Computer Science and Application 计算机科学与应用, 2026, 16(5), 312-326
Published Online May 2026 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2026.165186

Hans汉斯

基于分组蒸馏的可验证隐私保护公平个性化联邦学习方法

王江源

青海大学计算机技术与应用学院，青海 西宁

收稿日期：2026年4月23日；录用日期：2026年5月21日；发布日期：2026年5月28日

摘　要

个性化联邦学习(Personalized Federated Learning, PFL)允许每个用户在共享全局模型的基础上训练个性化模型，显著提升了模型在异构数据环境下的性能。然而也面临数据隐私泄露、客户端之间模型性能失衡以及聚合结果可信性等方面的挑战。为了解决上述问题，文章设计了基于用户数据相似性的层次聚类分组机制与跨组知识蒸馏机制，在不泄露原始梯度的前提下实现相似数据用户的精准聚类，将数据特征相近的用户归为同一组，共同开展联邦训练，实现数据层面的个性化；并缩小用户之间的模型性能差距，保障了系统的公平性。在隐私保护方面，设计了基于掩码的安全聚合机制，确保各用户上传的梯度参数在传输与聚合过程中始终处于隐私保护状态。同时，设计了验证机制，使诚实用户在接收聚合结果后能够独立校验其正确性。实验结果表明，本方案满足了高级别隐私保护要求，确保了用户之间的公平性，并且模型准确率优于现有方案。

关键词

可验证，个性化联邦学习，隐私保护

A Verifiable Privacy-Preserving Fair Personalized Federated Learning Method Based on Group Distillation

Jiangyuan Wang

School of Computer Technology and Application, Qinghai University, Xining Qinghai

Received: April 23, 2026; accepted: May 21, 2026; published: May 28, 2026

文章引用: 王江源. 基于分组蒸馏的可验证隐私保护公平个性化联邦学习方法[J]. 计算机科学与应用, 2026, 16(5): 312-326. DOI: 10.12677/csa.2026.165186

Abstract

Personalized Federated Learning (PFL) allows each user to train a personalized model based on a shared global model, significantly improving model performance in heterogeneous data environments. However, it also faces challenges such as data privacy leakage, imbalance in model performance among clients, and the trustworthiness of aggregation results. To address these issues, a hierarchical clustering grouping mechanism based on user data similarity and a cross-group knowledge distillation mechanism were designed. This enables precise clustering of users with similar data without revealing original gradients, allowing users with similar data to train together in the same group for federated training, achieving data-level personalization; it also reduces the performance gap between users, ensuring system fairness. In terms of privacy protection, a mask-based secure aggregation mechanism was designed to ensure that the gradient parameters uploaded by users remain protected throughout transmission and aggregation. At the same time, a verification mechanism was designed so that honest users can independently verify the correctness of the aggregated results once received. Experimental results show that this solution meets high-level privacy protection requirements, ensures fairness among users, and achieves model accuracy superior to existing solutions.

Keywords

Verifiable, Personalized Federated Learning, Privacy Protection

1. 引言

随着人工智能技术的快速发展，联邦学习(Federated Learning, FL) [1]作为一种分布式机器学习框架，日益受到关注。FL 通过让各个用户在本地训练模型并将模型更新发送至中央服务器进行聚合，从而避免了传统机器学习中的数据收集困难问题，在金融、医疗等领域广泛应用[2]-[6]。然而，如果每个用户的数据分布是高度非独立同分布(Non-IID)，则模型性能因客户端漂移而效率低下。尽管现有的个性化联邦学习(Personalized Federated Learning, PFL)研究在提升模型性能方面取得了一定进展[7]-[9]，但仍面临个性化模型性能差异显著、隐私泄露风险大、聚合结果验证困难等关键挑战。

个性化模型性能差异显著主要体现在不同用户间的收敛速度不一致，造成模型训练成本提升与模型准确性不足。现有的个性化方法，如 Li [10]等人使用局部批量归一化来缓解模型平均之前的特征偏移，虽然通过约束参数更新缓解了客户端漂移问题[11]，但难以平衡全局共享与局部特性。Chen [12]等人使用迁移学习策略，通过联邦学习进行数据聚合，然后通过迁移学习构建相对个性化的模型。Yang [13]等人提出 FedSteg 框架，通过联邦迁移学习训练安全、个性化的分布式模型。将预训练模型知识迁移到新的任务或用户场景中，可以减少训练开销并提升模型收敛速度。然而，由于用户间数据分布差异显著，当源任务与目标任务的特征分布不一致时，迁移效果会显著下降，甚至出现负迁移现象，即在跨用户验证时性能急剧退化。Fallah [14]等人研究了元学习方法，得到一个初始共享模型，当前或新用户可以通过对自己的数据执行一个或几个梯度下降步骤来轻松适应他们的本地数据集。这种方法保留了联合学习架构的所有优点，并且通过结构为每个用户提供了更加个性化的模型。Jiang [15]等人提出与模型无关的元学习

设置，即对任务的异构分布进行快速、基于梯度、少样本的适应进行优化，与 FL 的个性化目标有许多相似之处。通过在多任务环境中学习可快速迁移的初始化参数，从而使模型能够在面对新分布时迅速适应。然而，该类方法对数据质量和任务分布的一致性高度敏感，当用户本地数据量较少或包含较多噪声时，往往会导致模型在微调过程中出现过拟合现象，削弱其泛化能力。Zhang [16]等人提出了 KT-pFL 框架，由服务器为每个用户维护个性化软预测，用于指导其他用户的本地训练；再通过知识系数矩阵，对所有局部软预测做线性组合，完成每个用户个性化软预测的更新。FedMD [17]使用迁移学习和知识蒸馏设计了一个通用框架，当每个用户不仅拥有他们的私有数据，而且拥有独特设计的模型时，该框架可以实现联邦学习。这两个方案实现了隐私保护的个性化联邦学习，但其高度依赖服务器的诚信，如果服务器出现问题，其安全得不到保证。

众所周知，隐私泄露风险主要体现在梯度信息泄露造成的梯度反演攻击[18] [19]，这是当前联邦学习面临的最严峻安全挑战之一。在个性化联邦学习场景下，由于需要更频繁地交换细粒度的模型参数，隐私泄露风险被进一步放大。为确保联邦学习过程中梯度信息的安全性，目前学者主要从同态加密、差分隐私、掩码和多方安全计算[20]-[23]等方面展开安全聚合技术研究。尽管这些方法在隐私保护方面取得了显著进展，但仍面临计算效率、通信成本和实用性等方面的挑战。例如 Zhang [24]等人与 Park [25]等人使用同态加密支持在密文状态下直接计算梯度，但计算开销较大。Liao [26]等人与 Wei [27]等人使用差分隐私通过添加噪声实现隐私保护，但需要在隐私性和模型性能之间权衡。掩码方案通过巧妙的随机扰动机制，在计算效率与模型性能之间实现了较优的平衡，例如 Bonawitz [28]等人提出了 SecAgg 方案，用户在上传梯度前会施加特殊构造的随机掩码，这些掩码在服务器端聚合时能够相互抵消。服务器仅能获取聚合后的整体梯度信息，无法反推出任何单个用户的原始数据特征。同时，相较于差分隐私方案，掩码技术避免了噪声累积效应，不会随着训练轮次增加而导致模型性能持续衰减。在计算效率方面，掩码操作仅涉及简单的线性运算，使得整体计算复杂度保持在较低水平。但是在面对掉线时，整体运行时间随着掉线人数的增加而迅速增加。Liu [29]等人使用 HPRG 和秘密共享来实现安全聚合，提高了效率和抗掉线能力，但是解密过程涉及复杂的密码学运算，导致解密时产生巨额的计算开销。PHP-FL [30]采用知识矩阵进行用户分组，但其环状聚合算法存在明显的安全缺陷，一组只有一个掩码，攻击者仅需截获组内单个用户接收和发送的数据，即可成功破解该用户的原始梯度信息。

然而，保护隐私又增大了聚合结果验证难度。以差分隐私为例，差分隐私通过添加噪声保护数据隐私，但这些噪声会掩盖聚合结果的微小偏差，用户无法检测服务器对聚合结果的篡改。验证过程通常需要传输大量辅助信息，这些额外数据的规模会随着系统规模的扩大而增长。同时，验证操作通常具有严格的时序依赖关系，用户必须等待前一轮操作完成才能进行验证，导致通信延迟大幅增加。例如，VeriFL [31]虽然实现了可验证的隐私保护联邦学习，但其通信复杂度与用户数量呈线性增长关系，当参与用户规模扩大时会产生显著的通信开销。VCD-FL [32]同样试图在掩码隐私安全的基础上实现可验证性，设计了基于奇异矩阵的单向不可逆承诺，结合用户端本地生成的随机向量，试图阻止恶意服务器伪造验证信息。在 PFL 场景下，验证机制的实现更加困难。相较于传统联邦学习，个性化模型参数的异构性使得验证标准难以统一，不同用户可能采用完全不同的模型架构或超参数设置，无法建立统一的验证标准。

针对上述三个问题，本研究提出了一种基于分组蒸馏和哈希验证的隐私保护个性化联邦学习方法，通过构建软预测矩阵实现隐私保护下的用户分类，避免原始梯度传输导致的信息泄露。基于知识蒸馏技术，实现多模型知识安全共享并获得适配本地数据特性的个性化模型。为保障训练过程安全，设计了一种安全聚合与验证算法，在防止梯度泄露的同时支持聚合结果验证。本文的主要贡献总结如下：

1) 提出了一种基于分组蒸馏和哈希验证的隐私保护个性化联邦学习方法，通过隐私保护机制与知识蒸馏技术的结合，在严格保护用户数据隐私的前提下，实现了高效、可验证的个性化模型协同训练。

2) 设计了软预测的用户分组与跨组知识蒸馏方法。层次化聚类算法动态分析了用户模型的预测分布相似性，实现异构用户的智能分组，多教师蒸馏机制通过软标签传播与特征空间对齐，平衡了非独立同分布数据场景下各个用户模型的准确率，同时提升用户本地模型性能。

3) 设计了基于线性同态哈希的梯度盲化与验证算法，利用同态伪随机发生器对本地梯度进行动态随机掩码处理，确保服务器无法反推原始数据，同时支持密文状态下的安全聚合运算。构建了线性同态哈希验证机制，用户仅需存储传输固定大小的验证元数据即可高效检测服务器返回结果的准确性。

4) 对方案的有效性和隐私保护能力进行了分析，实验结果表明本方案在模型性能和隐私保护方面优于现有方案。

2. 预备知识

2.1. 同态伪随机发生器

同态伪随机发生器(Homomorphic Pseudorandom Generator, HPRG) [29]建立在伪随机发生器(PRG)的安全性与同态加密的计算特性之上。其核心思想是允许用户在仅持有种子密文的前提下，通过执行预定义的同态计算电路，自主扩展出长序列的伪随机比特密文流，本文使用的 HPRG 满足：$\prod_{i\in U}\mathrm{HPRG}(b_i)=\mathrm{HPRG}\left(\sum_{i\in U}b_i\right)$。

2.2. 线性同态哈希

线性同态哈希(Linear Homomorphic Hash, LHH)是一种特殊的哈希函数族，它除了具备常规哈希函数的单向性和抗碰撞性外，还具有线性同态性。对于若干输入消息的线性组合，其哈希值可以通过相应哈希值的线性组合计算得到，而无需访问原始消息。LHH 算法族由三个多项式时间算法组成：

1) 参数生成算法 $\mathrm{LHH.HGen}\left(1^{\lambda},1^{d}\right)\to\left(\mathbb{G},p,m,m_1,\cdots,m_d\right)$。该算法接受安全参数 λ 和本地梯度维度 d 作为输入，输出公共参数集合 $\left\{\mathbb{G},p,m,m_1,\cdots,m_d\right\}$。$p$ 是乘法循环群的阶；$m\mathbb{G}$ 是 $\mathbb{G}$ 的生成元；$\left\{m_1,\cdots,m_d\right\}$ 是群 $\mathbb{G}$ 内互不相同的 d 个元素。

2) 哈希计算算法 $\mathrm{LHH.Hash}\left(g_i\right)\to h_i$。该算法输入用户 i 的本地梯度向量 $g_i\in\mathbb{Z}_p^d$，其中 $g_i[l]$ 表示其第 l 维分量，输出对应的线性同态哈希值。计算公式为：$h_i=\prod_{l=1}^{d}m_l^{g_i[l]}\in\mathbb{G}$。

线性同态哈希的输出维度与梯度维度 d 无关，即通信开销始终固定为单个群元素的大小。

3) 聚合算法 $\mathrm{LHH.Eval}\left(h_1,\cdots,h_N;a_1,\cdots,a_N\right)\to h^*$。该算法输入来自 N 个用户的哈希结果 $\left\{h_1,\cdots,h_N\right\}$ 以及对应的权重系数 $\left\{a_1,\cdots,a_N\right\}$，输出聚合哈希结果 h^*，计算方式为 $h^*=\prod_{i=1}^{N}h_i^{a_i}$。在本方案中，为简化起见，各用户权重通常取 $a_i=1$。

3. 方案设计

3.1. 系统模型

系统由三个实体组成：聚合服务器、用户以及可信机构，如图 1 所示。

可信机构：主要负责系统初始化和用户注册。

服务器：主要负责对用户上传的梯度参数和验证信息进行聚合，并将聚合后的结果分发给各个用户。

用户：每个用户基于本地数据集进行训练以获得本地梯度更新，随后对本地梯度进行掩码处理并上传至服务器。同时，用户还会上传相应的验证信息，以便在服务器返回聚合结果时验证其正确性。

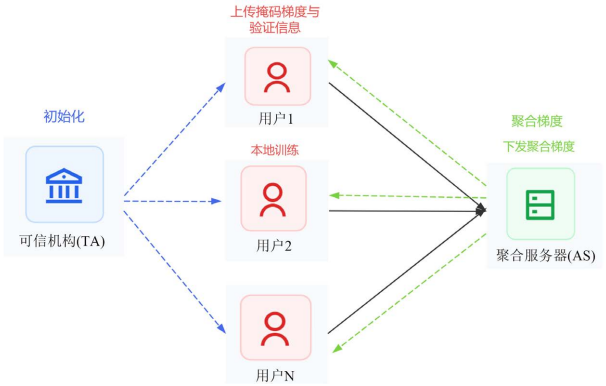


Figure 1. System model
图 1. 系统模型

3.2. 威胁模型

本方案的威胁模型中，TA 被视为完全可信。用户为半诚实实体，他们会严格遵循协议规定的计算流程，不会主动破坏训练过程；此外，所有用户可能通过协议交互中获取的信息来推断其他用户的私有数据。同时，部分用户可能与服务器合谋攻击来获取其他用户的私有数据。服务器为半诚实实体，虽然会正确执行聚合计算等基本功能，但会尝试从接收到的梯度信息中反推出用户敏感数据。

3.3. 方案流程

方案的流程如图 2 所示，共有三个过程：用户分组、组内联邦聚合、跨组知识蒸馏。

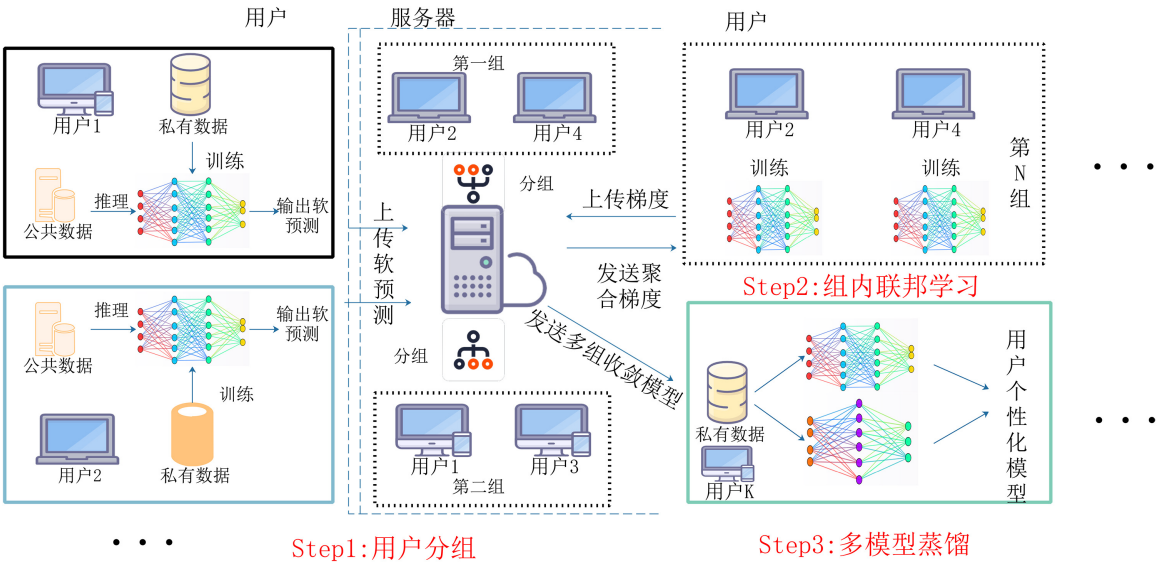


Figure 2. Scheme process
图 2. 方案流程

3.3.1. 本地模型训练与分组

在联邦学习系统中，由于用户数据的非独立同分布特性，直接进行全局模型聚合往往会导致模型性能下降。为了解决这一问题，本文首先提出根据数据分布相似性的层次聚类用户分组方法。

每个用户维护两个相互关联但功能不同的模型：其一为私有模型 M_u^{priv}，仅用户自己所有，用于捕捉

用户特有的数据分布与偏好特征，并结合蒸馏损失以吸收全局知识；其二为共享模型 M^{p}，用于服务器通过跨用户聚合。首先服务器初始化一个全局共享模型参数 w_0，该模型仅作为聚合与全局知识融合的载体。随后，服务器将该共享模型参数广播给所有用户终端，以作为各用户模型交互与全局更新的初始参数。用户使用共享模型 $M^{p}=w_0$ 在各自的本地数据集 $\mathcal{D}_u$ 上执行 K 轮随机梯度下降训练，以最小化本地损失：

$$\min_{\theta} F_u(\theta)=\min_{\theta}\frac{1}{|\mathcal{D}_u|}\sum_{(x,y)\in\mathcal{D}_u}\ell\left(f(x;\theta),y\right) \tag{1}$$

具体过程是以用户自己的共享模型参数 w_0 为初始点 $\theta_u^{(0)}=w_0$，第 k 轮本地更新($k=0,1,\cdots,K-1$)过程如下：

1) 采样小批量：$\mathcal{B}_u^{(k)}\subset\mathcal{D}_u$，$\left|\mathcal{B}_u^{(k)}\right|=B$；

2) 计算随机梯度：$g_u^{(k)}=\frac{1}{B}\sum_{(x,y)\in\mathcal{B}_u^{(k)}}\nabla_{\theta}\ell\left(f\left(x;\theta_u^{(k)}\right),y\right)$；

3) 更新模型参数：$\theta_u^{(k+1)}=\theta_u^{(k)}-\eta\cdot g_u^{(k)}$。

完成 K 轮更新后，得到本地模型参数 $\theta_u^{(K)}$，用于后续的软预测生成。为了获得更加精准的概率输出，该模型采用温度调节的 Softmax 输出层，其预测概率分布的计算公式为：

$$p_u^{(t)}(x)=\sigma_{\text{temp}}\left(z_u(x);t\right)=\left[\frac{\exp\left(z_u^{(1)}(x)/t\right)}{\sum_{j=1}^{C}\exp\left(z_u^{(j)}(x)/t\right)},\cdots,\frac{\exp\left(z_u^{(C)}(x)/t\right)}{\sum_{j=1}^{C}\exp\left(z_u^{(j)}(x)/t\right)}\right]^{T} \tag{2}$$

其中，$z_u(x)\in\mathbb{R}^{C}$ 表示模型对输入 x 的 logits 输出向量，$t>0$ 为温度参数，C 为分类类别数。温度参数 t 的引入可以调节输出概率分布的平滑程度，当 $t>1$ 时产生更加均匀的概率分布，有助于后续的分布相似性比较。

为了评估用户数据分布特征，系统维护一个具有标准分布的公共数据集 $\mathcal{D}_g=\{x_1,x_2,\cdots,x_N\}$。每个用户 u 使用其本地模型为该数据集生成软预测矩阵：

$$K_u=\left[k_u(x_1),k_u(x_2),\cdots,k_u(x_N)\right]^{\mathrm{T}}=\begin{bmatrix}k_u^{(1)}(x_1) & k_u^{(2)}(x_1) & \cdots & k_u^{(C)}(x_1)\\ k_u^{(1)}(x_2) & k_u^{(2)}(x_2) & \cdots & k_u^{(C)}(x_2)\\ \vdots & \vdots & \ddots & \vdots\\ k_u^{(1)}(x_N) & k_u^{(2)}(x_N) & \cdots & k_u^{(C)}(x_N)\end{bmatrix} \tag{3}$$

该矩阵 $K_u\in\mathbb{R}^{N\times C}$ 完整地捕获了用户 u 的本地模型在整个公共数据集上的预测行为特征。服务器收集所有用户的软预测矩阵 $\{K_1,K_2,\cdots,K_n\}$ 后，执行自底向上的层次聚类算法。层次聚类是一种自底向上的聚类方法，通过逐步合并或分割数据点来构建层次化的树状结构为树状图，从而反映数据点之间的相似性或差异性。将具有相似数据分布的用户分为同一组，从而更好地进行后续的模型聚合。用户间的分布相似性采用 Jensen-Shannon 散度进行度量：

$$D_{\text{JS}}(K_u\,\|\,K_v)=\frac{1}{2}D_{\text{KL}}(K_u\,\|\,M)+\frac{1}{2}D_{\text{KL}}(K_v\,\|\,M) \tag{4}$$

其中，$M=\frac{P_u+P_v}{2}$ 为两个概率矩阵的均值矩阵，D_{KL} 表示 Kullback-Leibler 散度，其计算公式为：

$$D_{\text{KL}}(A\,\|\,B)=\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}A_{n,c}\log\frac{A_{n,c}}{B_{n,c}} \tag{5}$$

在聚类过程中，簇间距离的计算采用平均链接法(Average Linkage)：

$$S\left(C_p,C_q\right)=\frac{1}{\left|C_p\right|\cdot\left|C_q\right|}\sum_{u\in C_p}\sum_{v\in C_q}D_{\mathrm{JS}}\left(K_u\,\|\,K_v\right) \tag{6}$$

基于上述相似性度量，层次聚类的算法实现流程如下：

算法 **1** 基于软预测矩阵的层次聚类算法

输入用户软预测矩阵 $\{\boldsymbol{K}_1,\cdots,\boldsymbol{K}_n\}$，目标组数 N；

输出用户分组 $\{\boldsymbol{G}_1,\cdots,\boldsymbol{G}_N\}$

1. 初始化相似度矩阵 $\boldsymbol{S}\in\mathbb{R}^{n\times n}$
2. for $i=1$ to n
3. for $j=1$ to n
4. 根据公式(4)更新 S
5. end for
6. end for
7. 初始化簇集合 $\mathcal{C}\leftarrow\{\{1\},\cdots,\{n\}\}$
8. while $|\mathcal{C}|>N$ do
9. 找到最小距离的簇对 $\left(C_p,C_q\right)$
10. 合并簇 $\mathcal{C}\leftarrow\mathcal{C}\setminus\{C_p,C_q\}\cup\{C_p\cup C_q\}$
11. 更新距离矩阵 $\boldsymbol{S}$
12. end while

3.3.2. 安全聚合系统设计与协议流程

组内联邦学习阶段主要负责同组用户间的局部模型聚合与安全验证。为了防止服务器或其他参与方在模型上传与聚合过程中获取用户的私有信息，本文设计了一种基于线性同态哈希的梯度盲化与可验证聚合机制。每个用户在上传其本地模型更新之前，首先对模型参数施加 HPRG 处理，以生成一个带有掩码的模型更新，从而在聚合阶段中有效隐藏真实梯度信息，使得服务器无法直接解析或推断单个用户的本地更新内容。随后，用户利用线性同态哈希函数对更新向量进行哈希运算，生成可验证的哈希标签，并连同掩码后的模型更新与相关辅助信息一并上传至服务器。服务器在接收所有用户的上传数据后，执行聚合操作，从哈希空间与参数空间同时进行累加，并对聚合结果进行承诺。在聚合完成后，服务器向用户广播承诺值并请求对应的掩码秘密分享值，最后将掩码、参数与哈希聚合结果发送给用户。用户可根据收到的承诺、聚合哈希值和恢复后的聚合结果，验证服务器是否在聚合过程中保持了数据完整性与计算一致性。具体流程如下：

1) 初始化阶段

在系统开始前，存在一个可信机构 *TA*，负责生成系统所需的全局公共参数、用户的随机种子以及线性同态哈希的参数。*TA* 执行下列操作：

a) 生成全局参数：选取安全素数模 p，生成元 g，并发布 (p,g)。

b) 生成线性同态哈希的参数：为梯度的每一维 $k=1,\cdots,d$ 生成基底值 $h_k=g^{\alpha_k}$，α_k 为 *TA* 随机选择的私有值，仅用于哈希参数生成，不泄露给任何其他实体。根据生成参数形成参数集合 $\mathrm{param}_{LHH}=\left(p,g,h_1,h_2,\cdots,h_d\right)$，并发布给所有参与方。

c) 向每个用户 u 分发私有掩码种子 b_u (基于用户标识安全派生)，并告知 Shamir 分享的阈值参数 t 与份额数 r。

TA 必须保证 h_k 的选择方式使得对于任意向量 $v \in \mathbb{Z}^d$，线性哈希满足线性同态性，即向量相加对应哈希值相乘。

$$\mathrm{LHH}(v) \triangleq \prod_{k=1}^{d} h_k^{v_k} = \prod_{k=1}^{d} g^{\alpha_k v_k} = g^{\sum_k \alpha_k v_k} \tag{7}$$

TA 在分发参数后可以退出，不参与后续计算，以避免单点信任风险。

服务器执行下列操作：

向每个用户单独发送其所属分组的标识信息及组内其他成员的列表，确保每个用户能够明确自身在分组结构中的位置，并准确获知其组内协作对象。

2) 本地训练与加密阶段

在某一训练轮次中，服务器首先从组内全体参与用户中随机选择 r 个用户，形成集合 $U_0 = \{1, 2, \cdots, r\}$，并通知这些用户参与当前轮次的模型更新。对于每个被选中的用户 $u \in U_0$，其本地计算过程如下所述。

用户 u 首先在其私有数据集 D_i 上使用共享模型 w_0 执行若干步本地训练，得到模型梯度向量

$$w_i = \nabla \mathcal{L}(D_i; \theta_i) \in \mathbb{Z}^d \tag{8}$$

其中，$\mathcal{L}$ 表示局部损失函数，θ_i 为当前本地模型参数。为了在后续全局聚合中验证梯度的正确性与一致性，用户计算对应的线性同态哈希值：

$$Z_i = \mathrm{LHH}(w_i) = \prod_{k=1}^{d} h_k^{w_{i,k}} = g^{\sum_{k=1}^{d} \alpha_k w_{i,k}} \tag{9}$$

由于哈希函数具备线性同态性，对于任意两个向量 w_i 与 w_j，有 $\mathrm{LHH}(w_i + w_j) = \mathrm{LHH}(w_i) \cdot \mathrm{LHH}(w_j)$，从而保证了梯度聚合后哈希值的可验证性。接下来，用户 u 需对其梯度进行随机掩码化处理以保护隐私。用户利用私有随机种子 b_u 通过同态伪随机生成器生成掩码向量 $m_i = \mathrm{HPRG}(b_i)$，并将其与梯度向量的群编码形式相结合：

$$y_i = g^{w_i} \times m_i = \left(g^{w_{i,1}} m_{i,1}, g^{w_{i,2}} m_{i,2}, \cdots, g^{w_{i,d}} m_{i,d} \right) \tag{10}$$

其中，g^{w_i} 表示将每个维度的梯度 $w_{i,k}$ 整数化后映射到生成元 g 所在群中的指数表示。由于 HPRG 具有可加性与同态性质，后续在聚合阶段可以直接对掩码后的梯度执行群内乘法运算而无需解掩码。为了防止单个用户的掩码种子 b_i 被泄露导致隐私破坏，用户采用 Shamir 秘密共享协议将 b_i 拆分为 r 份子份 $\{b_i^1, b_i^2, \cdots, b_i^r\} \leftarrow \mathrm{ss.share}(b_i, t, r)$，$t$ 表示恢复阈值，用户随机选取一个 $(t-1)$ 次多项式：

$$\begin{aligned} f_i(x) &= b_i + a_{i,1} x + a_{i,2} x^2 + \cdots + a_{i,t-1} x^{t-1} \pmod q \\ b_i^j &= f_i(x_j), \quad j \in \{1, 2, \cdots, r\} \end{aligned} \tag{11}$$

随后，用户 i 将 b_i^j 安全发送给相应的用户 j (当 $j \neq i$ 时)，并自行保留 b_i^i。在后续聚合阶段，任意不少于 t 个的用户即可通过 Lagrange 插值恢复原始的种子 b_i：

$$b_i = \sum_{j \in S} b_i^j \cdot \prod_{\substack{\ell \in S \\ \ell \neq j}} \frac{x_\ell}{x_\ell - x_j} \pmod q \tag{12}$$

其中，$S \subseteq \{1, 2, \cdots, r\}$ 且 $|S| = t$。该设计确保了在不满足阈值的情况下，任何用户或服务器均无法推断出 b_i 的真实值，从而实现梯度掩码的安全性与系统整体的隐私保护。

3) 服务器聚合阶段

服务器收集来自部分用户的加密上传值，记为 $U_1 \subseteq U_0$ (实际成功上传的用户集合)。服务器计算：

$$\begin{aligned} y_s &= \prod_{i\in U_1} y_i = \prod_{i\in U_1}\left(g^{w_i}\cdot \mathrm{HPRG}\left(b_i\right)\right) \\ &= \left(\prod_{i\in U_1} g^{w_i}\right)\cdot\left(\prod_{i\in U_1}\mathrm{HPRG}\left(b_i\right)\right) \\ &= g^{\sum_{i\in U_1} w_i}\cdot \mathrm{HPRG}\left(\sum_{i\in U_1} b_i\right) \end{aligned} \tag{13}$$

上式中第二步利用了 HPRG 的同态性 $\prod_i \mathrm{HPRG}\left(b_i\right)=\mathrm{HPRG}\left(\sum_i b_i\right)$。为保证聚合结果在广播时不可篡改，服务器选择随机数 m 并对 y_s 进行承诺 $\mathrm{Com}=\mathrm{Hash}\left(y_s\|m\right)$。随后将 Com 与 U_1 广播给所有用户 $j\in U_1$，以便用户启动秘密份额汇总与验证流程。

每个用户 $j\in U_1$ 在本地将其收到的来自集合 U_1 中各用户的份额求和 $b_s^j=\sum_{i\in U_1} b_i^j=\sum_{i\in U_1} f_i\left(x_j\right)$，用户将 $\left\{b_s^j,Z_j\right\}$ 发送给服务器(Z_j 用于后续哈希聚合校验)。服务器或任意能够收集到至少 t 个 b_s^j 的集合 $S\subseteq U_1$ ($\left|S\right|\geq t$)，可对函数 $F\left(x\right)\triangleq\sum_{i\in U_1} f_i\left(x\right)$ 在 $x=0$ 处进行插值来恢复掩码种子和 $b\triangleq F\left(0\right)=\sum_{i\in U_1} b_i$ (因为 $f_i\left(0\right)=b_i$)。按照公式(14)，通过 t 个或更多的 b_s^j 恢复出掩码种子和 b，服务器在恢复 b 后计算线性同态哈希的聚合值：

$$Z=\prod_{i\in U_1} Z_i=\mathrm{LHH}\left(\sum_{i\in U_1} w_i\right) \tag{14}$$

为节约通信，服务器将上面计算的 y_s 进行编码，将原始值化作 $y_s=y_s^{\alpha}\cdot p+M$ 的分解形式，并将 $\left\{m,Z,b,y_s^{\alpha},M\right\}$ 返回给 U_1 所有用户。

4) 用户验证阶段

在聚合阶段结束后，每个用户将从服务器接收到一组参数 $\left\{m,Z,b,y_s^{\alpha},M\right\}$ 与承诺 Com，用户在本地根据 y_s^{α} 与 M 解码出实际的聚合密文 $y_s=\mathrm{Dec}\left(y_s^{\alpha},M\right)$，得到 y_s 后，需依次完成承诺验证、掩码去除、明文恢复及哈希一致性校验等步骤，以确保服务器的计算正确性与结果完整性。用户首先执行承诺一致性验证，检验服务器返回的承诺值是否与聚合密文对应。该验证通过如下等式进行：

$$\mathrm{Com.ver}\left(\mathrm{Com},m,y_s\right):\quad \text{check } \mathrm{Hash}\left(y_s\|m\right)\overset{?}{=}\mathrm{Com} \tag{15}$$

若上式不成立，即哈希不匹配，则说明服务器可能篡改了结果或发生通信错误，此时用户将立即中止该轮训练并上报异常。由哈希函数的抗碰撞性可知，伪造满足等式的 $\left(y_s',m'\right)$ 对的概率可忽略不计。当承诺验证通过后，用户根据掩码种子和 b，使用同态伪随机生成器 HPRG 生成对应的群掩码 $\mathrm{HPRG}\left(b\right)=\prod_{i\in U_1}\mathrm{HPRG}\left(b_i\right)$，用于消除聚合结果中的加密随机性。根据公式(15)，通过除法操作与离散对数运算，从群表示中提取明文恢复未掩码化的聚合形式：

$$w=\log_g\left(\frac{y_s}{\mathrm{HPRG}\left(b\right)}\right)=\log_g\left(g^{\sum_{i\in U_1} w_i}\right)=\sum_{i\in U_1} w_i \tag{16}$$

在实际运算中，为避免直接求解大规模离散对数，可采用整数编码与分块解码机制，即将每个维度的梯度值嵌入到较小范围的整数域中，通过查表或安全协议逐步恢复。对于高维场景，可进一步采用分维解码 $w_k=\log_g\left(g^{\sum_{i\in U_1} w_{i,k}}\right),k=1,2,\cdots,d$ 以并行化计算过程并降低复杂度。

为确保服务器未在聚合或传输过程中篡改任何数据，用户执行线性同态哈希一致性校验。根据线性同态哈希的定义，判断 $Z \stackrel{?}{=} \mathrm{LHH}(w)$，若公式成立，则说明服务器公布的 Z 与明文聚合梯度 w 一致，系统聚合过程被正确执行。否则，若存在 $Z \neq \mathrm{LHH}(w)$，则判定服务器存在计算错误或篡改行为。由于 LHH 具有单向性与抗碰撞性，伪造合法哈希的成功概率可忽略，因此该验证过程可强有力地保证联邦聚合阶段的正确性与可追溯性。

3.3.3. 跨组知识蒸馏

在组内联邦学习收敛后，每个组 G_k 得到一个组级模型 w_{G_k}。为了进一步提升模型性能，使不同用户之间的模型相对公平，允许用户进行跨组知识蒸馏。每个用户 u 可以选择多个教师模型构成集合 $\mathcal{T}_u \subseteq \{1,2,\cdots,K\}$，学生模型为自己的私有模型通过优化以下损失函数来进行知识蒸馏：

$$\mathcal{L}_{\mathrm{KD}}(x,y;\theta_u) = \alpha \cdot \mathcal{L}_{\mathrm{CE}}\left(y, f_u(x)\right) + (1-\alpha) \cdot \sum_{k\in\mathcal{T}_u} \lambda_k \cdot \mathcal{L}_{\mathrm{KL}}\left(f_{G_k}(x/T) \,\|\, f_u(x/T)\right) \tag{17}$$

$\mathcal{L}_{\mathrm{CE}}(y,\hat{y}) = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$ 为交叉熵损失函数，$\mathcal{L}_{\mathrm{KL}}(p\|q) = \sum_{c=1}^{C} p_c \log \frac{p_c}{q_c}$ 为 KL 散度，度量教师模型与学生模型输出的分布差异，$\alpha \in [0,1]$ 为损失权重系数，平衡监督损失与蒸馏损失，$T>0$ 为温度参数，用于平滑教师模型输出分布以增强蒸馏信号，λ_k 为教师模型 w_{G_k} 的权重，满足归一化条件 $\sum_{k\in\mathcal{T}} \lambda_k = 1$。用户 u 可以根据其本地数据分布与各教师模型的匹配程度，自适应地设置权重 λ_k。而基于本地验证集上的模型性能设定权重，就是一类行之有效的权重分配策略：

$$\lambda_k = \frac{\exp\left(\eta \cdot \mathrm{Accuracy}_k\right)}{\sum_{j\in\mathcal{T}_u} \exp\left(\eta \cdot \mathrm{Accuracy}_j\right)} \tag{18}$$

其中，$\mathrm{Accuracy}_k$ 表示教师模型 w_{G_k} 在用户 u 的本地验证集上的分类准确率，η 为温度参数。当 η 较大时，用户倾向于选择性能最优的教师模型；当 η 较小时，多个教师模型的知识会以更均衡的方式融合。进一步地，用户还可以根据本地数据分布差异度 $D_u(G_k)$ 动态修正权重：

$$\lambda_{k'} = \frac{\lambda_k \cdot \exp\left(-\beta D_u(G_k)\right)}{\sum_{j\in\mathcal{T}_u} \lambda_j \cdot \exp\left(-\beta D_u(G_j)\right)} \tag{19}$$

其中，$D_u(G_k)$ 表示用户数据与组模型特征分布的差异，β 为调节系数。该修正机制使蒸馏过程更关注分布相近的教师模型，从而提升个性化适配效果。跨组知识蒸馏通过多教师协同、动态权重调整与温度平滑蒸馏，实现了用户模型的个性化优化，使各用户在保持隐私的前提下高效吸收跨域知识，显著提升了系统整体的个性化性能。

4. 理论分析

本节通过理论分析证明了方案的隐私保护能力，本文隐私性定义为：即使服务器与恶意用户串通，也不会破坏其他在线用户的数据隐私安全，下面给出并证明如下定理。

定理 1 服务器无法获得用户的原始梯度，恶意用户也无法获得其他用户的原始梯度。

证明：在本方案中，用户 u_i 上传给服务器的关于梯度的信息是掩码梯度 $y_i = g^{w_i} \cdot \mathrm{HPRG}(b_i)$ 以及线性同态哈希标签 $Z_i = \mathrm{LHH.Hash}(w_i)$。

对于 y_i：种子 b_i 仅用户 u_i 持有，未以任何形式直接暴露给服务器或其他用户。HPRG 以 b_i 为输入生成的掩码 $\mathrm{HPRG}(b_i)$ 对外表现为群上的伪随机元素，在不知道 b_i 的前提下，$y_i = g^{w_i} \cdot \mathrm{HPRG}(b_i)$ 与群上的

均匀随机元素无法区分，从中单独提取 g^{w_i} 进而求解 w_i，等价于在已知随机掩码乘积的情况下分离出其中一个因子，这在计算上是不可行的。

对于 $Z_i = \mathrm{LHH.Hash}(w_i)$：LHH.Hash 的实际计算为 $\mathrm{LHH}(w_i) = g^{\sum_l \alpha_l w_{i,k}}$，其指数中的系数 $\{\alpha_l\}$ 由 TA 保密生成且从不泄露。在离散对数困难假设下，从群元素 Z_i 反推 $\sum_l \alpha_l \theta_{i,l}^k$ 的值是不可行的，更无法由此还原各分量 $\theta_{i,l}^k$。

对于恶意用户：其他用户能直接观察到的同样只有 y_i 与 Z_i，分析与上述相同。此外，用户 i 虽向其他用户发送了 Shamir 份额 $b_i^j = f_i(x_j)$，但每个用户 $j \neq i$ 仅持有一个份额，远不足以达到恢复阈值 t，无法还原 b_i，进而无法去除 y_i 中的掩码。

综上，无论是服务器还是恶意用户，在协议执行过程中均无法从任何可见信息中恢复用户的原始梯度。定理得证。

定理 2 若串通的恶意用户数量不超过 $t-2$，则本文的方案能够保证本地梯度的安全性。

证明：本方案通过 Shamir 秘密共享对掩码种子 b_i 进行保护。用户 i 选取一个 $t-1$ 次随机多项式 $f_i(x) = b_i + a_{i,1}x + \cdots + a_{i,t-1}x^{t-1} \pmod q$，将份额 $b_i^j = f_i(x_j)$ 分别发送给用户 j，仅保留自身份额 b_i^i。要恢复 b_i，必须收集到不少于 t 个合法份额才能通过 Lagrange 插值求得 $f_i(0) = b_i$。现假设恶意用户的串通规模为 $c \le t-2$，则即便这 c 个恶意用户将各自持有的份额全部汇总，也至多掌握 $c \le t-2$ 个关于 b_i 的份额，不满足恢复所需的 t 份阈值。在 $t-1$ 次多项式的代数结构下，少于 t 个点不能唯一确定多项式，b_i 的取值在有限域 $\mathbb{F}_q$ 上仍有多种可能，恶意方无法确定其真实值。由于无法恢复 b_i，恶意合谋方同样无法计算掩码 $\mathrm{HPRG}(b_i)$，因而对 $y_i = g^{w_i} \cdot \mathrm{HPRG}(b_i)$ 中的 g^{w_i} 进行剥离是不可行的，本地梯度 w_i 的安全性得以保证。

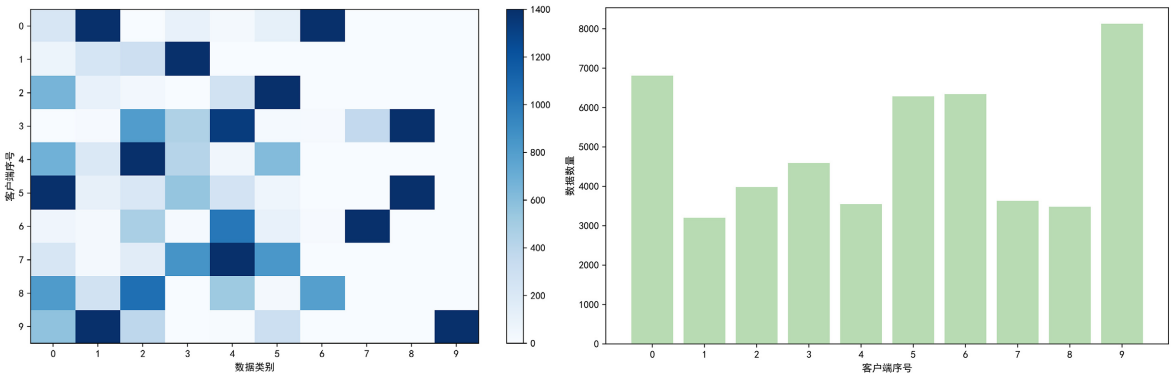
5. 实验结果与分析

所有实验均在统一的软硬件环境下完成，以保证实验结果的可复现性与公平性。实验采用一台配备高性能 GPU 的服务器。

数据集：本文在 MNIST 和 CIFAR-10 两个数据集上进行实验

MNIST 数据集包含 60,000 张手写数字的图像，每张图像为 28 × 28 像素，表示一个 0 到 9 之间的数字。数据集的每个样本都是一个灰度图像，并且每个图像都与一个数字标签对应，标签表示图像中的数字。测试集包含 10,000 张图像。

CIFAR 10 数据集包含 60,000 张彩色 32 × 32 图像，属于 10 个类别，包括猫、鸟、飞机等，其中 10,000 个是测试样本。CIFAR 10 是一个平衡的数据集，每个类有 6000 张图像。



(a) (b)

Figure 3. Visualization of partial results of Dirichlet partition
图 3. 狄利克雷划分部分结果可视化

为模拟真实联邦学习场景中数据异质性问题，本文采用 Dirichlet 分布对数据进行非独立同分布划分，通过调节 Dirichlet 分布的浓度参数α，控制各客户端本地数据的类别偏置程度。α越小，客户端间数据分布差异越大。本文对两个数据使用相同的$\alpha = 0.5$，将数据划分到客户端。图 3 显示了对 MNIST 数据集的划分，图 3(a)中的横坐标是标签编号，纵坐标是用户编号，颜色深浅代表数量多少；图 3(b)显示了客户端的总数据量。由于客户端太多，因此随机选取 10 个客户端进行展示。

训练配置：通信轮次为 100，学习率为 0.01，本地训练轮数设置为 5，本地批处理大小为 64，公共数据集大小为 500。

实验结果：

1) 隐私保护效能评估

本实验在 MNIST 数据集上实现了梯度泄露攻击，攻击结果如图 4 所示。第一行为原始图像，第二行是对原始梯度攻击迭代 10,000 次的结果，第三行是对本方案上传的加密梯度攻击迭代 10,000 次后的结果。可以看出我们的方案能显著提升隐私保护效果，攻击者难以从模型输出中恢复原始训练数据。

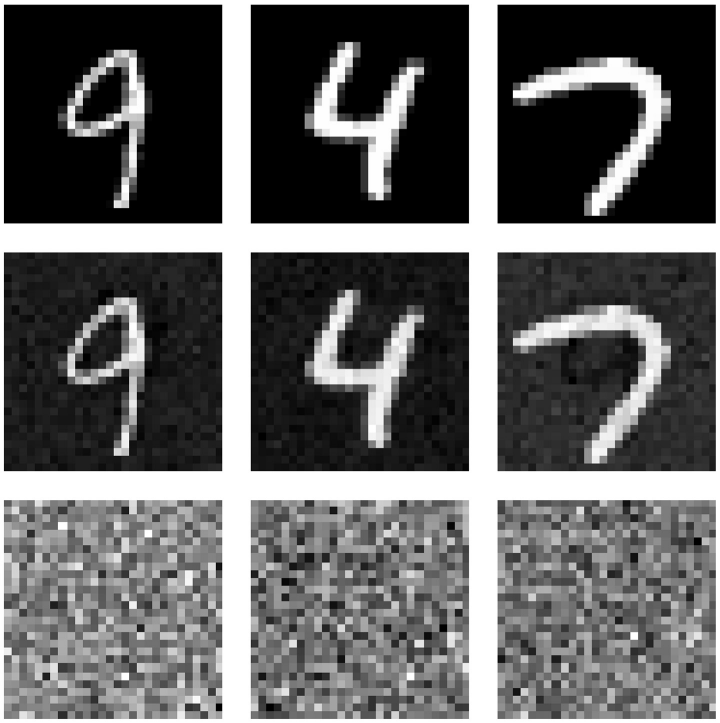


Figure 4. Privacy protection effectiveness evaluation
图 4. 隐私保护效能评估

2) 方案公平性效果验证

本实验在 100 轮训练后对比分析了 MNIST 数据集上 30 个用户的模型准确率。图 5 显示了准确率随轮次变换的图，不同组的准确率有较大差异，但是经过蒸馏处理后，模型的准确率有了显著提升，并且各个用户之间的准确率差距大大缩小，几乎达到了相同的水平。经过跨组知识蒸馏，不仅提升了模型的性能，还有效减小了用户之间由于数据分布差异所带来的性能差距，使不同用户之间的模型更加公平。

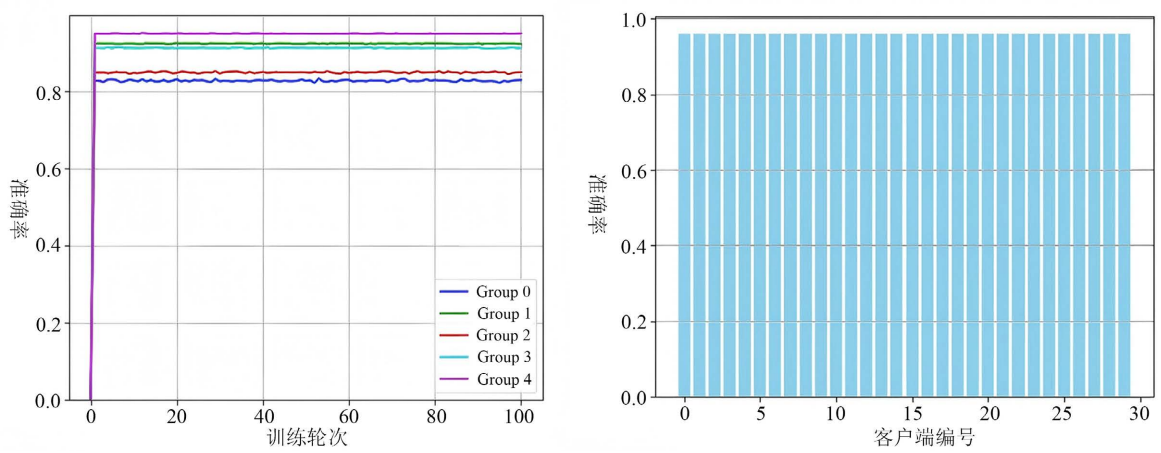


Figure 5. Accuracy before and after group distillation of the MNIST
图 5. MNIST 数据集分组蒸馏前后的准确率

3) 公共数据集规模影响

本实验在 MNIST 和 CIFAR-10 数据集上评估了三种不同规模的公共数据集(100、500 和 3000 样本量)。实验结果如图 6 所示，可以看出三种规模的数据集在收敛速度和最终性能上均未表现出显著差异。因此，在较小规模的公共数据集下仍实现优异的模型性能表现。

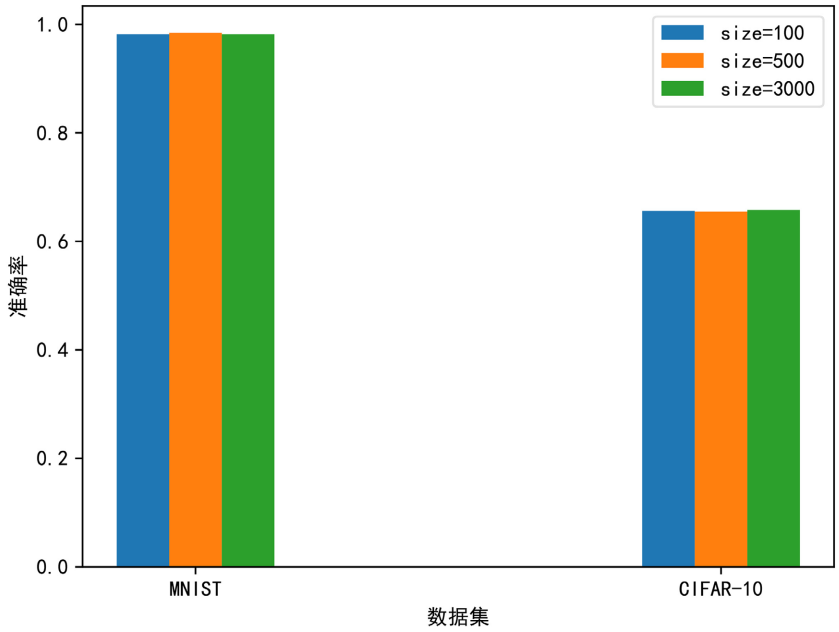


Figure 6. Average accuracy of different public dataset sizes
图 6. 不同公共数据集大小的平均准确率

4) 准确率对比

本实验在同质和异质模型下对比分析了本方案与 KT-pFL、PHP-FL、FedMD、FedAvg 的准确率表现。同质模型所有用户都使用 CNN-2 模型，结果如表 1 所示；异质模型是用户从 CNN-1、CNN-2、MLP 三个模型中随机选取，结果如表 2 所示。在 MNIST 数据集上，KT-pFL 在同质模型中表现最佳，而本方案在异质模型中以 95.94%的准确率领先其他方法。对于 CIFAR-10 数据集，本方案在同质和异质模型下均

表现最优，分别达到 65.61%和 56.72%，显著优于 KT-pFL、PHP-FL 和 FedMD 等方法。FedAvg 作为基线方法仅适用于同质模型，且性能相对较低。总体而言，本方案在两种数据集和模型架构下均展现出较强性能，特别是在处理异质模型和复杂数据时优势明显。

Table 1. Accuracy comparison under the homogeneous model
表 1. 同质模型下的准确率对比

方案	KT-pFL	PHP-FL	FedMD	FedAvg	Ours
MNIST	98.89%	98.58%	98.33%	95.74%	98.63%
CIFAR-10	61.34%	62.65%	60.29%	57.20%	65.61%

Table 2. Accuracy comparison under heterogeneous models
表 2. 异质模型下的准确率对比

方案	KT-pFL	PHP-FL	FedMD	Ours
MNIST	95.10%	94.27%	93.32%	95.94%
CIFAR-10	52.26%	53.16%	51.32%	56.72%

参考文献

[1] Mcmahan, B., Moore, E., Ramage, D., *et al*. (2017) Communication-Efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, **54**, 1273-1282.

[2] Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., *et al*. (2020) Federated Learning in Medicine: Facilitating Multi-Institutional Collaborations without Sharing Patient Data. *Scientific Reports*, **10**, Article No. 12598. https://doi.org/10.1038/s41598-020-69250-1

[3] Abdul Salam, M., Fouad, K.M., Elbably, D.L. and Elsayed, S.M. (2024) Federated Learning Model for Credit Card Fraud Detection with Data Balancing Techniques. *Neural Computing and Applications*, **36**, 6231-6256. https://doi.org/10.1007/s00521-023-09410-2

[4] Kairouz, P. and McMahan, H.B. (2021) Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, **14**, 1-210. https://doi.org/10.1561/2200000083

[5] Pei, J., Liu, W., Li, J., Wang, L. and Liu, C. (2024) A Review of Federated Learning Methods in Heterogeneous Scenarios. *IEEE Transactions on Consumer Electronics*, **70**, 5983-5999. https://doi.org/10.1109/tce.2024.3385440

[6] Dembani, R., Karvelas, I., Akbar, N.A., Rizou, S., Tegolo, D. and Fountas, S. (2025) Agricultural Data Privacy and Federated Learning: A Review of Challenges and Opportunities. *Computers and Electronics in Agriculture*, **232**, Article 110048. https://doi.org/10.1016/j.compag.2025.110048

[7] 郭倩, 赵津, 过弋. 基于分层聚类的个性化联邦学习隐私保护框架[J]. 信息网络安全, 2024, 24(8): 1196-1209.

[8] Sabah, F., Chen, Y., Yang, Z., Raheem, A., Azam, M., Ahmad, N., *et al*. (2025) FairDPFL-SCS: Fair Dynamic Personalized Federated Learning with Strategic Client Selection for Improved Accuracy and Fairness. *Information Fusion*, **115**, Article 102756. https://doi.org/10.1016/j.inffus.2024.102756

[9] Tan, A.Z., Yu, H., Cui, L. and Yang, Q. (2023) Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, **34**, 9587-9603. https://doi.org/10.1109/tnnls.2022.3160699

[10] Li, X., Jiang, M., Zhang, X., *et al*. (2021) FedBN: Federated learning on Non-IID Features via Local Batch Nor-Malization. arXiv:2102.07623.

[11] Wang, H., Kaplan, Z., Niu, D. and Li, B. (2020) Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. *IEEE INFOCOM* 2020-*IEEE Conference on Computer Communications*, Toronto, 6-9 July 2020, 1698-1707. https://doi.org/10.1109/infocom41043.2020.9155494

[12] Chen, Y., Qin, X., Wang, J., Yu, C. and Gao, W. (2020) FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *IEEE Intelligent Systems*, **35**, 83-93. https://doi.org/10.1109/mis.2020.2988604

[13] Yang, H., He, H., Zhang, W. and Cao, X. (2021) FedSteg: A Federated Transfer Learning Framework for Secure Image Steganalysis. *IEEE Transactions on Network Science and Engineering*, **8**, 1084-1094. https://doi.org/10.1109/tnse.2020.2996612

[14] Fallah, A., Mokhtari, A. and Ozdaglar, A. (2020) Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems*, **33**, 3557-3568.

[15] Jiang, Y., Konečný, J., Rush, K., *et al*. (2019) Improving Federated Learning Personalization via Model Agnostic Meta Learning. arXiv:1909.12488.

[16] Zhang, J., Guo, S., Ma, X., *et al*. (2021) Parameterized Knowledge Transfer for Personalized Federated Learning. *Advances in Neural Information Processing Systems*, **34**, 10092-10104.

[17] Li, D. and Wang, J. (2019) FedMD: Heterogenous Federated Learning via Model Distillation. arXiv:1910.03581.

[18] Zhu, L., Liu, Z. and Han, S. (2019) Deep Leakage from Gradients. arXiv:1906.08935.

[19] Geiping, J., Bauermeister, H., Dröge, H., *et al*. (2020) Inverting Gradients-How Easy Is It to Break Privacy in Federated Learning? *Advances in Neural Information Processing Systems*, **33**, 16937-16947.

[20] Mohassel, P. and Zhang, Y. (2017) SecureML: A System for Scalable Privacy-Preserving Machine Learning. 2017 *IEEE Symposium on Security and Privacy* (*SP*), San Jose, 22-26 May 2017, 19-38. https://doi.org/10.1109/sp.2017.12

[21] Byali, M., Chaudhari, H., Patra, A. and Suresh, A. (2020) FLASH: Fast and Robust Framework for Privacy-Preserving Machine Learning. *Proceedings on Privacy Enhancing Technologies*, **2020**, 459-480. https://doi.org/10.2478/popets-2020-0036

[22] Xu, R., Baracaldo, N., Zhou, Y., Anwar, A. and Ludwig, H. (2019) HybridAlpha: An Efficient Approach for Privacy-Preserving Federated Learning. *Proceedings of the* 12*th ACM Workshop on Artificial Intelligence and Security*, London, 15 November 2019, 13-23. https://doi.org/10.1145/3338501.3357371

[23] 徐茹枝, 仝雨蒙, 戴理朋. 基于异构数据的联邦学习自适应差分隐私方法研究[J]. 信息网络安全, 2025, 25(1): 63-77.

[24] Zhang, L., Xu, J., Vijayakumar, P., Sharma, P.K. and Ghosh, U. (2023) Homomorphic Encryption-Based Privacy-Preserving Federated Learning in IoT-Enabled Healthcare System. *IEEE Transactions on Network Science and Engineering*, **10**, 2864-2880. https://doi.org/10.1109/tnse.2022.3185327

[25] Park, J., Yu, N.Y. and Lim, H. (2022) Privacy-Preserving Federated Learning Using Homomorphic Encryption with Different Encryption Keys. 2022 13*th International Conference on Information and Communication Technology Convergence* (*ICTC*), Jeju Island, 19-21 October 2022, 1869-1871. https://doi.org/10.1109/ictc55196.2022.9952531

[26] Liao, J., Chen, Z. and Larsson, E.G. (2022) Over-the-Air Federated Learning with Privacy Protection via Correlated Additive Perturbations. 2022 58*th Annual Allerton Conference on Communication, Control, and Computing* (*Allerton*), Monticello, 27-30 September 2022, 1-8. https://doi.org/10.1109/allerton49937.2022.9929413

[27] Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., *et al*. (2020) Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *IEEE Transactions on Information Forensics and Security*, **15**, 3454-3469. https://doi.org/10.1109/tifs.2020.2988575

[28] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., *et al*. (2017) Practical Secure Aggregation for Privacy-Preserving Machine Learning. *Proceedings of the* 2017 *ACM SIGSAC Conference on Computer and Communications Security*, Dallas, 30 October 2017-3 November 2017, 1175-1191. https://doi.org/10.1145/3133956.3133982

[29] Liu, Z., Guo, J., Lam, K. and Zhao, J. (2023) Efficient Dropout-Resilient Aggregation for Privacy-Preserving Machine Learning. *IEEE Transactions on Information Forensics and Security*, **18**, 1839-1854. https://doi.org/10.1109/tifs.2022.3163592

[30] Pan, Y., Su, Z., Ni, J., Wang, Y. and Zhou, J. (2024) Privacy-Preserving Heterogeneous Personalized Federated Learning with Knowledge. *IEEE Transactions on Network Science and Engineering*, **11**, 5969-5982. https://doi.org/10.1109/tnse.2024.3386623

[31] Guo, X., Liu, Z., Li, J., Gao, J., Hou, B., Dong, C., *et al*. (2021) VeriFL: Communication-Efficient and Fast Verifiable Aggregation for Federated Learning. *IEEE Transactions on Information Forensics and Security*, **16**, 1736-1751. https://doi.org/10.1109/tifs.2020.3043139

[32] Gao, S., Luo, J., Zhu, J., Dong, X. and Shi, W. (2023) VCD-FL: Verifiable, Collusion-Resistant, and Dynamic Federated Learning. *IEEE Transactions on Information Forensics and Security*, **18**, 3760-3773. https://doi.org/10.1109/tifs.2023.3271268