arXiv:2301.07686v1 [cs.CR] 18 Jan 2023

通过私有集合并集实现私有联邦子模型学习*

Zhusheng Wang Sennur Ulukus

电气与计算机工程系 马里兰大学帕克分校，马里兰州 20742 zhusheng@umd.edu
ulukus@umd.edu

摘要

我们研究联邦子模型学习（FSL）问题，并提出一种方法，使客户端能够以信息论隐私的方式更新中央模型。我们的方法基于私有集合并（PSU），而私有集合并进一步基于多消息对称私有信息检索（MM-SPIR）。服务器有两个互不串通的数据库，它们以复制方式存储模型。使用我们的方案，服务器除了客户端所更新的子模型子集之外，无法获知任何其他信息：服务器既不知道哪个客户端更新了哪个子模型，也无法获知任何有关客户端本地数据的信息。与 Jia-Jafar 和 Vithana-Ulukus 提出的最先进私有 FSL 方案相比，我们的方案不要求数据库对模型进行噪声存储；与 Zhao-Sun 的安全聚合方案相比，我们的方案不要求预先分发客户端公共随机性，而是通过随机对称私人信息检索（RSPIR）和一次性密码本生成所需的客户端公共随机性。我们的系统初始化时，会在服务器端的两个数据库中复制存储子模型，并配置充足的公共随机性。协议从公共随机性生成（CRG）阶段开始，两个数据库使用 RSPIR 和一次性密码本在客户端侧建立公共随机性（此阶段称为 FSL-CRG）。接下来，客户端利用已建立的客户端公共随机性，让服务器以私密方式确定客户端要共同更新的子模型索引并集（此阶段称为 FSL-PSU）。随后，两个数据库向客户端广播集合并中的子模型当前版本。客户端根据本地训练数据更新子模型。最后，客户端使用 FSL-PSU 的一种变体，将更新私密地写回数据库（此阶段称为 FSL-write）。由于服务器上的数据库之间不进行通信，我们采用一种新方法，利用随机选出的在线客户端在两个数据库之间转发所需信息。我们提出的私有 FSL 方案能够抵御客户端掉线、客户端迟到以及数据库掉线。

*本研究得到 ARO资助项目 W911NF2010142 的支持。

1 引言

标准机器学习方法要求将全部训练数据集中存储在一台机器或服务器上，以学习目标模型。当今机器学习应用面临的一项实际挑战是，训练数据分散在多个彼此隔离的客户端中。例如，在电子商务中训练推荐系统，需要与大量由不同客户持有的移动设备交互，以利用这些设备上的数据。鉴于训练数据的分布式特性，一种称为联邦学习（FL）的新方法应运而生 [1, 2]。在联邦学习中，每个客户端被选中并从全局服务器获取当前模型后，便使用自己的数据进行一轮本地训练。在联邦学习场景下，如何协作完成学习任务，同时保护各客户端存储的本地数据不被全局服务器窥探，是一个重要问题。一种直观的隐私保护方式是使用安全聚合协议，使全局服务器无法检查任何单个客户端的更新 [3]。作为一个独立课题，安全聚合一直是计算机科学文献中的活跃研究主题，参见 [3–6]。近年来，针对客户端之间各种公共随机性分发设置，研究者提出了旨在实现最优通信成本的信息论安全聚合方案，参见 [7–9]。然而，这些论文分别考虑输入和公共随机性的通信成本，却没有明确说明具体方案实现中的公共随机性生成与分配过程。 [10] 研究了另一种信息论安全聚合方案，其中公共随机性的共享依赖于任意两个客户端之间的私有通信链路。

作为当前已成熟的联邦学习框架的一种扩展，近期提出了一种称为联邦子模型学习（FSL）的新框架，以进一步降低服务器端和客户端的通信与计算开销 [11]。在子模型框架中，存储于服务器上的完整学习模型会根据数据特征划分为多个子模型。每个选定客户端仅从服务器下载所需的子模型，然后根据其本地数据的类型上传相应的子模型更新。正如 [11], 所指出的，联邦子模型学习框架可以归纳出两个基本问题：其一，客户端如何在不向好奇的服务器泄露这些子模型索引的情况下，下载所需的子模型；其二，客户端如何在仍不向好奇的服务器泄露索引或已更新子模型内容的情况下，更新这些所需的子模型。第一个问题是私有读取问题，第二个问题是私有写入问题。

文献 [11] 提出了一种弱隐私方案，具体如下：首先，服务器以安全的方式获取所有选定客户端的目标子模型索引并集。如 [12, 13], 所述，这本质上是一个多方私有集合并（MP-PSU）问题，并通过 [11]中的布隆过滤器加以解决。随后，服务器将这个可能包含假阳性的不准确并集结果发送给每个选定客户端。每个客户端收到该信息后，会在此

并集范围内通过随机响应对其真实目标子模型索引集进行随机化。从服务器收到与该随机化后的目标子模型索引集相对应的子模型后，每个客户端仅训练其真实目标子模型索引集与随机化后的目标子模型索引集交集内的索引所对应的子模型，然后通过安全聚合上传更新。因此， [11] 部分解决了上述两个基本隐私问题，但牺牲了客户端的更新效率。此外，FSL 中的安全聚合问题与 [14], 中的私有多组聚合问题有关，其中不同的子模型被视为不同的组。

本文提出一种基于 [15] 交叉子空间对齐[16]、具有强隐私保护能力的 FSL 方法。在该方法中，每轮 FSL 流程仅有一个对特定子模型感兴趣的客户端参与。如果每个数据库都存储用于学习的明文完整模型，那么数据库便能通过比较训练前后存储的信息，获知该客户端所做的更新。因此，服务器中的多个分布式数据库会基于阈值秘密共享方案，以加噪方式存储完整模型。这种存储冗余有助于向数据库隐藏写入（更新）的子模型的索引和内容；也就是说，数据库无法判断哪个子模型被更新，也无法得知更新后的值。这种存储冗余也有助于保护子模型，使其不被数据库获知；也就是说，尽管数据库持有这些子模型，却无法得知其实际值。此外，文中还提出一种通信成本效率更高的改进方案 [17], ，并将其扩展至稀疏化更新的情形，从而进一步降低通信成本 [18]。

本文提出了一种新的 FSL 方案，保留了上述两种方法的主要优点，且隐私保护级别介于二者之间。具体而言，我们提出了一种两阶段方案。第一阶段，服务器安全地计算客户端目标子模型索引的并集；第二阶段，服务器在计算得到的集合并集中安全地聚合客户端生成的更新。第一阶段是众所周知的私有集合并（PSU）问题，本文称之为 FSL-PSU 阶段。在此阶段，服务器只能获知待客户端更新的子模型索引的并集，但无法获知各个客户端分别要更新哪些子模型索引。第二阶段是众所周知的安全聚合问题，本文称之为 FSL-write 阶段。在此阶段，服务器安全地聚合子模型更新，而无法查看单个更新。也就是说，在这两个阶段中，服务器只能获知最终结果，而无法知道哪个客户端为最终结果做出了何种贡献。我们采用同一核心思想的两种不同形式，一并解决这两个问题。为此，我们首先提出一种新颖的 PSU 方法，用于 FSL-PSU 阶段，且该方法具有信息论安全性；然后对其进行修改，得到一种具有信息论安全性的 FSL-write 阶段聚合方法。

在密码学领域，私人信息检索（PIR）指的是一个基本问题：用户希望从存储于多个互不串通且相互复制的数据库中的一组消息里检索特定消息，同时完全隐藏

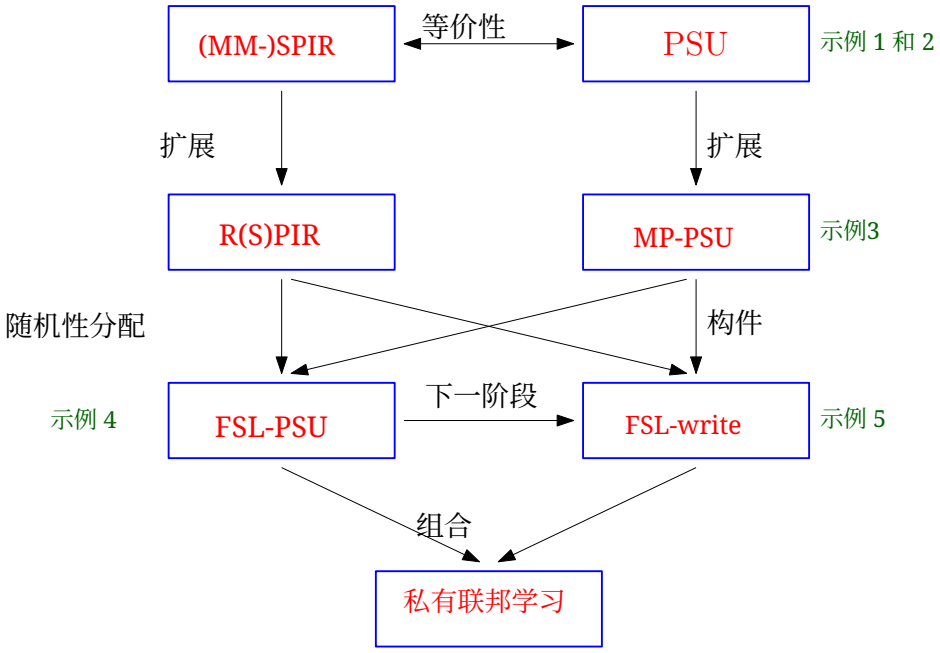


图 1：本文中使用的技术、它们之间的关系，以及私有联邦学习的发展路线图。方框旁的示例引用展示了本文后续给出的示例，用于说明各方框的功能。

所需消息的索引，使其不被每个单个数据库知晓 [19]。对称私有信息检索（SPIR）进一步要求，用户在下载所需消息后，无法获得有关数据库中其余消息的任何信息 [20]。继开创性论文研究多数据库私有信息检索的信息论容量之后 [21], ，PIR 和 SPIR 近年来在信息论领域引起了广泛关注，例如 [16, 22–41]。作为 SPIR 的一种非平凡变体，在多消息对称私有信息检索（MM-SPIR）中，用户希望一次检索多条消息 [41]。论文 [41] 还建立了 MM-SPIR 问题与私有集合求交（ PSI ）问题之间的等价性。在 PSI 问题中，两个参与方希望确定各自所拥有数据集合中的共同元素，同时不泄露有关其余元素的任何额外信息。注意，PSI 和 PSU 问题中的约束是类似的。在本文中，我们建立了 PSU 与 MM-SPIR 问题之间的等价性。

在本研究中，与典型的 PIR/SPIR 表述类似，我们考虑联邦学习服务器拥有多个数据库的情形，并聚焦于服务器拥有两个数据库这一最简单的情形。图 1 展示了本文采用的技术、它们之间的关系，以及本文的研究路线图。图 1 中方框旁的参考文献给出了本文后续介绍的示例，用以说明每个方框的功能。经典信息论对称私有信息检索是构建新的联邦学习可达方案的起点。我们首先展示多消息对称私有信息检索与私有集合并问题之间的等价性，随后将其扩展到 [41, 42] 中的多参与方私有集合并（MP-PSU）情形；因此，我们使用多消息对称私有信息检索方案来构建 MP-PSU 的信息论可达方案。我们使用这一

MP-PSU 框架作为构件，联合开发 FSL-PSU 和 FSL-write 阶段，并指出 FSL-PSU 与 FSL-write 本质上是类似的安全问题。对于 FSL-PSU 和 FSL-write 阶段，需要在客户端之间共享一定量的必要公共随机性，且这些随机性不能为任何单个数据库所知。仍以 SPIR 问题为起点，我们利用 [43] 中新提出的随机SPIR（RSPIR），在客户端之间完成公共随机性的分配。

在 [44] 中提出了一种经典多数据库安全聚合方案，其核心思想概述如下：在由 N 个相互独立的数据库组成的服务器，以及C个独立客户端的设置中，每个客户端 i将其输入 $z^{\langle i \rangle}$ 拆分为 N个份额[1] $z_{[N]}^{\langle i \rangle} = \{z_1^{\langle i \rangle}, \ldots, z_N^{\langle i \rangle}\}$，使这些份额之和恰好等于该输入，即 $\sum_{j\in[N]} z_j^{\langle i \rangle} = z^{\langle i \rangle}$。随后，每个客户端 i 将份额 $z_j^{\langle i \rangle}$ 发送至数据库 j，对所有 $j \in [N]$均如此。收集所有客户端的份额后，每个数据库都会公布收到的份额之和 $\sum_{i\in[C]} z_j^{\langle i \rangle}$。然后，服务器计算客户端持有的所有输入之和 $\sum_{j\in[N]} \sum_{i\in[C]} z_j^{\langle i \rangle} = \sum_{i\in[C]} \sum_{j\in[N]} z_j^{\langle i \rangle} = \sum_{i\in[C]} z^{\langle i \rangle}$。尽管这一简单方案实现了所需的隐私保护，即数据库只能获得总和，而无法获知各客户端私有输入的其他信息，但它也存在一些不可避免的缺点。首先，上传成本很高，因为每个客户端都需要向每个数据库上传一个答案。其次，一旦某个数据库无法正常运行，服务器只能收到一些随机值，与最终总和相去甚远。第三，服务器内各数据库之间需要进行通信。

本文提出了一种新的隐私分布式FSL可实现方案，主要通过在同一框架下统一 FSL-PSU 和 FSL-write 问题来实现。在实际的联邦学习实现中，选定客户端的索引集每轮都会变化，因此客户端之间的通信通常不稳定（如果存在通信的话）；而在客户端之间建立直接通信信道的成本极高。由于数据库间通信也不被允许，我们的方案仅依赖客户端与数据库之间通过若干条直接、安全且经过身份验证的客户端-数据库通信信道进行的诚实通信。此外，由于 FSL-write 阶段持续时间较长，随着该阶段的推进，部分客户端可能会掉线。因此，我们设计的方案能够在部分客户端失去与服务器的连接时继续正常运行。我们的方案在 FSL-PSU 阶段也能够抵御客户端掉线。此外，部分客户端生成的答案可能会迟到达其关联数据库，导致相应数据库错误地判定这些客户端已经掉线。我们的方案经过专门设计，可确保这些迟到的答案不会向数据库泄露有关这些迟到客户端的任何额外信息；这称为对客户端迟到的鲁棒性。此外，即使部分数据库变为非活动状态，我们的方案仍能正常运行，尤其是在数据库总数足够多时。我们方案的另一个关键方面

[1]为简洁起见，对于正整数 Z，本文采用记号 $[Z] = \{1, 2, \ldots, Z\}$。因此，此处的 $[N]$ 以及后文的 $[C]$、$[K]$、$[N_1]$、$[L]$ 均表示整数集。

是将产生的总通信成本纳入考量，其中包括客户端生成公共随机性的成本。最后，我们的 FSL 方案可以在多轮 FSL 流程中以迭代方式运行，直至满足预先设定的终止准则。

2 问题表述

2.1 多消息对称私有信息检索

如 [41], 所述，我们考虑 $N \geq 1$ 个非串通数据库，每个单个数据库都存储着相同的 $K \geq 2$ 个独立同分布消息 $W_{[K]} = \{W_1, \ldots, W_K\}$。每条消息中的 L 个独立同分布符号均从一个足够大的有限域 $\mathbb{F}_q$中均匀选取，因此，

$$H(W_k) = L, \quad \forall k \tag{1}$$

$$H(W_{[K]}) = H(W_1) + \cdots + H(W_K) = KL \tag{2}$$

多消息对称私有信息检索问题的目标是从消息集合 $W_{[K]}$ 中检索一组消息 W_Ω ，同时不向任何单个数据库泄露有关所检索索引集 $\Omega = \{i_1, i_2, \cdots, i_P\} \subseteq [K]$ 的信息，其基数为 $|\Omega| = P$（用户隐私约束）；并且除所需消息集合 W_Ω 外，不获取任何其他信息（数据库隐私约束）。所检索消息集合 P 的基数是公开信息，所有数据库均知晓。由于数据库隐私约束，数据库需要共享一定量的服务器端公共随机性 $\mathcal{R}_S$ ，且该随机性对用户不可见。服务器上的服务器端公共随机性 $\mathcal{R}_S$ 与消息集合 $W_{[K]}$ 相互独立。

期望消息索引集 Ω 是一个随机变量，对应于从集合 $[K]$ 中均匀地无放回选取元素；Ω 的样本空间是 $[K]$ 的幂集。我们用 $\mathcal{P}$ 表示随机变量 Ω 的一次实现。基于期望消息集合 Ω，用户在不知道数据库中存储的消息集合 $W_{[K]}$ 的情况下生成查询集合 $Q_{[N]}^{[\Omega]}$ ，因此，

$$I(W_{[K]}; Q_{[N]}^{[\Omega]}, \Omega) = 0 \tag{3}$$

对于任意期望消息索引集 $\mathcal{P}$，每个数据库在收到用户的查询后，都会根据所存储的消息集合和服务器端公共随机性给出真实答案，

$$\text{[多消息对称私有信息检索确定性答案]} \qquad H(A_n^{[\mathcal{P}]} | Q_n^{[\mathcal{P}]}, W_{[K]}, \mathcal{R}_S) = 0, \quad \forall n, \ \forall \mathcal{P} \tag{4}$$

随后，用户应能够在.....之后可靠地解码所需的消息集合

从所有数据库收集 N 个答案后，

$$[\text{MM-SPIR reliability}] \quad H(W_{\mathcal{P}}|Q_{[N]}^{[\mathcal{P}]}, A_{[N]}^{[\mathcal{P}]}, \mathcal{P}) = 0, \quad \forall \mathcal{P} \tag{5}$$

由于用户隐私约束，用于检索所需消息集的查询应与其他查询在统计上不可区分。因此，对于所有满足 $\mathcal{P} \neq \mathcal{P}'$ 和 $|\mathcal{P}| = |\mathcal{P}'| = P$ 的实现 $\mathcal{P}$ 和 $\mathcal{P}'$，

$$[\text{MM-SPIR user privacy}] \quad (Q_n^{[\mathcal{P}]}, A_n^{[\mathcal{P}]}, W_{[K]}, \mathcal{R}_S) \sim (Q_n^{[\mathcal{P}']}, A_n^{[\mathcal{P}']}, W_{[K]}, \mathcal{R}_S) \tag{6}$$

这等价于以下形式，

$$[\text{MM-SPIR user privacy}] \quad I(\Omega; Q_n^{[\Omega]}, A_n^{[\Omega]}, W_{[K]}, \mathcal{R}_S) = 0, \quad \forall n \tag{7}$$

由于数据库隐私约束，用户不应获知关于 $W_{\bar{\mathcal{P}}}$ 的任何信息，其中 $W_{\bar{\mathcal{P}}}$ 是 $W_{\mathcal{P}}$ 的补集，即 $W_{\bar{\mathcal{P}}} = W_{[K]\setminus\mathcal{P}}$，

$$[\text{MM-SPIR database privacy}] \quad I(W_{\bar{\mathcal{P}}}; Q_{[N]}^{[\mathcal{P}]}, A_{[N]}^{[\mathcal{P}]}, \mathcal{P}) = 0, \quad \forall \mathcal{P} \tag{8}$$

可实现的多消息对称私有信息检索方案是指满足可靠性约束 (5)、用户隐私约束 (6) 和数据库隐私约束 (8) 的方案。与单数据库对称私有信息检索类似，单数据库多消息对称私有信息检索也不可行。为了使单数据库多消息对称私有信息检索可行，我们在多消息对称私有信息检索设置中采用了 [36]，中扩展对称私有信息检索表述的多消息版本；在检索过程开始之前，用户可以获取共享的服务器端公共随机性的一个随机子集。

2.2 私有集合并

在 PSU 问题中，两个各自持有一个数据集的参与方希望共同计算各自集合的并集，同时不向任何一方透露其他信息。令 $\mathcal{A}$ 表示全局字母表。第一方 P_1 将数据集 Ω_1 存储在其自己的 $N_1 \geq 1$ 个副本数据库中，这些数据库互不串通；第二方 P_2 将数据集 Ω_2 存储在其自己的 $N_2 \geq 1$ 个副本数据库中，这些数据库互不串通。令 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 分别表示随机变量 Ω_1 和 Ω_2, 的取值。 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 中的所有元素均从 $\mathcal{A}$ 中选取，且服从任意统计分布，即 $\mathcal{P}_1, \mathcal{P}_2 \subseteq \mathcal{A}$。我们将其中一方称为领导方／服务器，另一方称为客户端／用户。不失一般性，设参与方 P_1 为服务器。那么，与 [41], P_1 类似，隐私约束、 P_2 隐私约束和 PSU 可靠性约束共同构成矛盾；与所有对称私有信息检索表述一样， [20, 35], 服务器数据库除各自的数据集外，还需要共享一定量的公共随机性 $\mathcal{R}_S$。然后，参与方 P_2 生成 N_1 查询 $Q_{[N_1]}^{[\mathcal{P}_2]}$，并将其发送给与参与方 P_1关联的数据库。收到查询$Q_{n_1}^{[\mathcal{P}_2]}$后，该参与方的第 n_1个数据库以答案 $A_{n_1}^{[\mathcal{P}_2]}$作出响应。

对于参与方 P_1中的每个数据库，答案 $A_{n_1}^{[\mathcal{P}_2]}$都应根据收到的查询、其自身的数据集和公共随机性如实生成，

$$\text{[PSU deterministic answer]} \quad H(A_{n_1}^{[\mathcal{P}_2]}|Q_{n_1}^{[\mathcal{P}_2]}, \Omega_1, \mathcal{R}_S) = 0, \quad \forall n_1, \ \forall \mathcal{P}_2 \tag{9}$$

私有集合并过程完成后，参与方 P_2 应能够根据发送的查询、收集的答案以及对 $\Omega_1 \cup \Omega_2$ 的了解，可靠地计算并集，而无需预先知道 $|\Omega_1 \cup \Omega_2|$ 。这由私有集合并可靠性约束表示：

$$\text{[PSU reliability]} \quad H(\Omega_1 \cup \Omega_2|Q_{[N_1]}^{[\mathcal{P}_2]}, A_{[N_1]}^{[\mathcal{P}_2]}, \Omega_2) = 0, \quad \forall \mathcal{P}_2 \tag{10}$$

私有集合并中的隐私要求可分为两部分，以保护每个参与方： P_1 隐私和 P_2 隐私。首先，参与方 P_2 希望保护 $\Omega_1 \cup \Omega_2$,。然而，由于参与方 P_2 在生成查询时并不知道 Ω_1 ，查询不能依赖于 Ω_1, ，因此， P_2 应仅在查询中保护 Ω_2 。因此， P_2 发送的查询不应泄露其自身数据集的任何信息；也就是说，与 P_1 相关联的任何单个数据库，都无法从其掌握的所有信息中获知有关 Ω_2 的任何信息：

$$\text{[PSU } P_2 \text{ privacy]} \qquad I(\Omega_2; Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) = 0, \quad \forall n_1 \tag{11}$$

由于全局字母表 $\mathcal{A}$已知且固定，显然有以下$^-$ $^-$两个约束 $H(\Omega_2|\Omega_2) = 0$ 和 $H(\Omega_2|\Omega_2) = 0$，由此可得以下关系：

$$\begin{aligned}
H(\Omega_2) &= H(\Omega_2) - H(\Omega_2|\bar{\Omega}_2) && (12)\\
&= I(\Omega_2; \bar{\Omega}_2) && (13)\\
&= H(\bar{\Omega}_2) - H(\bar{\Omega}_2|\Omega_2) && (14)\\
&= H(\bar{\Omega}_2) && (15)
\end{aligned}$$

因此，我们得到以下恒等式，

$$\begin{aligned}
&I(^-\Omega_2;\ Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S)\\
&\quad = H(^-\Omega_2) - H(^-\Omega_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) && (16)\\
&\quad = H(\bar{\Omega}_2) - H(\bar{\Omega}_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) + H(\bar{\Omega}_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \Omega_2, \mathcal{R}_S) && (17)\\
&\quad = H(\bar{\Omega}_2) - I(\bar{\Omega}_2; \Omega_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) && (18)\\
&\quad = H(\Omega_2) - H(\Omega_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) + H(\Omega_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \bar{\Omega}_2, \mathcal{R}_S) && (19)\\
&\quad = H(\Omega_2) - H(\Omega_2|Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) && (20)\\
&\quad = I(\Omega_2;\ Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) && (21)
\end{aligned}$$

因此，我们得到 P_2 隐私的以下等价表达式，

$$[\text{PSU } P_2 \text{ privacy}] \qquad I(\bar{\Omega}_2; Q_{n_1}^{[\Omega_2]}, A_{n_1}^{[\Omega_2]}, \Omega_1, \mathcal{R}_S) = 0, \quad \forall n_1 \tag{22}$$

根据并集结果 $\Omega_1 \cup \Omega_2$, ，参与方 P_2 始终知道参与方 P_1 包含 $(\Omega_1 \cup \Omega_2)\backslash\Omega_2$ 中的元素，且不包含 $(\Omega_1 \cup \Omega_2)$中的元素。注意到 $\overline{(\Omega_1 \cup \Omega_2)\backslash\Omega_2 \cup (\Omega_1 \cup \Omega_2)} = \overline{\Omega_2}$, ，因此， P_2 不应从生成的查询、收集的答案及其自身数据集中获知 P_1 是否包含 Ω_2 中的任何元素（我们将此信息记为 $E_{1,\ \Omega_2}$），

$$[\text{PSU } P_1 \text{ privacy}] \quad I(E_{1,\Omega_2}; Q_{[N_1]}^{[\mathcal{P}_2]}, A_{[N_1]}^{[\mathcal{P}_2]}, \Omega_2) = 0, \quad \forall \mathcal{P}_2 \tag{23}$$

定理 1 PSU 与多消息对称私有信息检索等价，其中 $L = 1$ 且 $P = |\mathcal{A}| - |\Omega_2|$。

证明：在将各参与方的数据集映射为相应的关联向量后，我们按照 [41] 中 PSI 与多消息对称私有信息检索等价性证明的思路，证明 PSU 与多消息对称私有信息检索的等价性。具体而言，若将 PSU 中的 Ω_2 视为多消息对称私有信息检索中的 Ω，则 PSU 问题中的 P_1 隐私、 P_2 隐私和 PSU 可靠性约束，与多消息对称私有信息检索问题中的数据库隐私、用户隐私和可靠性约束一致。相比之下，若将 PSI 中的 Ω_2 视为多消息对称私有信息检索中的 Ω，则 PSI 与多消息对称私有信息检索的三项约束一致。 ■

注释 1 根据 *[41],* ，我们注意到，PSI 等价于满足 $L = 1$ 且 $P = |\Omega_2|$ 的多消息对称私有信息检索。根据上述定理 1，我们注意到，PSU 等价于满足 $L = 1$ 且 $P = |\mathcal{A}| - |\Omega_2|$ 的多消息对称私有信息检索。[2] 根据德摩根定律，即 $\overline{A \cup B} = \overline{A} \cap \overline{B}$，我们有 $A \cup B = \overline{\overline{A} \cap \overline{B}}$，因此，集合并可以通过集合交集与集合补集的组合来实现。这表明了 PSU 与 PSI 问题之间的对偶性。不过，我们注意到，参与方应就执行 PSU 还是 PSI 达成一致，因为具体协议取决于此。本文将重点设计具体的 PSU 协议。

注释 2 在 PSU 的某些应用中，一方或双方可能只有一个数据库。由于根据定理 1，PSU 等价于多消息对称私有信息检索，且在这种情况下，单数据库多消息对称私有信息检索不可行 *[35],* ，因此，一方可以在 PSU 过程开始前，从另一方获取（提取）共享的服务器端公共随机性的一个随机子集，如 *[36]* 所示。这样一来，多消息对称私有信息检索便可行，进而 PSU 也可行。

注释 3 随着 PSI 在 *[42],* 中被推广到多方隐私保护集合交集（MP-PSI），PSU 也可以推广到 MP-PSU。与多方隐私保护集合交集类似，MP-PSU 需要在客户端之间额外分配公共随机性。为避免重复，我们略去 MP-PSU 的详细推导；不过，在下一小节中，我们将详细介绍一种特殊的 MP-PSU，其中一方

[2]这两个结论均基于参与方 P_2 是用户这一假设。作为另一种情况，如果将参与方 P_1 视为用户，我们只需在这两个陈述中将 Ω_1 替换为 Ω_2 。

没有输入。一个关键区别是，在可靠性验证阶段，对于多方私有集合交集问题，如果所有客户端都包含相同的元素，则需要使 *[42,* 式（56）*]* 中的和等于 0；而对于多方私有集合并集问题，如果所有客户端都不包含该元素，则该和应为 0。详见示例3。在 MP-PSU 中，如果所有参与方都拥有一个单数据库，我们可以像 *[36]* 中那样，利用从领导方预先获取的服务器端公共随机性，构造一个可达方案。此外，在客户端之间分配公共随机性时，我们利用非串通数据库的分布式特性，以及 *[43]* 中介绍的 RSPIR方法。

2.3 私有分布式联邦学习

我们考虑一个分布式联邦子模型学习问题，其中有一台服务器，包含 $N = 2$ 个互不串通的复制数据库[3], ，以及 C 个由服务器选定、参与一轮 FSL 流程的客户端；见图 2。按照惯例，每个客户端都与两个数据库分别建立直接、安全且经过认证的通信信道。服务器端存储的待学习完整模型由 K 个子模型组成，每个子模型包含 L个从有限域 $\mathbb{F}_q$中均匀选取的独立同分布符号。因此，服务器中的每个数据库都包含完整模型 $M_{[K]}$，且有：

$$H(M_k) = L, \quad \forall k \tag{24}$$

$$H(M_{[K]}) = H(M_1) + \cdots + H(M_K) = KL \tag{25}$$

这两个数据库还共享一定量的服务器端公共随机性 $\mathcal{R}_S$ ，客户端并不知道这些随机性。每个选定客户端都希望根据其本地训练数据更新一个或多个子模型。具体而言，对于 $i \in [C]$，第 i 个客户端希望更新索引集由随机变量 $\Gamma^{\langle i \rangle}$ 表示的子模型，其取值记为 $\gamma^{\langle i \rangle}$。对于 $i \in [C]$，随机变量 $Y^{\langle i \rangle} = \{Y_1^{\langle i \rangle}, Y_2^{\langle i \rangle}, \ldots, Y_K^{\langle i \rangle}\}$ 用于表示将 $\Gamma^{\langle i \rangle}$ 映射到字母表后（如 [41,42]所示）对应的关联向量。

我们参照 [11] 中开创性的 FSL 工作来表述 FSL 流程。首先，服务器中的每个单个数据库都需要计算选定客户端所需子模型索引集的并集 $\Gamma^{\langle 1 \rangle} \cup \Gamma^{\langle 2 \rangle} \cup \cdots \cup \Gamma^{\langle C \rangle}$ ，记为 Γ。这一阶段称为 FSL-PSU 阶段。由于服务器中的两个数据库无法直接相互通信，我们的解决方案是随机选择在线客户端作为中间节点，转发两个数据库接收到的信息，而不是要求每个客户端向两个数据库发送相同的答案。这种新方法的主要目标是降低总通信成本和所需通信时间。因此，我们将 C 个客户端分为两组：一组客户端的索引集

[3]我们从最简单的情形开始研究，即两个数据库的情形。稍作修改后，我们的可行方案适用于任意数量的数据库。然而，如何通过增加数据库数量来提升性能，仍需进一步研究。

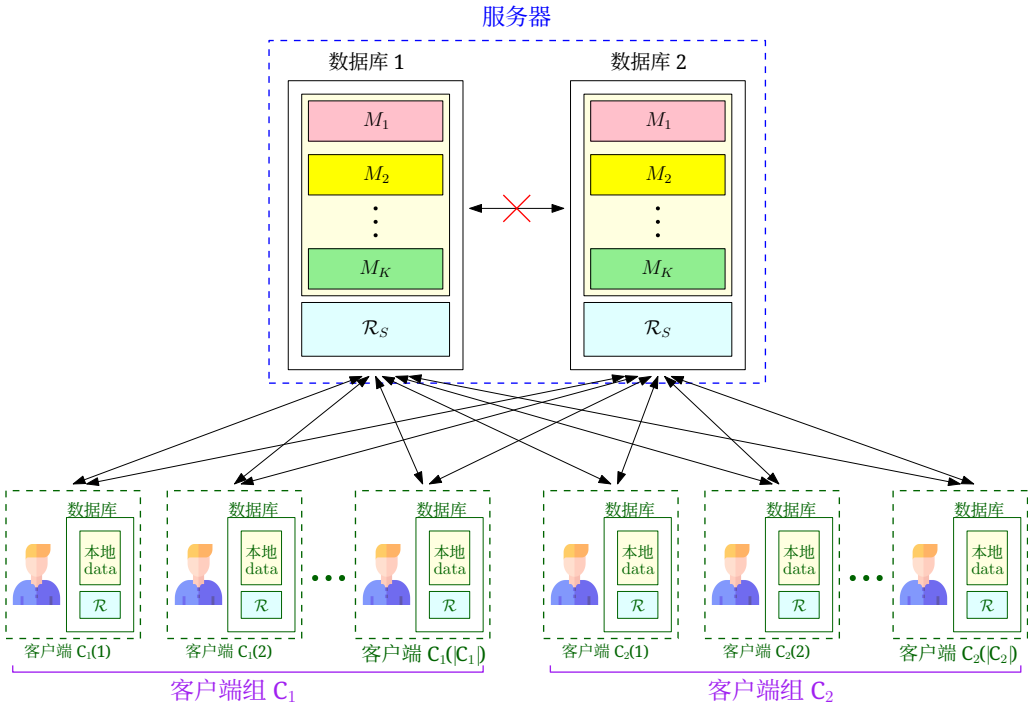
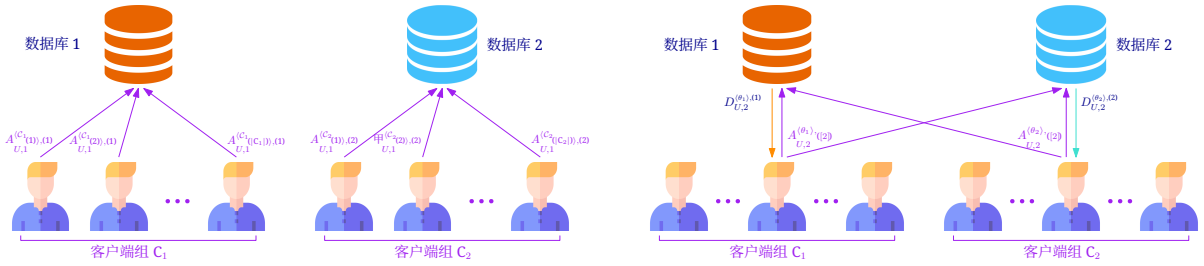


图 2：分布式联邦子模型学习（联邦子模型学习）系统模型。

记为 $\mathcal{C}_1 = \{\mathcal{C}_1(1)\ \mathcal{C}_1(2) \ldots, \mathcal{C}_1(|\mathcal{C}_1|)\}$，与数据库 1 关联；另一组客户端的索引集记为 $\mathcal{C}_2 = \{\mathcal{C}_2(1)\ \mathcal{C}_2(2) \ldots, \mathcal{C}_2(|\mathcal{C}_2|)\}$，与数据库 2 关联。一种可行的划分方法是依据每个客户端与两个数据库之间通信信道的带宽（或质量）。例如，如果客户端与数据库 1 之间信道的带宽（质量）高于其与数据库 2 之间信道的带宽（质量），则将该客户端划分至 $\mathcal{C}_1$。否则，该客户端归入 $\mathcal{C}_2$。注意，$\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ 和 $\mathcal{C}_1 \cup \mathcal{C}_2 = [C]$。示意图见图 2 和图 3。

鉴于两个随机客户端（每个客户端组各一个）被用于在数据库之间中继信息，FSL-PSU 阶段进一步分为两个步骤；见图 3。该信息由两个数据库分别从其关联的客户端组收集的答案生成。在第一步中，由于服务器本身不参与私有集合并集计算，因此每个客户端无需从数据库下载任何信息，即对于所有 $j \in [2]$，下载内容 $D^{\langle \mathcal{C}_j \rangle}_{(j)\,U,\,1}$ 均为空[4]。因此，该步骤中唯一的操作是让客户端将其精心设计的答案 $A^{\langle \mathcal{C}_j \rangle}_{(j)\,U,\,1}$ 发送至关联的数据库。在第二步中，对于所有 $j \in [2]$，数据库 j 借助其自身的

[4]在本文中，为清晰起见，我们用 $\langle\rangle$ 中的值表示客户端的索引，用 () 中的值表示数据库的索引。下文中，下载内容 D 或答案 A 的上标表示客户端与数据库通信期间的信息流。D 或 A 的第一个下标用于表示其处于 FSL-PSU 阶段还是 FSL-write 阶段（字母 U 表示并集，字母 W 表示写入），而第二个下标则用于表示该阶段中的步骤编号。特别地，此处“$D^{\langle \mathcal{C}_j \rangle}_{(j)\,U,\,1}$ 均为空”表示，在 FSL-PSU 阶段的第一步中，$\mathcal{C}_j$ 中的任意客户端与数据库 j 之间的通信始终为空。



(a) FSL-PSU 阶段步骤 1。 (b) FSL-PSU 阶段步骤 2。

图 3：FSL 系统模型中 FSL-PSU 阶段的数据流。

服务器端公共随机性处理从其关联客户端接收到的答案，然后生成的 $D^{\langle\theta_j\rangle}$, (j) U, 2 仅由一名随机选定的客户端下载，该客户端的索引为 θ_j，且属于其关联的客户端组 $\mathcal{C}_j$。最后，客户端 θ_j 根据接收到的下载内容，将相同的处理后的答案 $A^{\langle\theta_j\rangle}$, $([2])$ U, 2 转发至两个数据库；见图 3。

与 [45,46], 中所定义的传统多用户 PIR/SPIR 问题类似，FSL-PSU 阶段的约束包含三部分。首先，每个数据库 j 都应能够在两个 FSL-PSU 步骤内，利用收集到的所有答案 $\{A_{U1}^{\langle\mathcal{C}_j\rangle,}{}_{(j)}, A^{\langle\theta_{[2]}\rangle}{}_{,(j)\ U,2}\}$ 以及自身的服务器端公共随机性 $\mathcal{R}_S$，可靠地确定并集 Γ，这可表示为，

$$[\text{FSL-PSU reliability}] \quad H(\Gamma|A_{U,1}^{\langle\mathcal{C}_j\rangle,(j)}, A_{U,2}^{\langle\theta_{[2]}\rangle,(j)}, \mathcal{R}_S) = 0, \quad \forall j \tag{26}$$

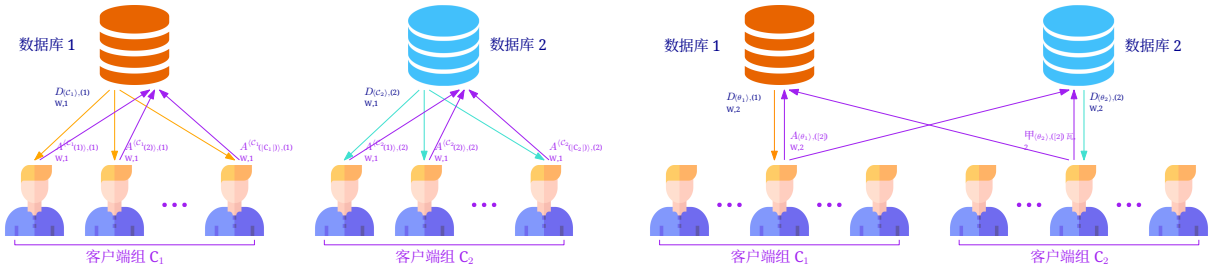
其次，数据库不应获知有关集合 $\Gamma^{\langle[C]\rangle}$ 或 $Y^{\langle[C]\rangle}$ 的任何额外信息，除并集 Γ 之外。注意，如果某个元素不在并集 Γ 中，则每个数据库都可得出结论：没有客户端包含该元素。否则，每个数据库会得知至少有一个客户端包含该元素。令 $Y_\Gamma^{\langle i\rangle} = \{Y_k^{\langle i\rangle} : k \in \Gamma\}$，我们定义一个新集合 $Y_\Gamma = Y_\Gamma^{\langle[C]\rangle}$，于是，

$$[\text{FSL-PSU privacy}] \quad I(Y_\Gamma; A_{U,1}^{\langle\mathcal{C}_j\rangle,(j)}, A_{U,2}^{\langle\theta_{[2]}\rangle,(j)}, \mathcal{R}_S | \sum_{i\in[C]} Y_k^{\langle i\rangle} > 0, \forall k \in \Gamma) = 0, \quad \forall j \tag{27}$$

第三，获取数据库 j 下载内容 $D^{\langle\theta_j\rangle}{}_{,(j)\ U,2}$ 的客户端 θ_j 不应了解到其他客户端的目标子模型索引。因此，我们有以下约束，

$$[\text{FSL-PSU inter-client privacy}] \quad I(Y_\Gamma^{\langle[C]\setminus\theta_j\rangle}; D_{U,2}^{\langle\theta_j\rangle,(j)}, Y^{\langle\theta_j\rangle}) = 0, \quad \forall j \tag{28}$$

有效的 FSL-PSU 阶段需满足 FSL-PSU 可靠性 (26)、FSL-PSU 隐私 (27) 和 FSL-PSU 客户端间隐私 (28)。FSL-PSU 阶段的效率以其中涉及的通信字符串的位数来衡量。因此，对于 FSL-PSU 阶段本身，我们希望尽可能减少答案 $\{A_{U1}^{\langle\mathcal{C}_1\rangle,}{}_{(1)}, , A^{\langle\mathcal{C}_2\rangle,}{}_{(2)\ U,1}, A^{\langle\theta_1\rangle,}{}_{([2])\ U,\ 2}, A^{\langle\theta_2\rangle,}{}_{([2])\ U,\ 2}\}$ 以及下载内容 $\{D_{U2}^{\langle\theta_1\rangle,}{}_{(1)}, , D^{\langle\theta_2\rangle,}{}_{(2)\ U,\ 2}\}$ 中的总位数。



(a) FSL-write 阶段步骤 1。 (b) FSL-write 阶段步骤 2。

图4：我们的 FSL 系统模型中 FSL-write 阶段的数据流。

FSL-PSU 阶段完成后，每个数据库都会获知 Γ，即待更新子模型索引的并集。接下来，我们进入 FSL-write 阶段，在此阶段，每个数据库将同步更新完整学习模型。FSL-write 阶段与 FSL-PSU 阶段类似，因此也像 FSL-PSU 阶段一样分为两个步骤。不同之处在于，第一步中，两个数据库都会向各自关联的客户端广播相同的子模型集合$M_\Gamma = \{M_k: k \in \Gamma\}$，之后每个客户端利用其本地数据训练所需的子模型集合 $M_{\Gamma^{\langle i \rangle}}$。因此，对于所有 $j \in [2]$，下载内容 $D^{\langle \mathcal{C}_j \rangle,(j)}{}_{W,1}$ 始终呈 M_Γ 的形式。随后，客户端将与子模型更新对应的、经过充分处理的答案 $A^{\langle \mathcal{C}_j \rangle,(j)}{}_{W,1}$ 发回各自关联的数据库。第二步中，对于所有 $j \in [2]$，数据库 j 通过不同的服务器端公共随机性处理其关联客户端的答案，随后生成的 $D^{\langle \theta_j \rangle,(j)}{}_{W,2}$ 再次由第 θ_j 个客户端下载。最后，第 θ_j 个客户端处理新收到的下载内容后，将相同的结果答案 $A^{\langle \theta_j \rangle,([2])}{}_{W,2}$ 转发给两个数据库；见图 4。

同样，FSL-write 阶段的约束由三部分组成。首先，每个数据库 j 都应能够根据收集的答案 $\{A^{\langle \mathcal{C}_j \rangle,}_{W1}{}_{(j)}, A^{\langle \theta_{[2]} \rangle}{}_{,(j)\,W,2}\}$，在两个 FSL-write 步骤内可靠地获得所有子模型更新的聚合；同时还需依据其自身当前的完整模型 $M_{[K]}$ 和自身的服务器端公共随机性 $\mathcal{R}_S$。当使用本地数据完成第 i个客户端的子模型训练后，对于所有 $k \in \Gamma^{\langle i \rangle}$，该客户端将为子模型$M_k = \{M_{k,1}, M_{k,2}, \ldots, M_{k,L}\}$中的每个符号生成增量 $\Delta_k = \{\Delta^{\langle i \rangle}_{k,1}, \Delta^{\langle i \rangle}_{k,2}, \ldots, \Delta^{\langle i \rangle}_{k,L}\}$。因此，对于第 k个子模型，设 Φ_k为其所需子模型索引集 $\Gamma^{\langle i \rangle}$ 包含 k的客户端集合，则其正确的更新后版本应为 $M'_k = \{M'_{k,1}, M'_{k,2}, \ldots, M'_{k,L}\} = \{M_{k,1} + \sum_{i \in \Phi_k} \Delta^{\langle i \rangle}_{k,1}, M_{k,2} + \sum_{i \in \Phi_k} \Delta^{\langle i \rangle}_{k,2}, \ldots, M_{k,L} + \sum_{i \in \Phi_k} \Delta^{\langle i \rangle}_{k,L}\}$。对于服务器中的每个数据库，正确的更新后子模型聚合应为 $M'_\Gamma = \{M'_k: k \in \Gamma\}$，因此第一项约束可表示为：

$$[\text{FSL-write reliability}] \quad H(M'_\Gamma | A^{\langle \mathcal{C}_j \rangle,(j)}_{W,1}, A^{\langle \theta_{[2]} \rangle,(j)}_{W,2}, M_{[K]}, \mathcal{R}_S) = 0, \quad \forall j \tag{29}$$

其次，任何数据库都不应获知各客户端所需子模型索引集的任何信息，也不应获知关于各客户端子模型增量的、超出更新后的子模型聚合所包含内容的任何其他信息。对于第 i 个客户端的子模型增量，令 $\Delta^{\langle i \rangle}_\Gamma = \{\Delta^{\langle i \rangle}_{k,l}: k \in$

Γ、$l \in [L]\}$，我们定义一个新集合 $\Delta_\Gamma = \Delta_\Gamma^{\langle [C] \rangle}$，那么，[5]

$$[\text{FSL-write privacy}]\ I(\Delta_\Gamma; A_{W,1}^{\langle \mathcal{C}_j \rangle,(j)}, A_{W,2}^{\langle \theta_{[2]} \rangle,(j)}, M_{[K]}, \mathcal{R}_S | \sum_{i \in \Phi_k} \Delta_{k,l}^{\langle i \rangle}, \forall k \in \Gamma, \forall l \in [L]) = 0,\ \forall j \quad (30)$$

第三，依据从数据库j获得的下载内容 $D^{\langle \theta_j \rangle,\ (j)\ W,\ 2}$，第 θ 个客户端不应获知其他客户端的目标子模型索引或子模型增量。因此，我们有以下约束，

$$[\text{FSL-write inter-client privacy}] \quad I(\Delta_\Gamma^{\langle [C] \setminus \theta_j \rangle}; D_{W,2}^{\langle \theta_j \rangle,(j)}, \Gamma^{\langle \theta_j \rangle}, \Delta_{\Gamma^{\langle \theta_j \rangle}}) = 0 \qquad (31)$$

有效的 FSL-write 阶段是指满足 FSL 写入可靠性（29）、FSL-write 隐私（30）和 FSL 写入客户端间隐私（31）的阶段。对于初始参数固定的任意特定 FSL 问题，从两个数据库向每个客户端发送子模型集合 M_Γ 的通信成本是一个常数。因此，FSL-write 阶段的效率也以答案 $\{A_{W1}^{\langle \mathcal{C}_1 \rangle,\ (1)\ ,\ ,\ } A^{\langle \mathcal{C}_2 \rangle,\ (2)\ W,\ 1}, A^{\langle \theta_1 \rangle,\ ([2])\ W,\ 2}, A^{\langle \theta_2 \rangle,\ ([2])\ W,\ 2}\}$ 以及下载内容 $\{D_{W2}^{\langle \theta_1 \rangle,\ (1)\ ,\ ,\ } D^{\langle \theta_2 \rangle,\ (2)\ W,\ 2}\}$中的总比特数来衡量，并且我们希望尽可能将其最小化。如果不考虑执行 FSL 所需的客户端侧公共随机数生成，则一个完整的 FSL 轮次由两个阶段组成：FSL-PSU 阶段和 FSL-write 阶段。我们的目标是尽可能减少这两个阶段交换的通信比特总数。此外，该 FSL 轮次可以迭代执行，直至满足预定义的终止准则，例如更新后的全局模型精度超过预设阈值，或达到预设的最大迭代次数。

3 主要结果

我们的主要结果是上述一种新的隐私保护 FSL 算法。以下定理给出了该算法的性能，衡量指标是整个过程的总通信成本，其中包括 FSL-PSU、FSL-write 以及在客户端生成所需公共随机性的成本。定理的证明见第 5.2 节和第 5.3 节。

定理 2：所提出的私有 FSL 方案在一轮中的总通信成本为 $\mathcal{O}(CK + C|\Gamma|L)$ 个 q 进制位，其中 C 是选定客户端的数量，K 是子模型总数

[5]通常，以下条件互信息中的第一项应为 Y_Γ, Δ_Γ，而非Δ_Γ。在 FSL-write 阶段，我们注意到，信息传输仅涉及关于 M_Γ 的子模型增量，与关联向量 $Y^{\langle [C] \rangle}$无关。这意味着，如果某个数据库除了从所有选定客户端获得的聚合增量之外一无所知，那么该数据库必然对关联向量 Y_Γ一无所知。因此，表达式 Δ_Γ 取代了 Y_Γ, Δ_Γ。这一观察也适用于 FSL-write 客户端间隐私约束 (31)，其中表达式 $\Delta_\Gamma^{\langle [C] \setminus \theta_j \rangle}$ 取代了 $Y_\Gamma^{\langle [C] \setminus \theta_j \rangle}$，$\Delta_\Gamma^{\langle [C] \setminus \theta_j \rangle}$。

，而 $|\Gamma|$ 是给定轮次中更新的子模型数量。这里，$\mathcal{O}(CK)$ 来自 FSL-PSU 阶段，而 $\mathcal{O}(C|\Gamma|L)$ 来自 FSL-write 阶段。

备注 4 定理的可实现性始于一种具有多个副本且彼此不串通的数据库的多消息对称私有信息检索。数据库中存储的数据未经编码且没有噪声。我们在统一的信息论框架下结合私有集合并与安全聚合，并提出一种新颖的私有 FSL 方案。我们利用数据库彼此不串通这一特点，在选定客户端之间实现简单的公共随机数生成与分发。

备注 5 我们提出的 FSL 实现了无条件信息论隐私。这不同于大多数以计算安全为重点的既有安全聚合研究，例如 *[3, 6, 11, 14, 44]*。这也不同于既有的私有读取、更新和写入（PRUW）研究 *[15, 17, 47–50]*，在这些研究中，尽管满足信息论安全性，但每个 FSL 轮次中一次只有一个客户端更新完整模型。我们提出的私有 FSL 方案能够抵御客户端掉线、客户端迟到以及数据库掉线。此外，FSL 流程中允许掉线的客户端数量不受限制。

备注 6 我们所提出的私有 FSL 的通信成本 $\mathcal{O}(CK + C|\Gamma|L)$优于现有文献中已知的最佳通信成本 *[3–6]*，，后者至少为$\mathcal{O}(CKL)$。在开创性的 FSL 工作 *[11]*, 中，PSU 阶段的通信成本为 $\mathcal{O}(C|\Gamma|)$，而整个 FSL 流程的通信成本为 $\mathcal{O}(C|\Gamma|L)$，但其隐私保证弱得多。尽管就 PSU 阶段而言，该通信成本略优于我们的通信成本，但 PSU *[11]* 会产生错误结果，而我们的 PSU 能够产生完全准确的结果。此外，*[11]*将 PSU 问题与后续的安全聚合问题分开考虑。我们注意到，在 *[11]*的电子商务推荐系统中，当每种商品均由一个独立的子模型表示时，子模型总数 K 会非常庞大。因此，鉴于完整学习模型的规模以及实践中客户端所需商品的一般平均数量，我们可以通过调整 K的大小来进一步优化通信成本，例如将相关商品合并到同一商品类别中。具体而言，当我们减小 K时，$|\Gamma|$ 与 L 的乘积很可能会增大，从而使 K 与$|\Gamma|L$ 处于同一数量级。因此，我们的方案在通信成本方面优于现有方案，并且还可以通过优化系统模型参数进一步改进。然而，很难找到一个公平的指标，将我们的通信成本与 *[15, 18]*中的通信成本进行比较。主要原因是，*[15, 18]*中的方案要求每次只有一个客户端更新一个子模型，并且严重依赖于数量足够多的数据库 N。也就是说，*[15, 18]* 中的方案至少需要 $N \geq 4$ 个数据库，无法与本文方案进行比较，因为本文方案中的数据库数量为 $N = 2$。如果遵循 $L \gg K$ 这一渐近假设，并令 C 取值为 1，我们唯一能够得出的结论是：这两种不同方案的通信成本都是子模型大小 L的线性函数。

备注 7 通常，计算机科学文献中的现有私有联邦学习方案依赖大量密码计算，而我们提出的 FSL 方案在客户端和服务器端均仅依赖有限域 $\mathbb{F}_q$中的简单加法和乘法计算。此外，由于实践中客户端间通信不稳定，且在我们的假设中不允许数据库间通信，我们的 FSL 方案仅依赖客户端与数据库间通信。为了缓解缺乏数据库间通信所带来的挑战，作为我们 FSL 方案中的一种新方法，我们利用随机客户端在服务器中的数据库之间转发所需信息。转发的信息来自各数据库从其关联客户端收集的答案，我们还进一步保护客户端之间的这一信息（客户端间隐私）。

备注 8 在实际实现中，对于每个客户端，在客户端与数据库间通信期间，上传速度通常远低于下载速度。与 *[3],* 中的经典安全聚合方案不同，我们的 FSL 流程进一步缩短了总通信时间，因为在每个阶段，几乎所有在线客户端都只向一个数据库发送一个答案。此外，在确定要连接到两个数据库的两个客户端组时，我们还可以根据每条客户端与数据库间通信信道的实际带宽/质量，进一步缩短总通信时间。

备注 9 通过刷新服务器端和客户端的公共随机性，所提出的私有 FSL 方案可在 FSL 流程的多个轮次中迭代使用。

4 私有分布式 FSL 模块示例

本节通过示例说明图 1 中各模块（方框）的功能。这些示例的复杂程度逐步提升：示例 1 考虑一个双方 PSU 场景，其中客户端参与方拥有多个数据库，而领导方拥有一个数据库。示例 2 考虑示例 1 的一个稍复杂版本，其中客户端参与方也只有一个数据库。在这种情况下，单数据库 SPIR 不可行，领导方需要获取客户端公共随机性，并将其用作领导方侧信息，如 [36] 所示。示例 3 将示例 2 推广到多方（MP）场景；具体而言，共有 5 个参与方，且每个参与方都只有一个数据库。示例 4 是示例 3 的扩展版本，其中领导方拥有两个数据库。该示例体现了所提出的私有 FSL 方案中的 FSL-PSU 阶段如何运作。最后，示例 5 展示了私有写入的工作方式。示例 4 中的 FSL-PSU 与示例 5 中的 FSL-write 共同构成了我们提出的私有 FSL 方案。

示例 1 双方 PSU；双数据库客户端；单数据库领导方：考虑一个全局字母表为 $\mathcal{A} = \{1, 2, 3, 4\}$ 的双方 PSU 问题。第一方 P_1 包含元素 1 和元素 2，即 $\mathcal{P}_1 = \{1, 2\}$。第二方 P_2 包含元素 1 和

元素 3，即 $\mathcal{P}_2 = \{1, 3\}$。为方便起见，每个参与方所拥有的元素总数是公开的。参与方希望共同计算各自元素集合的并集，同时不向对方透露任何其他信息。通过将各自的元素集合映射为相应的关联向量，双方构造的向量如下：

$$Party\ P_1: \quad \mathcal{P}_1 = \{1,2\} \quad \Rightarrow \quad X^{\langle 1\rangle} = [X_1^{\langle 1\rangle}\ X_2^{\langle 1\rangle}\ X_3^{\langle 1\rangle}\ X_4^{\langle 1\rangle}]^T = [1\ 1\ 0\ 0]^T \tag{32}$$

$$Party\ P_2: \quad \mathcal{P}_2 = \{1,3\} \quad \Rightarrow \quad X^{\langle 2\rangle} = [X_1^{\langle 2\rangle}\ X_2^{\langle 2\rangle}\ X_3^{\langle 2\rangle}\ X_4^{\langle 2\rangle}]^T = [1\ 0\ 1\ 0]^T \tag{33}$$

首先， P_2 （领导方）使用对称私有信息检索方法从 P_1 请求 $X_2^{\langle 1\rangle}$ 的值。 P_1（客户端参与方）拥有两个复制的非串通数据库。这两个数据库共享一个服务器端公共随机性符号 S_1 ，该符号从有限域$\mathbb{F}_2$ 中均匀选取，且 P_2不知道该符号。因此， P_1 为两个单个数据库生成如下形式的答案表，

$$A_U^{(1)}(1) = S_1, \qquad A_U^{(2)}(1) = X_1^{\langle 1\rangle} + S_1 \tag{34}$$

$$A_U^{(1)}(2) = X_1^{\langle 1\rangle} + X_2^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(2) = X_2^{\langle 1\rangle} + S_1 \tag{35}$$

$$A_U^{(1)}(3) = X_1^{\langle 1\rangle} + X_3^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(3) = X_3^{\langle 1\rangle} + S_1 \tag{36}$$

$$A_U^{(1)}(4) = X_1^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(4) = X_4^{\langle 1\rangle} + S_1 \tag{37}$$

$$A_U^{(1)}(5) = X_2^{\langle 1\rangle} + X_3^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(5) = X_1^{\langle 1\rangle} + X_2^{\langle 1\rangle} + X_3^{\langle 1\rangle} + S_1 \tag{38}$$

$$A_U^{(1)}(6) = X_2^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(6) = X_1^{\langle 1\rangle} + X_2^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1 \tag{39}$$

$$A_U^{(1)}(7) = X_3^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(7) = X_1^{\langle 1\rangle} + X_3^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1 \tag{40}$$

$$A_U^{(1)}(8) = X_1^{\langle 1\rangle} + X_2^{\langle 1\rangle} + X_3^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1, \qquad A_U^{(2)}(8) = X_2^{\langle 1\rangle} + X_3^{\langle 1\rangle} + X_4^{\langle 1\rangle} + S_1 \tag{41}$$

按照第2节中的记号， A 的上标表示 P_1的数据库索引，而 A 右侧的索引表示 P_2可以选择的潜在查询选项。

为了检索 $X_2^{\langle 1\rangle}$ ， P_2 为 P_1的第一个数据库随机选择一个查询选项，并为 P_1的第二个数据库选择与之配对的查询。例如， P_2 为 P_1 的第一个数据库选择 1，为 P_1的第二个数据库选择 2。收到查询符号 1 后， P_1 的第一个数据库以答案 $A_U^{(1)}(1) = S_1$作出响应。同时，收到查询符号 2 后， P_1 的第二个数据库以答案 $A_U^{(2)}(2) = X_2^{\langle 1\rangle} + S_1$ 作出响应。接下来， P_2 以相同的方式向 P_1 询问 $X_4^{\langle 1\rangle}$ 的值。

由于对 P_1中的每个数据库都有 8 种可能的查询，从 P_2到 P_1 的通信成本为 $3 + 3 = 6$ 比特；又由于 P_1 中的每个数据库都会返回一个比特的答案，从 P_1 到 P_2 的通信成本为 $1 + 1 = 2$ 比特。因此，通过这种多消息对称私人信息检索方法，学习 $X_2^{\langle 1\rangle}$和 $X_4^{\langle 1\rangle}$的总通信成本为 $2 \cdot (6+2) = 16$比特。获知 $X_2^{\langle 1\rangle}$ 和 $X_4^{\langle 1\rangle}$ 的值后， P_2 知道 P_1 拥有元素 2，但不拥有元素 4。结合自身拥有的元素， P_2 便能计算出并集，即 $\{1, 2, 3\}$。因此，私有集合并可靠性约束得以满足。关于两项隐私约束，由于用户

在 SPIR 问题中的隐私约束， P_1 中的每个单个数据库只能知道P_2 拥有两个元素，而无法获知这两个具体元素是什么。由于 SPIR 问题中的数据库隐私约束， P_2 只能知道 P_1拥有元素 2 而不拥有元素 4，无法额外获知 P_1 是否拥有元素 1、3。特别是， P_1 是否拥有元素 4，可以由 P_2 根据最终并集结果及其自身拥有的元素推断出来。因此， P_1 和 P_2 的隐私约束均得到保障。由此，这是一个有效的双方隐私并集方案。

示例 2 双方私有集合并；单数据库客户端；单数据库领导方：与示例 1 相比，设置中唯一的修改是参与方 P_1 现在只有一个数据库。参与方 P_1 还持有四个服务器端公共随机符号 S_1、 S_2、 S_3 和 S_4 ，它们均从有限域 $\mathbb{F}_2$ 中均匀选取。为了实现如 *[36],* 所示的可行单数据库 SPIR 方案，参与方 P_2 会预先获得一个随机服务器端公共随机符号。因此， P_1 为唯一的数据库生成如下答案表，

$$A_U^{(1)}(1) = \{X_1^{\langle 1\rangle} + S_1, X_2^{\langle 1\rangle} + S_2, X_3^{\langle 1\rangle} + S_3, X_4^{\langle 1\rangle} + S_4\} \tag{42}$$
$$A_U^{(1)}(2) = \{X_1^{\langle 1\rangle} + S_2, X_2^{\langle 1\rangle} + S_3, X_3^{\langle 1\rangle} + S_4, X_4^{\langle 1\rangle} + S_1\} \tag{43}$$
$$A_U^{(1)}(3) = \{X_1^{\langle 1\rangle} + S_3, X_2^{\langle 1\rangle} + S_4, X_3^{\langle 1\rangle} + S_1, X_4^{\langle 1\rangle} + S_2\} \tag{44}$$
$$A_U^{(1)}(4) = \{X_1^{\langle 1\rangle} + S_4, X_2^{\langle 1\rangle} + S_1, X_3^{\langle 1\rangle} + S_2, X_4^{\langle 1\rangle} + S_3\} \tag{45}$$

随后， P_2 选择与其预先获取的服务器端公共随机符号相匹配的查询选项。例如，如果其预先获取的符号是 S_1，为了检索 $X_2^{\langle 1\rangle}$ ， P_2 选择 4，然后将此查询符号发送给 P_1。属于 P_1 的数据库以答案 $A_U^{(1)}(4) = \{X_1^{\langle 1\rangle} + S_4, X_2^{\langle 1\rangle} + S_1, X_3^{\langle 1\rangle} + S_2, X_4^{\langle 1\rangle} + S_3\}$作出响应。同样， P_2 也以相同方式向 P_1 请求 $X_4^{\langle 1\rangle}$ 的值。由于向 P_1的数据库发起查询共有 4 种可能， P_2 到 P_1 的通信成本为 2 比特；由于 P_1 发回包含 4 个分量的答案， P_1 到 P_2 的通信成本为 4 比特。因此，通过此多消息对称私有信息检索方法，学习 $X_2^{\langle 1\rangle}$和 $X_4^{\langle 1\rangle}$的总通信成本为 $2\cdot(2+4) = 12$比特，且不计预先获取服务器端公共随机符号所产生的通信成本。与示例 1 中类似，可以验证此可实现方案满足隐私集合并集可靠性、 P_1 隐私约束和 P_2 隐私约束。

示例3 五方隐私集合并；每个客户端一个数据库；单数据库领导方：作为示例 1 和 2 的推广，在本示例中，我们考虑一种多方设置，同样使用全局字母表 $\mathcal{A} = \{1, 2, 3, 4\}$。这里，第一方 P_1 包含元素 1，即 $\mathcal{P}_1 = \{1\}$。第二方 P_2 包含元素 1 和元素 3，即 $\mathcal{P}_2 = \{1, 3\}$。第三方 P_3 包含元素 1 和元素 4，即 $\mathcal{P}_3 = \{1, 4\}$。第四方 P_4包含元素 1、元素 3 和元素 4，即 $\mathcal{P}_4 = \{1, 3, 4\}$。第五方 P_5 不包含任何元素，即 $\mathcal{P}_5 = \emptyset$。与之前一样，我们假设每个参与方中的元素总数为

公开知识。参与方按如下方式构造相应的关联向量 $X^{\langle[5]\rangle}$，

$$\text{参与方 } P_1\text{: } \mathcal{P}_1=\{1\} \quad \Rightarrow \quad X^{\langle1\rangle}=[X_1^{\langle1\rangle}\ X_2^{\langle1\rangle}\ X_3^{\langle1\rangle}\ X_4^{\langle1\rangle}]^T=[1\ 0\ 0\ 0]^T \tag{46}$$

$$\text{参与方 } P_2\text{: } \mathcal{P}_2=\{1,3\} \quad \Rightarrow \quad X^{\langle2\rangle}=[X_1^{\langle2\rangle}\ X_2^{\langle2\rangle}\ X_3^{\langle2\rangle}\ X_4^{\langle2\rangle}]^T=[1\ 0\ 1\ 0]^T \tag{47}$$

$$\text{参与方 } P_3\text{: } \mathcal{P}_3=\{1,4\} \quad \Rightarrow \quad X^{\langle3\rangle}=[X_1^{\langle3\rangle}\ X_2^{\langle3\rangle}\ X_3^{\langle3\rangle}\ X_4^{\langle3\rangle}]^T=[1\ 0\ 0\ 1]^T \tag{48}$$

$$\text{参与方 } P_4\text{: } \mathcal{P}_4=\{1,3,4\} \quad \Rightarrow \quad X^{\langle4\rangle}=[X_1^{\langle4\rangle}\ X_2^{\langle4\rangle}\ X_3^{\langle4\rangle}\ X_4^{\langle4\rangle}]^T=[1\ 0\ 1\ 1]^T \tag{49}$$

$$\text{参与方 } P_5\text{: } \mathcal{P}_5=\emptyset \quad \Rightarrow \quad X^{\langle5\rangle}=[X_1^{\langle5\rangle}\ X_2^{\langle5\rangle}\ X_3^{\langle5\rangle}\ X_4^{\langle5\rangle}]^T=[0\ 0\ 0\ 0]^T \tag{50}$$

以 *[42]* 中的多方隐私保护集合交集可实现方案为参考，我们选择参与方 P_5 作为领导方，并将其余参与方作为客户端方，因为参与方 P_5 在全局上已知为空方。因此，无需 P_5 向其余参与方发送任何查询，前两个示例中使用的服务器端公共随机性也不再需要。在本示例中，所有参与方都拥有单数据库。除各自的关联向量外，每个客户端参与方还持有相同的公共随机符号[6] $\{u_\alpha^{\langle[4]\rangle}$: 来自有限域 $\mathbb{F}_5$ 的 $\alpha\in[4]\}$，以及相同的、在 $\{1,2,3,4\}$ 上均匀分布的全局公共随机符号 c 。此外， $\{u_\alpha^{\langle[4]\rangle}$: $\alpha\in[4]\}$ 满足其总和$\sum_{i\in[4]}u_\alpha^{\langle i\rangle}$对于所有 $\alpha\in[4]$始终等于 0。来自客户端参与方的答案为:

$$A_U^{\langle1\rangle}=\{c(X_1^{\langle1\rangle}+u_1^{\langle1\rangle}),c(X_2^{\langle1\rangle}+u_2^{\langle1\rangle}),c(X_3^{\langle1\rangle}+u_3^{\langle1\rangle}),c(X_4^{\langle1\rangle}+u_4^{\langle1\rangle})\} \tag{51}$$

$$A_U^{\langle2\rangle}=\{c(X_1^{\langle2\rangle}+u_1^{\langle2\rangle}),c(X_2^{\langle2\rangle}+u_2^{\langle2\rangle}),c(X_3^{\langle2\rangle}+u_3^{\langle2\rangle}),c(X_4^{\langle2\rangle}+u_4^{\langle2\rangle})\} \tag{52}$$

$$A_U^{\langle3\rangle}=\{c(X_1^{\langle3\rangle}+u_1^{\langle3\rangle}),c(X_2^{\langle3\rangle}+u_2^{\langle3\rangle}),c(X_3^{\langle3\rangle}+u_3^{\langle3\rangle}),c(X_4^{\langle3\rangle}+u_4^{\langle3\rangle})\} \tag{53}$$

$$A_U^{\langle4\rangle}=\{c(X_1^{\langle4\rangle}+u_1^{\langle4\rangle}),c(X_2^{\langle4\rangle}+u_2^{\langle4\rangle}),c(X_3^{\langle4\rangle}+u_3^{\langle4\rangle}),c(X_4^{\langle4\rangle}+u_4^{\langle4\rangle})\} \tag{54}$$

*Reg*关于可靠性：领导方 P_5 计算以下表达式，

$$\begin{aligned}\textit{Element } 1: \quad & c(X_1^{\langle1\rangle}+u_1^{\langle1\rangle})+c(X_1^{\langle2\rangle}+u_1^{\langle2\rangle})+c(X_1^{\langle3\rangle}+u_1^{\langle3\rangle})+c(X_1^{\langle4\rangle}+u_1^{\langle4\rangle})\\ & =c(\sum_{i\in[4]}X_1^{\langle i\rangle}+\sum_{i\in[4]}u_1^{\langle i\rangle})=c(\sum_{i\in[4]}X_1^{\langle i\rangle})=c\cdot4\neq0\end{aligned} \tag{55}$$

$$\begin{aligned}\textit{Element } 2: \quad & c(X_2^{\langle1\rangle}+u_2^{\langle1\rangle})+c(X_2^{\langle2\rangle}+u_2^{\langle2\rangle})+c(X_2^{\langle3\rangle}+u_2^{\langle3\rangle})+c(X_2^{\langle4\rangle}+u_2^{\langle4\rangle})\\ & =c(\sum_{i\in[4]}X_2^{\langle i\rangle}+\sum_{i\in[4]}u_2^{\langle i\rangle})=c(\sum_{i\in[4]}X_2^{\langle i\rangle})=c\cdot0=0\end{aligned} \tag{56}$$

$$\begin{aligned}\textit{Element } 3: \quad & c(X_3^{\langle1\rangle}+u_3^{\langle1\rangle})+c(X_3^{\langle2\rangle}+u_3^{\langle2\rangle})+c(X_3^{\langle3\rangle}+u_3^{\langle3\rangle})+c(X_3^{\langle4\rangle}+u_3^{\langle4\rangle})\\ & =c(\sum_{i\in[4]}X_3^{\langle i\rangle}+\sum_{i\in[4]}u_3^{\langle i\rangle})=c(\sum_{i\in[4]}X_3^{\langle i\rangle})=c\cdot2\neq0\end{aligned} \tag{57}$$

*El*式 4: $c(X_4^{\langle1\rangle}+u_4^{\langle1\rangle})+c(X_4^{\langle2\rangle}+u_4^{\langle2\rangle})+c(X_4^{\langle3\rangle}+u_4^{\langle3\rangle})+c(X_4^{\langle4\rangle}+u_4^{\langle4\rangle})$

[6]此 MP-PSU 实现中的公共随机符号 $u_\alpha^{\langle i\rangle}$与 MP-PSI 论文 [42]中的公共随机符号 $t_{i,\ j}$ 作用完全相同。后续回写实现中的公共随机符号 $w_\alpha^{\langle i\rangle}$也是如此。

$$= c(\sum_{i\in[4]} X_4^{\langle i\rangle} + \sum_{i\in[4]} u_4^{\langle i\rangle}) = c(\sum_{i\in[4]} X_4^{\langle i\rangle}) = c\cdot 2 \neq 0 \tag{58}$$

因此， P_5 得出结论， 并集 $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{P}_4$ 为 $\{1,3,4\}$，因为第一、第三和第四个表达式均不等于 0。

关于隐私：领导方的隐私约束显然得到满足，因为领导方为空，且没有向客户端发送任何查询。公共随机符号 $\{u_\alpha^{\langle[4]\rangle}\colon\ \alpha\in[4]\}$ 和 c保护了客户端的隐私。首先，关联向量 $\{X_\alpha^{\langle[4]\rangle}\colon\ \alpha\in[4]\}$ 各个分量的值通过添加的随机性符号 $\{u_\alpha^{\langle[4]\rangle}\colon\ \alpha\in[4]\}$对 P_5 保密。当来自客户端的分量相加时，这些耦合（即相关）随机变量会消失，因为对于所有 $\alpha\in[4]$, $\sum_{i\in[4]} u_\alpha^{\langle i\rangle}$始终为 0。最后，全局公共随机符号 c 保护了所有 α对应的和 $\sum_i X_\alpha^{\langle i\rangle}$的实际值。也就是说，领导方 P_5 只能获知这些和是否为零，除此之外一无所知。因此，这是一个满足可靠性和隐私要求的有效方案。

示例 4 五方隐私集合并；每个客户端一个数据库；双数据库领导方：相较于示例3中的 MP-PSU 配置，我们只更改领导方 P_5 中数据库的数量，现在其中包含两个副本数据库，且二者互不串通。由于这两个数据库不会直接相互通信，一个直接的方法是让每个客户端将其在 (51)-(54) 中给出的答案发送给领导方的两个数据库。这样，每个数据库都能单独获知并集，同时仍满足隐私约束。这里，我们提出另一种方法：利用两个随机客户端方作为中介，在领导方的两个非串通数据库之间转发信息，这样客户端就无需将重复的答案发送给 P_5 中的两个数据库。为此，每个客户端参与方还持有另一组公共随机符号副本 $\{u_\alpha\colon\ \alpha\in[4]\}$，这些符号均根据现有公共随机符号 $\{u_\alpha^{\langle[4]\rangle}\colon \alpha\in[4]\}$ 从 $\mathbb{F}_5$ 中均匀选取。由于 P_5 不包含任何元素，因此其他参与方一开始无需从 P_5 下载任何信息，即 $D^{\langle\mathcal{C}_1\rangle,\ (1)\,U,\ 1}$ 和 $D^{\langle\mathcal{C}_2\rangle,\ (2)\,U,\ 1}$ 均为空。此时，令客户端参与方 P_1 和 P_2 组成第一组。他们将各自的答案 $A^{\langle1\rangle,\ U\,1}{}^{(1)}$，和 $A^{\langle2\rangle,\ U\,1}{}^{(1)}$，（如 (51)-(52) 所示）发送给 P_5 的第一个数据库，因为他们与数据库 1 相关联。该数据库生成响应 $D^{\langle2\rangle,\ U\,2}{}^{(1)}$，，客户端 2 通过逐元素相加其收到的答案，并附加从有限域 $\mathbb{F}_5$ 中均匀选取的领导方公共随机符号，来下载该响应，

$$\begin{aligned} D_{U,2}^{\langle2\rangle,(1)} = \{&c(X_1^{\langle1\rangle} + X_1^{\langle2\rangle} + u_1^{\langle1\rangle} + u_1^{\langle2\rangle}) + S_1, c(X_2^{\langle1\rangle} + X_2^{\langle2\rangle} + u_2^{\langle1\rangle} + u_2^{\langle2\rangle}) + S_2,\\ &c(X_3^{\langle1\rangle} + X_3^{\langle2\rangle} + u_3^{\langle1\rangle} + u_3^{\langle2\rangle}) + S_3, c(X_4^{\langle1\rangle} + X_4^{\langle2\rangle} + u_4^{\langle1\rangle} + u_4^{\langle2\rangle}) + S_4\} \end{aligned} \tag{59}$$

随后，该数据库将此响应发送回 P_2。添加额外的公共随机性后

收到的响应中， P_2 将以下答案转发给 P_5 中的两个数据库,

$$A_{U,2}^{\langle 2\rangle,([2])} = \{c(X_1^{\langle 1\rangle} + X_1^{\langle 2\rangle} + u_1^{\langle 1\rangle} + u_1^{\langle 2\rangle}) + u_1 + S_1, c(X_2^{\langle 1\rangle} + X_2^{\langle 2\rangle} + u_2^{\langle 1\rangle} + u_2^{\langle 2\rangle}) + u_2 + S_2, \\ c(X_3^{\langle 1\rangle} + X_3^{\langle 2\rangle} + u_3^{\langle 1\rangle} + u_3^{\langle 2\rangle}) + u_3 + S_3, c(X_4^{\langle 1\rangle} + X_4^{\langle 2\rangle} + u_4^{\langle 1\rangle} + u_4^{\langle 2\rangle}) + u_4 + S_4\} \tag{60}$$

与此同时，组成第二组的客户端参与方 P_3 和 P_4将各自的答案 $A^{\langle 3\rangle,\ U1}{}^{(2)}$ ，和 $A^{\langle 4\rangle,\ U1}{}^{(2)}$ ，（如式（53）–（54）所示）发送到 P_5的第二个数据库。类似地，该数据库生成响应 $D^{\langle 3\rangle,\ U2}{}^{(2)}$ ，供客户端 3 下载（如下所示），并将其发送回 P_3,

$$D_{U,2}^{\langle 3\rangle,(2)} = \{c(X_1^{\langle 3\rangle} + X_1^{\langle 4\rangle} + u_1^{\langle 3\rangle} + u_1^{\langle 4\rangle}) - S_1, c(X_2^{\langle 3\rangle} + X_2^{\langle 4\rangle} + u_2^{\langle 3\rangle} + u_2^{\langle 4\rangle}) - S_2, \\ c(X_3^{\langle 3\rangle} + X_3^{\langle 4\rangle} + u_3^{\langle 3\rangle} + u_3^{\langle 4\rangle}) - S_3, c(X_4^{\langle 3\rangle} + X_4^{\langle 4\rangle} + u_4^{\langle 3\rangle} + u_4^{\langle 4\rangle}) - S_4\} \tag{61}$$

然后, P_3 将以下进一步处理后的答案转发给 P_5中的两个数据库 ,

$$A_{U,2}^{\langle 3\rangle,([2])} = \{c(X_1^{\langle 3\rangle} + X_1^{\langle 4\rangle} + u_1^{\langle 3\rangle} + u_1^{\langle 4\rangle}) - u_1 - S_1, c(X_2^{\langle 3\rangle} + X_2^{\langle 4\rangle} + u_2^{\langle 3\rangle} + u_2^{\langle 4\rangle}) - u_2 - S_2, \\ c(X_3^{\langle 3\rangle} + X_3^{\langle 4\rangle} + u_3^{\langle 3\rangle} + u_3^{\langle 4\rangle}) - u_3 - S_3, c(X_4^{\langle 3\rangle} + X_4^{\langle 4\rangle} + u_4^{\langle 3\rangle} + u_4^{\langle 4\rangle}) - u_4 - S_4\} \tag{62}$$

收集第二轮通信中的答案后， P_5 中的每个单个数据库 j 通过逐元素相加 $A^{\langle 2\rangle,\ U2}{}^{(j)}$ ，和 $A^{\langle 3\rangle,\ U2}{}^{(j)}$ ，， 找到所需的子模型并集

$$A_{U,2}^{\langle 2\rangle,(j)} + A_{U,2}^{\langle 3\rangle,(j)} = \left\{c(\sum_{i\in[4]} X_1^{\langle i\rangle}), c(\sum_{i\in[4]} X_2^{\langle i\rangle}), c(\sum_{i\in[4]} X_3^{\langle i\rangle}), c(\sum_{i\in[4]} X_4^{\langle i\rangle})\right\}, \quad \forall j \tag{63}$$

关于可靠性：领导方 P_5 中的 MP-PSU可靠性继承自示例3中的 MP-PSU可靠性分析。具体而言， P_5 中的每个单个数据库都可以针对字母表中的每个元素进行与式（55）–（58）所示相同的分析，从而得出并集 $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{P}_4$。此外，如有需要， P_5 可以将此并集结果发送给任意客户端参与方。

关于隐私：客户端方 P_1 和 P_4 的隐私分析是显而易见的，因为它们都没有从其余参与方接收到任何信息。对于客户端参与方 P_2，由于附加了领导方的公共随机性 $\{S_\alpha, \alpha \in [4]\}$，该参与方无法从其唯一接收到的信息 $D^{\langle 2\rangle,\ U2}{}^{(1)}$ ，中获知其余参与方的关联向量符号。此分析同样适用于客户端参与方 P_3。对于领导方 P_5，显然，第一数据库中接收到的信息 $\{A_{U,1}^{\langle 1\rangle,(1)}, A_{U,1}^{\langle 2\rangle,(1)}, A_{U,2}^{\langle 2\rangle,(1)}, A_{U,2}^{\langle 3\rangle,(1)}\}$ 和第二数据库中接收到的信息 $\{A_{U,1}^{\langle 3\rangle,(2)}, A_{U,1}^{\langle 4\rangle,(2)}, A_{U,2}^{\langle 2\rangle,(2)}, A_{U,2}^{\langle 3\rangle,(2)}\}$ 分别包含的客户端参与方关联向量符号信息，都少于示例3中 P_5 收到的答案集 $A_U^{\langle [4]\rangle}$ 。因此，领导方 P_5 只能获知并集，除此之外一无所知。由此，这是一个有效的 MP-PSU方案。

接下来，我们考虑实际实现中常见的情形。

首先，在 MP-PSU 过程中，一个或多个客户端参与方可能会掉线。例如，P_1 可能会与 P_5失去连接，此时，P_5 的第一个数据库将只收到来自 P_2的答案。原始方法生成的下载内容现在变为，

$$\begin{aligned} D'^{\langle 2\rangle,(1)}_{U,2} = \{&c(X_1^{\langle 2\rangle} + u_1^{\langle 2\rangle}) + S_1, c(X_2^{\langle 2\rangle} + u_2^{\langle 2\rangle}) + S_2, \\ &c(X_3^{\langle 2\rangle} + u_3^{\langle 2\rangle}) + S_3, c(X_4^{\langle 2\rangle} + u_4^{\langle 2\rangle}) + S_4\} \end{aligned} \tag{64}$$

很容易看出，这两个下载内容中的公共随机符号 $u_\alpha^{\langle [4]\rangle}$无法像之前那样完全抵消。不过请注意，$P_2$ 拥有因 P_1 掉线而缺失的公共随机符号。因此，只要 P_2 从数据库 1 获知 P_1 已掉线，就可以自行添加所需的公共随机性。因此，第二步中调整后的答案 $A'^{\langle 2\rangle, {}^{([2])}_{U\,2}}$，如下，并将通过 P_5发送回两个数据库，

$$\begin{aligned} A'^{\langle 2\rangle,([2])}_{U,2} = \{&c(X_1^{\langle 2\rangle} + u_1^{\langle 1\rangle} + u_1^{\langle 2\rangle}) + u_1 + S_1, c(X_2^{\langle 2\rangle} + u_2^{\langle 1\rangle} + u_2^{\langle 2\rangle}) + u_2 + S_2, \\ &c(X_3^{\langle 2\rangle} + u_3^{\langle 1\rangle} + u_3^{\langle 2\rangle}) + u_3 + S_3, c(X_4^{\langle 2\rangle} + u_4^{\langle 1\rangle} + u_4^{\langle 2\rangle}) + u_4 + S_4\} \end{aligned} \tag{65}$$

此外，如果 P_2 与 P_5失去连接，那么第一个客户端组中仍然活跃的客户端参与方 P_1 将充当路由器。由于第二个客户端组中没有任何参与方掉线，下载内容 $D^{\langle 3\rangle, {}^{(2)}_{U\,2}}$，保持不变，

$$\begin{aligned} D^{\langle 3\rangle,(2)}_{U,2} = \{&c(X_1^{\langle 3\rangle} + X_1^{\langle 4\rangle} + u_1^{\langle 3\rangle} + u_1^{\langle 4\rangle}) - S_1, c(X_2^{\langle 3\rangle} + X_2^{\langle 4\rangle} + u_2^{\langle 3\rangle} + u_2^{\langle 4\rangle}) - S_2, \\ &c(X_3^{\langle 3\rangle} + X_3^{\langle 4\rangle} + u_3^{\langle 3\rangle} + u_3^{\langle 4\rangle}) - S_3, c(X_4^{\langle 3\rangle} + X_4^{\langle 4\rangle} + u_4^{\langle 3\rangle} + u_4^{\langle 4\rangle}) - S_4\} \end{aligned} \tag{66}$$

P_3 转发并由 P_5 中的数据库接收的结果保持不变，

$$\begin{aligned} A^{\langle 3\rangle,([2])}_{U,2} = \{&c(X_1^{\langle 3\rangle} + X_1^{\langle 4\rangle} + u_1^{\langle 3\rangle} + u_1^{\langle 4\rangle}) - u_1 - S_1, c(X_2^{\langle 3\rangle} + X_2^{\langle 4\rangle} + u_2^{\langle 3\rangle} + u_2^{\langle 4\rangle}) - u_2 - S_2, \\ &c(X_3^{\langle 3\rangle} + X_3^{\langle 4\rangle} + u_3^{\langle 3\rangle} + u_3^{\langle 4\rangle}) - u_3 - S_3, c(X_4^{\langle 3\rangle} + X_4^{\langle 4\rangle} + u_4^{\langle 3\rangle} + u_4^{\langle 4\rangle}) - u_4 - S_4\} \end{aligned} \tag{67}$$

现在可以验证，P_5 中的两个数据库无需 P_1参与即可确定并集 $\mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{P}_4$。

其次，由 P_1 生成的答案 $A^{\langle 1\rangle, {}^{(1)}_{U\,1}}$，可能会非常迟才到达数据库 1 的 P_5，以至于数据库 1 可能会认为 P_1 已掉线。在这种情况下，我们的 MP-PSU 仍能保证隐私。查看 P_5 的数据库 1 中收到的信息 $\{A^{\langle 1\rangle,(1)}_{U,1}, A^{\langle 2\rangle,(1)}_{U,1}, A'^{\langle 2\rangle,(1)}_{U,2}, A^{\langle 3\rangle,(1)}_{U,2}\}$，由于存在额外的公共随机符号 $\{u_\alpha\colon\ \alpha \in [4]\}$，关联向量 $X^{\langle 1\rangle}$ 的任何信息都不会泄露。此外，为避免进一步泄露 $X^{\langle 1\rangle}$，P_5 绝不会将这个迟到的答案 $A^{\langle 1\rangle, {}^{(1)}_{U\,1}}$，传输给任何其他客户端方。此处使用额外公共随机性 u_α 的方式类似于双重掩码思想

在 *[3]* 中解决这个迟到问题，但方式非常简单。

第三，在 P_5 中，两个数据库之一也可能在实现过程中掉线。例如，如果数据库 2 掉线，数据库 1 仍可在 P_5 中从 P_1 和 P_2收到相同的答案 $\{A_{U,1}^{\langle 1\rangle,(1)}, A_{U,1}^{\langle 2\rangle,(1)}, A_{U,2}^{\langle 2\rangle,(1)}\}$，但无法像往常一样从另一个客户端组收到 $\{A_{U,2}^{\langle 3\rangle,(1)}\}$。相应的解决办法是，存活的数据库再进行一轮通信，通过 P_2 向 $\alpha \in [4]\}$ 请求 $\{c\ (u_\alpha^{\langle 3\rangle} + u_\alpha^{\langle 4\rangle})$ 的取值。由此很容易看出，P_5中的第一个数据库可以推导出与第一个客户端组相关的并集 $\mathcal{P}_1 \cup \mathcal{P}_2$。

示例 5 五方隐私集合并集；每个客户端一个数据库；双数据库领导方；结合 FSL-write：考虑一个分布式联邦子模型学习问题，其中服务器由两个副本数据库组成，这两个数据库彼此不串通；本轮联邦子模型学习流程中有四个选定客户端。每个单独的数据库存储 4 个相互独立的子模型，每个子模型包含 2 个从足够大的有限域 $\mathbb{F}_q$中均匀选取的独立同分布符号 $q \geq 5$，即$M_1 = [M_{1,1}, M_{1,2}]$、$M_2 = [M_{2,1}, M_{2,2}]$、$M_3 = [M_{3,1}, M_{3,2}]$、$M_4 = [M_{4,1}, M_{4,2}]$，以及一些所需的服务器端公共随机符号。根据客户端各自的本地训练数据，客户端 1 可用于更新子模型 1，客户端 2 可用于更新子模型 1、3，客户端 3 可用于更新子模型 1、4，客户端 4 可用于更新子模型 1、3、4，即 $\Gamma^{\langle 1\rangle} = \{1\}$、$\Gamma^{\langle 2\rangle} = \{1, 3\}$、$\Gamma^{\langle 3\rangle} = \{1, 4\}$、$\Gamma^{\langle 4\rangle} = \{1, 3, 4\}$。服务器中的两个数据库都可以通过安全认证信道与每个客户端通信。我们进一步假设，对于客户端 1、2，连接数据库 1 的信道带宽高于连接数据库 2 的信道；对于客户端 3、4 则相反。因此，FSL-PSU 阶段完全按照示例 4 中 MP-PSU 的方式执行，该阶段完成后，服务器中的每个数据库都会获知所需的子模型并集 $\Gamma = \{1, 3, 4\}$。

由于 FSL-PSU 阶段和 FSL-write 阶段的表达形式相似，我们再次采用示例 4 中的思路来执行 FSL-write 阶段。数据库 1 将子模型 1、3、4 发送给客户端 1 和客户端 2，而数据库 2 将子模型 1、3、4 发送给客户端 3 和客户端 4，即，下载内容 $D^{\langle 1}$、$^{2\rangle\,(1)}$、、和 $D^{\langle 3}$、$^{4\rangle\,(2)}$、、均为 $\{M_1, M_3, M_4\}$。从服务器收到所需的子模型后，客户端 1 ^{W_1}在完成本地训练后^{W_1}为子模型 1 生成增量$\{\Delta_{1,1}^{\langle 1\rangle}, \Delta_{1,2}^{\langle 1\rangle}\}$，客户端 2 为子模型 1、3 生成增量 $\{\Delta_{1,1}^{\langle 2\rangle}, \Delta_{1,2}^{\langle 2\rangle}, \Delta_{3,1}^{\langle 2\rangle}, \Delta_{3,2}^{\langle 2\rangle}\}$，客户端 3 为子模型 1、4 生成增量 $\{\Delta_{1,1}^{\langle 3\rangle}, \Delta_{1,2}^{\langle 3\rangle}, \Delta_{4,1}^{\langle 3\rangle}, \Delta_{4,2}^{\langle 3\rangle}\}$，客户端 4 为子模型 1、3、4 生成增量 $\{\Delta_{1,1}^{\langle 4\rangle}, \Delta_{1,2}^{\langle 4\rangle}, \Delta_{3,1}^{\langle 4\rangle}, \Delta_{3,2}^{\langle 4\rangle}, \Delta_{4,1}^{\langle 4\rangle}, \Delta_{4,2}^{\langle 4\rangle}\}$。此外，我们假设每个客户端都已从有限域 $\mathbb{F}_q$ 中获得两组公共随机符号 $\{w_{k,l}^{\langle \Gamma\rangle}:\ k \in [4]$、$l \in [2]\}$ 和 $\{w_{k,l}:\ k \in [4]$、$l \in [2]\}$，与前面的示例一样。对于所有 $k \in \Gamma$ 和所有 $l \in [2]$，总和 $\sum_{i\in[4]} w_{k\,l}^{\langle i\rangle}$、始终等于 0。因此，客户端 1 和客户端 2 发送到服务器中数据库 1 的答案如下，

$$A_{W,1}^{\langle 1\rangle,(1)} = \{\Delta_{1,1}^{\langle 1\rangle} + w_{1,1}^{\langle 1\rangle}, \Delta_{1,2}^{\langle 1\rangle} + w_{1,2}^{\langle 1\rangle}, w_{3,1}^{\langle 1\rangle}, w_{3,2}^{\langle 1\rangle}, w_{4,1}^{\langle 1\rangle}, w_{4,2}^{\langle 1\rangle}\} \tag{68}$$

$$A_{W,1}^{\langle 2\rangle,(1)} = \{\Delta_{1,1}^{\langle 2\rangle} + w_{1,1}^{\langle 2\rangle}, \Delta_{1,2}^{\langle 2\rangle} + w_{1,2}^{\langle 2\rangle}, \Delta_{3,1}^{\langle 2\rangle} + w_{3,1}^{\langle 2\rangle}, \Delta_{3,2}^{\langle 2\rangle} + w_{3,2}^{\langle 2\rangle}, w_{4,1}^{\langle 2\rangle}, w_{4,2}^{\langle 2\rangle}\} \tag{69}$$

收集客户端 1、2 的答案后，数据库 1 借助自身的服务器端公共随机符号进行逐元素求和，并向其关联客户端组中的客户端 2 发送以下响应，

$$\begin{aligned} D_{W,2}^{\langle 2\rangle,(1)} = \{&\Delta_{1,1}^{\langle 1\rangle} + \Delta_{1,1}^{\langle 2\rangle} + w_{1,1}^{\langle 1\rangle} + w_{1,1}^{\langle 2\rangle} + S_{1,1}, \Delta_{1,2}^{\langle 1\rangle} + \Delta_{1,2}^{\langle 2\rangle} + w_{1,2}^{\langle 1\rangle} + w_{1,2}^{\langle 2\rangle} + S_{1,2}, \\ &\Delta_{3,1}^{\langle 2\rangle} + w_{3,1}^{\langle 1\rangle} + w_{3,1}^{\langle 2\rangle} + S_{3,1}, \Delta_{3,2}^{\langle 2\rangle} + w_{3,2}^{\langle 1\rangle} + w_{3,2}^{\langle 2\rangle} + S_{3,2}, \\ &w_{4,1}^{\langle 1\rangle} + w_{4,1}^{\langle 2\rangle} + S_{4,1}, w_{4,2}^{\langle 1\rangle} + w_{4,2}^{\langle 2\rangle} + S_{4,2}\} \end{aligned} \tag{70}$$

随后，客户端 2 通过添加额外的公共随机性来处理收到的响应，然后将以下答案转发给服务器中的两个数据库，

$$\begin{aligned} A_{W,2}^{\langle 2\rangle,([2])} = \{&\Delta_{1,1}^{\langle 1\rangle} + \Delta_{1,1}^{\langle 2\rangle} + w_{1,1}^{\langle 1\rangle} + w_{1,1}^{\langle 2\rangle} + w_{1,1} + S_{1,1}, \Delta_{1,2}^{\langle 1\rangle} + \Delta_{1,2}^{\langle 2\rangle} + w_{1,2}^{\langle 1\rangle} + w_{1,2}^{\langle 2\rangle} + w_{1,2} + S_{1,2}, \\ &\Delta_{3,1}^{\langle 2\rangle} + w_{3,1}^{\langle 1\rangle} + w_{3,1}^{\langle 2\rangle} + w_{3,1} + S_{3,1}, \Delta_{3,2}^{\langle 2\rangle} + w_{3,2}^{\langle 1\rangle} + w_{3,2}^{\langle 2\rangle} + w_{3,2} + S_{3,2}, \\ &w_{4,1}^{\langle 1\rangle} + w_{4,1}^{\langle 2\rangle} + w_{4,1} + S_{4,1}, w_{4,2}^{\langle 1\rangle} + w_{4,2}^{\langle 2\rangle} + w_{4,2} + S_{4,2}\} \end{aligned} \tag{71}$$

与此同时　　时，客户端 3 和客户端 4 发送给服务器中数据库 2 的答案为，

$$A_{W,1}^{\langle 3\rangle,(2)} = \{\Delta_{1,1}^{\langle 3\rangle} + w_{1,1}^{\langle 3\rangle}, \Delta_{1,2}^{\langle 3\rangle} + w_{1,2}^{\langle 3\rangle}, w_{3,1}^{\langle 3\rangle}, w_{3,2}^{\langle 3\rangle}, \Delta_{4,1}^{\langle 3\rangle} + w_{4,1}^{\langle 3\rangle}, \Delta_{4,2}^{\langle 3\rangle} + w_{4,2}^{\langle 3\rangle}\} \tag{72}$$

$$A_{W,1}^{\langle 4\rangle,(2)} = \{\Delta_{1,1}^{\langle 4\rangle} + w_{1,1}^{\langle 4\rangle}, \Delta_{1,2}^{\langle 4\rangle} + w_{1,2}^{\langle 4\rangle}, \Delta_{3,1}^{\langle 4\rangle} + w_{3,1}^{\langle 4\rangle}, \Delta_{3,2}^{\langle 4\rangle} + w_{3,2}^{\langle 4\rangle}, \Delta_{4,1}^{\langle 4\rangle} + w_{4,1}^{\langle 4\rangle}, \Delta_{4,2}^{\langle 4\rangle} + w_{4,2}^{\langle 4\rangle}\} \tag{73}$$

收集和计算完成后，数据库 2 向属于其关联客户端组的客户端 3 发送以下响应，

$$\begin{aligned} D_{W,2}^{\langle 3\rangle,(2)} = \{&\Delta_{1,1}^{\langle 3\rangle} + \Delta_{1,1}^{\langle 4\rangle} + w_{1,1}^{\langle 3\rangle} + w_{1,1}^{\langle 4\rangle} - S_{1,1}, \Delta_{1,2}^{\langle 3\rangle} + \Delta_{1,2}^{\langle 4\rangle} + w_{1,2}^{\langle 3\rangle} + w_{1,2}^{\langle 4\rangle} - S_{1,2}, \\ &\Delta_{3,1}^{\langle 4\rangle} + w_{3,1}^{\langle 3\rangle} + w_{3,1}^{\langle 4\rangle} - S_{3,1}, \Delta_{3,2}^{\langle 4\rangle} + w_{3,2}^{\langle 3\rangle} + w_{3,2}^{\langle 4\rangle} - S_{3,2}, \\ &\Delta_{4,1}^{\langle 3\rangle} + \Delta_{4,1}^{\langle 4\rangle} + w_{4,1}^{\langle 3\rangle} + w_{4,1}^{\langle 4\rangle} - S_{4,1}, \Delta_{4,2}^{\langle 3\rangle} + \Delta_{4,2}^{\langle 4\rangle} + w_{4,2}^{\langle 3\rangle} + w_{4,2}^{\langle 4\rangle} - S_{4,2}\} \end{aligned} \tag{74}$$

类似地，客户端 3 再次通过添加额外的公共随机性来处理收到的响应，然后将以下答案转发给服务器中的两个数据库，

$$\begin{aligned} A_{W,2}^{\langle 3\rangle,([2])} = \{&\Delta_{1,1}^{\langle 3\rangle} + \Delta_{1,1}^{\langle 4\rangle} + w_{1,1}^{\langle 3\rangle} + w_{1,1}^{\langle 4\rangle} - w_{1,1} - S_{1,1}, \Delta_{1,2}^{\langle 3\rangle} + \Delta_{1,2}^{\langle 4\rangle} + w_{1,2}^{\langle 3\rangle} + w_{1,2}^{\langle 4\rangle} - w_{1,2} - S_{1,2}, \\ &\Delta_{3,1}^{\langle 4\rangle} + w_{3,1}^{\langle 3\rangle} + w_{3,1}^{\langle 4\rangle} - w_{3,1} - S_{3,1}, \Delta_{3,2}^{\langle 4\rangle} + w_{3,2}^{\langle 3\rangle} + w_{3,2}^{\langle 4\rangle} - w_{3,2} - S_{3,2}, \\ &\Delta_{4,1}^{\langle 3\rangle} + \Delta_{4,1}^{\langle 4\rangle} + w_{4,1}^{\langle 3\rangle} + w_{4,1}^{\langle 4\rangle} - w_{4,1} - S_{4,1}, \Delta_{4,2}^{\langle 3\rangle} + \Delta_{4,2}^{\langle 4\rangle} + w_{4,2}^{\langle 3\rangle} + w_{4,2}^{\langle 4\rangle} - w_{4,2} - S_{4,2}\} \end{aligned} \tag{75}$$

此时，两个数据库在收到第二步的答案并通过逐元素求和移除所有涉及的公共随机性符号后，均可更新相应的子模型，

$$M_1' = \{M_{1,1} + \Delta_{1,1}^{\langle 1\rangle} + \Delta_{1,1}^{\langle 2\rangle} + \Delta_{1,1}^{\langle 3\rangle} + \Delta_{1,1}^{\langle 4\rangle}, M_{1,2} + \Delta_{1,2}^{\langle 1\rangle} + \Delta_{1,2}^{\langle 2\rangle} + \Delta_{1,2}^{\langle 3\rangle} + \Delta_{1,2}^{\langle 4\rangle}\} \tag{76}$$

$$M'_3 = \{M_{3,1} + \Delta_{3,1}^{\langle 2\rangle} + \Delta_{3,1}^{\langle 4\rangle}, M_{3,2} + \Delta_{3,2}^{\langle 2\rangle} + \Delta_{3,2}^{\langle 4\rangle}\} \tag{77}$$

$$M'_4 = \{M_{4,1} + \Delta_{4,1}^{\langle 3\rangle} + \Delta_{4,1}^{\langle 4\rangle}, M_{4,2} + \Delta_{4,2}^{\langle 3\rangle} + \Delta_{4,2}^{\langle 4\rangle}\} \tag{78}$$

在本示例中，我们注意到，FSL-write 阶段所使用的方案是示例 4 中所使用的 MP-PSU 方案的简化版本，其中未考虑全局公共随机符号 c。因此，对于这一 FSL-write 方案，参照示例 4 中的领导方 P_5，我们可以轻易验证服务器端每个单个数据库的 FSL 写入可靠性约束和 FSL 写入隐私约束，以及参照示例 4 中的客户端方 P_2、P_3，验证客户端 2、3 的 FSL 写入客户端间隐私约束。同样地，这一 FSL-write 方案所具备的针对客户端掉线、迟到客户端和数据库掉线的鲁棒性，也继承自示例 4 中的方案。最后，按照 FSL 流程本身的要求，如果选出一组新的客户端来执行新一轮 FSL 流程，本示例中介绍的 FSL-PSU 阶段和 FSL-write 阶段可以重复执行。每一轮都需要使用新组的公共随机符号和服务器端公共随机符号，以确保隐私。

5 通用联邦子模型学习可实现方案

本节介绍适用于具有任意初始参数的分布式 FSL 模型的通用可实现方案；模型表述见第 2.3 节。我们的通用可实现方案分为三个阶段：公共随机性生成阶段（FSL-CRG）、待更新子模型索引并集的私密确定（FSL-PSU），以及将并集中的已更新子模型私密写回数据库（FSL-write）。第 4 节中，我们给出了 FSL-PSU 的示例，以及 FSL-PSU 与 FSL-write 结合使用的示例。FSL-PSU 和 FSL-write 阶段会用到客户端预先建立的公共随机数。在第 4 节中，我们假设 FSL-PSU 和 FSL-write 所需的公共随机数已预先建立。本节首先介绍如何在客户端之间建立所需的公共随机数。我们的 FSL-CRG 方案利用服务器数据库的分布式特性，并使用一次性密码本和 [43] 中介绍的 RSPIR 方案来生成公共随机性。然后，我们介绍最一般情况下的 FSL-PSU 和 FSL-write。

5.1 公共随机性生成（FSL-CRG）阶段

服务器中的两个数据库旨在为客户端建立两种类型的公共随机性：第一种是全局公共随机符号 c，它从集合 $\mathbb{F}_q \backslash \{0\}$ 中均匀选取。第二种是一组通用公共随机符号 $\{R_0, R_1, \ldots, R_C\}$，其集合长度 $C+1$ 可灵活调整，其中每个符号都从 $\mathbb{F}_q$ 中均匀选取，且最后 C 个符号之和等于 0，即 $\sum_{i\in[C]} R_i = 0$。因此， R_0

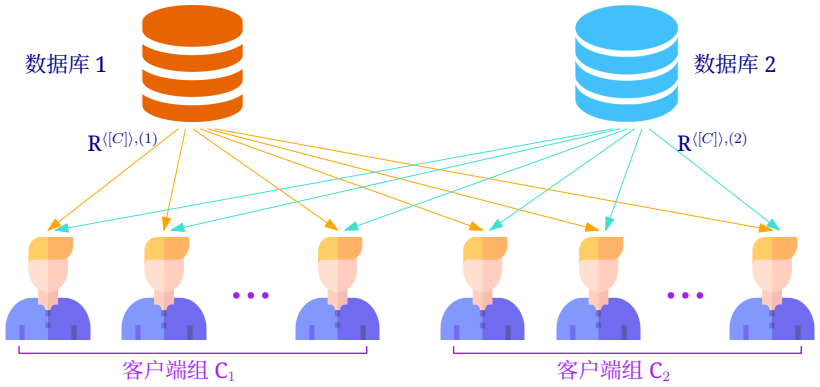


图 5：FSL 系统模型中公共随机性分配阶段的数据流。

可用作 u_k 或$w_{k,\ l}$，而 $R_{[C]}$ 可在接下来的两个阶段中用作 $u_k^{\langle [C] \rangle}$ 或 $w_{k\ l}^{\langle [C] \rangle}$，。在实际实现中，FSL-CRG 阶段独立于 FSL-PSU 和 FSL-write 阶段，因此可以安排在非高峰时段执行。

我们首先介绍第二类公共随机性分配的一种可实现方案。服务器中的数据库 1 从 $\mathbb{F}_q$ 中按照均匀分布随机选择一组符号 $\{R_0^{(1)}, R_1^{(1)}, \ldots, R_{C-1}^{(1)}\}$，然后将该集合广播给每个选定客户端。同样，数据库 2 执行与数据库 1 相同的选择过程，然后也将一组符号 $\{R_0^{(2)}, R_1^{(2)}, \ldots, R_{C-1}^{(2)}\}$ 广播给每个选定客户端。因此，每个客户端通过在有限域内逐元素求和，得到一个长度为 C 的新符号集合，

$$R_i = R_i^{(1)} + R_i^{(2)}, \quad i = 0, 1, \ldots, C-1 \tag{79}$$

随后，每个客户端再向现有集合中添加一个符号 R_C，使得总和等于零，即 $\sum_{i \in [C]} R_i = 0$。由于从服务器发送给每个客户端的两个随机集合相同，这些选定客户端最终共享一个公共随机性集合 $\{R_0, R_1, \ldots, R_C\}$。此外，由于采用了一次一密加密，服务器中的单个数据库并不了解该集合。

接下来，我们考虑第一类公共随机性分配，即分配 c。注意，c 需要在 $\mathbb{F}_q \backslash \{0\}$ 中服从均匀分布。我们可以按如下方式使用上述方法的一个修改版本：每个数据库独立地从 $\mathbb{F}_q$ 中按均匀分布随机选择一个符号，然后将所选符号广播给每个客户端。全局公共随机符号 c 由两个数据库传输的两个随机值相加得到。然而，c 现在以 $\frac{1}{|q|}$ 的概率取值为 0，因此，c 在 $\mathbb{F}_q \backslash \{0\}$ 上均匀分布这一约束并未立即得到满足。服务器中的两个数据库可以重复此过程，直到 c 落入允许区域；如上所述，这需要客户端提供反馈。

为克服这一缺点，我们通过对 [43] 中介绍的 RSPIR 方案采用广播变体，提出了一种新颖的公共随机性分配方法。我们考虑一个

具有 $N=2$、$K=|q|-1$、$L=1$ 的 RSPIR 问题，并使用 [43, 第 V 节] 中给出的可能并非最优的 [7] RSPIR 可实现方案。两个数据库中存储的相应消息集合设为 $W_1=1$、$W_2=2$、.....、$W_{|q|-1}=|q|-1$。我们注意到，此时无需保护其余消息的隐私，因为客户端都可以全局获知这些消息。因此，原始 RSPIR 问题 [43] 中所需的服务器端公共随机性可以舍弃。[8] 数据库 1 包含以下消息集合，并随机向活跃客户端广播其中一条消息，

$$c_1^{(1)}=\emptyset \tag{80}$$
$$c_2^{(1)}=(W_1+W_2,W_2+W_3,\ldots,W_{|q|-2}+W_{|q|-1}) \tag{81}$$
$$c_3^{(1)}=(W_1+W_3,W_2+W_4,\ldots,W_{|q|-2}+W_1) \tag{82}$$
$$\vdots$$
$$c_{|q|-1}^{(1)}=(W_1+W_{|q|-1},W_2+W_1,\ldots,W_{|q|-2}+W_{|q|-3}) \tag{83}$$

同样，数据库 2 有以下一组消息，并从中随机选取一条广播给活跃客户端，

$$c_1^{(2)}=W_1 \tag{84}$$
$$c_2^{(2)}=W_2 \tag{85}$$
$$\vdots$$
$$c_{|q|-2}^{(2)}=W_{|q|-2} \tag{86}$$
$$c_{|q|-1}^{(2)}=W_{|q|-1} \tag{87}$$

通过应用 [43], 中 RSPIR 方法的解码过程，所有客户端都能够获得相同的全局公共随机符号 c，该符号在集合 $\mathbb{F}_q\backslash\{0\}$ 上服从均匀分布。此外，由于 RSPIR 中的用户隐私约束 [43], 每个单独的服务器数据库都无法得知所获得的随机符号 c。

5.2 私有集合并集（FSL-PSU）阶段

FSL-CRG 阶段完成后，每个选定客户端将获得一个在 $\mathbb{F}_q\backslash\{0\}$ 上均匀分布的全局公共随机符号 c，以及一组公共随机符号 $\{u_k^{\langle[C]\rangle}:\ k\in[K]\}$，其中恒等式 $\sum_{i\in[C]}u_k^{\langle i\rangle}=0$ 对所有 $k\in[K]$ 均成立；此外，还将获得另一组公共随机符号 $\{u_k:\ k\in[K]\}$，这些符号均匀分布在 c 上。

[7] 已证明该方案对于 $K=2$ 是最优的，但对于任意 K 都是有效方案。[8] 与 [43] 中研究的随机对称私有信息检索（RSPIR）相比，此处提出的新思路似乎更接近随机私人信息检索（RPIR）的定义。可以进一步研究 RPIR 和 RSPIR，以设计客户端之间更高效、更强大的公共随机性构造。我们将此留作一个有趣的未来研究方向。

按照第 2.3 节中的分布式 FSL 模型，本轮 FSL 流程中的 C 个选定客户端被分为两组 $\mathcal{C}_1$ 和 $\mathcal{C}_2$。 $\mathcal{C}_1$ 中的客户端将其答案发送至服务器中的数据库 1，而 $\mathcal{C}_2$ 中的客户端将其答案发送至服务器中的数据库 2。随后， $\mathcal{C}_1$中的第 i 个客户端将其所需的子模型索引集 $\Gamma^{\langle i\rangle}$映射为相应的关联向量 $Y^{\langle i\rangle}$后，构造其答案如下：

$$A_{U,1}^{\langle i\rangle,(1)} = \{c(Y_1^{\langle i\rangle} + u_1^{\langle i\rangle}), c(Y_2^{\langle i\rangle} + u_2^{\langle i\rangle}), \ldots, c(Y_K^{\langle i\rangle} + u_K^{\langle i\rangle})\} \tag{88}$$

类似地， $\mathcal{C}_2$ 中的第 i 个客户端将其答案构造为：

$$A_{U,1}^{\langle i\rangle,(2)} = \{c(Y_1^{\langle i\rangle} + u_1^{\langle i\rangle}), c(Y_2^{\langle i\rangle} + u_2^{\langle i\rangle}), \ldots, c(Y_K^{\langle i\rangle} + u_K^{\langle i\rangle})\} \tag{89}$$

服务器中的第一个数据库收到来自其在$\mathcal{C}_1$中关联客户端的所有答案后，将生成相应的响应供下载：

$$D_{U,2}^{\langle\theta_1\rangle,(1)} = \left\{ c\sum_{i\in\mathcal{C}_1}(Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) + S_k \colon k \in [K] \right\} \tag{90}$$

其中， $\{S_k \colon \ k \in [K]\}$ 是服务器端共享公共随机符号，也从 $\mathbb{F}_q$ 中均匀选取。随后，某个索引 θ_1 属于 $\mathcal{C}_1$的随机客户端将下载生成的响应 $D^{\langle\theta_1\rangle,\ (1)\ U,\ 2}$。之后，第 θ_1 个客户端通过向收到的响应中添加额外的公共随机性来处理该响应，然后将以下答案转发给服务器中的两个数据库：

$$A_{U,2}^{\langle\theta_1\rangle,([2])} = \left\{ c\sum_{i\in\mathcal{C}_1}(Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) + u_k + S_k \colon k \in [K] \right\} \tag{91}$$

同样，第二个数据库在收到客户端组 $\mathcal{C}_2$在第一步中发送的所有答案后，将生成响应供下载，如下所示：

$$D_{U,2}^{\langle\theta_2\rangle,(2)} = \left\{ c\sum_{i\in\mathcal{C}_2}(Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) - S_k \colon k \in [K] \right\} \tag{92}$$

随后， $\mathcal{C}_2$ 中的某个随机客户端将下载此生成的响应，其索引记为 θ_2。之后，与第 θ_1个客户端一样，该客户端也会将以下进一步处理后的答案转发给服务器中的两个数据库：

$$A_{U,2}^{\langle\theta_2\rangle,([2])} = \left\{ c\sum_{i\in\mathcal{C}_2}(Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) - u_k - S_k \colon k \in [K] \right\} \tag{93}$$

在第二个通信轮次收集到这两个答案集后，服务器中的每个单个数据库 j 都可以通过执行以下操作，得出所需的子模型并集 Γ

经过逐元素求和后，

$$A_{U,2}^{\langle\theta_1\rangle,(j)}+A_{U,2}^{\langle\theta_2\rangle,(j)}=\Bigg\{c\sum_{i\in\mathcal{C}_1}(Y_1^{\langle i\rangle}+u_1^{\langle i\rangle})+u_1+S_1+c\sum_{i\in\mathcal{C}_2}(Y_1^{\langle i\rangle}+u_1^{\langle i\rangle})-u_1-S_1,$$
$$c\sum_{i\in\mathcal{C}_1}(Y_2^{\langle i\rangle}+u_2^{\langle i\rangle})+u_2+S_2+c\sum_{i\in\mathcal{C}_2}(Y_2^{\langle i\rangle}+u_2^{\langle i\rangle})-u_2-S_2,$$
$$\vdots$$
$$c\sum_{i\in\mathcal{C}_1}(Y_K^{\langle i\rangle}+u_K^{\langle i\rangle})+u_K+S_K+c\sum_{i\in\mathcal{C}_2}(Y_K^{\langle i\rangle}+u_K^{\langle i\rangle})-u_K-S_K\Bigg\} \quad (94)$$
$$=\left\{c\sum_{i\in[C]}Y_1^{\langle i\rangle},c\sum_{i\in[C]}Y_2^{\langle i\rangle},\ldots,c\sum_{i\in[C]}Y_K^{\langle i\rangle}\right\} \quad (95)$$

FSL-PSU 可靠性：服务器中的每个单个数据库 j 都利用 $A^{\langle\theta_1\rangle,(}{}_{2}^{j}+A_{U}^{\langle\theta_2\rangle)\,U,\,,\,(j)\,,\,2}$ 中的 K 个元素，判断集合 $[K]$ 中的任意元素是否属于最终子模型索引并集，进而确定 Γ。下面以任意索引 k 为例，分析上述论述。一方面，如果任何客户端所需的子模型索引集包含 k，那么总和 $\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 必定不为零，且表达式 $c\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 必须属于 $\mathbb{F}_q\backslash\{0\}$。另一方面，如果这些客户端所需的子模型索引集均不包含 k，那么总和 $\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 及其相关表达式 $c\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 均等于零。因此，每个数据库都利用其计算所得表达式 $c\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 的值（无论是否为零）来判断索引 k 是否属于并集 Γ。对每个 $k\in[K]$ 进行同样的分析后，两个数据库最终都能得到正确的子模型并集 Γ。由此，FSL-PSU 可靠性得以满足。

FSL-PSU 隐私：我们根据答案集 $\{A_{U\,1}^{\langle\mathcal{C}_1\rangle,}$(1) , , $A^{\langle\mathcal{C}_2\rangle}$,(2) $_{U,1}$ $\}$ 的可用性分析 FSL-PSU 隐私。对于所有 $i\in[C]$ 和所有 $k\in\Gamma$，使用公共随机性 $u_k^{\langle i\rangle}$ 来保护 $Y_k^{\langle i\rangle}$ 的隐私，使得由于一次一密加密，每个数据库都无法获知 $Y_k^{\langle i\rangle}$ 的值。此外，对于所有 $k\in\Gamma$，使用公共随机性 c 来保护 $\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 的隐私，使得由于 $\mathbb{F}_q\backslash\{0\}$ 中乘法下的有限循环群，每个数据库除了知道该和是否为零之外，无法获知 $\sum_{i\in[C]}Y_k^{\langle i\rangle}$ 的值。因此，每个数据库收到答案集 $\{A_{U\,1}^{\langle\mathcal{C}_1\rangle,}$(1) , , $A^{\langle\mathcal{C}_2\rangle}$,(2) $_{U,1}$ $\}$ 时，FSL-PSU 隐私得以保留。具体证明可由 [42, 第 V.B 小节] 中客户端隐私的证明推出。实际上，数据库 1 收到的答案为 $\{A_{U\,1}^{\langle\mathcal{C}_1\rangle,}$(1) , , $A^{\langle\theta_1\rangle}$, (1) $_{U}$,2 , $A^{\langle\theta_2\rangle}$,(1) $_{U,2}$ $\}$，这等价于 $A^{\langle\mathcal{C}_1\rangle}$,(1) $_{U}$,1 和 $\{c\sum_{i\in[C]}Y_k^{\langle i\rangle}:\ k\in[K]\}$，这是因为未知的额外公共随机性 $\{u_k:\ k\in[K]\}$。与此同时，数据库 2 收到的答案为 $\{A_{U\,1}^{\langle\mathcal{C}_2\rangle,}$(2) , , $A^{\langle\theta_1\rangle}$, (2) $_{U}$,2 , $A^{\langle\theta_2\rangle}$,(2) $_{U,2}$ $\}$，出于同样的原因，这等价于 $A^{\langle\mathcal{C}_2\rangle}$,(2) $_{U}$,1 和 $\{c\sum_{i\in[C]}Y_k^{\langle i\rangle}:\ k\in[K]\}$。这意味着，相较于答案集 $\{A_{U\,1}^{\langle\mathcal{C}_1\rangle,}$(1) , , $A^{\langle\mathcal{C}_2\rangle}$,(2) $_{U,1}$ $\}$，每个数据库获取的关于关联向量 $Y^{\langle[C]\rangle}$ 的信息更少。因此，FSL-PSU 隐私约束得以满足。

FSL-PSU 客户端间隐私：只有客户端 θ_1 和 θ_2 能从外部接收信息。由于下载内容 $D^{\langle\theta_1\rangle,\ (1)\ U,\ 2}$ 和 $D^{\langle\theta_2\rangle,\ (2)\ U,\ 2}$ 中存在未知的服务器端公共随机性，第 θ_1 个客户端和第 θ_2 个客户端都无法获知其他客户端的关联向量。因此，FSL-PSU 客户端间隐私约束也得以满足。

FSL-PSU 通信成本：不考虑公共随机性生成阶段产生的通信成本，本阶段的通信成本为 $(C+6)K$ 个 q 进制位。此外，根据第 5.1 节提供的公共随机性生成方法，公共随机性集合 $\{u_k^{\langle[C]\rangle}: k \in [K]\}$ 和 $\{u_k: k \in [K]\}$ 还会产生 $2CK$ 的额外通信成本。对于全局公共随机符号 c，所需通信成本约为 $2(|q|-1)C$ 个 q 进制位；由于 K 通常非常大，因此该成本可以忽略不计。因此，本阶段的总通信成本为 $(3C+6)K$ 个 q 进制位。

FSL-PSU 客户端掉线鲁棒性：在 FSL-PSU 阶段的第一步中，对于所有 $i \in [C]$，客户端 i 会将其生成的答案发送至服务器中与之关联的数据库。不失一般性，我们假设一组属于第一客户端组 $\mathcal{C}_{1,\ \mathcal{D}}$ $\mathcal{C}_1$ 的客户端，以及另一组属于第二客户端组 $\mathcal{C}_{2,\ \mathcal{D}}$ $\mathcal{C}_2$ 的客户端在此步骤中掉线。因此，数据库 1 生成的待下载响应如下，

$$D_{U,2}^{\prime\langle\theta_1\rangle,(1)} = \left\{ c \sum_{i \in \mathcal{C}_1 \setminus \mathcal{C}_{1,\mathcal{D}}} (Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) + S_k \colon k \in [K] \right\} \tag{96}$$

收到响应 $D^{\prime\langle\theta_1\rangle,\ (1)\ U,\ 2}$ 以及停止运行的客户端的索引集$\mathcal{C}_{1,\ \mathcal{D}}$后，客户端 θ_1 可以通过额外附加所有 $k \in [K]$缺失的公共随机符号之和 $\sum_{i \in \mathcal{C}_{1,\ \mathcal{D}}} u_k^{\langle i\rangle}$ 来调整答案。因此，客户端 θ_1 在第二步生成的答案如下，

$$A_{U,2}^{\prime\langle\theta_1\rangle,([2])} = \left\{ c \sum_{i \in \mathcal{C}_1 \setminus \mathcal{C}_{1,\mathcal{D}}} Y_k^{\langle i\rangle} + c \sum_{i \in \mathcal{C}_1} u_k^{\langle i\rangle} + u_k + S_k \colon k \in [K] \right\} \tag{97}$$

Like数据库 2 生成的待下载响应如下，

$$D_{U,2}^{\prime\langle\theta_2\rangle,(2)} = \left\{ c \sum_{i \in \mathcal{C}_2 \setminus \mathcal{C}_{2,\mathcal{D}}} (Y_k^{\langle i\rangle} + u_k^{\langle i\rangle}) + S_k \colon k \in [K] \right\} \tag{98}$$

客户端 θ_2 在第二步生成的答案如下，

$$A_{U,2}^{\prime\langle\theta_2\rangle,([2])} = \left\{ c \sum_{i \in \mathcal{C}_2 \setminus \mathcal{C}_{2,\mathcal{D}}} Y_k^{\langle i\rangle} + c \sum_{i \in \mathcal{C}_2} u_k^{\langle i\rangle} - u_k - S_k \colon k \in [K] \right\} \tag{99}$$

收集到答案 $A'^{\langle\theta_1\rangle,\ (j)\ U,\ 2}$ 和 $A'^{\langle\theta_2\rangle,\ (j)\ U,\ 2}$ 后，通过逐元素相加，服务器中的每个单个数据库 j 都将得到并集结果 $\Gamma_{\mathcal{C}_1,\ \mathcal{D}} \cup \Gamma_{\mathcal{C}_2,\ \mathcal{D}}$，其中包含按照 FSL-SPU 可靠性约束中的步骤选出的所有活跃客户端。另一个不容忽视的问题是，随机选定的客户端 θ_1 和 θ_2 也可能在 FSL-PSU 阶段第 2 步的实现过程中掉线。在实际应用中，一种可能的解决方案是，每个数据库分别随机选出一小组选定客户端，并行转发信息，例如客户端 θ_1 和 θ_2。此外，我们可以利用观测到的客户端掉线率来确定这个小型中继集合的基数。

FSL-PSU 客户端迟到鲁棒性：不失一般性，假设第一步中索引为 $i \in \mathcal{C}_j$ 的任意客户端生成的答案迟到，才到达数据库 j。尽管数据库 j 分别收到了信息 $A^{\langle i\rangle,\ U\,1}{}^{(j)}$，但由于存在未知的额外公共随机性 $\{u_k\colon\ k \in [K]\}$，它仍无法从 FSL-PSU 阶段两步中收到的答案提取有关关联向量 $Y^{\langle i\rangle}$ 的任何信息。这一结论可以推广到任意一组迟到并到达同一数据库的客户端。此外，我们可以轻松确保这一迟到的答案不会被传输给任何其他客户端，从而避免信息泄漏。

FSL-PSU 数据库掉线鲁棒性：如果数据库 1 掉线而无法正常运行，数据库 2 仍能正常收到答案 $A^{\langle\mathcal{C}_2\rangle,\ (2)\ U,\ 1}$ 和 $A^{\langle\theta_2\rangle,\ (2)\ U,\ 2}$，但无法收到中继客户端在 $\mathcal{C}_1$ 中发送的任何答案。为了通过解码集合 $\{c\sum_{i\in\mathcal{C}_2} Y_k^{\langle i\rangle}\colon\ k \in [K]\}$，利用现有信息推导出并集 $\Gamma_{\mathcal{C}_2}$，数据库 2 需要再次与客户端 θ_2 通信，以获取 $\{c\sum_{i\in\mathcal{C}_1} u_k^{\langle i\rangle}\colon\ k \in [K]\}$ 的值。同样，如果数据库 2 无法正常运行，数据库 1 此时仍可用相同方法推导出并集 $\Gamma_{\mathcal{C}_1}$。此外，如果数据库掉线的同时还发生客户端退出或客户端迟到，则可以相应地运用前述两种鲁棒性分析，使该方案正常运行。

5.3 私有写入（FSL-write）阶段

FSL-PSU 阶段完成后，服务器便从本轮 FSL 中所有选定客户端处获知所需的子模型并集 Γ。随后，服务器中的每个数据库都会单独向其关联的客户端发送子模型集合 M_Γ。在先于 FSL-PSU 阶段的 FSL-CRG 阶段中，每个选定客户端还从 $\mathbb{F}_q$ 获得了两组公共随机符号 $\{w_{k,l}^{\langle[C]\rangle}\colon\ k \in \Gamma、\ l \in [L]\}$，以及 $\{w_{k,l}\colon\ k \in \Gamma、\ l \in [L]\}$。同样，对于所有 $k \in \Gamma$ 和 $l \in [L]$，我们始终有 $\sum_{i\in[C]} w_k^{\langle i\rangle},_l = 0$。因此，$\mathcal{C}_1$ 中的第 i 个客户端将根据其本地训练数据，为索引属于 $\Gamma^{\langle i\rangle}$ 的每个所需子模型生成增量，并据此构造经过妥善处理的答案。具体而言，对于所有

$k \in \Gamma^{\langle i \rangle}$，答案符号按以下形式生成，

$$A_{W,1}^{\langle i \rangle,(1)}(k) = \{\Delta_{k,1}^{\langle i \rangle} + w_{k,1}^{\langle i \rangle}, \Delta_{k,2}^{\langle i \rangle} + w_{k,2}^{\langle i \rangle}, \ldots, \Delta_{k,L}^{\langle i \rangle} + w_{k,L}^{\langle i \rangle}\} \tag{100}$$

此外，对于所有 $k \in \Gamma \backslash \Gamma^{\langle i \rangle}$，答案符号均按如下方式生成，且不对当前子模型进行任何更新，

$$A_{W,1}^{\langle i \rangle,(1)}(k) = \{w_{k,1}^{\langle i \rangle}, w_{k,2}^{\langle i \rangle}, \ldots, w_{k,L}^{\langle i \rangle}\} \tag{101}$$

因此，该客户端在第一步中生成的最终答案为 $A^{\langle i \rangle}$、$W^{(1)}$、$_1$ $= \{A_{W,1}^{\langle i \rangle,(1)}(k)\colon\ k \in \Gamma\}$。$\mathcal{C}_2$ 中第 i 个客户端将以相同方式生成最终答案 $A^{\langle i \rangle}$、$W^{(2)}$、$_1$ $= \{A_{W,1}^{\langle i \rangle,(2)}(k)\colon\ k \in \Gamma\}$，其中

$$A_{W,1}^{\langle i \rangle,(2)}(k) = \{\Delta_{k,1}^{\langle i \rangle} + w_{k,1}^{\langle i \rangle}, \Delta_{k,2}^{\langle i \rangle} + w_{k,2}^{\langle i \rangle}, \ldots, \Delta_{k,L}^{\langle i \rangle} + w_{k,L}^{\langle i \rangle}\}, \quad k \in \Gamma^{\langle i \rangle} \tag{102}$$

$$A_{W,1}^{\langle i \rangle,(2)}(k) = \{w_{k,1}^{\langle i \rangle}, w_{k,2}^{\langle i \rangle}, \ldots, w_{k,L}^{\langle i \rangle}\}, \quad k \in \Gamma \backslash \Gamma^{\langle i \rangle} \tag{103}$$

随后，每个客户端都会将其答案发送到服务器中与之关联的数据库。这两个数据库还共享另一组服务器端公共随机符号 $\{S_{k,l}\colon\ k \in [K],\ l \in [L]\}$，这些符号来自 $\mathbb{F}_q$。令 $C_k^{(1)}$ 为第一客户端组 $\mathcal{C}_1$ 中所需子模型索引集包含索引 k 的客户端索引集，即 $C_k^{(1)} = \{i \in \mathcal{C}_1 | k \in \Gamma^{\langle i \rangle}\}$。类似地，$C_k^{(2)}$ 和 C_k 分别定义为 $\{i \in \mathcal{C}_2 | k \in \Gamma^{\langle i \rangle}\}$ 和 $\{i \in [C] | k \in \Gamma^{\langle i \rangle}\}$。收集来自 $\mathcal{C}_1$ 的所有答案 $A^{\langle \mathcal{C}_1 \rangle}$、$_{(1)}$ W、$\mathbf{1}$ 后，数据库 $\mathbf{1}$ 计算第 k 个子模型的第 l 个符号对应的聚合增量，其中 k 属于并集 Γ，

$$\sum_{i \in C_k^{(1)}} \left(\Delta_{k,l}^{\langle i \rangle} + w_{k,l}^{\langle i \rangle}\right) + \sum_{i \in \mathcal{C}_1 \backslash C_k^{(1)}} w_{k,l}^{\langle i \rangle} = \sum_{i \in C_k^{(1)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_1} w_{k,l}^{\langle i \rangle} \tag{104}$$

与上一个 FSL-PSU 阶段一样，在加入服务器端公共随机性后，将生成相应的响应，供客户端 θ_1 下载，

$$D_{W,2}^{\langle \theta_1 \rangle,(1)} = \left\{ \sum_{i \in C_k^{(1)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_1} w_{k,l}^{\langle i \rangle} + S_{k,l}\colon k \in \Gamma, l \in [L] \right\} \tag{105}$$

客户端 θ_1 收到此响应后，只需添加额外的公共随机性，然后将以下答案转发给两个数据库，

$$A_{W,2}^{\langle \theta_1 \rangle,([2])} = \left\{ \sum_{i \in C_k^{(1)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_1} w_{k,l}^{\langle i \rangle} + w_{k,l} + S_{k,l}\colon k \in \Gamma, l \in [L] \right\} \tag{106}$$

与此同时，数据库 2 收集完来自 $\mathcal{C}_2$的所有答案 $A^{\langle \mathcal{C}_2 \rangle,\ (2)\,W,\ 1}$ 后，会生成以下响应，客户端 θ_2 将下载此响应，

$$D_{W,2}^{\langle \theta_2 \rangle,(2)} = \left\{ \sum_{i \in C_k^{(2)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_2} w_{k,l}^{\langle i \rangle} - S_{k,l} \colon k \in \Gamma, l \in [L] \right\} \tag{107}$$

客户端 θ_2 转发给两个数据库的答案如下，

$$A_{W,2}^{\langle \theta_2 \rangle,([2])} = \left\{ \sum_{i \in C_k^{(2)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_2} w_{k,l}^{\langle i \rangle} - w_{k,l} - S_{k,l} \colon k \in \Gamma, l \in [L] \right\} \tag{108}$$

此时，服务器中的每个单个数据库都已准备好，按需聚合本轮联邦子模型学习中所有选定客户端的更新。对于 M_Γ 中第 k 个子模型的第 l 个符号，最终的聚合增量计算如下，

$$\begin{aligned} &\sum_{i \in C_k^{(1)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_1} w_{k,l}^{\langle i \rangle} + w_{k,l} + S_{k,l} + \sum_{i \in C_k^{(2)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_2} w_{k,l}^{\langle i \rangle} - w_{k,l} - S_{k,l} \\ &\quad = \sum_{i \in C_k^{(1)} \cup C_k^{(2)}} \Delta_{k,l}^{\langle i \rangle} + \sum_{i \in \mathcal{C}_1 \cup \mathcal{C}_2} w_{k,l}^{\langle i \rangle} \qquad (109) \\ &\quad = \sum_{i \in C_k} \Delta_{k,l}^{\langle i \rangle} \qquad (110) \end{aligned}$$

执行本轮 FSL-write 后，服务器中存储的第 k 个子模型的更新后第 l 个符号 $M'_{k,\ l}$ 最终应为

$$M'_{k,l} = M_{k,l} + \sum_{i \in C_k} \Delta_{k,l}^{\langle i \rangle} \tag{111}$$

显然，该方案满足 FSL-write 可靠性约束。需要注意的是，FSL-write 阶段中的方案本质上是对 FSL-PSU 阶段方案的简化版本进行重复应用，且不涉及全局公共随机符号 c。因此，FSL-write 方案满足 FSL 写入隐私约束和联邦子模型学习写入客户端间隐私约束，同时还能应对客户端掉线、客户端迟到和数据库退出事件。

FSL-write 通信开销：如果也不考虑伴随公共随机性生成所产生的通信成本，则通信成本为 $(2C+6)|\Gamma|L$ 个 q 进制位，其中 $C|\Gamma|L$ 是客户端从服务器下载子模型所需的通信成本。获取公共随机性集合的通信成本为$\{w_{k,l}^{\langle [C] \rangle} \colon\ k \in \Gamma、l \in [L]\}$ 和 $\{w_{k,l} \colon\ k \in \Gamma,\ l \in [L]\}$ 个客户端所需的通信成本为 $2C|\Gamma|L$ 个 q 进制位。因此，此阶段的总通信成本为 $(4C+6)|\Gamma|L$ 个 q 进制位。

在本轮 FSL 流程中，涉及 FSL-PSU 阶段和 FSL-write 阶段的完整过程可以重复执行，以迭代更新完整学习模型，直至满足预先指定的终止准则。上述所有特性在所有 FSL 轮次中均得以保留。

6 结论与讨论

我们提出了一种新的隐私分布式FSL可达方案，其通信成本的量级与现有方案相近，而现有方案提供的隐私保证要弱得多。与隐私保证相近的现有方案相比，我们提出的方案不需要在数据库中对子模型进行噪声存储。该方案能够抵御客户端掉线、客户端迟到以及数据库掉线。该方案的主要思想基于私有集合并集（PSU）及其私有写入变体，并结合随机私人信息检索和一次性密码本，在客户端生成所需的公共随机性。

我们的方案首先将子模型复制存储在服务器上的两个非串通数据库中，并配备一定数量的服务器端公共随机性。我们的方案在客户端私下生成所需的公共随机性，私下确定待更新子模型的索引并集，并将更新后的子模型私下写回数据库。子模型并集中被更新的索引及其更新后的数值都不会泄露给数据库。

在本研究中，我们考虑了这一新表述中最简单的情形。需要进一步研究的问题包括：1）服务器拥有两个以上数据库的情形。2）针对数据库的存储数据隐私。3）串通数据库。4）发送错误信息的拜占庭数据库。5）客户端之间的串通。6）向学习过程发送错误更新以进行投毒的拜占庭客户端。7）降低通信和存储成本的方案，以及潜在的通信与存储权衡。8）MDS 编码存储和/或 MDS 编码的用户端公共随机性。9）在数据库之间最优地划分客户端，尤其是在已知串通结构且存在串通数据库的情况下。

参考文献

[1]B. McMahan 和 D. Ramage。《联邦学习：无需集中式训练数据的协作式机器学习》。可访问：
https://ai.googleblog.com/2017/04/federated-learning-collaborative.html。

[2] Q. Yang、Y. Liu、T. Chen 和 Y. Tong。《联邦机器学习：概念与应用》。ACM智能系统与技术汇刊，10(2):1–19，2019年3月。

[3] K. Bonawitz、V. Ivanov 等。《面向隐私保护机器学习的实用安全聚合》。载于2017年ACM SIGSAC计算机与通信安全会议论文集，第1175–1191页，2017年。[4] J. Bell、K. A. Bonawitz、A. Gasc´on、T. Lepoint 和 M. Raykova。《具有（多项式）对数级开销的安全单服务器聚合》。载于密码学电子印刷档案，2020年。[5] J. So、B. Guler 和 A. S. Avestimehr。《Turbo-aggregate：突破安全联邦学习中的二次聚合壁垒》。载于密码学电子印刷档案，2020年。[6]S. Kadhe、K. Ramchandran、N. Rajaraman abd O. O. Koyluoglu。《Fastsecagg：面向隐私保护联邦学习的可扩展安全聚合》。可见 arXiv:2009.11248。[7] Y. Zhao 和 H. Sun。《存在用户掉线时的信息论安全聚合》。IEEE信息论汇刊，68(11):7471–7484，2022年11月。[8] K. Wan、H. Sun、M. Ji 和 G. Caire。《采用未编码分组密钥的信息论安全聚合》。可见 arXiv:2204.11364。[9] Y. Zhao 和 H. Sun。《安全求和：容量区域、分组密钥与可行性》。可见 arXiv:2205.08458。[10] J. So、C. He 等。《Lightsecagg：联邦学习中安全聚合的轻量且通用设计》。载于机器学习与系统会议论文集，第694–720页，2022年。[11] C. Niu、F. Wu、S. Tang、L. Hua、R. Jia、C. Lv、Z. Wu 和 G. Chen。《移动客户端上的十亿规模联邦学习：具有可调隐私的子模型设计》。载于第26届移动计算与网络年度国际会议论文集，第1–14页，2020年。[12] L. Kissner 和 D. Song。《隐私保护集合运算》。载于密码学进展——CRYPTO 2005，第241–257页，2005年。[13] K. Frikken。《隐私保护集合并》。载于《应用密码学与网络安全》，第237–252页，2007年。[14] C. Naim、R. G. L. D'Oliveira 和 S. El Rouayheb。《私有多组聚合》。IEEE通信领域精选期刊，40(3):800–814，2022年3月。[15] Z. Jia 和 S. A. Jafar。《 X-安全、 T-隐私的隐私联邦子模型学习，具有弹性掉线鲁棒性》。IEEE信息论汇刊，68(8):5418–5439，2022年8月。

[16] Z. Jia、H. Sun 和 S. A. Jafar。《交叉子空间对齐与 X安全的 T隐私信息检索的渐近容量》。IEEE信息论汇刊，65(9)：5783–5798，2019年9月。[17] S. Vithana 和 S. Ulukus。《高效隐私联邦子模型学习》。载于 IEEE国际通信会议，第3394–3399页，2022年5月。[18] S. Vithana 和 S. Ulukus。《联邦子模型学习（FSL）中的私有读取、更新和写入（PRUW）：有无稀疏化的通信高效方案》。可通过 arXiv:2209.04421 获取。[19] B. Chor、E. Kushilevitz、O. Goldreich 和 M. Sudan。《隐私信息检索》。《ACM学报》，45(6)：965–981，1998年11月。[20] Y. Gertner、Y. Ishai、E. Kushilevitz 和 T. Malkin。《保护隐私信息检索方案中的数据隐私》。《计算机与系统科学杂志》，60(3)：592–629，2000年6月。[21] H. Sun 和 S. A. Jafar。《隐私信息检索的容量》。IEEE信息论汇刊，63(7)：4075–4088，2017年7月。[22] K. Banawan 和 S. Ulukus。《从编码数据库进行隐私信息检索的容量》。IEEE信息论汇刊，64(3)：1945–1956，2018年3月。[23] H. Sun 和 S. A. Jafar。《具有串通数据库的鲁棒隐私信息检索的容量》。IEEE信息论汇刊，64(4)：2361–2370，2018年4月。[24] K. Banawan 和 S. Ulukus。《从拜占庭数据库和串通数据库进行隐私信息检索的容量》。IEEE信息论汇刊，65(2)：1206–1219，2019年2月。[25] Q. Wang、H. Sun 和 M. Skoglund。《存在窃听者时隐私信息检索的容量》。IEEE信息论汇刊，65(5)：3198–3214，2019年5月。[26] K. Banawan 和 S. Ulukus。《多消息隐私信息检索：容量结果与近最优方案》。IEEE信息论汇刊，64(10)：6842–6862，2018年10月。[27] R. Tandon。《缓存辅助隐私信息检索的容量》。载于 Allerton会议，第1078–1082页，2017年10月。[28] Y.-P. Wei、K. Banawan 和 S. Ulukus。《具有未知且未编码预取的缓存辅助隐私信息检索的基本极限》。IEEE信息论汇刊，65(5)：3215–3232，2019年5月。[29] Z. Chen、Z. Wang 和 S. A. Jafar。《具有隐私侧信息的 T隐私信息检索的容量》。IEEE信息论汇刊，66(8)：4761–4773，2020年8月。

[30] M. A. Attia、D. Kumar 和 R. Tandon。《从未编码存储的存储受限数据库进行隐私信息检索的容量》。IEEE信息论汇刊，66(11)：6617–6634，2020年11月。[31] C. Tian。《论隐私信息检索的存储成本》。IEEE信息论汇刊，66(12)：7539–7549，2020年12月。[32] C. Tian、H. Sun 和 J. Chen。《具有最优消息大小和上传成本的容量可达隐私信息检索码》。IEEE信息论汇刊，65(11)：7613–7627，2019年11月。[33] I. Samy、M. Attia、R. Tandon 和 L. Lazos。《非对称泄漏式私有信息检索》。IEEE信息论汇刊，67(8)：5352–5369，2021年8月。[34] S. Vithana、K. Banawan 和 S. Ulukus。《语义私有信息检索》。IEEE信息论汇刊，68(4)：2635–2652，2022年4月。[35] H. Sun 和 S. A. Jafar。《对称私有信息检索的容量》。IEEE信息论汇刊，65(1)：322–329，2019年1月。[36] Z. Wang 和 S. Ulukus。《在隐私信息检索速率下的对称私有信息检索》。IEEE信息论精选领域期刊，3(2)：350–361，2022年6月。[37] Z. Wang 和 S. Ulukus。《双数据库对称私有信息检索的通信成本：多重秘密条件披露视角》。载于 IEEE 信息论国际研讨会，第402–407页，2022年6月。[38] Q. Wang 和 M. Skoglund。《关于具有对手和窃听者的串通数据库中的隐私信息检索与对称隐私信息检索》。IEEE信息论汇刊，65(5)：3183–3197，2019年5月。[39] Q. Wang 和 M. Skoglund。《来自 MDS 编码分布式存储的对称隐私信息检索：非串谋与串谋服务器情形》。IEEE信息论汇刊，65(8)：5160–5175，2019年8月。[40] J. Cheng、N. Liu、W. Kang 和 Y. Li。《任意串谋与窃听模式下对称隐私信息检索的容量》。IEEE信息取证与安全汇刊，17：3037–3050，2022年8月。[41]Z. Wang、K. Banawan 和 S. Ulukus。《私有集合求交：多消息对称私有信息检索视角》。IEEE信息论汇刊，68(3)：2001–2019，2022年3月。

[42]Z. Wang、K. Banawan 和 S. Ulukus。多方私有集合求交：一种信息论方法。《IEEE 期刊：精选领域信息论》，2(1):366–379，2021年3月。[43] Z. Wang 和 S. Ulukus。数字盲盒：随机对称私人信息检索。发表于 IEEE 信息论研讨会，第95–100页，2022年11月。[44] H. Corrigan-Gibbs 和 D. Boneh。Prio：聚合统计量的私有、稳健且可扩展的计算。发表于《第14届 USENIX 网络系统设计与实现会议论文集》，第259–282页，2017年。[45] Y. Lu、Z. Jia 和 S. A. Jafar。双盲 T-私有信息检索。《IEEE 期刊：精选领域信息论》，2(1):428–440，2021年3月。[46] J. Zhu、Q. Yan 和 X. Tang。来自编码服务器的多用户盲对称私有信息检索。《IEEE 精选通信领域期刊》，40(3):815–831，2022年3月。[47] M. Kim 和 J. Lee。联邦子模型学习中的信息论隐私。可见 arXiv:2008.07656。[48] S. Vithana 和 S. Ulukus。存储受限数据库中的私有读取、更新和写入（PRUW）。发表于 IEEE 信息论国际研讨会，第2391–2396页，2022年6月。[49] S. Vithana 和 S. Ulukus。通过稀疏化实现隐私联邦子模型学习。发表于 IEEE 信息论研讨会，第410–415页，2022年11月。[50] S. Vithana 和 S. Ulukus。联邦子模型学习中私有读取、更新和写入的率失真权衡。发表于 Allerton会议，2022年10月。