面向时空开销均衡的轻量级联邦遗忘学习框架

唐湘云[1)] 武杭[1)] 王亚杰[2)] 沈蒙[2)] 翁彧[1)] 祝烈煌[2)]

[1)](中央民族大学信息工程学院北京 100081)

[2)](北京理工大学网络空间安全学院北京 100081)

摘要 联邦遗忘学习已成为实现用户“被遗忘权”的一种有效范式，能够从全局模型中消除特定用户的数据贡献，以满足数据隐私保护法规的要求。然而，现有的联邦遗忘学习方法通常以牺牲存储空间换取遗忘速度，或以增加遗忘时间降低存储需求，难以平衡时间开销与存储开销。此外，这些方法大多仅支持特定粒度的遗忘，如客户端级遗忘或类别级遗忘，这限制了它们在现实应用场景中的实用性。本文提出了一种轻量级的联邦遗忘学习框架 FedUR，能够同时支持客户端级、样本级和类别级三种不同粒度的遗忘需求。FedUR 在平衡时间开销与存储开销的前提下，保持全局模型性能的同时实现了有效的遗忘效果。具体地，为实现对三种遗忘粒度的支持并平衡性能与资源消耗，FedUR 将联邦遗忘学习过程解耦为自适应遗忘和优化恢复两个阶段。在自适应遗忘阶段，目标客户端采用随机梯度上升策略实现对目标数据的遗忘，同时最小化存储开销；若遗忘后的模型性能未满足约束条件，则使用投影梯度下降防止模型退化为随机状态。在优化恢复阶段，服务器基于外包的标记数据集，利用知识蒸馏方法恢复全局模型的性能，相较于传统的后训练或者将遗忘过程集成到联邦学习的方法恢复更快。本文在真实数据集上开展了广泛的实验，针对三种遗忘粒度将 FedUR 与五种现有先进的遗忘方法进行了比较。实验结果表明，FedUR 在时间和空间开销方面表现优越的同时，FedUR 模型恢复准确率比其他方法高出 1%至 10%。

关键词 联邦学习；联邦遗忘学习；隐私保护；自适应遗忘；优化恢复

中图法分类号 TP18

A Lightweight Federated Unlearning Framework for Balanced Time-Storage Overhead

TANG Xiang-Yun[1)] WU Hang[1)] WANG Ya-Jie[2)] SHEN Meng[2)]

WENG Yu[1)] ZHU Lie-Huang[2)]

[1)](School of Information Engineering, Minzu University of China, Beijing 100081)

[2)](School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081)

Abstract In the era of booming data-driven artificial intelligence technologies, societal concerns about data privacy protection are escalating. Federated Learning (FL) has emerged as a promising distributed machine learning paradigm, with its core objective centered on addressing data privacy challenges through decentralized collaborative training mechanisms. While FL effectively mitigates data leakage risks and complies with regulations like the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) that mandate local data storage, it fails to satisfy the "right to be forgotten" granted to users under these regulations. To achieve compliance with the "right to be forgotten," Federated Unlearning (FU) has been

本课题得到国家自然科学基金国家自然科学基金面上项目（62572502）、国家自然科学基金国家自然科学基金青年科学基金项目（C 类）[原青年科学基金项目]（62302539）资助。**唐湘云**，博士，副教授，硕士生导师，中国计算机学会(CCF)会员，主要研究领域为人工智能安全、联邦学习、数据安全与隐私保护。**武杭**，硕士研究生，主要研究领域为联邦学习。**王亚杰**，助理教授，硕士生导师，中国计算机学会(CCF)会员，主要研究领域为数据安全、人工智能安全。**沈蒙**，教授，博士生导师，中国计算机学会(CCF)会员，主要研究领域为数据安全、人工智能安全、区块链安全。**翁彧**，教授，博士生导师，主要研究领域为人工智能、深度学习。**祝烈煌**，教授，博士生导师，中国计算机学会(CCF)会员，主要研究领域为密码算法、安全协议、区块链技术、云计算安全。

proposed. Currently, federated unlearning has become a promising paradigm to fulfill users' "right to be forgotten," enabling the elimination of users' data contributions from global models while meeting data privacy protection requirements. However, existing federated unlearning methods typically either sacrifice storage space to reduce unlearning time or sacrifice unlearning time to reduce storage space, failing to balance time overhead and storage overhead. Beyond introducing significant time overheador storage overhead, current FU methods generally support only specific unlearning levels—such as client-level or class-level unlearning—limiting their practicality in real-world scenarios. For instance, in a smart healthcare setting, multiple hospitals may collaboratively train disease diagnosis models via federated learning while preserving data privacy. However, if a hospital needs to withdraw from the collaboration due to policy changes, identifies mislabeled data, or encounters sensitive disease categories in the model, it would require client-level, sample-level, and class-level unlearning, respectively. Therefore, in this paper, we propose FedUR, a lightweight federated unlearning framework that supports three unlearning levels: client-level, sample-level, and class-level. FedUR achieves clean unlearning effect while preserving global model performance and effectively balancing time overhead and storage overhead. To accommodate the three unlearning levels and optimize resource trade-offs, FedUR decouples the federated unlearning process into two stages: adaptive unlearning and optimized recovery. In the stage of adaptive unlearning, the target client employs stochastic gradient ascent (SGA) on target data to achieve unlearning while minimizing storage overhead. If the unlearning model violates constraints, projected gradient descent (PGD) is applied to prevent degradation into a random model. In the stage of optimized recovery, the server leverages knowledge distillation with an outsourced labeled dataset to restore the performance of the global model, achieving faster recovery compared to traditional post-training or methods that integrate the unlearning process into federated learning. Extensive experiments on real-world datasets compare FedUR against five state-of-the-art FU methods across three unlearning levels using four metrics: accuracy (reflecting the post-unlearning model performance), backdoor attack success rate (reflecting the unlearning effect), time overhead, and storage overhead (reflecting the efficiency). Results demonstrate that FedUR achieves the highest accuracy (reflecting the superior model performance), lowest backdoor attack success rate (reflecting the cleanest unlearning effect), minimal storage overhead, and second-lowest time overhead (reflecting the high efficiency). While the FUG method is the fastest, it shows a 1%-10% accuracy gap compared to FedUR, and its unlearning effect is incomplete. Overall, FedUR optimizes model performance, unlearning effect, and efficiency. These findings validate FedUR's effectiveness and efficiency, enabling robust unlearning, preserving model performance, and balancing time overhead and storage overhead. It addresses the typical performance degradation and spatiotemporal trade-off challenges associated with federated unlearning.

Key words federated learning; federated unlearning; privacy protection; adaptive unlearning; optimized Recovery

1 引言

联邦学习(Federated Learning，FL)[1]是一种有前途的分布式机器学习范式，其核心目标是通过去中心化的协作训练机制解决数据隐私保护问题[2-4]。联邦学习中各客户端基于本地数据训练本地模型，仅将更新后的本地模型参数上传至中央服务器；中央服务器通过聚合算法聚合各客户端上传的本地模型参数，更新全局模型并下发给各客户端继续迭代。因此，联邦学习有效避免了数据泄露风险，被广泛应用于数据隐私要求高或者存在数据孤岛问题的医疗、金融等领域[5-6]。然而，随着“被遗忘权”日益成为数据隐私保护的核心诉求，现有的联邦学习框架因缺乏高效的数据遗忘机制，难以满足《中华人民共和国数据安全法》《中华人民共和国个人信息保护法》、欧盟《通用数据保护条例》(GDPR)[7]以及美国《加州消费者隐私法》(CCPA)[8]等法规对

用户“被遗忘权”的合规性要求。

因此，联邦遗忘学习(Federated Unlearning，FU)被提出，其核心机制是消除目标数据在全局模型中的历史贡献，以实现用户“被遗忘权”的合规要求[9]。根据遗忘粒度，联邦遗忘学习主要包括三种级别[10-11]：客户端级遗忘，即删除某一客户端的全部数据贡献，适用于用户主动退出系统或检测到恶意客户端的场景；样本级遗忘，即删除客户端本地数据中的特定样本，适用于数据标注错误修正或用户请求删除部分敏感数据的场景；类别级遗忘，即删除与特定类别相关的所有数据影响，常见于模型合规性调整，例如自动驾驶模型需移除涉及隐私的街景特征[12]、内容推荐系统需过滤违规类别的训练痕迹等等。联邦遗忘学习的应用价值不仅体现在满足隐私合规需求，还在提升联邦学习系统安全性方面发挥关键作用。当检测到全局模型因恶意客户端发起后门攻击（如注入特定触发模式）而遭受污染时，联邦遗忘学习可快速消除污染数据的潜在影响[13-17]，从而增强模型的鲁棒性和防御能力。

然而，现有的联邦遗忘学习方法无法平衡时间开销和空间开销，主要分为牺牲存储空间换取遗忘时间和牺牲遗忘时间换取存储空间两类。第一类联邦遗忘学习方法通过校准客户端的历史参数信息来重建全局模型以实现遗忘，用存储空间换取遗忘时间[18-21]。虽然这类方法减少了时间开销，但它们通常需要存储客户端的历史参数信息，增加了存储开销和隐私泄露的风险。第二类联邦遗忘学习方法，如从头开始重新训练和通道修剪结合微调[22]等，不需要存储客户端的历史参数信息，但是时间开销增加。除了上述两个类别以外，只有少数联邦遗忘学习方法同时考虑了时间开销和存储开销。FUG[23]虽然考虑了同时优化时间开销和存储开销，但它并不能保证全局模型的遗忘效果和遗忘后的模型性能，导致遗忘后的全局模型不可用，违背了联邦遗忘学习算法的核心目标。

因此，亟需设计一种轻量级的联邦遗忘学习框架，该框架能够在时间开销和存储开销之间取得平衡，适用于三种遗忘级别，并且在实现有效遗忘的同时维持全局模型在剩余数据上的性能。然而，设计一个能够平衡时间开销和存储开销、维持全局模型性能以及实现有效遗忘的轻量级联邦遗忘学习框架存在很大挑战。首先，从全局模型中消除遗忘数据的贡献通常需要利用客户端的历史参数信息或使用剩余数据从头开始重新训练，这两种方法分别增加了存储开销和时间开销。其次，遗忘操作可能会降低全局模型的性能，从而需要后训练过程来恢复全局模型性能，这会进一步增加时间开销。最后，即使提出了轻量级的联邦遗忘学习方法，但该方法仍需克服同时适用于客户端级、样本级和类别级三种遗忘级别的技术挑战。这是因为优化时间开销或存储开销的技术通常是针对特定级别的遗忘定制的，而不是三种遗忘级别[18][22][24]。

在本文中，我们提出了一种新颖的适用于三种遗忘级别的轻量级联邦遗忘学习框架，名为FedUR。该框架平衡时间开销和存储开销的同时，保证了全局模型性能合理恢复，并实现了有效的遗忘。现有的联邦遗忘学习方法通常在联邦遗忘学习过程中引入大量的空间开销或者时间开销。为了使本文框架适用于三种遗忘级别并实现轻量级特性，我们在 FedUR 中将联邦遗忘学习过程解耦成自适应遗忘和优化恢复两个阶段。自适应遗忘阶段采用无历史信息依赖的随机梯度上升与投影梯度下降，避免存储多轮参数，最小化遗忘过程中的存储开销。优化恢复阶段通过知识蒸馏恢复模型性能，相较于基于后训练或者将遗忘过程集成到联邦学习的方法恢复性能速度更快，减少时间开销。

具体地，为了最小化存储开销，在自适应遗忘阶段，目标客户端在遗忘数据上执行随机梯度上升，并采用投影梯度下降满足约束防止执行随机梯度上升的模型退化为随机模型，从而实现遗忘而无需存储客户端的历史参数信息。这避免了通常与消除遗忘数据贡献相关联的额外存储开销。而且目标客户端在自适应遗忘阶段只需执行很少轮次的迭代即可实现遗忘目标数据，且投影梯度下降操作只在遗忘模型不满足约束时执行，因此 FedUR 的时间开销主要取决于优化恢复阶段。为了减少全局模型

性能恢复的时间，在优化恢复阶段，服务器使用外包标记的蒸馏数据集进行知识蒸馏。相较于后训练或者将遗忘过程集成到联邦学习的方法，知识蒸馏恢复性能速度更快，减少恢复模型性能通常所需的时间开销。因为后训练过程需要剩余客户端的参与，本质是剩余客户端的联邦学习过程，需要本地训练、上传、服务器聚合和下发，而知识蒸馏只需服务器的参与，在服务器端训练遗忘模型，计算开销更小，时间开销更少，恢复性能速度更快。

FedUR 通过自适应遗忘和优化恢复两阶段的协同设计，实现了时间开销与存储开销的协同优化，在时间开销和存储开销之间实现了有效平衡。在自适应遗忘阶段，目标客户端通过随机梯度上升更新本地模型参数实现遗忘；在优化恢复阶段，服务器通过知识蒸馏调整全局模型实现性能恢复。FedUR 在自适应遗忘和优化恢复阶段中使用的策略独立于模型结构和遗忘级别，确保 FedUR 适用于三种遗忘级别，解决同时适用于客户端级遗忘、样本级遗忘和类别级遗忘的核心挑战，能够有效响应客户端的遗忘请求。

本文的主要贡献总结如下：

（1）本文提出了一种轻量级的适用于三种遗忘级别的联邦遗忘学习框架 FedUR，该框架平衡了时间开销和存储开销，同时保证了有效的遗忘和遗忘后的全局模型性能合理恢复。

（2）本文在 FedUR 方法中将联邦遗忘学习过程解耦成自适应遗忘和优化恢复两个阶段，在自适应遗忘阶段通过随机梯度上升结合投影梯度下降将存储开销最小化，在优化恢复阶段通过知识蒸馏减少性能恢复期间的时间开销，实现了存储开销和时间开销的均衡。

（3）本文在 MNIST 和 CIFAR-10 两个数据集上，通过系统性实验评估了 FedUR 和五种前沿的联邦遗忘学习方法在客户端级、样本级和类别级三个遗忘级别下的性能表现，结果表明 FedUR 实现了时间开销和空间开销的均衡。在时间开销和空间开销方面表现优越的同时，FedUR 的全局模型恢复准确率比其他方法高出 1%至 10%。

本文第 2 节介绍遗忘学习的相关工作；第 3 节介绍联邦遗忘学习问题的形式化定义以及目标；第 4 节详细介绍本文提出的轻量级联邦遗忘学习框架 FedUR；第 5 节对 FedUR 和比较的联邦遗忘学习方法进行时间复杂度和空间复杂度分析；第 6 节进行实验设计并验证 FedUR 的有效性和高效性；第 7 节总结本文的工作并进行展望。

2 相关工作

本文的目标是设计一个适用于三种遗忘级别的轻量级联邦遗忘学习框架，因此在本节中系统地回顾机器遗忘学习和联邦遗忘学习的研究进展。

2.1机器遗忘学习

机器遗忘学习（Machine Unlearning，MU）的概念首先由 Cao 和 Yang[25]提出，他们从隐私、安全和可用性的角度分析了遗忘的必要性，并且通过将学习算法转换为求和形式来实现遗忘。之后，一些针对特定机器学习模型的机器遗忘学习方法被提出，例如 k-means 聚类[26]、随机森林[27]以及变分贝叶斯[28]。后来，一种通用的机器遗忘学习方法 SISA[29]被提出，其核心思想是将训练数据划分为多个不重叠的分片，并为每个分片训练一个子模型。当某个数据点需要被遗忘时，只需要删除该数据所在的分片，然后在受影响的部分重新训练，而不需要对整个数据集进行重新训练。ARCANE[30]架构通过选择代表性数据去除冗余，并分割数据训练多个子模型。在遗忘学习阶段，它将朴素重训练转化为多个单分类任务，从而在维持模型性能的同时降低了重新训练成本。性能不变模型增强（PUMA）[31]设计目标是保障遗忘后模型的性能，其通过在训练期间记录模型更新信息来提高遗忘效率。

尽管这些机器遗忘学习方法可以应用于各种机器学习模型，但是它们不适用于联邦学习场景，因为机器遗忘学习假设的是传统的集中式设置，训练数据是普遍可访问的。而联邦学习假设的是分布式环境设置，在分布式环境中没有任何一个客户端或服务器可以访问整个数据集。

2.2联邦遗忘学习

近些年来已经有许多研究专注于在联邦学习场景中实现对特定数据的遗忘。现有的联邦遗忘学习方法主要分为牺牲存储空间换取遗忘时间和牺牲遗忘时间换取存储空间两类。牺牲存储空间换取遗忘时间的方法主要包括 FedEraser[18]、FUKD[19]、FedRecovery[20]和 FedAU[21]等，这类方法通过存储客户端历史参数信息来减少遗忘目标数据所需的时间。FedEraser[18]是第一个提出的联邦遗忘学习方法，通过服务器以规则的轮次间隔保留客户端的模型参数信息和相应轮次的索引，以便校准保留的更新来重建全局模型，而不是从头开始重新训练。FUKD[19]通过采用懒惰学习策略，从全局模型中减去发出遗忘请求的目标客户端的历史参数更新来消除其贡献，并使用知识蒸馏来恢复模型性能。FedRecovery[20]引入梯度残差的概念来量化增量效应，通过从全局模型中删除梯度残差的加权和来消除某个客户端的影响，并利用高斯噪声使得遗忘后的模型和重新训练的模型在统计上不可区分。FedAU[21]通过将一个轻量级的辅助遗忘模块集成到联邦学习训练过程，并采用线性操作结合辅助遗忘模块和学习模块来实现遗忘。

牺牲遗忘时间换取存储空间的方法主要包括重新训练、Class-disc[22]、FedU[32]和 FedOSD[33]等，这类方法通过避免存储客户端历史参数信息的方式实现遗忘，以此减少遗忘所需的存储空间开销，但会增加遗忘所需的时间。其中重新训练是联邦遗忘学习问题的基准方法，即删除遗忘数据从头开始重新训练。Class-disc[22]通过引入词频-逆文档频率的概念来评估卷积神经网络的通道和类别之间的相关值，将相关值作为修剪的依据以消除遗忘类别数据的贡献，最后通过微调恢复模型性能。FedU[32]通过量化客户端对全局模型的动态影响，设计近似算法估算目标客户端的历史影响，从而反向调整模型参数以消除其数据痕迹。FedOSD[33]引入遗忘交叉熵损失克服梯度上升的收敛问题，通过正交最速下降法和动态梯度投影策略分别解决遗忘阶段的更新冲突可用性下降问题和后训练恢复阶段的模型后退问题。除了上述两个类别以外，只有少数联邦遗忘学习方法同时考虑了优化时间开销和存储开销，例如 FUG[23]通过将遗忘过程表述为约束问题，并通过随机梯度上升和弹性权重巩固在实现遗忘的同时保持模型在剩余数据上的性能。

尽管这些联邦遗忘学习方法能够响应遗忘请求，但是除了 FedAU[21]和 FUG[23]以外，其他方法只能实现客户端级遗忘或者类别级遗忘。而且现有的大多数联邦遗忘学习方法在实现遗忘和恢复模型性能过程中引入了大量的时间开销或存储开销。虽然 FUG 实现了三种级别的遗忘并且平衡了时间开销和空间开销，但是不能保证遗忘效果和遗忘后的全局模型性能。因此，本文提出了一种轻量级联邦遗忘学习框架 FedUR，该框架适用于三种遗忘级别，而且实现了时间开销和存储开销和均衡，在保证全局模型性能的同时实现了有效的遗忘。

3 问题定义

本节从形式化定义与目标两个角度系统阐述联邦遗忘学习的核心问题，详细介绍联邦遗忘学习问题的定义、分类以及目标。

3.1 联邦遗忘学习问题的形式化定义

联邦遗忘学习关注的重点是从全局模型中消除遗忘数据的贡献，同时维持全局模型在剩余数据上的性能。具体地，在正式定义联邦遗忘学习问题之前，我们引入一些必要的符号。我们考虑一个拥有K个客户端的联邦学习设置，每一个客户端拥有数据集$D_k = \{(x_j, y_j)\}_{j=1}^{n_k}$，其中$k$是客户端的序号，$n_k$是客户端$k$的样本数量。对于模型参数$\omega$和本地数据集$D_k$，$F_k(\omega)$代表客户端$k$的损失函数，$f(\omega)$代表服务器端的损失函数。

当目标客户端i发出遗忘请求时，我们将其请求

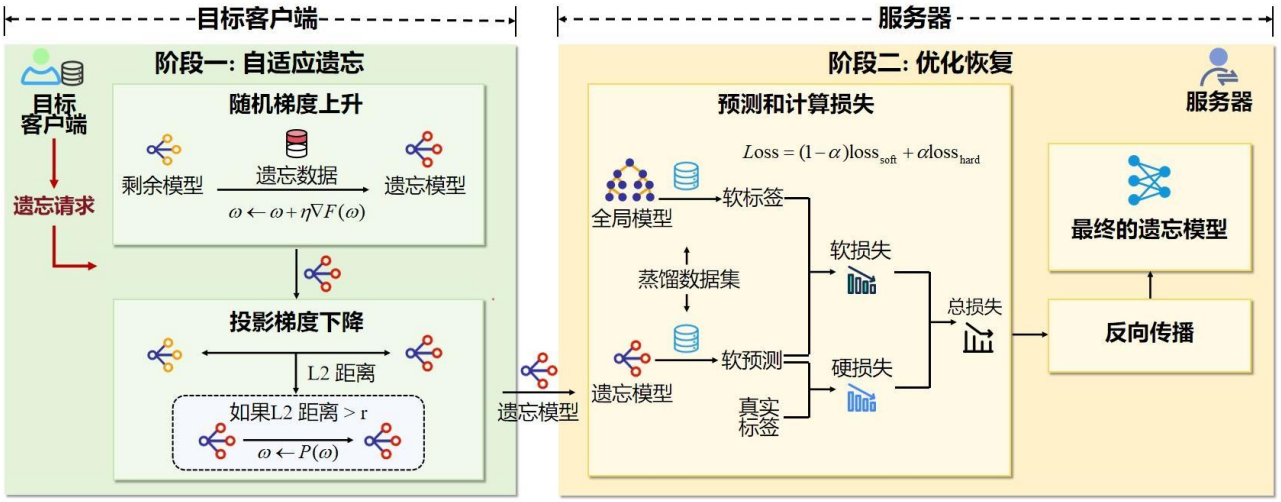


图 1 轻量级的联邦遗忘学习框架 FedUR

遗忘的数据定义为D^u，剩余不需要遗忘的数据定义为$D^r = D - D^u$，遗忘数据和剩余数据的样本数量分别是n_u和n_r。联邦遗忘学习的任务是学习一个新的模型参数$\hat{\omega}$，具体通过优化下面两个目标函数实现：

为了保证联邦遗忘学习方法能够实现有效的遗忘，遗忘之后的本地模型对于遗忘数据的预测输出应该是错误的，如公式(1)所示：

$$maxF_i(\omega) = \frac{1}{n_u} \sum_{j \in D^u} \ell_j(\omega) \tag{1}$$

其中ℓ是损失例如交叉熵损失。

为了维持全局模型在剩余数据上的性能，遗忘之后的全局模型在剩余数据上的预测结果应该是正确的，如公式(2)所示：

$$minf(\omega) = \frac{1}{n_r} \sum_{j \in D^r} \ell_j(\omega) \tag{2}$$

根据遗忘请求遗忘的数据的不同，联邦遗忘学习主要包括客户端级遗忘、样本级遗忘和类别级遗忘。我们详细介绍联邦遗忘学习的遗忘级别如下：

（1）客户端级联邦遗忘学习：遗忘数据D^u是某个客户端的所有本地数据，该客户端的所有数据的贡献应该从全局模型中移除。

（2）样本级联邦遗忘学习：遗忘数据D^u是某个客户端本地数据中的部分样本，这些样本的贡献应当从全局模型中移除。

（3）类别级联邦遗忘学习：遗忘数据D^u是某个类的相关数据，即需要从全局模型中移除该类别相关的所有数据的影响。

3.2联邦遗忘学习的目标

本文的目标是设计一种轻量级的联邦遗忘学习框架，该框架能够适用于三种遗忘级别，平衡时间开销和存储开销，保证遗忘后的全局模型性能并实现有效的遗忘。本文提出的联邦遗忘学习方法要实现的目标介绍如下：

（1）目标数据有效遗忘。当某些数据需要被移除时，系统需要确保被遗忘数据的影响从全局模型中消除，好像遗忘数据从未参与过训练过程那样，即尽可能接近从头开始重新训练的方法。

（2）全局模型性能合理恢复。联邦遗忘学习方法应确保在执行遗忘操作后，全局模型的性能尽可能接近或达到从头开始重新训练方法的水平，确保模型实用性不受显著影响[21]。

（3）轻量级的遗忘框架。本文的目标是提出一个轻量级的联邦遗忘学习框架，该框架可以平衡时间开销和存储开销，同时确保实现有效遗忘并维持全局模型性能，避免因遗忘操作引入过高的系统开销。

4 轻量级联邦遗忘学习框架 FedUR

现有的联邦遗忘学习方法主要包括牺牲遗忘

时间换取存储空间和牺牲存储空间换取遗忘时间两类，不能平衡时间开销和存储开销。因此，本文提出了一个轻量级的联邦遗忘学习框架 FedUR，该框架适用于三种遗忘级别，包括客户端级遗忘、样本级遗忘和类别级遗忘。而且 FedUR 能够平衡时间开销和存储开销，维持全局模型性能的同时实现有效遗忘。本文在 FedUR 方法中将联邦遗忘学习过程解耦成自适应遗忘和优化恢复两个阶段。

（1）自适应遗忘阶段。在自适应遗忘阶段，当目标客户端发出遗忘请求时，目标客户端首先在本地遗忘数据上执行随机梯度上升，执行后模型如果不满足约束条件则采用投影梯度下降来满足约束条件防止遗忘模型退化成随机模型，以此消除遗忘数据的影响，然后目标客户端将遗忘模型上传到服务器。因此自适应遗忘阶段无需存储历史参数信息，仅需较少轮次且计算开销较小的随机梯度上升和投影梯度下降操作，避免了通常与遗忘相关的较大的存储开销。

（2）优化恢复阶段。在优化恢复阶段，服务器获取目标客户端上传的遗忘模型后，利用外包标记的蒸馏数据集和知识蒸馏技术来恢复遗忘模型的性能。传统的后训练方法需要其他客户端参与本地训练、上传更新并经过多轮服务器聚合迭代，时间开销较大。而知识蒸馏只需要在服务器端训练遗忘模型，时间开销较小。

对于三种级别的遗忘，它们都需要首先在目标客户端上执行自适应遗忘阶段，然后在服务器上执行优化恢复阶段。我们将 FedUR 的整体流程展示在图 1 中。FedUR 的计算流程如协议 1 所示。接下来我们将详细介绍 FedUR 的两个阶段，并详细阐述如何基于这两个阶段实现三种级别的遗忘。

协议 1.轻量级联邦遗忘学习框架 FedUR

输入：全局联邦学习模型ω^T，客户端级别的遗忘数据D_i，样本级别的遗忘数据D_i^u，类别级别的遗忘数据D_i^c，遗忘轮次T_u，学习率η，半径r，蒸馏数据集D_d，蒸馏轮次T_d，蒸馏温度T_t

输出：最终遗忘模型M

1. IF 客户端级遗忘
2.计算剩余模型$\omega_{remain} = \frac{1}{K-1}(K\omega^T - \omega_i^T)$
3.$\omega_i^u = Unlearning(\omega_{remain}, D_i, T_u, \eta, r)$
4.$M = Recovery(\omega^T, \omega_i^u, D_d, T_d, T_t)$
5. ELSE IF 样本级遗忘
6.计算剩余模型$\omega_{remain} = \frac{1}{K-1}(K\omega^T - \omega_i^T)$
7.$\omega_i^u = Unlearning(\omega_{remain}, D_i^u, T_u, \eta, r)$
8.$M = \boldsymbol{Recovery}(\omega^T, \omega_i^u, D_d, T_d, T_t)$
9. ELSE 类别级遗忘
10.计算剩余模型$\omega_{remain} = \omega^T$
11.$\omega_i^u = Unlearning(\omega_{remain}, D_i^c, T_u, \eta, r)$
12.$M = Recovery(\omega^T, \omega_i^u, D_d, T_d, T_t)$
13. RETURN 最终遗忘模型M

Unlearning$(\omega_{remain}, D^u, T_u, \eta, r)$
15. 初始化投影算子P:$\| \omega - \omega_{remain} \|_2 \leq r$，$\omega \leftarrow \omega_{remain}$
16. FOR $t = 1,2,\ldots,T_u$
17.$\omega \leftarrow \omega + \eta\nabla F(\omega)$
18. IF L2 距离$(\omega, \omega_{remain}) > r$
19.$\omega \leftarrow P(\omega)$
20.$\omega_i^u \leftarrow \omega$
21. RETURN ω_i^u

Recovery$(\omega^T, \omega_i^u, D_d, T_d, T_t)$
23. 初始化全局联邦学习模型M_{global},遗忘模型M
24. FOR $t = 1,2,\ldots,T_d$
25.$y_{teacher} \leftarrow M_{global}(D_d), T_t$
26.$y_{student} \leftarrow M(D_d), T_t$
27.通过$y_{teacher}$和$y_{student}$计算$loss_{soft}$
28.通过y_{D_d}和$y_{student}$计算$loss_{hard}$
29.$Loss = (1-\alpha)loss_{soft} + \alpha loss_{hard}$
30.根据$Loss$反向传播
31. RETURN 最终遗忘模型M

4.1自适应遗忘

在自适应遗忘阶段，我们的目标是从全局模型中消除遗忘数据的影响。为了避免因存储模型参数信息而带来的存储开销，本文提出使用随机梯度上升来实现遗忘。目前深度学习中的大多数模型依赖梯度下降进行训练和优化，梯度下降可以理解为模型从训练数据中逐渐学习到类内数据的共性和类间数据的特征，从而扩展泛化边界以包含更多数据的过程。梯度下降可以理解为一个学习过程，那么梯度上升可以理解为一个遗忘过程，通过缩小泛化

边界来消除模型对特定数据的分类能力。因此在遗忘阶段，我们的目标不是学习经验损失最小化的模型参数，而是学习经验损失最大化的模型参数。

然而，仅仅在遗忘数据上执行梯度上升是存在局限的，因为本文的目标是遗忘整个客户端或者部分样本或者某个类别的数据，而不是遗忘个别样本。常用的损失函数如交叉熵损失是无界的，对于无界损失函数，每一次随机梯度上升都推动模型参数向损失更大的方向变化，并且在几轮之后模型可能会退化为随机初始化的模型[34]。为了维持模型在剩余数据上的性能，防止退化为随机模型，我们希望在自适应遗忘阶段中获得的遗忘模型足够接近于已经学习了剩余数据分布的剩余模型，即ω_{remain}。我们将自适应遗忘阶段总结如下：当客户端i提出遗忘请求时，客户端i首先使用剩余模型作为初始模型，然后客户端i在遗忘数据D^u上执行随机梯度上升。为了防止遗忘模型退化为随机模型，我们将遗忘模型约束在ω_{remain}周围半径为r的l_2范数球内。因此，遗忘过程可以表述为如下约束优化问题：

$$maxF_i(\omega)=\frac{1}{n_u}\sum_{j\in\mathcal{D}^u}\ell_j(\omega)$$

$$s.t.\|\omega-\omega_{remain}\|_2\leq r \quad (3)$$

其中D^u代表客户端i的遗忘数据，n^u代表遗忘数据的样本数量。这一约束不仅防止模型退化为随机模型，还通过限制参数变化的幅度平衡了不同样本或类别的梯度更新强度。若某样本的梯度更新方向导致参数偏离剩余模型过远，投影梯度下降会将其投影回约束范围内，避免局部过大的遗忘强度。所有遗忘数据的梯度更新被约束至同一参数空间范围，从而缓解了遗忘强度的差异性。

我们采用投影梯度下降来解决上述约束优化问题。具体来说，我们使用投影算子$P:\|\omega-\omega_{remain}\|_2\leq r$来代表剩余模型周围半径为$r$的$l_2$范数球。如果执行完随机梯度上升的遗忘模型与剩余模型之间的距离大于半径r，则需要通过投影算子将结果投影回约束集。客户端i在随机梯度上升的基础上采用投影梯度下降来更新模型，防止模型退化为随机初始化的模型，如公式(4)所示：

$$\omega\leftarrow P(\omega+\eta\nabla F(\omega)) \quad (4)$$

其中η是学习率，$\nabla F(\omega)$是损失函数F相对于模型ω的梯度。在解决完上述优化约束问题后，目标客户端i将得到的遗忘模型上传到服务器。

4.2优化恢复

在优化恢复阶段，由于在自适应遗忘阶段获得的遗忘模型在剩余数据上的性能显著降低，因此我们需要采取一些策略来恢复其性能。然而，我们不能利用常规的恢复模型性能的后训练方法。一方面，联邦学习不保证客户端始终保留本地数据用于处理遗忘请求，这意味着我们不能依赖于使用这些数据进行重新训练来恢复遗忘模型的性能。另一方面，对于类别级联邦遗忘学习，要求其他客户端只在排除目标类别的剩余类别数据上训练是不可行的，这表明后训练方法不适用于类别级遗忘，不是适用于三种遗忘级别的性能恢复方法。

为了增加可移植性，本文提出使用知识蒸馏方法来恢复遗忘模型的性能。知识蒸馏的核心思想是从一个复杂的参数量较大的模型（通常称为教师模型）中提取知识，并把知识转移给一个更紧凑、更精简的模型（通常称为学生模型）[35]。该过程有效降低了模型复杂度和计算开销。因此，与后训练相比，知识蒸馏可以更快地恢复遗忘模型的性能。

为了将知识蒸馏应用于联邦遗忘学习任务，服务器将全局联邦学习模型M_{global}视为教师模型，并且将在自适应遗忘阶段获得的遗忘模型M视为学生模型。服务器使用外包标记的蒸馏数据集来训练遗忘模型并恢复其性能。知识蒸馏的细节如下：教师模型和学生模型分别对蒸馏数据集进行预测，预测输出除以温度参数T_t，然后进行 Softmax 计算以获得软标签和软预测。区别于 FUKD[19]仅依赖教师软标签的蒸馏损失设计，本文基于外包标记数据集的双重监督特性，利用融合蒸馏损失（软目标）与学生损失（硬目标）的联合优化框架。具体而言，蒸

馏损失通过 KL 散度最小化教师与学生的软化输出分布，传递教师模型隐含的类别相关性知识；学生损失通过交叉熵直接对齐学生预测与真实标签，强化任务特异性判别特征。总损失设计为软目标和硬目标的加权平均[36]，硬目标通过真实标签提供高置信度监督信号，与软目标的类间关系知识形成互补，缓解教师模型错误传播风险，如公式(5)所示。

$$Loss=(1-\alpha)loss_{soft}+\alpha loss_{hard} \quad (5)$$

其中α为软硬目标的权重系数，$loss_{soft}$为蒸馏损失，$loss_{hard}$为学生损失。训练初期阶段知识迁移更加依赖教师网络的贡献，因此软目标的加权系数要较大，因为这样可以帮助学生网络识别相对简单的样本，但在训练后期需要适当减小软目标的比重，这样真实标签可以帮助学生网络识别相对困难的样本。另外，教师网络的精度通常高于学生网络，而且教师网络精度越高，越有利于学生网络的学习，蒸馏效果越好。通过优化恢复阶段，服务器获得性能合理恢复的最终遗忘模型，系统以此继续运行。

4.3 FedUR实例化应用

在本节中，我们将详细介绍如何利用自适应遗忘和优化恢复两个阶段来实现三个级别的遗忘，包括客户端级遗忘、样本级遗忘和类别级遗忘。

4.3.1 客户端级遗忘

客户端级遗忘即消除某个客户端的所有数据贡献，利用 FedUR 实现的过程总结如下：当客户端i发出客户端级遗忘请求时，客户端i首先计算剩余模型并将其作为初始模型。对于客户端级遗忘，目标客户端通过从第T轮聚合之后的全局联邦学习模型中减去第T轮的目标客户端i的本地模型来计算剩余模型，即$\omega_{remain}=\frac{1}{K-1}(K\omega^T-\omega_i^T)$。其中$\omega^T$是$T$轮之后的全局联邦学习模型，$\omega_i^T$是第$T$轮的目标客户端$i$的本地模型。然后，客户端$i$执行自适应遗忘阶段，包括对整个本地数据的随机梯度上升和不满足约束条件而执行的投影梯度下降，避免退化为随机模型。执行完自适应遗忘阶段后，客户端i将遗忘模型上传到服务器。最后，在服务器端进行优化恢复阶段。服务器中的蒸馏数据集与客户端本地数据集保持相同的类别均衡结构，即每个数据类别在蒸馏数据集中和客户端本地数据集中的样本数量完全相同。在优化恢复阶段通过知识蒸馏成功恢复遗忘模型性能后，客户端获得服务器下发的最终遗忘模型，以此继续运行。

4.3.2 样本级遗忘

当遗忘部分敏感或者标注错误的样本时即样本级遗忘场景，FedUR 实现样本级遗忘的过程如下：当客户端i发出样本级遗忘请求时，客户端i首先计算剩余模型并将其作为初始模型。对于样本级遗忘，剩余模型和蒸馏数据集是和客户端级遗忘一致的，因为样本级遗忘和客户端级遗忘的剩余数据分布是一致的。所以样本级遗忘的剩余模型同样为$\omega_{remain}=\frac{1}{K-1}(K\omega^T-\omega_i^T)$，蒸馏数据集与客户端本地数据集也拥有相同的数据分布，即蒸馏数据集包括全部类别数据且每个类别数据的样本数量相等。目标客户端i在目标样本上执行自适应遗忘阶段，并把获得的遗忘模型上传到服务器。在优化恢复阶段，服务器借助蒸馏数据集利用知识蒸馏方法恢复遗忘模型的性能。最后可以通过优化恢复阶段得到最终的遗忘模型。

4.3.3 类别级遗忘

本文提出的 FedUR 可以解决遗忘某个类别数据贡献的任务，过程总结如下：当客户端i发出类别级的遗忘请求时，客户端i使用全局联邦学习模型作为初始模型和剩余模型，即$\omega_{remain}=\omega^T$，因为第T轮聚合之后的全局联邦学习模型学习了剩余类的数据分布。客户端i对目标类数据执行自适应遗忘阶段以获得遗忘模型，即对目标类别数据先执行随机梯度上升，如果不满足约束条件执行投影梯度下降将结果投影回约束集，避免遗忘模型退化为随机模型。执行完自适应遗忘阶段，目标客户端将遗忘模型上传到服务器。在优化恢复阶段中，为恢复模型在剩余类别数据上的性能，蒸馏数据集只包括剩余

类的数据，而不包括目标遗忘类别的数据。服务器使用剩余类的数据来恢复性能。执行完优化恢复阶段后，服务器和客户端获得最终的遗忘模型。

5 理论分析

本文提出了轻量级联邦遗忘学习框架 FedUR，该框架旨在均衡时间开销和空间开销，提升数据遗忘效率。因此本节我们对 FedUR 以及比较的联邦遗忘学习方法的时间复杂度和空间复杂度进行分析。

5.1时间复杂度分析

FedUR 包括自适应遗忘阶段和优化恢复阶段。在自适应遗忘阶段，客户端执行随机梯度上升与投影梯度下降，时间复杂度主要取决于梯度计算和投影操作。对于梯度计算，每个遗忘轮次需要对遗忘数据计算一次梯度，复杂度为$O(T_u \cdot n_u \cdot d)$，其中T_u是遗忘轮次，n_u为遗忘数据量，d为模型参数量。对于投影操作，每次随机梯度上升后如果模型参数不满足约束条件则需要执行一次投影梯度下降，复杂度为$O(d)$，代表向量范数计算。这里我们考虑最坏的情况，即每次随机梯度上升后都需要进行投影梯度下降。因此自适应遗忘阶段的总时间复杂度为$O(T_u \cdot (n_u \cdot d + d)) = O(T_u \cdot n_u \cdot d)$。在优化恢复阶段，服务器使用知识蒸馏恢复模型性能，时间复杂度主要取决于教师模型推理和学生模型训练。对于教师模型推理，每个轮次对蒸馏数据集推理，复杂度为$O(T_d \cdot |D_d| \cdot d)$，其中$T_d$为蒸馏轮次，$|D_d|$为蒸馏数据集数据量。对于学生模型训练，每个轮次计算软目标和硬目标损失，复杂度为$O(T_d \cdot |D_d| \cdot d)$。因此，优化恢复阶段总的时间复杂度为$O(T_d \cdot |D_d| \cdot d)$。因此 FedUR 的总时间复杂度为$O(T_u \cdot n_u \cdot d + T_d \cdot |D_d| \cdot d)$，如表 1 所示。

FedUR 通过优化恢复阶段的设计降低时间开销。相较于依赖后训练微调或联邦学习过程进行性能恢复的 FedEraser[18]、Class-disc[22]和 FedAU[21]等方法，FedUR 展现出更快的恢复速度。同时，FedUR 利用带标记的蒸馏数据集进行知识蒸馏，相比 FUKD[19]基于无标记数据集的方法，所需训练轮次更少，进一步缩短了时间开销。上述的遗忘方法的总时间复杂度总结如表 1 所示。具体分析如下：

表 1　时间复杂度和空间复杂度分析

算法名称	时间复杂度	空间复杂度
FedEraser[18]	$O(T \cdot K \cdot d + T_r \cdot \|D_r\| \cdot d)$	$O(T \cdot K \cdot d)$
FUKD[19]	$O(T_d \cdot \|D_d\| \cdot d)$	$O(T \cdot d)$
Class-disc[22]	$O(T_p + T_r \cdot \|D_r\| \cdot d)$	$O(d)$
FUG[23]	$O(T_u \cdot n_u \cdot d)$	$O(d)$
FedAU[21]	$O(T_u \cdot \|D_i\| \cdot d + T_u \cdot \|D\| \cdot d)$	$O(d + m)$
FedUR	$O(T_u \cdot n_u \cdot d + T_d \cdot \|D_d\| \cdot d)$	$O(d)$

FedEraser[18]的时间复杂度取决于模型重建和微调过程。模型重建通过校准历史更新来重建全局模型以移除特定客户端的影响，复杂度主要来自参数更新校准为$O(T \cdot K \cdot d)$，其中 T 是联邦学习轮次。微调过程即在剩余数据上执行多轮联邦学习以恢复性能，复杂度为$O(T_r \cdot |D_r| \cdot d)$，其中$T_r$是微调轮次，$|D_r|$为剩余数据的数据量。因此，FedEraser 的总时间复杂度为$O(T \cdot K \cdot d + T_r \cdot |D_r| \cdot d)$。

FUKD[19]通过线性操作从全局模型中减去目标客户端的历史参数更新实现遗忘，并采用知识蒸馏恢复性能，因此时间复杂度主要取决于知识蒸馏过程。FUKD 的知识蒸馏过程分析同 FedUR，教师模型推理和学生模型训练的复杂度均为$O(T_d \cdot |D_d| \cdot d)$，因此 FUKD 的总时间复杂度为$O(T_d \cdot |D_d| \cdot d)$。

Class-disc[22]主要包括通道剪枝和微调过程。通道剪枝过程中模型通过选择性移除通道来实现遗忘目标类，剪枝的时间开销依赖于模型结构，复杂度为$O(T_p)$。微调过程中剪枝后的模型通过后训练恢复性能，复杂度为$O(T_r \cdot |D_r| \cdot d)$。因此，Class-disc 的总时间复杂度为$O(T_p + T_r \cdot |D_r| \cdot d)$。

FUG[23]的时间复杂度主要取决于在遗忘数据上执行梯度上升和弹性权重巩固操作。在遗忘数据上执行梯度上升的复杂度为$O(T_u \cdot n_u \cdot d)$；弹性权重巩固操作用于防止灾难性遗忘，在此方法中与梯度上升结合，复杂度仍由梯度上升主导为$O(T_u \cdot n_u \cdot d)$。因此，FUG 的总时间复杂度为$O(T_u \cdot n_u \cdot d)$。

FedAU[21]需要训练辅助遗忘模块，并将遗忘过程集成到联邦学习过程中。辅助遗忘模块的训练通

过在修改的本地数据上局部训练实现，复杂度为$O(T_u \cdot |D_i| \cdot d)$，其中$|D_i|$是目标客户端的数据量。联邦学习过程的复杂度为$O(T_u \cdot |D| \cdot d)$，其中$|D|$是所有客户端的数据量。因此，FedAU 的总时间复杂度为$O(T_u \cdot |D_i| \cdot d + T_u \cdot |D| \cdot d)$。

5.2空间复杂度分析

对于空间复杂度分析，本文同样分别分析 FedUR 算法的自适应遗忘和优化恢复阶段。在自适应遗忘阶段，仅需要存储当前模型参数和剩余模型参数，无需保留历史更新，复杂度为$O(d)$。在优化恢复阶段，需要存储教师模型参数$O(d)$、学生模型参数$O(d)$以及蒸馏数据集$O(|D_d|)$。考虑到蒸馏数据集所需要的存储空间大小相较于模型参数可以忽略不计，因此优化恢复阶段的空间复杂度为$O(d)$。由上述分析可知，FedUR 的空间复杂度为$O(d)$，如表 1 所示。FedUR、Class-disc[22]和 FUG[23]相较于 FedEraser[18]、FUKD[19]和 FedAU[21]无需存储历史参数更新信息或者辅助模块，仅需存储当前模型参数。上述方法的空间复杂度分析如下，总结如表 1 所示。FedEraser[18]需存储所有客户端在历史轮次中的模型参数，空间复杂度为$O(T \cdot K \cdot d)$，T是联邦学习轮次。FUKD[19]需存储目标客户端在历史轮次中的模型参数，空间复杂度为$O(T \cdot d)$。Class-disc[22]和 FUG[23]仅需存储当前模型参数，空间复杂度为$O(d)$。FedAU[21]需存储主模型和辅助模块参数，空间复杂度为$O(d + m)$，m为辅助模块参数量。

6 实验

本节先介绍实验环境设置，包括实验数据集、对比方法、超参数设置以及评价指标。然后从遗忘效果、模型性能、可扩展性、遗忘强度以及时间开销和空间开销六个角度评估本文提出的轻量级联邦遗忘学习框架 FedUR。最后通过消融实验探究重要因素对 FedUR 遗忘效果、性能和效率的影响。

6.1实验设置

本文实验代码的实验环境基于 Python 3.8、PyTorch 1.10.0 和 CUDA 11.3 构建，运行环境配置为 Intel Xeon Platinum 8352V 处理器（2.10 GHz，12 个 vCPU）、NVIDIA RTX 4090 GPU 和 60GB 内存，操作系统为 Ubuntu 20.04。

6.1.1 数据集

本文在联邦遗忘学习常用的两个数据集 MNIST[37]和 CIFAR-10[38]上进行实验，以验证本文提出的 FedUR 算法的有效性和高效性。

（1）MNIST：手写数字识别数据集，所有图像都是 28×28 的灰度图像，每张图像包含一个手写数字。MNIST 数据集分为训练集和测试集，训练集 60,000 张图片，测试集 10000 张图片。

（2）CIFAR-10：深度学习图像分类常用数据集，包含飞机、汽车、鸟等 10 个类别物体的 32×32 大小的彩色图片，每个类别有 6000 张图片。CIFAR-10 数据集的训练集包含 50000 张图片，测试集包含 10000 张图片。

本文设置联邦学习客户端数目为 10，每个客户端均分全部训练数据，服务器拥有全部测试数据用以验证模型。以 MNIST 数据集为例，每个客户端拥有 6000 张训练图像，服务器拥有 10000 张测试图像，我们假设客户端的数据是独立同分布的。

MNIST 数据集的实验中使用了联邦遗忘学习任务中常用的卷积神经网络，具体来说该网络包括两个卷积层、两个最大池化层以及两个全连接层。在 CIFAR-10 数据集的实验中，考虑到 CIFAR-10 数据集相较于 MNIST 更复杂，本文使用同样在联邦遗忘学习中常用的 VGG11 卷积神经网络[39]。因此，不同遗忘方法在相同数据集上的神经网络相同，模型规模相同，以便于比较不同遗忘方法的有效性。

6.1.2 对比方法

为了评估 FedUR 方法的有效性和高效性，本文将 FedUR 与五个前沿的联邦遗忘学习方法进行了

比较，这些方法包括牺牲存储空间换取遗忘时间的 FedEraser、FUKD 和 FedAU，牺牲遗忘时间换取存储空间的 Class-disc 以及同时考虑了时间开销和空间开销的 FUG 方法。

表 2 三种遗忘级别下的联邦遗忘学习方法比较

方法	MNIST				CIFAR-10			
	准确率/%	后门攻击成功率/%	时间开销/s	空间开销/MB	准确率/%	后门攻击成功率/%	时间开销/s	空间开销/MB
客户端级联邦遗忘学习								
FedAvg	98.54	94.58	--	--	79.53	94.26	--	--
Retraining	98.69	0.29	41.8	~ 10^0	80.43	0.83	101.6	~ 10^1
FedEraser[18]	97.41	5.80	16.1	~ 10^3	76.18	9.46	42.6	~ 10^3
FUKD[19]	98.28	0.65	8.3	~ 10^2	78.60	1.42	17.0	~ 10^3
Class-disc[22]	--	--	--	--	--	--	--	--
FUG[23]	88.37	2.29	**0.7**	~ 10^0	75.26	88.54	**2.0**	~ 10^1
FedAU[21]	97.21	0.52	7.0	~ 10^1	78.32	1.60	16.9	~ 10^2
FedUR	**98.62**	**0.30**	5.6	~ **10^0**	**79.35**	**1.11**	15.2	~ **10^1**
样本级联邦遗忘学习								
FedAvg	98.53	92.33	--	--	79.41	93.18	--	--
Retraining	98.62	0.22	42.6	~ 10^0	80.19	0.61	112.0	~ 10^1
FedEraser[18]	--	--	--	--	--	--	--	--
FUKD[19]	--	--	--	--	--	--	--	--
Class-disc[22]	--	--	--	--	--	--	--	--
FUG[23]	85.90	2.15	**0.7**	~ 10^0	77.05	88.06	**1.0**	~ 10^1
FedAU[21]	97.03	0.59	7.0	~ 10^1	78.48	1.02	16.9	~ 10^2
FedUR	**98.47**	**0.50**	5.4	~ **10^0**	**78.98**	**0.90**	14.6	~ **10^1**
类别级联邦遗忘学习								
FedAvg	98.31	99.74	--	--	79.36	97.90	--	--
Retraining	98.43	0.26	39.3	~ 10^0	80.42	0.60	94.7	~ 10^1
FedEraser[18]	--	--	--	--	--	--	--	--
FUKD[19]	--	--	--	--	--	--	--	--
Class-disc[22]	98.02	7.29	14.9	~ 10^0	78.38	1.50	33.3	~ 10^1
FUG[23]	83.54	7.40	**0.4**	~ 10^0	61.21	85.38	**0.9**	~ 10^1
FedAU[21]	97.11	0.54	7.0	~ 10^1	78.29	1.32	16.9	~ 10^2
FedUR	**98.18**	**0.44**	4.9	~ **10^0**	**78.48**	**1.00**	13.1	~ **10^1**

注：“--”表示对应的方法在论文中没有实现相应的遗忘级别。FedAvg 和 Retraining 是两个基准方法。FedAvg 是遗忘前的联邦学习状态，FedAvg 与联邦遗忘学习方法在后门攻击成功率上的差异反映了联邦遗忘学习方法的遗忘效果。Retraining 即从头开始重新训练，是联邦遗忘学习问题的基准方法，其他联邦遗忘学习方法的准确率和后门攻击成功率应尽可能接近 Retraining。

（1）FedEraser[18]

这是第一个提出的联邦遗忘学习方法，用于实现客户端级别的遗忘。FedEraser 主要思想是服务器以规则的轮次间隔保留客户端的模型参数更新和相应轮次的索引，以重建全局模型，而不需要从头开始重新训练。

（2）FUKD[19]

这是一种基于懒惰学习策略的联邦遗忘学习方法，通过从全局模型中减去目标客户端的平均历史模型参数更新来实现客户端级别的遗忘，并使用

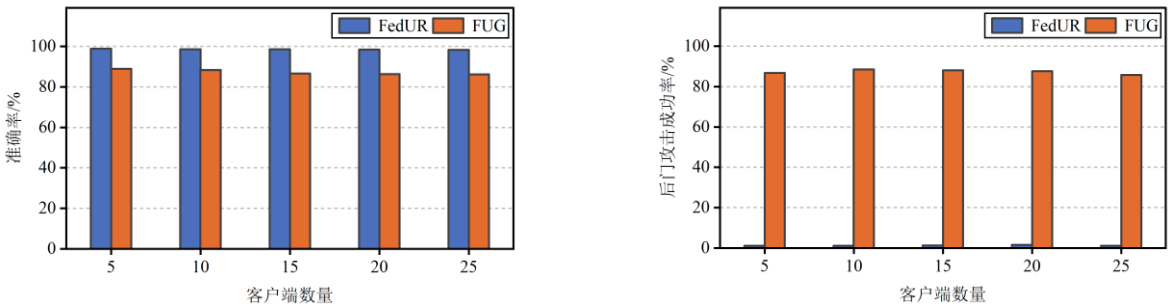
知识蒸馏来恢复模型性能。

（3）FedAU[21]

这是一种实现三种遗忘级别的联邦遗忘学习方法，通过将一个轻量级的辅助遗忘模块集成到联邦学习过程中，并采用简单的线性操作结合辅助遗忘模块和学习模块来促进遗忘。

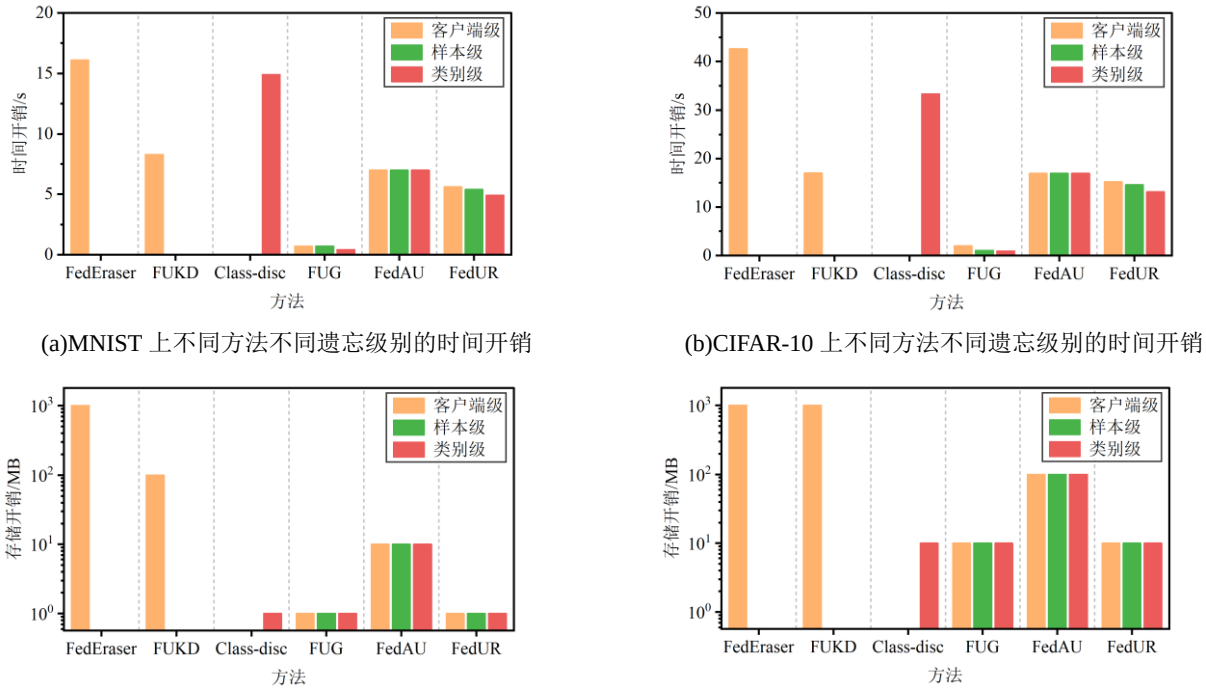
（4）Class-disc[22]

这是一种基于模型剪枝的类别级联邦遗忘学习方法，通过引入词频-逆文档频率



(a)客户端数量对准确率的影响 (b)客户端数量对后门攻击成功率的影响

图 2 MNIST 数据集上客户端数量对 FedUR 和 FUG 的影响



(a)MNIST 上不同方法不同遗忘级别的时间开销 (b)CIFAR-10 上不同方法不同遗忘级别的时间开销

(c)MNIST 上不同方法不同遗忘级别的存储开销 (d)CIFAR-10 上不同方法不同遗忘级别的存储开销

图 3 不同联邦遗忘学习方法在不同遗忘级别的时间开销和存储开销

的概念来评估卷积神经网络的通道和类别之间的相关值，然后将相关值作为通道修剪的依据来消除需要遗忘的类别的数据贡献，最后通过微调恢复因通道修剪而受损的模型性能。

（5）FUG[23]

这是一种时间开销和存储开销都较小的联邦遗忘学习方法，通过结合随机梯度上升和弹性权重巩固实现客户端级联邦遗忘学习、样本级联邦遗忘学习和类别级联邦遗忘学习。

6.1.3 超参数设置

在本文的实验中，我们考虑一个有 10 个客户端的联邦学习场景，每个客户端本地的数据是独立同分布的。在训练过程中，所有客户端都参与每个联邦学习轮次，并且服务器采用 FedAvg 作为聚合算法。对于样本级遗忘，我们将遗忘样本的比例设

置为目标客户端的 20%。对于类别级的遗忘，我们将 MNIST 中的遗忘类设置为数字“1”，将 CIFAR-10 中的遗忘类别设置为“汽车”。对于 MNIST，我们将剩余模型和遗忘模型的距离阈值半径设置为 150，知识蒸馏温度为 7，权重参数为 0.5。对于 CIFAR-10，我们设置剩余模型和遗忘模型的距离阈值半径为 600，温度为 5，权重系数为 0.5。在本地训练中，学习率为 0.01，批次大小是 128。

方法	遗忘级别	准确率/%	后门攻击成功率/%	时间开销/s	存储开销/MB
Retraining	客户端级联邦遗忘学习	94.90	0.37	142.2	~ 10^2
	样本级联邦遗忘学习	95.00	0.30	154.7	~ 10^2
	类别级联邦遗忘学习	95.14	0.00	143.0	~ 10^2
FedUR	客户端级联邦遗忘学习	94.32	0.53	20.4	~ 10^2
	样本级联邦遗忘学习	94.30	0.44	19.8	~ 10^2
	类别级联邦遗忘学习	94.39	0.21	18.7	~ 10^2

表 3 FedUR 在 GTSRB 数据集上与重新训练方法的对比

效性。为了评估遗忘效果，我们在客户端的更新过程中引入了后门攻击[40]。一个成功的遗忘模型应该在正常测试集上表现良好，但在由后门输入触发时降低后门攻击成功率。因此，我们使用模型在正常测试集上的准确率来衡量模型性能，使用模型在带有后门的测试集上的后门攻击成功率来衡量遗忘效果[41-43]。另外，为了验证我们提出的联邦遗忘学习方法 FedUR 的轻量级，我们使用时间开销和存储开销指标来验证 FedUR 的效率。

6.2总体评估

本文分别在 MNIST 和 CIFAR-10 数据集上对 FedUR 和五种前沿的联邦遗忘学习方法进行比较，评价指标为准确率、后门攻击成功率、时间开销和空间开销，实验结果如表 2 所示。

6.2.1 遗忘效果评估

为了评估联邦遗忘学习方法的遗忘效果，我们测量了在 MNIST 和 CIFAR-10 数据集上的后门攻击成功率。根据表 2 的结果，我们可以观察到 FedUR 在两个数据集上的后门攻击成功率均低于比较的方法，说明 FedUR 有最有效的遗忘效果。FedUR 的遗忘效果最接近于重新训练，这表明 FedUR 能够有效地遗忘数据。FedEraser 和 FUG 没有实现有效的遗忘效果，因为它们的后门攻击成功率和重新训练方法之间存在差距，在 MNIST 数据集上客户端级遗忘的后门攻击成功率分别有 5%和 2%的差异。Class-disc 在 CIFAR-10 数据集上有较好的遗忘效果，但在 MNIST 数据集上遗忘效果欠佳，与重新训练相比后门攻击成功率有 5%的差异。FUKD 和 FedAU 的遗忘效果接近重新训练，后门攻击成功率相比重新训练方法的差距均不到 1%。

4 自适应遗忘阶段后不同类别的后门攻击成功率

类别	后门攻击成功率/%
0	0.00
1	0.00
2	0.00
3	0.00
4	0.00
5	0.00
6	0.00
7	0.00
8	0.00

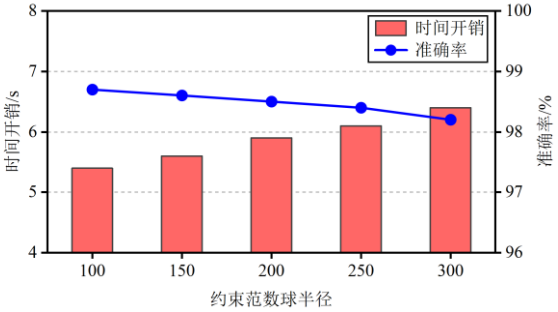
6.2.2 模型性能评估

我们通过测量最终的遗忘模型在正常测试集上的准确率来评估模型性能恢复。如表 2 所示，我们可以观察到 FedUR 在准确率方面优于这些前沿

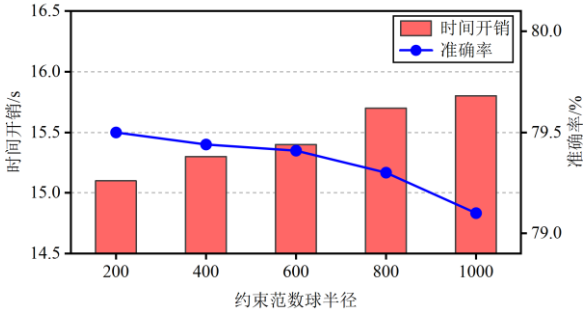
6.1.4 评价指标

本文使用准确率、后门攻击成功率、时间开销和空间开销四个指标来评估 FedUR 的有效性和高

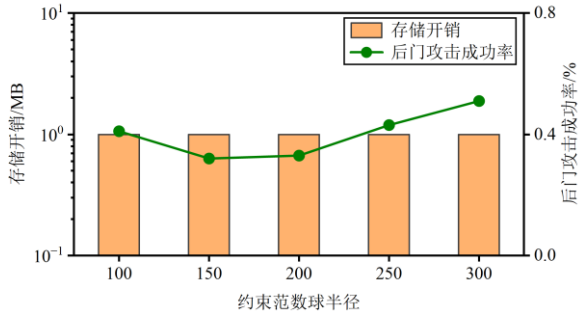
的方法。FedUR 表现出最接近重新训练的准确率，在 MNIST 数据集上相较于重新训练下降幅度小于 0.3%，在 CIFAR-10 数据集上下降幅度小于 2%，说明 FedUR 的最终遗忘模型有很好的实用性。与重新训练相比，FUG 的准确率下降很多，在 MNIST 数据集上下降幅度超过 10%。类似地，FedEraser 和 FedAU 的准确率也有一定程度下降，在 MNIST 数据集上下降幅度在 1%以上，在 CIFAR-10 数据集上下降幅度在 2%以上。FUKD 和 Class-disc 的准确率虽然下降较少，但在 MNIST 数据集上的下降幅度大于 0.4%，下降幅度高于 FedUR。



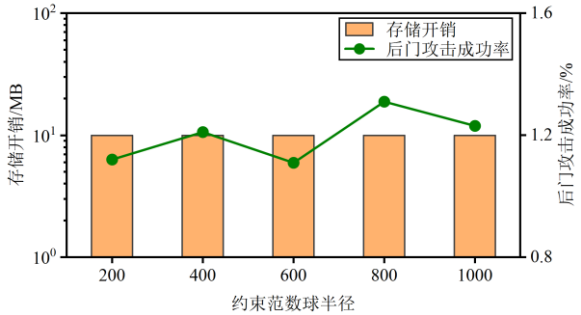
(a)MNIST 上半径对准确率和时间开销的影响



(b)CIFAR-10 上半径对准确率和时间开销的影响

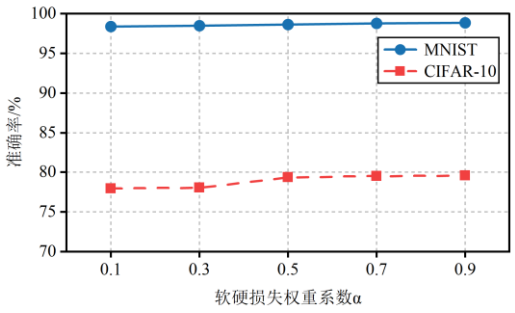


(c)MNIST 上半径对后门攻击成功率和存储开销的影响

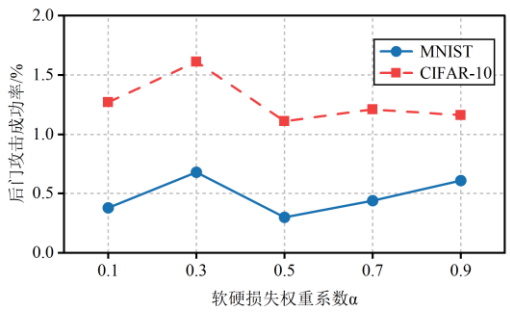


(d)CIFAR-10 上半径对后门攻击成功率和存储开销的影响

图 4 约束范数球半径对 FedUR 的影响



(a)软硬损失权重系数对准确率的影响



(b)软硬损失权重系数对后门攻击成功率的影响

图 5 软硬损失权重系数对 FedUR 的影响

6.2.3 可扩展性评估

为了验证 FedUR 的可扩展性，本文在复杂度相对较高的 GTSRB 数据集上将 FedUR 与联邦遗忘学习中的基准方法重新训练进行对比，其中 GTSRB 数据集的构建基于德国交通标志识别基准，包含 43 类交通标志。如表 3 所示，FedUR 在三个遗忘级别上的准确率与后门攻击成功率均接近于重新训练方法，准确率与后门攻击成功率的差异分别在 1%和 0.3%以内。而且 FedUR 时间开销远小于重新训练方法，空间开销则和重新训练同样不需要存储额外的历史参数。结果表明，FedUR 在 GTSRB 数据集上保持轻量级的同时能够有效地遗忘目标数据且在剩余数据上保持了较高的准确率，验证了本文

所提出的 FedUR 方法的可扩展性。

数据集	方法	遗忘级别	准确率/%	后门攻击成功率/%	时间开销/s	存储开销/MB
MNIST	FUG[23]	客户端级	88.37	2.29	0.7	~ 10^0
		样本级	85.90	2.15	0.7	~ 10^0
		类别级	83.54	7.40	0.4	~ 10^0
	FedUR	客户端级	98.62	0.30	5.6	~ 10^0
		样本级	98.47	0.50	5.4	~ 10^0
		类别级	98.18	0.44	4.9	~ 10^0
CIFAR-10	FUG[23]	客户端级	75.26	88.54	2.0	~ 10^1
		样本级	77.05	88.06	1.0	~ 10^1
		类别级	61.21	85.38	0.9	~ 10^1
	FedUR	客户端级	79.35	1.11	15.2	~ 10^1
		样本级	78.98	0.90	14.6	~ 10^1
		类别级	78.48	1.00	13.1	~ 10^1

表 5 FedUR 与 FUG 的实验结果对比

6.2.4 遗忘强度评估

为了验证 FedUR 中的投影梯度下降过程能够保证随机遗忘过程中不同样本或不同类别的遗忘程度一致，我们在 MNIST 数据集的客户端级遗忘实验中补充了自适应遗忘阶段后不同类别的后门攻击成功率。我们设计的后门攻击策略为：向标签非 9 的样本注入后门特征，并将其标签篡改为 9。为验证自适应遗忘阶段的遗忘强度一致性，我们建了仅包含单类别样本的后门测试集，通过测试遗忘阶段完成后的模型在各测试集上的后门攻击成功率展开分析。实验结果如表 4 所示，模型在所有类别后门测试集上的攻击成功率均为 0，这一结果充分表明自适应遗忘阶段的随机遗忘过程具备均匀一致的遗忘强度，不存在遗忘强度不均的问题。

6.2.5 时间开销和空间开销评估

为了验证 FedUR 是轻量级的，我们测量了联邦遗忘学习方法的时间开销和存储开销。根据表 2 中的结果，我们将不同方法在不同遗忘级别下的时间开销和存储开销可视化成柱状图。如图 3 所示，本文方案 FedUR 具有最小的空间开销，而时间开销仅次于 FUG。FedEraser，FUKD 和 FedAU 由于需要存储额外的模型参数更新，与其他联邦遗忘学习方法相比产生更高的存储开销，而且时间开销也高于本文方案 FedUR。Retraining，FedEraser 和 Class-disc 相比其他联邦遗忘学习方法因为后训练或者微调过程具有更大的时间开销。

为了进一步比较本文方案 FedUR 与 FUG，我们将两个方法在两个数据集三个遗忘级别上的实验结果展示如表 5 所示。实验结果表明，在 MNIST 数据集上 FUG 的准确率较 FedUR 低约 10%，后门攻击成功率高约 2%；而在 CIFAR-10 数据集上，FUG 在类别级遗忘上的准确率较 FedUR 低约 17%，在三个遗忘级别上的后门攻击成功率均高达 85%以上，远高于 FedUR 的 1%左右。另外，我们评估了 FedUR 和 FUG 在联邦学习客户端数量变化情况下的准确率和后门攻击成功率，实验结果如图 2 所示。FUG 在联邦学习客户端数量变化的情况下准确率相较于 FedUR 始终低 10%左右，而后门攻击成功率始终在 80%以上，FedUR 却只有不到 1%。说明 FUG 通过修改损失函数并只训练两个轮次不能实现有效遗忘，且不能维持全局模型在剩余数据上的准确率，模型实用性大大降低。因此综合准确率、后门攻击成功率、时间开销和空间开销，只有 FedUR 实现了有效遗忘、模型性能保持和轻量级。

6.3消融实验

本节介绍对以下三个重要因素的消融研究：阈值半径r、软硬损失权重系数α以及自适应遗忘阶段和优化恢复阶段的影响。

6.3.1 阈值半径r的影响

为了评估自适应遗忘阶段中约束条件中的范数球半径r对 FedUR 性能的影响，我们在 MNIST 和 CIFAR-10 数据集上进行实验，对于 MNIST 数据集r从 100 变化到 300，对于 CIFAR-10 数据集r从 200 变化到 1000。我们在四个维度上说明了半径r对 FedUR 的影响：准确率、后门攻击成功率、时间开销和存储开销。图 4 展示了半径r对 FedUR 在 MNIST 和 CIFAR-10 数据集上的准确率和时间开销的影响。总体来说，随着r的增加，剩余数据上的准确率会降低，但是时间开销增加。具体地，对于 MNIST 数据集，当r从 100 变化到 300 时，精度降低 0.5%，并且时间开销增加 1.0s。对于 CIFAR-10 数据集，当r从 200 变化到 1000 时，精度降低 0.4%，并且时间开销增加了 0.7s。这个实验结果是合理的，半径r的设定本质上反映了遗忘过程对模型知识结构的保护强度，r决定了遗忘模型与剩余模型之间的允许偏差范围，当r增大时，遗忘模型在执行随机梯度上升以遗忘目标数据时，被允许偏离剩余模型更远，更新幅度更大，优化恢复阶段需补偿更大的参数偏离，时间开销增加。同时r的增大使得模型在遗忘过程中能更自由地调整参数，但破坏了模型对剩余数据的学习稳定性，对剩余数据的拟合能力下降，导致准确率下降。图 4 还展示了半径r对 FedUR 的后门攻击成功率和存储开销的影响。根据结果，我们观察到半径r对后门攻击成功率和存储开销没有明显影响。

表 6　自适应遗忘阶段和优化恢复阶段的影响

数据集	方法	准确率/%	后门攻击成功率/%
MNIST	FedUR（仅自适应遗忘阶段）	20.03	0.00
	FedUR（仅优化恢复阶段）	98.42	9.41
	完整的 FedUR（自适应遗忘阶段+优化恢复阶段）	98.62	0.30
CIFAR-10	FedUR（仅自适应遗忘阶段）	10.01	0.00
	FedUR（仅优化恢复阶段）	75.32	6.20
	完整的 FedUR（自适应遗忘阶段+优化恢复阶段）	79.35	1.11

6.3.2 软硬损失权重系数α的影响

为了评估软硬损失的权重系数α对 FedUR 性能的影响，我们在 MNIST 和 CIFAR-10 数据集上进行实验，观察α从 0.1 变化到 0.9 对模型准确率和后门攻击成功率的影响。图 5 的结果说明：随着α的增大，模型的准确率也呈上升趋势，而α过大或者过小都会导致后门攻击成功率变大。这是因为α本质涉及到模型对不同信息源的依赖程度和学习策略的平衡。当α增大时，模型更依赖真实标签，这有助于模型更快地拟合剩余数据的分布，从而提高准确率。但当α过大时，后门攻击成功率上升，这是因为过度依赖真实标签使得模型对目标数据的遗忘效果被削弱。模型在学习剩余数据时，可能会重新学习到部分被遗忘目标数据的特征，导致后门残留。反之，当α较小时，模型更依赖教师模型的软标签。软标签包含了教师模型的泛化知识，但教师模型可能引入噪声。如果教师模型在某些样本上的判断不准确，软标签携带的噪声会传递给学生模型，降低模型准确率。同时，较小的α可能导致目标数据的影响未彻底消除，后门攻击成功率反弹。因此，选择合适的α值（如 0.5），能够平衡真实标签和软标签的优势，使模型在泛化与拟合之间达到最佳状态，既保证对剩余数据的准确分类，又能有

效遗忘目标数据，降低后门攻击成功率。

6.3.3 自适应遗忘阶段和恢复阶段的影响

为了评估 FedUR 中每个阶段的有效性，我们进行了一项消融研究。在表 6 中，对于每个数据集，前两行分别展示了移除优化恢复阶段和自适应遗忘阶段的结果。最后一行展示了完整 FedUR 过程，作为我们消融研究的基线[44]。根据结果，我们得出以下两个结论：缺乏自适应遗忘阶段阻止了遗忘模型有效地消除遗忘数据的影响，这一结论在 MNIST 和 CIFAR-10 数据集的后门攻击实验中得到了验证，其后门攻击成功率分别达到 9.41%和 6.20%，明显高于基线。这强调了遗忘阶段在实现有效遗忘中的关键作用。优化恢复阶段的缺乏导致遗忘模型在剩余数据上的准确率显著下降，在 MNIST 和 CIFAR-10 数据集上分别下降到 20.03%和 10.01%。虽然后门攻击成功率接近于零，表明有效的遗忘效果，但在剩余数据上的准确率很低，模型变得无用。总体而言，FedUR 通过自适应遗忘阶段和优化恢复阶段的合作和互补实现遗忘。

7 结论

在本文中，我们提出了一个轻量级的联邦遗忘学习框架 FedUR，该框架适用于客户端级、样本级和类别级三种遗忘级别。FedUR 实现时间开销和存储开销均衡的同时，确保全局模型性能合理恢复并实现有效的遗忘效果。FedUR 的自适应遗忘阶段和优化恢复阶段分别通过随机梯度上升结合投影梯度下降最小化存储开销和知识蒸馏减少恢复时间。通过在两个真实数据集上进行广泛实验，我们将 FedUR 与五种前沿的联邦遗忘学习方法在三种遗忘级别下进行了系统的对比。实验结果表明，FedUR 在保持模型准确率、实现有效遗忘效果的同时显著提升了遗忘效率。

由于优化恢复阶段需要使用到外包标记的数据集，本文提出的轻量级联邦遗忘学习框架 FedUR 还有不足之处。所以在未来的工作中，考虑引入无数据蒸馏的思想，在不需要外包数据集的情况下实现更加有效和高效的联邦遗忘学习。

致谢感谢《计算机学报》编辑和审稿专家，他们付出了辛勤工作。

参考文献

[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, USA, 2017: 1273-1282

[2] Sun P, Che H, Wang Z, et al. Pain-FL: Personalized privacy-preserving incentive for federated learning. IEEE Journal on Selected Areas in Communications, 2021, 39(12): 3805-3820

[3] Zhao B, Sun P, Wang T, et al. FedInv: Byzantine-robust federated learning by inversing local model updates//Proceedings of the 36th AAAI Conference on Artificial Intelligence. Palo Alto, USA, 2022: 9171-9179

[4] He Z, Wang Z, Dong X, et al. Towards fair federated learning via unbiased feature aggregation. IEEE Transactions on Dependable and Secure Computing, 2025, 22(4): 3795-3807

[5] Sun P, Wu L, Wang Z, et al. A Profit-Maximizing Data Marketplace with Differentially Private Federated Learning under Price Competition//Proceedings of the ACM on Management of Data. Santiago, Chile, 2024, 2(4): 1-27

[6] Sun P, Liao G, Huang J, et al. Socially Optimal Mechanism Design for Relay-assisted Asynchronous Federated Learning. IEEE Journal on Selected Areas in Communications, 2025

[7] Voigt P, Von Dem Bussche A. The EU General Data Protection Regulatio. Cham, Switzerland: Springer International Publishing, 2017

[8] Pardau S. The California Consumer Privacy Act: Towards a European-style privacy regime in the United States? Journal of Technology Law & Policy, 2022, 23(1): 2

[9] Wang F, Li B, Li B. Federated unlearning and its privacy threats. IEEE Network, 2024, 38(2): 294-300

[10] Wang Peng-Fei, Wei Zong-Zheng, Zhou Dong-Sheng, et al. A survey on federated unlearning. Chinese Journal of Computers, 2024, 47(2): 396-422 (in Chinese)
(王鹏飞, 魏宗正, 周东生, 等. 联邦忘却学习研究综述. 计算机学报, 2024, 47(2): 396-422)

[11] Romandini N, Mora A, Mazzocca C, et al. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(7): 11697-11717

[12] Zaman M M U, Sun X, Yao J. Sky of Unlearning (SoUL): Rewiring federated machine unlearning via selective pruning. arXiv preprint arXiv:2504.01705, 2025

[13] Wu C, Zhu S, Mitra P, et al. Unlearning backdoor attacks in federated

learning//Proceedings of the 2024 IEEE Conference on Communications and Network Security. Taipei, China, 2024: 1-9

[14] Xu J, Zhang Z, Hu R. Identify backdoored model in federated learning via individual unlearning//Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision. Tucson, USA, 2025: 7960-7969

[15] Nguyen T D, Nguyen T, Nguyen P L, et al. Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions. Engineering Applications of Artificial Intelligence, 2024, 127: 107166

[16] Sheng X, Bao W, Ge L. Robust federated unlearning//Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. Boise, USA, 2024: 2034-2044

[17] Wu C. Backdoor attacks and defenses in federated machine learning[PhD Thesis]. Centre County, PA, USA: The Pennsylvania State University, 2024

[18] Liu G, Ma X, Yang Y, et al. FedEraser: Enabling efficient client-level data removal from federated learning models//Proceedings of the 2021 IEEE/ACM 29th International Symposium on Quality of Service. Tokyo, Japan, 2021: 1-10

[19] Wu C, Zhu S, Mitra P. Federated unlearning with knowledge distillation. arXiv preprint arXiv:2201.09441, 2022

[20] Zhang L, Zhu T, Zhang H, et al. FedRecovery: Differentially private machine unlearning for federated learning frameworks. IEEE Transactions on Information Forensics and Security, 2023, 18: 4732-4746

[21] Gu H, Zhu G, Zhang J, et al. Unlearning during learning: An efficient federated machine unlearning method//Proceedings of the 33rd International Joint Conference on Artificial Intelligence. Jeju, Republic of Korea, 2024: 4035-4043

[22] Wang J, Guo S, Xie X, et al. Federated unlearning via class-discriminative pruning//Proceedings of the 2022 ACM Web Conference. Virtual, 2022: 622-632

[23] Wu L, Guo S, Wang J, et al. Federated unlearning: Guarantee the right of clients to forget. IEEE Network, 2022, 36(5): 129-135

[24] Liu Y, Xu L, Yuan X, et al. The right to be forgotten in federated learning: An efficient realization with rapid retraining//Proceedings of the 2022 IEEE Conference on Computer Communications. London, UK, 2022: 1749-1758

[25] Cao Y, Yang J. Towards making systems forget with machine unlearning//Proceedings of the 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015: 463-480

[26] Ginart A A, Guan M Y, Valiant G, et al. Making AI forget you: Data deletion in machine learning//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook, USA, 2019: 3518-3531

[27] Brophy J, Lowd D. Machine unlearning for random forests//Proceedings of the 38th International Conference on Machine Learning. Virtual, 2021: 1092-1104

[28] Jose S T, Simeone O. A unified PAC-Bayesian framework for machine unlearning via information risk minimization//Proceedings of the 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing. Gold Coast, Australia, 2021: 1-6

[29] Bourtoule L, Chandrasekaran V, Choquette-Choo C A, et al. Machine unlearning//Proceedings of the 2021 IEEE Symposium on Security and Privacy. San Francisco, USA, 2021: 141-159

[30] Yan H, Li X, Guo Z, et al. ARCANE: An efficient architecture for exact machine unlearning//Proceedings of the 31st International Joint Conference on Artificial Intelligence. Vienna, Austria, 2022: 4006-4013

[31] Wu G, Hashemi M, Srinivasa C. PUMA: Performance unchanged model augmentation for training data removal//Proceedings of the 36th AAAI Conference on Artificial Intelligence. Virtual, 2022: 8675-8682

[32] Wang W, Zhang C, Tian Z, et al. FedU: Federated unlearning via user-side influence approximation forgetting. IEEE Transactions on Dependable and Secure Computing, 2025, 22(3): 2550-2562

[33] Pan Z, Wang Z, Li C, et al. Federated unlearning with gradient descent and conflict mitigation//Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia, USA, 2025: 19804-19812

[34] Halimi A, Kadhe S, Rawat A, et al. Federated unlearning: How to efficiently erase a client in FL? arXiv preprint arXiv:2207.05521, 2023

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

[36] Huang Zhen-Hua, Yang Shun-Zhi, Lin Wei, et al. knowledge distillation: A survey. Chinese Journal of Computers, 2022, 45(3): 624-653 (in Chinese)
(黄震华, 杨顺志, 林威, 等. 知识蒸馏研究综述. 计算机学报, 2022, 45(3): 624-653)

[37] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324

[38] Krizhevsky A. Learning multiple layers of features from tiny images[Master thesis]. Toronto, Canada: University of Tront, 2009

[39] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2015

[40] Gu T, Dolan-Gavitt B, Garg S. BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733, 2019

[41] Alam M, Lamri H, Maniatakos M. Get rid of your trail: Remotely erasing backdoors in federated learning. IEEE Transactions on Artificial Intelligence, 2024, 5(12): 6683-6698

[42] Li Y, Zhang J, Liu Y, et al. Class-wise federated unlearning: Harnessing active forgetting with teacher – student memory generation. Knowledge-Based Systems, 2025, 316: 113353

[43] Liu T, Zhang Y, Feng Z, et al. Beyond traditional threats: A persistent backdoor attack on federated learning//Proceedings of the 38th AAAI Conference on Artificial Intelligence. Vancouver, Canada, 2024: 21359-21367

[44] Zhao Y, Wang P, Qi H, et al. Federated unlearning with momentum

degradation. IEEE Internet of Things Journal, 2024, 11(5): 8860-8870

TANG Xiang-Yun, Ph.D., associate professor. Her main research interests include artificial intelligence security, federated learning, data security, and privacy protection.

WU Hang,M.S. candidate. His main research interest is federated learning.

WANG Ya-Jie, Ph.D., assistant professor. His main research interests include data security, and artificial intelligence security.

SHEN Meng, Ph.D., Professor. His main research interests include data security, artificial intelligence security, and blockchain security.

WENG Yu, Ph.D., professor. His main research interests include artificial intelligence, and deeplearning.

ZHU Lie-Huang, Ph.D., professor. His main research interests include cryptozoological algorithms, security protocols, blockchain technology, and cloud computing security.