地数据学习一个共同的全局模型，以充分利用全局知识共享所带来的共性。

7.3 分层联邦学习中的客户端-边缘节点关联

由于分散的客户端之间存在数据和资源异质性，分层联邦学习中的客户端-边缘关联策略会显著影响模型学习性能 [47]。HierFAVG 忽略了分布式客户端训练速度的差异，并将它们随机分配到不同集群中，这可能会延长每个训练轮次的通信时间 [14]。考虑到拖尾效应，TiFL [47] 和 FedAT [16]均根据测得的时延将客户端划分到不同层级，以缓解资源异质性 [47]。为解决联邦学习中的数据异质性，Duan 等人提出了 FedGroup，该方法根据客户端参数更新的余弦相似度将客户端聚类为多个组 [48]。在 FedCluster 中，作者提供了几种具有代表性的聚类方法，包括随机均匀聚类、基于时区的聚类和基于可用性的聚类，以适应各种应用场景 [49]。遗憾的是，当前的客户端-边缘关联方案只关注异质性的一个维度，没有考虑数据和资源异质性之间的权衡。

现有分层联邦学习方案的收敛性分析主要聚焦于采用两级同步模型聚合的联邦学习。例如，HierFAVG 的收敛性分析衡量了分层系统中数据分布的两级非独立同分布性（即客户端层级和边缘层级），并为如何选择两个层级的聚合频率提供了定性指导 [14]。而 [22] 中的理论分析进一步证明，降低边缘层级的非独立同分布性对模型收敛至关重要。尽管 FedAT [16] 采用同步客户端-边缘聚合和异步边缘-云端聚合，但 FedAT 中的异步聚合机制是对所有最新边缘模型进行加权和，这与我们的异步更新机制不同，此外，FedAT 的收敛性分析忽略了陈旧度效应，因此无法为缓解陈旧度效应提供见解。尽管如此，我们对 HiFL 的收敛性分析同时考虑了异步模型聚合引入的陈旧度和联邦学习固有的非独立同分布性，并进一步提出了有关陈旧度控制和非独立同分布性降低的见解，以实现更快的收敛速率和更低的收敛界。

8 结论

本文采用 HiFL，这是一种分层联邦学习方法，通过协同使用同步的客户端-边缘模型聚合和异步的边缘-云模型聚合，实现通信高效的模型学习。基于对 HiFL 的收敛分析，我们确定了影响模型收敛的可控因素，并进一步提出 HiFlash——一种具有自适应陈旧度控制和异质性感知的客户端-边缘关联的增强型 HiFL，以支持现实场景中的大规模部署。我们提出了一种基于深度强化学习的陈旧阈值决策算法，以实现准确且具成本效益的联邦学习模型学习。为应对客户端间固有的资源和数据异质性，我们设计了一种异构性感知的客户端-边缘关联策略，在通信延迟与边缘数据分布的异质性之间取得良好平衡。基于三个图像分类数据集的实证评估验证了我们的理论分析，并表明，与现有联邦学习方法相比，HiFlash 在不同程度的数据异质性下均能取得令人满意的预测性能，同时具备较高的通信效率。

参考文献

[1] Brendan McMahan、Eider Moore、Daniel Ramage、Seth Hampson 和 Blaise Aguera y Arcas。基于去中心化数据的深度网络通信高效学习。载于人工智能与统计，第 1273–1282 页。PMLR，2017年。[2] Peter Kairouz、H Brendan McMahan、Brendan Avent、Aur´elien Bellet、Mehdi Bennis、Arjun Nitin Bhagoji、Kallista Bonawitz、Zachary Charles、Graham Cormode、Rachel Cummings 等。联邦学习的进展与开放问题。《机器学习基础与趋势》，14(1–2)：1–210，2021年。[3] Yang Liu、Anbu Huang、Yun Luo、He Huang、Youzhi Liu、Yuanyuan Chen、Lican Feng、Tianjian Chen、Han Yu 和 Qiang Yang。Fedvision：由联邦学习驱动的在线视觉目标检测平台。载于*AAAI* 人工智能会议论文集，第 34 卷，第 13172–13179 页，2020年。[4] Timothy Yang、Galen Andrew、Hubert Eichner、Haicheng Sun、Wei Li、Nicholas Kong、Daniel Ramage 和 Franc¸oise Beaufays。联邦学习应用：改进 Google键盘的查询建议。*arXiv* 预印本 *arXiv:1812.02903*，2018年。[5] Yang Chen、Xiaoyan Sun 和 Yaochu Jin。通过逐层异步模型更新和时间加权聚合实现通信高效的联邦深度学习。*IEEE*神经网络与学习系统汇刊，31(10)：4229–4238，2019年。[6] Shiqiang Wang、Tiffany Tuor、Theodoros Salonidis、Kin K Leung、Chris-tian Makaya、Ting He 和 Kevin Chan。边缘与学习相遇：面向资源受限分布式机器学习的自适应控制。载于*IEEE INFOCOM 2018-IEEE* 国际通信会议，第 63–71 页。IEEE，2018年。[7] Anit Kumar Sahu、Tian Li、Maziar Sanjabi、Manzil Zaheer、Ameet Tal-walkar 和 Virginia Smith。异构网络中联邦优化的收敛性。*arXiv* 预印本 *arXiv:1812.06127*，3，2018年。[8] Cong Xie、Sanmi Koyejo 和 Indranil Gupta。异步联邦优化。载于*NeurIPS*机器学习优化研讨会 *(OPT)*，2020年。[9] Wei Yang Bryan Lim、Nguyen Cong Luong、Dinh Thai Hoang、Yutao Jiao、Ying-Chang Liang、Qiang Yang、Dusit Niyato 和 Chunyan Miao。移动边缘网络中的联邦学习：全面综述。*IEEE*通信调查与教程，22(3)：2031–2063，2020年。

[10] Wentai Wu、Ligang He、Weiwei Lin 和 Rui Mao。在移动边缘计算系统中加速可靠性无关客户端的联邦学习。IEEE 并行与分布式系统汇刊，32(7)：1539–1551，2020年。[11] Mengwei Xu、Zhe Fu、Xiao Ma、Li Zhang、Yanan Li、Feng Qian、Shangguang Wang、Ke Li、Jingyu Yang 和 Xuanzhe Liu。从云端到边缘：首次考察公共边缘平台。载于 IMC '21：ACM 互联网测量会议，虚拟活动，美国，2021年11月2日至4日，第37–53页。ACM，2021年。[12] Jinliang Yuan、Mengwei Xu、Xiao Ma、Ao Zhou、Xuanzhe Liu 和 Shangguang Wang。通过局域网-广域网编排实现分层联邦学习。arXiv 预印本 arXiv:2010.11612，2020年。[13] Qiong Wu、Xu Chen、Zhi Zhou 和 Junshan Zhang。Fedhome：基于云端与边缘的个性化联邦学习，用于居家健康监测。IEEE 移动计算汇刊，21(8)：2818–2832，2022年。[14] Lumin Liu、Jun Zhang、SH Song 和 Khaled B Letaief。客户端-边缘-云端分层联邦学习。载于 ICC 2020-2020 IEEE 国际通信会议，第 1–6 页。IEEE，2020年。[15] Siqi Luo、Xu Chen、Qiong Wu、Zhi Zhou 和 Shuai Yu。HFEL：面向高成本效益分层联邦边缘学习的边缘关联与资源分配联合优化。IEEE 无线通信汇刊，19(10)：6535–6548，2020年。[16] Zheng Chai、Yujing Chen、Ali Anwar、Liang Zhao、Yue Cheng 和 Huzefa Rangwala。FedAT：具有异步层级的高性能、通信高效联邦学习系统。SC '21，美国纽约州纽约市，2021年。计算机协会。[17] Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Alex Graves、Ioannis Antonoglou、Daan Wierstra 和 Martin Riedmiller。使用深度强化学习玩 Atari 游戏。arXiv 预印本 arXiv:1312.5602，2013年。[18] Richard S Sutton 和 Andrew G Barto。《强化学习导论》。麻省理工学院出版社，2018年。[19] Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A Rusu、Joel Veness、Marc G Bellemare、Alex Graves、Martin Riedmiller、Andreas K Fidjeland、Georg Ostrovski 等。通过深度强化学习实现人类水平控制。《自然》，518(7540)：529–533，2015年。[20] Georgios Damaskinos、Rachid Guerraoui、Anne-Marie Kermarrec、Vlad Nitu、Rhicheek Patra 和 Francois Taiani。FLeet：通过陈旧度感知和性能预测实现在线联邦学习。载于第 21 届国际中间件会议论文集，第 163–177 页，2020年。[21] Yann LeCun、L´eon Bottou、Yoshua Bengio 和 Patrick Haffner。应用于文档识别的基于梯度的学习。IEEE 会刊，86(11)：2278–2324，1998年。[22] Jin-woo Lee、Jaehoon Oh、Yooju Shin、Jae-Gil Lee 和 Se-Young Yoon。通过独立同分布与通信感知分组实现准确且快速的联邦学习。arXiv 预印本 arXiv:2012.04857，2020年。[23] Jie Feng、Lei Liu、Qingqi Pei 和 Keqin Li。无线边缘网络中高效分层联邦学习的极小极大成本优化。IEEE 并行与分布式系统汇刊，33(11)：2687–2700，2021年。[24] Yufeng Zhan、Peng Li 和 Song Guo。通过深度强化学习进行联邦学习的经验驱动计算资源分配。载于 2020 IEEE 国际并行与分布式处理研讨会 (IPDPS)，第 234–243 页。IEEE，2020年。[25] Minh N. H. Nguyen、Nguyen H. Tran、Yan Kyaw Tun、Zhu Han 和 Choong Seon Hong。面向移动边缘网络中多种联邦学习服务的资源共享。IEEE 移动计算汇刊，22(1)：541–555，2023年。[26] Yuchang Sun、Jiawei Shao、Yuyi Mao、Jessie Hui Wang 和 Jun Zhang。面向非独立同分布数据快速收敛的半去中心化联邦边缘学习。载于 2022 IEEE 无线通信与网络会议 (WCNC)，第 1898–1903 页。IEEE，2022年。[27] Hado Van Hasselt、Arthur Guez 和 David Silver。基于双重 Q 学习的深度强化学习。载于第 13 届 AAAI 人工智能会议论文集，第 2094–2100 页，2016年。[28] Jie Zhang、Zhiqi Li、Bo Li、Jianghe Xu、Shuang Wu、Shouhong Ding 和 Chao Wu。通过 logits 校准应对标签分布偏斜的联邦学习。载于国际机器学习会议，第 26311–26329 页。PMLR，2022年。[29] AP Majtey、PW Lamberti 和 DP Prato。Jensen-Shannon散度作为衡量混合量子态可区分性的指标。《物理评论 A》，72(5)：052310，2005年。[30] Xiaoxiao Li、Meirui JIANG、Xiaofei Zhang、Michael Kamp 和 Qi Dou。FedBN：通过本地批归一化处理非独立同分布特征的联邦学习。载于国际学习表征会议，2021年。[31] Rashid Sheikh、Beerendra Kumar 和 Durgesh Kumar Mishra。保护隐私的 k 安全求和协议。arXiv 预印本 arXiv:0912.0956，2009年。

[32] Takayuki Nishio 和 Ryo Yonetani。移动边缘中异构资源场景下的联邦学习客户端选择。载于*2019*年 *IEEE* 国际通信会议 *(ICC)*，第 1–7 页。IEEE，2019年。[33] Georgios Damaskinos、Rachid Guerraoui、Anne-Marie Kermarrec、Vlad Nitu、Rhicheek Patra 和 Francois Taiani。FLeet：通过陈旧度感知和性能预测实现在线联邦学习。*ACM* 智能系统与技术汇刊 *(TIST)*，第 13 卷，文章 30，2022年，第 5 期，1–?页。[34] Alex Krizhevsky 、 Geoffrey Hinton 等。学习微型图像的多层特征。2009年。[35]Sebastian Caldas、Sai Meher Karthik Duddu、Peter Wu、Tian Li、Jakub Koneˇcn`y、H Brendan McMahan、Virginia Smith 和 Ameet Talwalkar。Leaf：联邦学习场景的基准。*arXiv* 预印本 *arXiv:1812.01097*，2018年。[36] Gregory Cohen、Saeed Afshar、Jonathan Tapson 和 Andre Van Schaik。EMNIST：将 MNIST 扩展至手写字母。载于国际神经网络联合会议 *(IJCNN)*，第 2921–2926 页。IEEE，2017年。[37] Yann LeCun、Bernhard E Boser、John S Denker、Donnie Henderson、Richard E Howard、Wayne E Hubbard 和 Lawrence D Jackel。基于反向传播网络的手写数字识别。载于神经信息处理系统进展，第 396–404 页，1990年。[38] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第 770–778 页，2016年。[39] Yuang Jiang、Shiqiang Wang、V´ıctor Valls、Bong Jun Ko、Wei-Han Lee、Kin K. Leung 和 Leandros Tassiulas。模型剪枝使边缘设备上的联邦学习更高效。*IEEE*神经网络与学习系统汇刊，第 1–13 页，2022年。[40] Dan Alistarh、Demjan Grubic、Jerry Li、Ryota Tomioka 和 Milan Vojnovic。QSGD：通过梯度量化和编码实现通信高效的随机梯度下降。神经信息处理系统进展，第 30 卷：1709–1720，2017年。[41] WANG Luping、WANG Wei 和 LI Bo。CMFL：降低联邦学习的通信开销。载于*2019*年 *IEEE* 第*39*届国际分布式计算系统会议 *(ICDCS)*，第 954–964 页。IEEE，2019年。[42] Jin-Hyun Ahn、Osvaldo Simeone 和 Joonhyuk Kang。面向异构数据分布式边缘学习的无线联邦蒸馏。载于*2019*年 *IEEE* 第*30*届个人、室内和移动无线电通信国际年会 *(PIMRC)*，第 1–6 页。IEEE，2019年。[43] Wei Zhang、Suyog Gupta、Xiangru Lian 和 Ji Liu。面向分布式深度学习的陈旧度感知异步随机梯度下降。载于第二十五届国际人工智能联合会议论文集，第 2350–2356 页，2016年。[44] Ligeng Zhu、Hongzhou Lin、Yao Lu、Yujun Lin 和 Song Han。延迟梯度平均：容忍联邦学习中的通信延迟。神经信息处理系统进展，第 34 卷：29995–30007，2021年。[45] Ming Chen、Bingcheng Mao 和 Tianyi Ma。FedSA：一种面向非独立同分布数据的陈旧度感知异步联邦学习算法。未来一代计算机系统，第 120 卷：1–12，2021年。[46] Yu Zhang、Morning Duan、Duo Liu、Li Li、Ao Ren、Xianzhang Chen、Yujuan Tan 和 Chengliang Wang。CSAFL：聚类半异步联邦学习框架。载于*2021*年国际神经网络联合会议 *(IJCNN)*，第 1–10 页，2021年。[47] Zheng Chai、Ahsan Ali、Syed Zawad、Stacey Truex、Ali Anwar、Nathalie Baracaldo、Yi Zhou、Heiko Ludwig、Feng Yan 和 Yue Cheng。TIFL：分层联邦学习系统。载于第*29*届国际高性能并行与分布式计算研讨会论文集，第 125–136 页，2020年。[48] Moming Duan、Duo Liu、Xinyuan Ji、Renping Liu、Liang Liang、Xianzhang Chen 和 Yujuan Tan。FedGroup：通过基于分解相似度的聚类实现高效联邦学习。载于*2021*年 *IEEE* 并行与分布式处理及应用、大数据与云计算、可持续计算与通信、社交计算与网络国际会议 *(ISPA/BDCloud/SocialCom/SustainCom)*，第 228–237 页，2021年。[49] Cheng Chen、Ziyi Chen、Yi Zhou 和 Bhavya Kailkhura。FedCluster：通过聚类循环加速联邦学习收敛。载于*2020*年 *IEEE* 国际大数据会议 *(Big Data)*，第 5017–5026 页。IEEE，2020年。