arXiv:2303.02842v1 [cs.DB] 6 Mar 2023

隐私保护空间数据联盟上的高效、准确范围计数

Maocheng Li[1], Yuxiang Zeng[1,2],和 Lei Chen[1,3]

[1]香港科技大学，中国香港{csmichael,leichen}@cse.ust.hk[2]北京航空航天大学计算机科学与工程学院，中国 turf1013@buaa.edu.cn[3]香港科技大学（广州），中国广州

摘要

空间数据联邦由多个数据所有者组成（例如，出租车公司联盟），整体上能够提供更优质的基于位置的服务（LBS）。例如，空间数据联邦中的网约车服务可让终端用户轻松挑选最佳方案。我们聚焦于范围计数查询。这类查询是空间数据库中的基本操作，但在相关研究中鲜少受到关注，尤其是考虑到数据所有者的隐私要求——他们不愿披露自己的专有数据。我们提出了一种名为 FedGroup 的基于分组的技术框架，在不损害隐私的前提下对数据所有者进行分组。与直接应用现有差分隐私（DP）隐私机制相比，该框架的查询准确度更高（最高提升 50%）。实验结果还表明，FedGroup 的运行速度比传统的基于安全多方计算（MPC）的方法快几个数量级；而且 FedGroup 甚至可以扩展至数百万个数据所有者，这在移动设备无处不在的时代是常见的应用场景。

1 引言

联邦数据库系统由多个相互协作但彼此自治的数据库系统组成 [17]。它反映了现实中的一种情况：客户记录的整个数据库通常分散在不同的公司中，每家公司都拥有整个数据库中独立的一部分。联邦数据库系统引起了广泛的研究兴趣 [18, 3, 2]，并已应用于现实场景。例如，多家医院加入一个联盟，共同提供数据以发现新药物 [2]。

在基于位置的服务（LBS）系统这一特定领域中，空间数据联邦是指由空间数据构成的联邦，例如位置或轨迹。与通用联邦数据库类似，空间数据联邦也有着广泛的现实应用。例如，阿里巴巴旗下的高德地图 [21] 在中国通过一个

[1]版权可能在未通知的情况下转让，此后此版本将不再可访问。

由多家出租车公司组成的联邦提供网约车服务，让用户拥有更大的选择空间，轻松挑选最优惠的服务。

为支持上述基于位置的服务应用，需要针对各种空间查询采用查询处理技术。其中，范围计数查询是最重要的基本操作之一。范围计数查询返回位于给定范围内的空间对象数量。网约车联邦系统可能会频繁发出以下范围计数查询：“某位客户的位置周围*500*米内有多少辆车？”

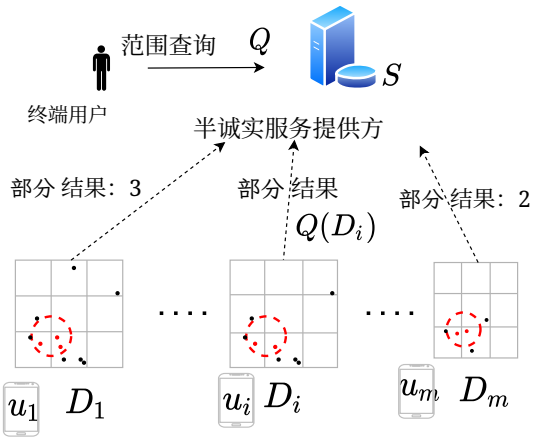


图1：FPRC问题。每个 m 数据所有者 u_i 都拥有一个空间数据库 D_i。终端用户通过空间数据联邦发出范围计数查询 Q： $D_1 \cup \ldots \cup D_m$。

本文研究联邦隐私保护范围计数（*FPRC*）问题（图1），重点关注大规模空间数据联邦，其中数据所有者的数量非常庞大（例如，超过1万）。由于移动设备的广泛普及，此类大规模场景近年来受到了越来越多的关注。例如，在去中心化的众包平台中， [24], 每个数据所有者都是一台移动设备，每位用户都持有自己的数据，并且不愿意将任何未经保护的私有数据披露给其他方。在此类应用中，数据所有者的数量很容易达到数百万量级。

在空间数据联邦上支持范围计数查询颇具挑战，因为每个数据所有者通常都不愿披露其专有数据。直接采用传统的安全多方计算（MPC）技术或差分隐私（DP）方案，要么计算开销过高，要么会引入过多噪声，导致基于位置的服务应用无法获得有意义的结果。正如我们在第4节的实验所示，MPC基线比我们的方案慢几个数量级，在输入规模较大时需要 13 小时以上；而差分隐私基线的准确率则损失了 50% 以上。鉴于基于位置的服务应用具有交互性，理想的方案应具备出色的准确率、实用的效率和经过验证的隐私保证。

为应对 FPRC问题带来的挑战，我们提出了一种基于分组的技术框架，名为 FedGroup。为减少对每个数据所有者直接应用差分隐私所引入的大量噪声，我们将相似的数据所有者（在我们的场景中为移动用户）划分到同一组中。我们认为，在每个组中注入差分隐私噪声（而非为每个数据所有者分别注入噪声）足以提供隐私保证。这是因为某些用户可能与其他数据所有者存在社会关系（例如，他们属于同一个家庭），并且拥有相似的轨迹和行踪。

为实现有效分组，我们采用图分析中一个广为接受的概念——(k, r)-核 [29]，以确保组内连接紧密（此处图中的节点代表数据所有者，节点之间的边表示他们具有较强的空间

相似性，即，他们的轨迹相似）。我们构建了一个优化问题，以寻找将用户划分为各组的最佳方式，从而尽量减少差分隐私噪声在 FPRC问题中引入的误差。我们证明该问题是 NP困难的，并提出了一种高效的贪心算法，且具有性能保证。

最后但同样重要的是，用户分组取决于如何衡量用户之间的空间相似度（例如，他们的轨迹）。考虑到轨迹属于各数据所有者的隐私信息，在每对用户之间构建相似性图并非易事。我们设计了一种新颖的混合方案，结合差分隐私和 MPC，以实现准确的相似性图构建。

总而言之，我们的贡献如下:

• 我们提出了一种新颖的技术框架 FedGroup，通过采用离线分组步骤，大幅减少联邦隐私保护范围计数（FPRC）问题所需的噪声量。我们在第2节介绍 FPRC问题。

• 为了在 FedGroup 中实现有效分组，我们证明，要使 FPRC问题所需的噪声量最小化而实现最优分组，是一个 NP难问题。随后，我们提出了一种高效的贪心算法，并为其提供性能保证，以实现有效分组。

• 考虑到数据所有者的轨迹属于隐私信息，我们还设计了结合差分隐私和 MPC技术来构建用户间空间相似性图的有效技术。详情见第3节。

• 我们进行了大量实验，以验证所提出方案的有效性和效率。结果如第4节所示。

此外，我们在第5节回顾相关工作，并在第6节进行总结。

2 问题定义

本节将介绍一些基本概念、对手模型以及联邦隐私保护范围计数（FPRC）问题的定义。由于篇幅限制，附录中提供了一个玩具示例。

2.1 基本概念

定义 1（位置）。 位置 $l =$（x、 y） 表示欧几里得空间中坐标为 （x、 y） 的二维空间点。

现有研究中通常会涉及欧几里得空间中的位置 [16, 10, 20]。

定义 2 (数据所有者). 共有 m 个数据所有者 u_1，.....， u_m。每个数据所有者 u_i 拥有一个空间数据库 D_i。每个空间数据库 D_i 由多条数据记录（如定义 *1* 中的位置）组成，即，$D_i = \{l_1, \ldots, l_{|D_i|}\}$。

我们还将每个数据所有者的数据库 D_i 称为一个数据孤岛。我们令 $D = \cup_{i=1}^{m} D_i$ 表示所有数据孤岛的集合，即每个数据孤岛中所有数据记录的并集。

定义 3（范围计数查询）。 范围计数查询 Q 询问距离查询位置 r 以内有多少条数据记录，即，

$$Q(r, q_0) = \sum_{l \in D} \mathbb{I}(d(l, q_0) < r) \tag{1}$$

其中， $\mathbb{I}(\cdot)$ 是指示函数：当谓词 $d(l, q_0) < r$ 为真时取值为 *1*，否则取值为 *0*。 $d(\cdot)$ 是欧氏距离函数。 $l \in D$ 是来自所有数据孤岛中全部数据记录并集的任意一条记录。

2.2 隐私与对手模型

我们的空间数据联邦中主要有三方（角色）（见图1）：（1）发出查询的**终端用户**；（2）接收查询并协调查询执行的**服务提供商**；以及（3）拥有私有空间数据库的每位**数据所有者** u_i D_i。

我们假设所有参与方都是半诚实的，即他们好奇但不恶意。他们会好奇 其他参与方的隐私信息，但会诚实地遵循并执行系统协议。这种设定已广泛应用于近期与隐私保护位置服务相关的应用中 [22, 19, 25]。

2.3 联邦隐私保护范围计数问题

基于前述概念和对手模型，我们现在将联邦隐私保护范围计数（FPRC）问题定义如下。

定义 4（联邦隐私保护范围计数问题）。 给定一个由m 个空间数据库 D_1、.....、D_m组成的联邦、一项范围计数查询 Q（q_0， r）以及一个隐私参数~（又称隐私预算） ϵ，*FPRC*问题要求给出查询答案 Q，并满足以下隐私要求：

• *R1.* 计算出的最终结果 Q 满足 $\tilde{\epsilon}$-差分隐私（*DP*），其中，相邻数据库的定义是指任意数据孤岛 D_i中仅有一条记录发生变化。

• *R2.* 任意数据孤岛 D_i 披露的中间结果 D_i 满足 ϵ-*DP*。

• *R3.* 如果涉及多方计算，则每个数据孤岛 D_i 的私有输入均为机密。

本文提出了两种基线方案，分别基于MPC和差分隐私。MPC基线直接应用多方计算技术，对各数据孤岛的查询结果进行安全求和。差分隐私基线则向每个数据孤岛注入一次拉普拉斯噪声，并收集聚合噪声求和结果。

如第 1 节所述，解决 FPRC问题面临三大**关键挑战**：1）隐私：每位数据所有者都不愿披露其敏感数据；2）效率：直接应用 MPC 会导致方案不切实际；3）准确性：由于数据所有者的数量可能达到百万级，差分隐私基线注入的总噪声规模过大。

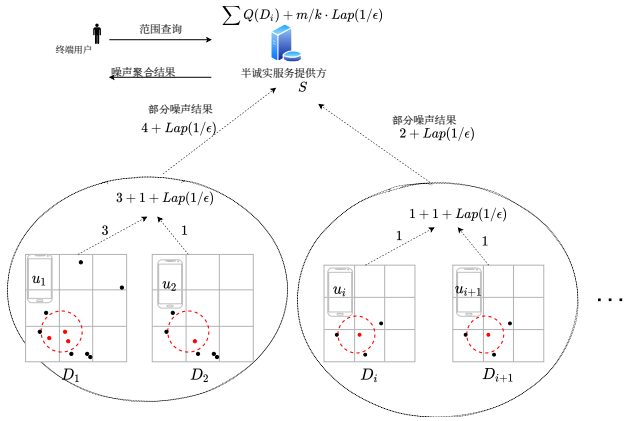


图2：FedGroup工作流程。

3 我们的解决方案 FedGroup

本节介绍 FedGroup 的技术细节，该方法旨在应对 FPRC问题所面临的挑战。我们首先概述其整体流程（图 2 为示意图），然后介绍各个步骤的细节：1）构建空间相似性图；2）根据相似性图发现组；3）部分答案与聚合。由于篇幅有限，更多示例和详细证明请参阅附录。

关键思想与直觉。在我们所研究的联邦隐私保护范围计数（FPRC）问题（如第2节所述）中，数据所有者的数量（m）可能非常庞大（例如，数百万），因为每个数据所有者都可能是移动用户。FedGroup的核心思想基于以下观察：某些移动用户彼此之间联系紧密（源于社会关系或其他协作关系）。如果我们将这类数据所有者对视为属于同一个组，那么组内的隐私保护要求就可以放宽。例如，无需考虑家庭内部用户轨迹的隐私保护，因为家庭成员很可能知道其他成员一天中的行踪。因此，只需在不同组之间添加差分隐私噪声，而差分隐私基线则是为每个移动用户注入一份拉普拉斯噪声。因此，针对FPRC问题的查询结果所注入的总噪声可以大幅减少。

3.1 空间相似性图构建

在这一步中，我们的目标是在数据所有者之间构建一个无向图 $G_s = (V, E)$。图 V中的每个节点代表一个数据所有者 u_i，而 E 中连接两个节点 u_i 和u_j 的边表示分别由 u_i 和 u_j拥有的空间数据库 D_i 和 D_j 之间的相似度。边的权重衡量 u_i 与 u_j之间连接的强度，我们使用余弦相似度作为边的权重。

我们首先介绍相似性图的细节，包括如何衡量数据所有者之间的空间相似度。然后，我们考虑隐私保护设置，其中每个数据所有者的空间数据库都被视为私有，并介绍以差分隐私方式计算数据所有者之间相似度函数的方法。

空间相似性图。构建空间相似性图 $G_s = (V, E)$ 的关键在于测量数据所有者 u_i, $u_j \in V$ 之间每条无向边 $e = (u_i, u_j)$ 的权重。下面将介绍如何计算两个数据孤岛 D_i 和 D_j 之间的相似度函数。

对于每个数据孤岛 D_i，我们使用网格结构 T 将空间域分解为若干个网格，其中 $|T|$ 表示网格数量。然后，在每个网格内计算一个计数，以表示 D_i 中落入该网格的空间点（记录）数量。因此，我们为数据孤岛 D_i获得一个 $|T|$ 维计数向量。

同样，对于数据孤岛 D_j，我们可以得到另一个计数向量 $v_j = [c_{j,1}, c_{j,2}, \ldots, c_{j,|T|}]$。由于这两个向量 v_i和 v_j的长度相同 $|T|$，我们可以使用 余弦 e 相似度来衡量它们的相似度：

$$\text{sim}(D_i, D_j) = \cos(v_i, v_j) = v_i \cdot v_j / \|v_i\| \|v_j\| \tag{2}$$

余弦相似度的取值范围为 0 到 1，值越大，两个计数向量就越相似。在我们的应用中，两个计数向量之间的余弦相似度越高，它们所对应的空间数据库就越相似，而这些数据库由不同的数据所有者拥有。

构建图。要构建空间相似性图 $G_s = (V, E)$，首先将所有数据所有者加入节点集 V。然后，我们遍历所有数据所有者对 $u_i, u_j \in V$，并根据公式（2）衡量其空间数据库 D_i 和 D_j之间的空间相似度。对应边 $e = (u_i, u_j)$ 的权重设为 $\text{sim}(D_i, D_j)$，即，$w(e) = \text{sim}(D_i, D_j)$。

如果不考虑每个数据孤岛 D_i 的隐私，构建空间相似性图 G_s 就很直接。该方法在服务提供商 S上运行。首先，我们将所有数据所有者 $u_1, \ldots, u_m$ 插入 V，从而创建 G_s 的节点集 V。然后，该方法只需遍历数据所有者对 u_i, u_j（其中 $i, j \in [m]$ 且 $i < j$），并计算 $\text{sim}(D_i, D_j)$。如果相似度大于给定阈值 r，则将边 (u_i, u_j) 插入 G_s 的边集 E，并将边权重设为 $w(u_i, u_j) = \text{sim}(D_i, D_j)$。

时间复杂度为 $O(m^2)$，因为我们遍历了所有可能的数据所有者对。所需的空间复杂度也为 $O(m^2)$，因为我们需要在 S 上存储 G_s，包括所有节点和边（以及它们的权重）。

隐私保护计算。 尽管在不考虑每个数据孤岛 D_i 隐私的情况下，构建 G_s 很直接，但在衡量每对数据孤岛之间的空间相似度时，提供经过验证的隐私保证至关重要。这项工作的首要动机是在空间联邦场景中考虑隐私保护，其中每个数据所有者的数据孤岛 D_i 都被视为敏感信息。

显然，非隐私的图构建方式无法满足定义 4 中问题定义的隐私要求 R2。R2 要求每个数据所有者共享的中间结果满足 ϵ-差分隐私。然而，在非隐私版本中，每个数据所有者都会直接向服务提供商 S 发布所提供的 D_i（或其计数向量 v_j）。显然，这会导致隐私泄露。此外，定义 4 中的 R1 也可能无法满足。尚不清楚所构建的相似性图 G_s 是否能针对任一数据孤岛中的一条记录变更 D_i 提供任何形式的隐私保证关于。

因此，我们提出一种混合方案，将安全多方计算（MPC）与标准的拉普拉斯机制用于差分隐私相结合。

混合方案。 混合方案结合了两种方法各自的优势：MPC 的高准确性（因为计算是精确的），以及差分隐私的高效率（因为每个数据孤岛都独立执行拉普拉斯计算，而且注入拉普拉斯噪声本身也是一种高效操作）。

混合方案的核心思路是：我们使用差分隐私中的高效拉普拉斯机制，快速计算出一个噪声权重 $\tilde{w}$。我们不会直接将这个噪声权重作为替代

算法1： 构建 G_s - 混合方案

输入： $u_1, \ldots, u_m, \tilde{v_1}, \ldots, \tilde{v_m}, r, r_l, r_u$.

输出： $\tilde{G_s} = (V, \tilde{E})$.

```
1  V := {u_1, ..., u_m}
2  Ẽ := {}
3  对每个 u_i ∈ V 执行
4      对每个 u_j ∈ V 且 j > i 执行
5          e := (u_i, u_j)
6              w̃ := cos(ṽ_i, ṽ_j)
7          如果 e.权重 < r_l 则
8              继续
9          否则如果 w̃ > r_u then
10             e.权重：= w̃
11             插入 e 到 Ẽ
12         else
13             e.权重：= 安全相似度(D_i, D_j)
14             如果 e.权重 > r 那么
15                 插入 e 到 Ẽ
16 返回 G_s = (V, Ẽ)
```

真实图中边权重的值，而只将其用作过滤机制。如果噪声权重 $\tilde{w}$ 与给定阈值 r相比过小，我们就可以迅速判断该边不应被加入图中。反之，如果噪声权重 $\tilde{w}$ 看起来较大，我们就可以迅速判断应将该边加入图中。如果噪声权重处于临界范围，表明结果存在不确定性且可能引入错误，我们便调用计算开销较大的MPC方法，根据两个数据所有者的私有输入计算出精确的边权重。

算法 1 展示了详细步骤。输入 $\tilde{v_1}$，.....，$\tilde{v_m}$ 是噪声计数向量，通过向计数向量 v_1，.....，v_m添加拉普拉斯噪声得到。

隐私分析。 我们证明，我们的混合方案满足定义 4 中的隐私要求。

定理 1。 算法 *1* 中的混合方案满足定义 *4* 中的所有隐私要求（*R1-R3*）。

证明。 由于唯一发布的中间结果是噪声计数 $\tilde{v_i}$，因此满足 R2。根据差分隐私的后处理和并行组合定理，满足 R1，因为输出图 G_s 仅取决于噪声计数。满足 R3 是因为多方计算仅在第 12 行执行，且 MPC 协议保证了数据的机密性。

□

3.2 群组发现

(k, r)-核。 我们采用图挖掘文献中普遍接受的概念（k，r）-核 [29] 来定义这些组。**(k, r)-核**的概念考虑了两个重要标准：

用于确定关系密切的组（又称社区检测）：参与度和相似度，即组成员不仅彼此之间具有紧密的社交连接（参与度高），而且还具有高度相似性（即，他们的轨迹相似）。

在我们的设定中，由于我们关注的是数据所有者的空间数据库，而非社交图，因此要求所形成的组能够体现数据所有者之间的强相似性。因此，我们要求每个组都是一个 r-团，这意味着每个组都是一个团，且数据所有者之间的边权重（即第 3.1 节中介绍的空间相似度）大于或等于 r。

定义 5（r-组）。*A* r-组 g 是空间相似性图 G_s中的一个团，其中属于该组的任意两个数据所有者之间都存在一条边，且该边权重高于阈值 r，即，对于任意两个数据所有者 u_i、$u_j \in g$，

$$\exists e = (u_i, u_j) \in E \, , w(e) > r \tag{3}$$

优化分组。 r-组的概念赋予了将数据所有者归为一组以意义，因为他们彼此相似。然而，将数据所有者划分为不同组的方式可能有多种。例如，一个数据所有者可能属于多个 r-组，那么我们应该将该数据所有者归入哪个组呢?

在本节中，我们将构建一个优化问题，将本文最终问题——FPRC问题的效用目标与分组策略联系起来。随后，我们证明该优化问题是一个NP困难问题，并提出一种有效且高效的贪心算法。

我们首先定义数据所有者分组（DOG）问题。

定义 6（数据所有者分组（DOG）问题）。 给定定义 *4* 中*FPRC*问题的输入，*DOG*问题要求将数据所有者分配到 λ 个互不相交的组中：g_1、g_2、.....、g_λ，其中每个组 g_i 都是一个 r-组（根据定义 *5*），使得*FedGroup*针对*FPRC*问题注入的噪声总量最小。

接下来，我们将证明，要使FPRC问题的误差（注入的噪声总量）最小化，等价于使组数 λ最小化。

引理 1. 在 *FPRC*问题中，将误差最小化等价于将 λ 最小化，也就是 *DOG*问题中的组总数。

证明。 我们回顾一下 FedGroup 在 FPRC问题中的误差。FedGroup 将数据所有者分配到互不相交的组中，每个组向查询结果注入一个拉普拉斯噪声实例。随后，从每个组收集部分含噪结果，并将其汇总为˜最终答案。因此，以方差衡量的最终答案 Q 的总误差为:

$$\mathrm{Var}(\tilde{Q}) = \mathrm{Var}(\sum_{i=1}^{\lambda} \mathrm{Lap}(1/\epsilon)) = \lambda \cdot \mathrm{Var}(\mathrm{Lap}(1/\epsilon)) = 2\lambda/\epsilon^2. \tag{4}$$

因此，要使公式 **(4)** 中的总误差最小化，我们需要将 λ最小化。 □

现在，我们的 DOG问题转化为寻找一种最优方式，将数据所有者分配到互不相交的组中，使每个组都是一个 r-团，并使组数最少。事实上，该问题可以从最小团划分（MCP）问题归约而来，其困难性结果如下所示。

定理2。 *DOG*问题是*NP*困难的。

算法 2： 贪心查找组

输入： $\tilde{G_s}$ =(V, $\tilde{E}$).
输出： $g_1, g_2, \ldots, g_\lambda$。

G_c :=补图($\tilde{G_s}$)
已分配：= 初始化整数数组(size=$|V|$, initValue=0)
$\lambda := 1$; 已分配$[u_1] = 1$； g_1.insert(u_1)
u := 下一个未分配顶点(V, 已分配)
当 u 存在 **时执行**
    gid :=下一个组编号(u.邻居(), 已分配)
    已分配$[u] = gid$； g_{gid}.insert(u)
    如果 $gid > \lambda$ **则**
        $\lambda = gid$
    u := 下一个未分配顶点(V, assigned)
返回 g_1， g_2、.....、 g_λ

证明。 我们在附录中将NP难问题——最小团划分（MCP）归约为DOG问题。因此，DOG具有相同的难度。 □

贪心算法。 由于 DOG问题是 NP困难问题，我们设计了一种高效的贪心解（算法 2），以实现有效分组，并为返回的解提供有界的噪声尺度。该解的灵感来自最小图着色问题的贪心解；最小图着色问题已被证明是 MCP问题的等价问题。

性能保证。 在定理3中，我们证明了该贪心解的组数有界，并且能为 FPRC问题提供有界噪声。

定理3。 如果 d 是补图 G_c中节点的最大度，那么算法 *2* 至多返回 $d+1$ 组。

证明。 在算法 2 的执行过程中，我们关注一个度为 d （图中最大度）的数据所有者 u。我们最多需要避免让 u 的 d 个邻居使用相同的组标识符。由于我们使用的是其邻居尚未使用过的编号最小的组标识符，因此在组标识符 1，2、.....、 $d+1$中，至少有一个标识符可供 u 使用（例如，前 d 个组标识符已被使用，现在使用 $d+1$）。因此，我们得出结论：算法 2 至多返回 $d+1$ 组。 □

时间复杂度。算法 2 每次处理一个顶点时，都会遍历该顶点的邻居，以找到可用的最小组标识符。因此，运行该算法总共需要 $O(|E|)$时间。初始化时的图补步骤也需要 $O(|E|)$时间。

3.3 部分答案与聚合

第3节.2描述了FedGroup最关键的一步，即为数据所有者寻找组。完成分组后，属于同一组的数据所有者可以避免分别向各自的查询答案中注入拉普拉斯噪声。相反，每个组会汇总来自数据所有者的部分答案，并向部分汇总答案中注入一次拉普拉斯噪声。

最后一步，服务提供商 S 从所有组收集部分噪声答案g_1、.....、 g_λ。这些噪声答案经过汇总（执行求和）后，再返回给终端用户。

3.4 扩展至范围聚合查询

本文聚焦于基于位置的服务系统中最基础的原语——范围计数查询。不过，我们提出的技术也可以轻松扩展到其他聚合查询，包括范围求和和平均值。

要将 FedGroup 扩展到求和，重要的一项扩展是为每个组注入的噪声尺度（即敏感度）。由于现在空间数据库中的每条数据记录对最终查询结果的影响不再仅为 +/-1，因此我们需要注入最坏情况下的噪声尺度，以满足差分隐私。为避免注入无界的拉普拉斯噪声，可以进行截断，即，将所有数据记录截断，使某个属性小于或等于截断参数 θ。

扩展到平均值很直接，因为平均值可以通过求和/计数计算得出。

4 实验研究

本节首先在第 4.1 节介绍实验设置，并在第 4.2 节呈现详细的实验结果。

4.1 实验设置

数据集。 我们在实验中使用了真实世界地理社交网络数据集Gowalla [6]以及一个随机生成的合成数据集。
基线方案。 我们将所提出的FedGroup（简称**FG**）方法与差分隐私基线（简称**DP**）和MPC基线（简称**MPC**）进行比较。

对于FedGroup，我们还实现了一种基于团大小的变体（一种基于启发式的解决方案）。该变体按大小降序列出图中的团，并不断将最大的团加入一个新组，直到所有节点都被分配到某个组中。我们将该变体命名为FedGroup-穷举法（简称**FG-Ex**）。请注意，在图中寻找最大团通常是一个NP难问题，因此我们预计该变体仅适用于小型图。

指标。 我们关注以下端到端指标：

• 平均相对误差（MRE）：与真实结果相比的误差比率，即， |返回结果 - 真实结果|/真实结果，并在重复查询中取平均值。

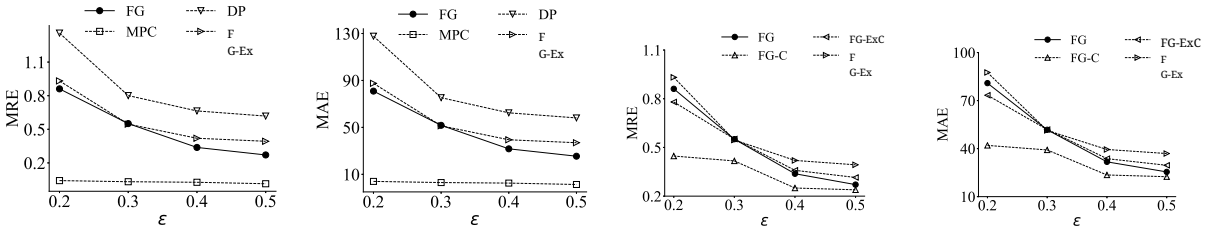
• 平均绝对误差（MAE）：返回结果与真实结果相比的绝对误差，即， |返回结果 - 真实结果|，在重复查询中取平均值。

• 查询评估时间：执行空间计数查询所需的时间（以秒为单位），在重复查询中取平均值。

控制变量。

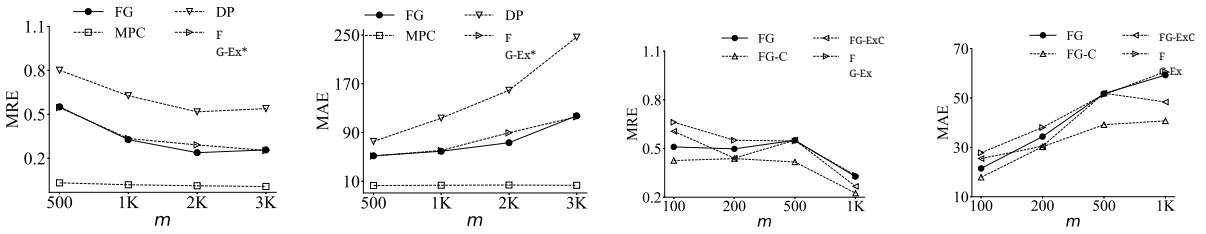
• 数据所有者数量： $m \in [\mathbf{500}, 1K, 2K, 3K]$。对于可扩展性测试： $m \in [100K, 250K, 500 \quad K, \ 1M]$

• 隐私预算： $\epsilon \in [0.2, \mathbf{0.3}, \mathbf{0.4}, 0.5]$。所考虑的隐私预算相对较小，因为这是每个数据所有者的个人预算，而重复查询可能会导致预算线性组合。



（a）平均相对误差与 ϵ的关系。　（b）平均绝对误差与 ϵ的关系。　（c）噪声与清晰。　（d）噪声与清晰。

图3：不同方法的端到端查询准确率与隐私预算 ϵ的关系。（Gowalla数据集，$m = 500$.）



（a）平均相对误差与 m的关系。　(b) 平均绝对误差与 m。　(c) 噪声与清晰。　(d) 噪声与清晰。

图4：不同方法的端到端查询准确率与联邦数量 m 的关系。（Gowalla数据集，$\epsilon = 0.3$。）

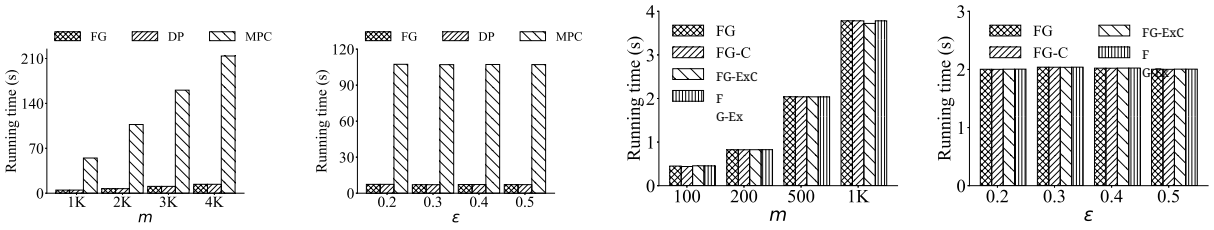
4.2 实验结果

在此，我们展示查询准确度方面的结果，以首先验证我们工作的动机——与差分隐私基线相比，我们提出的解决方案 FedGroup 能够提供更高的查询准确度（最高提升50%）。我们还比较了查询执行的运行时间，以说明即使数据规模适中，MPC基线也难以实际应用，而我们的解决方案 FedGroup 则具有出色的可扩展性。总体而言，与MPC基线相比，我们在效用与效率权衡方面表现更佳。

查询准确度。我们测试了不同方法的查询准确度（使用平均相对误差和平均绝对误差）。结果如图3和图4所示。在不同隐私预算下，FedGroup始终大幅优于差分隐私基线（提升30-50%）。在不同输入规模（联邦数量 m）下，FedGroup也显著提升了准确度。

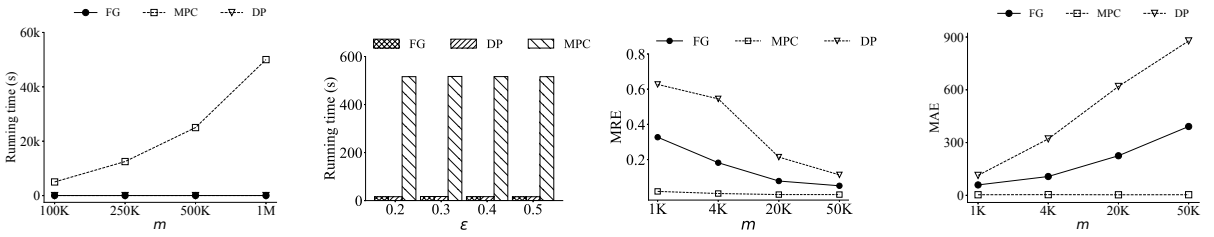
变化 ϵ。图 3 展示了我们在采用 Gowalla 数据集的、由 $m = 500$ 个数据所有者组成的空间联邦中，针对不同隐私预算 ϵ 测试各对比方法时得到的查询准确度结果。FedGroup（FG）的表现明显优于差分隐私基线。具体而言，如图 3a 所示，差分隐私基线在 $\epsilon = 0.3$ 时的平均相对误差为 0.801（80.1%），而 FG 将误差降至 0.551（55.1%），提升了 $(0.801 - 0.551)/0.801 * 100\% = 31.2\%$。

随着 ϵ 增大，表明隐私要求更加宽松，所有方法的查询准确度通常都会提高，表现为 MRE 和 MAE 更小。在 $\epsilon = 0.5$ 处（如图 3a 和图 3b 所示），差分隐私基线的 MRE 和 MAE 分别为 0.618 和 58.069。相比之下，FG 的 MRE 和 MAE 分别降至 0.271 和 25.448。FG 相较于差分隐私的查询准确度提升约为 56.1%。这一准确度提升比 $\epsilon = 0.3$ 的提升（31.2%）更显著，原因在于 ϵ 增大时，构建出的空间相似性图更加准确（从而能够正确识别更多边），而 FG 可以将数据所有者分配到数量更少的组中，最终



(a) 变化 m。 (b) 变化 ϵ。 (c) 噪声与清晰。 (d) 噪声与清晰。

图5：不同控制变量下的效率（运行时间）。（Gowalla数据集）



(a) 时间与 m。 (b) 时间与 ϵ。 (c) 平均相对误差与 m。 (d) 平均绝对误差与 m。

图6：可扩展性测试。（合成数据集）

使端到端查询误差更小。

值得注意的是，计算开销较大的 MPC基线在所有 ϵ 上都具有最佳的查询准确度（得益于 $O(1)$ 噪声）。尽管它不适用于真实世界数据规模，我们仍在此列出其结果，以求完整。

变化的 m。 在固定隐私预算 ϵ 的情况下，我们测试了不同方法在不同 m 下的查询准确度。结果（图4）表明，FG 始终显著优于差分隐私基线。随着 m 从 500 增至 3K，差分隐私基线的平均绝对误差从 75.3 增至 246.5。相比之下，我们的 FG 平均绝对误差更低，仅从 51.8 增至 117.4。从图4b可以看出，FG 的平均绝对误差增长速度比差分隐私基线更慢。

随着 m增加，真实查询计数也会增加。如图4a所示，这使所有方法的平均相对误差呈下降趋势。当 $m = 2K$时，差分隐私基线的平均相对误差为0.517，而我们的FG的平均相对误差为0.238，提升了53.97%。对于不同的 m，差分隐私的平均相对误差约为我们FG的 2×倍（表现更差）。

查询效率。 随后，我们进行了实验，以测试不同方法的查询效率。结果如图5所示。合成数据集上的可扩展性测试结果如图6所示。

Gowalla数据集。 对于范围为1K–4K的较小 m ，我们使用Gowalla数据集，在图5中展示了不同方法的查询执行时间。使用MPC方法时，查询时间随 m 增长而线性增加（图5a），并且在不同 ϵ下保持在同一水平（图5b），但其耗时比FG和差分隐私高出数个数量级。

图5c和图5d表明，使用真实图还是噪声图对运行时间的影响不大。随着 m 增加，运行时间大致呈线性增长；但当 $m = 1K$时，所有FG变体均能在4秒内完成，而耗时较长的MPC基线则无法与之相比。

可扩展性测试。 我们还通过在合成数据集上进行测试，进一步验证 FG方法的可扩展性。图6a 展示了当我们将 m 扩展到一百万时的查询执行时间。

尽管 MPC 的运行时间呈线性增长，但当 $m = 1M$时，执行一次查询需要 13 小时以上。这在现实世界的基于位置的服务应用中是不可接受的。相比之下，在最大输入规模下，差分隐私方法和我们的FG方法都能在 20 秒内完成查询。

对于给定的 m，不同方法的运行时间在不同 ϵ 下变化不大（图6b）。除了效率结果之外，我们还测试了这些方法在合成数据集中的超大输入上的表现，并给出了部分查询准确度结果（如图6c–图6d所示）。这些结果进一步验证了我们的FG方法始终优于效率相当的差分隐私方法。

总结。 我们开展了广泛的实验，以评估所提出的FedGroup方法的有效性（以查询准确度衡量）和效率。在查询准确度方面，FedGroup相比差分隐私基线最高提升了50%。尽管计算开销较大的MPC基线能够提供最佳准确度，但其低效率阻碍了其在实际场景中的应用。FedGroup的运行速度比MPC基线快数个数量级，并且能够在20秒内为一百万个数据所有者执行一次查询。

5 相关工作

本节首先回顾通用数据联邦方面的研究工作，然后讨论空间数据联邦领域的具体研究，以及其他兼顾数据隐私的空间数据管理技术。

通用数据联邦。数据联邦的概念可追溯至 20 世纪 90 年代，当时出现了管理自治但相互协作的数据库系统的需求 [17]。近年来，越来越多的研究致力于优化并采用安全多方计算来构建隐私保护数据联邦 [2, 3, 4]。Bater 等人首先实现了一个基于 MPC 的数据联邦系统 SMCQL [2], ，随后又将其改进为 Shrinkwrap [3] ，通过差分隐私减少查询执行期间插入的虚拟记录数量，从而使计算不可观测。其后续工作 SAQE [4],通过考虑近似查询处理，进一步改善了效用、效率和隐私之间的权衡；这是实现差分隐私所需注入噪声所带来的附带效果。Conclave [26] 是另一个采用安全查询处理的系统，即查询执行期间私有数据是机密的。

这些基于安全性的系统能够提供强有力的隐私保证，但效率问题是其瓶颈。大多数研究仅限于在安全计算中涉及两个参与方，实验结果 [23] 表明，当前系统在真实世界数据规模下仍远未达到可扩展的水平。例如，对私有表 L（由多个参与方共享）和公共表 R 进行连接，即使 R 仅包含 100 个对象，也需要 25 分钟以上。

专用空间数据联邦。为构建更专用、优化程度更高的空间数据联邦，Shi 等人 [18]首先研究了如何在不考虑隐私的情况下高效执行近似范围聚合查询。随后，研究人员构建了一个基于 MPC 的系统 HuFu [23]，并针对空间算子进行了专门优化。尽管与其他现有工作（例如 Conclave ）相比，该系统取得了显著改进，但运行时间仍是一个主要问题。在我们的实验中，我们证明，即使是最简化的 MPC基线，其运行速度也比我们提出的解决方案慢了几个数量级。

隐私保护空间数据管理。 与此同时，也有大量研究利用差分隐私或本地差分隐私（LDP）进行空间数据管理。差分隐私模型会掩盖单条位置记录 [7] 或整条轨迹 [11]的存在。还有其他研究致力于保护

位置数据 [27, 19]，或采用 LDP 及其变体，为保护位置数据 [5, 1, 8]提供理论保障。差分隐私和本地差分隐私模型都不同于我们的隐私模型，因为在我们的模型中存在多个数据所有者。

6 结论

本文针对联邦隐私保护范围计数（FPRC）问题展开研究，其中联邦（数据所有者）的数量众多。我们利用数据所有者之间的社会关系（例如家庭成员之间的关系），提出了一种基于分组的框架，称为 FedGroup。该框架只需为每个组注入一次拉普拉斯噪声，从而减少所需的噪声量；而差分隐私基线要求为每个数据所有者注入一次噪声，导致噪声尺度大得难以接受，甚至超过真实查询答案。此外，FedGroup 的效率远高于基于多方计算的基线，尤其是在数据所有者数量扩展至百万级时。大量实验结果验证了我们提出的方案在准确性方面的提升和效率方面的优势。未来，我们将探索更先进的数据压缩方法 [15, 14, 13]，以实现更好的隐私与效用权衡。

参考文献

[1] Miguel E. Andres、Nicolas E. Bordenabe、Konstatinos Cjhatzikokolakis 和 Catuscia Palamidessi。地理不可区分性：基于位置的系统的差分隐私。发表于*CCS*，2013年。[2] Johes Bater、Gregory Elliott、Craig Eggen、Satyender Goel、Abel N Kho 和 Jennie Rogers。SMCQL：面向私有数据网络的安全查询处理。*VLDB*，2017年。[3]Johes Bater、Xi He、William Ehrich、Ashwin Machanavajjhala 和 Jennie Rogers。Shrinkwrap：差分隐私数据联邦中的高效 SQL 查询处理。*VLDB*，2018年。[4] Johes Bater、Yongjoo Park、Xi He、Xiao Wang 和 Jennie Rogers。SAQE：面向数据联邦的实用隐私保护近似查询处理。*VLDB*，2020年。[5] Rui Chen、Haoran Li、A Kai Qin、Shiva Prasad Kasiviswanathan 和 Hongxia Jin。本地设置下的私有空间数据聚合。发表于*ICDE*，2016年。[6] Eunjoon Cho、Seth A. Myers 和 Jure Leskovec。友谊与移动性：基于位置的社交网络中的用户移动。发表于*KDD*，2011年。[7] Graham Cormode、Cecilia Procopiuc、Divesh Srivastava、Entong Shen 和 Ting Yu。差分隐私空间分解。发表于*ICDE*，2012年。[8] Teddy Cunningham、Graham Cormode、Hakan Ferhatosmanoglu 和 Divesh Srivastava。采用本地差分隐私实现真实场景中的轨迹共享。*VLDB*，2021年。[9] Bowen Du、Yongxin Tong、Zimu Zhou、Qian Tao 和 Wenjun Zhou。电动汽车共享的需求感知充电桩规划。发表于*KDD*，页码 1330–1338，2018年。[10] Dawei Gao、Yongxin Tong、Jieying She、Tianshu Song、Lei Chen 和 Ke Xu。空间众包中的 Top-k 团队推荐。发表于*WAIM*，2016年。

[11] Xi He、Graham Cormode、Ashwin Machanavajjhala、Cecilia M Procopiuc 和 Divesh Srivastava。DPT：使用分层参考系统进行差分隐私轨迹合成。*VLDB*，2015年。[12] Xi He、Jennie Rogers、Johes Bater、Ashwin Machanavajjhala、Chenghong Wang 和 Xiao Wang。数据库系统的实用安全与隐私。发表于*SIGMOD*，2021年。[13] Qiyu Liu、Yanyan Shen 和 Lei Chen。Lhist：面向海量空间数据的多维直方图学习。发表于*ICDE*，2021年。[14] Qiyu Liu、Yanyan Shen 和 Lei Chen。HAP：基于增强鸽巢原理的高效汉明空间索引。发表于*SIGMOD*，2022年。[15] Qiyu Liu、Libin Zheng、Yanyan Shen 和 Lei Chen。面向数据流的稳定学习布隆过滤器。*VLDB*，2020年。[16] Jieying She、Yongxin Tong、Lei Chen 和 Tianshu Song。感知反馈的社交活动参与者安排。发表于*SIGMOD*，2017年。[17] Amit P Sheth 和 James A Larson。用于管理分布式、异构和自治数据库的联邦数据库系统。*CSUR*，1990年。[18] Yexuan Shi、Yongxin Tong、Yuxiang Zeng、Zimu Zhou、Bolin Ding 和 Lei Chen。在大规模空间数据联邦上进行高效近似范围聚合。*IEEE Trans.Knowl. Data Eng.*，35(1):418–430，2023年。[19] Qian Tao、Yongxin Tong、Zimu Zhou、Yexuan Shi、Lei Chen 和 Ke Xu。空间众包中的差分隐私在线任务分配：一种基于树的方法。发表于*ICDE*，2020年。[20] Qian Tao、Yuxiang Zeng、Zimu Zhou、Yongxin Tong、Lei Chen 和 Ke Xu。面向多工作者的实时空间众包任务规划。发表于*DASFAA*，2018年。[21] Technode。阿里巴巴旗下高德地图于北京推出出租车网约车平台，2021年。[22] Hien To、Cyrus Shahabi 和 Li Xiong。通过不可信服务器实现空间众包中的隐私保护在线任务分配。发表于*ICDE*，2018年。[23] Yongxin Tong、Xuchen Pan、Yuxiang Zeng、Yexuan Shi、Chunbo Xue、Zimu Zhou、Xiaofei Zhang、Lei Chen、Yi Xu、Ke Xu 等。Hu-fu：面向数据联邦的高效安全空间查询。*VLDB*，2022年。[24] Yongxin Tong、Ye Yuan、Yurong Cheng、Lei Chen 和 Guoren Wang。时空众包数据管理技术综述。*J.Softw.*，28(1):35–58，2017年。[25] Yongxin Tong、Zimu Zhou、Yuxiang Zeng、Lei Chen 和 Cyrus Shahabi。空间众包：综述。*VLDBJ*，2020年。[26] Nikolaj Volgushev、Malte Schwarzkopf、Ben Getchell、Mayank Varia、Andrei Lapets 和 Azer Bestavros。Conclave：面向大数据的安全多方计算。发表于*EuroSys*，2019年。[27] Yonghui Xiao 和 Li Xiong。在时间相关性下利用差分隐私保护位置。发表于*CCS*，2015年。

[28] Samee Zahur 和 David Evans。Obliv-c：一种用于可扩展数据无关计算的语言。*IACR Cryptol. ePrint Arch.*，2015年：1153，2015年。[29] Fan Zhang、Ying Zhang、Lu Qin、Wenjie Zhang 和 Xuemin Lin。参与度与相似度相遇：社交网络上的高效（k，r）-核计算。*VLDB*，2017年。

表1：本文使用的主要符号。

符号	说明
m	数据所有者（数据孤岛）的数量
u_1、...、u_m	数据所有者
$D_1, \ldots, D_m$	数据所有者 i 的数据库 D_i （又称数据孤岛）
D	$D = \cup_{i=1}^{m} D_i$ 是空间数据联邦。
ϵ	该查询的隐私预算
$G_s = (V, E)$	空间相似性图
$w, w(e)$ $\tilde{w}$	边的权重 w e，以及噪声权重 $\tilde{w}$
$T, \lvert T \rvert$	网格结构和网格数量
$v_i, \tilde{v_i}$	数据所有者 u_i 的计数向量 v_i 及其噪声版本。
r	用于判断两个数据所有者是否相似的阈值（根据其数据库的空间相似度）数据库）
g	一个 r-组
$g_1, g_2, \ldots, g_\lambda$	分配给数据所有者的组
λ	组数
d	图的最大度

A 其他技术细节

为便于阐述，本文使用的主要符号汇总于表1。

A.1 问题定义

为便于理解，我们使用图1中的一个玩具示例来说明上述概念。

示例 1。 如图*1*所示，共有 m 个数据所有者（以移动设备图标表示），每个数据所有者拥有一个数据库 D_i。所有数据孤岛的并集共同构成空间数据联邦 $D = \cup_{i=1}^{m} D_i$。每个数据孤岛 D_i 都有一个空间数据库 *[23, 18, 9]*, ，图中以网格上的黑点表示。此示例中的范围计数查询 Q 以红色虚线圆表示（图中省略了中心点 q_0 和范围 r）。圆内的红点是落在该查询范围内的数据记录 Q。例如，对于第一个数据所有者 u_1 的数据库 D_1，计数为 $Q(D_1) = 3$。对于第 i 个数据所有者 u_i 的数据库 D_i，计数为 $Q(D_i) = 2$。此示例中的范围计数查询应返回：

$$Q = \sum_{i=1}^{m} Q(D_i). \tag{5}$$

这是每个数据孤岛 D_i 的部分范围计数 $Q(D_i)$ 之和。

A.1.1 数据隐私和计算隐私

在我们的问题中，我们同时考虑输入数据隐私（即，数据隐私）和空间数据联邦中的无感知查询执行（即，计算隐私） [12] 。

对于数据隐私，我们首先确定需要保护的私有数据。在我们的问题中，所涉及的私有数据是每个数据所有者 u_i 的空间数据库 D_i 。

关于计算隐私，我们指的是计算过程中发生的隐私泄露。它指基于私有输入的函数的指令轨迹、程序计数器和执行时间，而私有输入指的是每位数据所有者 u_i 的 D_i 。

A.1.2 对手模型

对于（1）终端用户：终端用户向服务提供商发出范围计数查询 $Q(r,\ q_0)$，˜并接收查询结果 Q。终端用户会好奇，并试图从查询结果 Q中获取尽可能多的隐私˜信息。这里的隐私信息指每位数据所有者 u_i 拥有的私有数据库 D_i。

对于（2）服务提供商 S：服务提供商从终端用户处接收到查询 $Q(\cdot)$ 后，负责协调并执行该查询，并获得一个答案 Q，以返回给终端用户。在查询执行期间，服务提供商可能会与数据所有者u_1、. . . 、 u_m 进行交互，并接收中间结果。例如，在图1中，S 分别从 D_1接收部分结果 3，从 D_i接收 $Q(D_i)$，并从 D_m接收 2。 S 也被假设为半诚实的，因此 S 会尽可能获取有关 D_1、. . . 、 D_m 的更多隐私信息。此外，S 还会试图在计算过程中获取隐私泄露期间，这里的计算过程是指评估过程，包括其与数据所有者协调执行查询执行期间的程序计数器、执行时间和指令轨迹。

对于（3）数据所有者：每个数据所有者 u_i 也被假设为半诚实的，并且好奇其他数据所有者的隐私信息。例如，u_1 可能会对 u_3感到好奇，并试图在查询执行期间获取有关 D_3 的隐私信息。当 S 协调各个数据所有者共同计算结果时，泄露可能会在 计算过程中发生。

A.2 基线方案

我们先通过一个案例研究说明图1为何不满足定义 4 中的隐私要求。然后，我们介绍两个分别采用差分隐私和 MPC 的简易基线方案。

A.2.1 隐私失效案例研究

不难看出，图1所示的计算过程不满足定义 4 中 FPRC 的隐私要求，因为各方计算出的真实答案会直接共享给半诚实服务提供方 S。事实上，R1 和 R2 均遭到违反，而 R3 得以满足，因为其中没有涉及多方计算。详细分析将在附录中给出。

A.2.2 差分隐私基线

我们通过应用拉普拉斯机制来满足隐私要求，从而引入一种差分隐私简易基线方案，随后分析由此引入的误差规模。

对于每个数据所有者。在接收到查询 $Q(r,\ q_0)$ 后，每个数据所有者 u_i 通过在其私有数据库上运行该查询来计算部分真实结果，从而获得 $Q(D_i)$。随后，注入单个拉普拉斯噪声实例以获得：

$$\tilde{Q}(D_i) = Q(D_i) + \text{Lap}(1/\epsilon). \tag{6}$$

将带噪声的部分结果发送给服务提供商 S。

˜**服务提供商** S。在从数据所有者处接收到带噪部分计数 $Q(D_i)$ 后，S 通过求和汇总所有部分计数，以获得：

$$\tilde{Q} = \sum_{i=1}^{m} \tilde{Q}(D_i). \tag{7}$$

˜最后， S 将该值 Q 返回给终端用户。

隐私分析。 不难看出，R1-R3 均得到满足，因为对所有数据所有者而言，每个部分计数 $Q(D_i)$的计数查询敏感度均为 1。

˜**效用分析**。 首先，我们注意到， Q 的随机性取决于公式（6）中随机生成的拉普拉斯噪声。由于每个拉普拉斯噪声随机变量的均值为 0， m 个此类随机变量的求和均值也为 0。因此，公式（6）中返回的噪声答案 $\tilde{Q}$ 是真实答案 Q的无偏估计量。

因此，为衡量效用，我们将误差定义为返回的噪声答案 $\tilde{Q}$的方差。

$$\text{Var}(\tilde{Q}) = \text{Var}(\sum_{i=1}^{m} \text{Lap}(1/\epsilon)) = m \cdot \text{Var}(\text{Lap}(1/\epsilon)) = 2m/\epsilon^2. \tag{8}$$

˜由此可知， Q 的误差随联邦数量 m线性增长。在我们的设定中， m 可能很大（例如，数百万台移动设备），因此我们预计差分隐私简易基线会产生显著误差。

A.2.3 MPC基线

然后，我们展示一种通过利用 MPC技术，让不同数据所有者共同计算查询结果的简单 MPC基线。其基本思路是， m 数据所有者与服务提供商 S共同计算带噪声的求和：

$$\tilde{Q} = \sum_{i=1}^{m} Q(D_i) + \text{Lap}(1/\epsilon), \tag{9}$$

在多方计算协议中，将各自的私有数据 D_i 作为私有输入。

真实查询结果中只注入一次拉普拉斯噪声。然后，˜服务提供商 S 将计算结果 Q 返回给终端用户。

隐私分析。 公式（9）中注入的拉普拉斯噪声确保最终返回的查询结果满足定义 4中的 R1。由于使用了多方计算协议，R2 和 R3 也均得到满足，并且每个数据所有者的私有数据库 D_i 均严格保密，不会向其他参与方披露任何信息。

效用分析。 简单安全多方计算基线的效用严格优于差分隐私基线，因为其中只注入了一次拉普拉斯噪声。与差分隐私基线的 O（m）误差规模相比，MPC基线的误差为 O(1)。

时间复杂度。 MPC基线对 m 个数据所有者执行安全求和，并在最后加入一次拉普拉斯噪声。时间复杂度与数据所有者的数量呈线性关系： $O(m)$，但数据所有者之间的通信成本很高。每个数据所有者 u_i 都需要等待前面 $i-1$ 个数据所有者返回部分结果，才能继续计算过程。

A.2.4 隐私失效案例研究

如图1所示，每个数据所有者 u_i 都会计算与 相关 的部分范围计数，数据孤岛˜D_i。服务提供商 S 随后通过求和 $Q =\sim\sum_{i=1}^{m} Q(D_i)$ 汇总所有部分计数。然后， S 将 Q 返回给终端用户。如下所示，这个示例违反了定义 4 中的多项隐私要求。

首先，定义 4 中的 R2 显然遭到了违反。部分计数被直接返回给 S。由于我们假设 S 是半诚实的，并且对手可能存在于 S，因此访问部分查询结果 $Q(D_i)$会直接威胁私有数据库 D_i的隐私。更改 D_i 中的单条记录会直接影响部分计数结果 $Q(D_i)$，显然这一步不满足 R2 中的隐私保证。

~接下来，定义 4 中的 R1 不成立，因为 $Q = \sum_{i=1}^{m} Q(D_i)$ 会被直接返回给最终~用户。请注意，Q 等于真实计数，如示例 1 所述。在任意数据孤岛中更改任何特定的~记录，都会以敏感度 1 直接影响查询结果 D_i Q，因此无法提供任何隐私保证。由于原因显而易见，我们省略证明其不满足 ϵ-差分隐私的形式化论证。

最后，我们注意到定义 4 中的 R3 实际上是满足的，因为计算完全由各数据所有者独立执行。严格来说，半诚实服务提供商 S 仍可通过侧信道攻击观察各数据所有者的查询执行时间，以推断私有数据库 D_i的规模。不过，由于此步骤不涉及多方计算，我们认为 R3 得到了满足。在我们的对手模型中，只有当服务提供商 S 与其他数据所有者之间涉及多方计算时，才会发生侧信道攻击。

A.2.5 DPbaseline 隐私分析

首先，我们证明定义 4 中的 R2 得以满足。原因很简单：对于所有数据孤岛，部分计数 $Q(D_i)$的计数查询敏感度均为 1。加入与相应敏感度相对应的拉普拉斯噪声，即可提供 ϵ-差分隐私。

然后，我们通过应用差分隐私的并行组合定理，证明定义 4 中的 R1 得以满足。考察整个数据库 $D = \cup_{i=1}^{m} D_i$，即所有数据孤岛的并集，那么每个数据孤岛 D_i 都是 D 的不相交子集。更改任何数据孤岛 D_i 中的一条记录，只会更改 D 中的一条记录，反之亦然。因此，差分隐私的并行组合定理适用，也就是说，公式（7）计算出的最终结果满足 ϵ-差分隐私，因为每个数据所有者的部分结果（由公式（6）计算）都满足 ϵ-差分隐私。

稻草人差分隐私基线也满足 R3，因为在查询评估的整个过程中不涉及多方计算。

A.3 FedGroup 细节

下面我们提供一个玩具示例，以更好地说明这一概念。

示例 2（空间相似度）。在图7中，D_1 是数据所有者 u_1 的空间数据库，D_2 是数据所有者 u_2的空间数据库。图中展示了网格结构 T，其在空间域上包含 $|T| = 9$ 个网格。因此，我们可以将 D_1的计数向量构建为$v_1 = [0\ 1\ 1\ 1\ 0\ 0\ 2\ 5\ 0]$，其中每个条目对应于特定网格的计数，从左上角开始，逐行向下排列。类似地，D_2的计数向量为 $v_2 = [0\ 0\ 1\ 1\ 0\ 0\ 1\ 4\ 0]$。然后，$D_1$ 与 D_2 之间的相似度为 $sim(D_1, D_2) = \cos(v_1, v_2) = v_1 \cdot v_2 / \|v_1\| \|v_2\| = 24/24.658 = 0.973$。

下面给出一个空间相似性图的玩具示例。

示例 3（空间相似性图 G_s）。图7给出了一个包含 4 个数据所有者的空间相似性图玩具示例。在 G_s 中，每对数据所有者之间都有一条边。边的权重设为公式（2）中定义的空间相似度。如示例 2 所示，u_1 与 u_2 之间的相似度等于 $sim(D_1, D_2) = 0.973$，如上所示

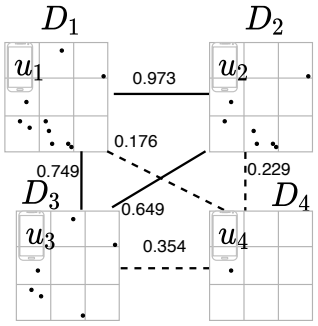


图7：空间相似性图示例。

算法3： 非隐私地构造 G_s

输入： $u_1, \ldots, u_m, D_1, \ldots, D_m, r$.

输出： $G_s = (V, E)$。

```
V := {u_1, ..., u_m}
E := {}
对每个 u_i ∈ V 执行
    对每个 u_j ∈ V 且 j > i 执行
        e := (u_i, u_j)
        e.权重 := sim(D_i, D_j)
        如果 e.权重 > r 则
            插入 e 到 E
返回 G_s = (V, E)
```

对应的边 $e = (u_1, u_2)$。类似地，$w(u_1, u_3) = 0.749$, $w(u_3, u_2) = 0.649$。请注意，这三条边的边权重都大于阈值 $r = 0.5$（该阈值 r 将在第 *3.2* 节的 r-团中使用），在图中以实线表示。其余三条边，即i.e.(u_1, u_4)、(u_3, u_4)、(u_2, u_4)，其边权重小于 r，在图中以虚线表示。

MPC选项。确保满足所有隐私要求的最直接方法是使用 MPC技术，在保证机密性的同时计算边权重。每对数据所有者 u_i和 u_j 使用 MPC 根据公式（2）安全地计算其相似度，其中其私有数据库 D_i, D_j 和计数向量 v_i, v_j 被视为混淆电路的私有输入。因此，每对数据所有者 u_i 和 u_j 使用两方MPC，在保证机密性的情况下计算$w = \mathsf{sim}(D_i, D_j)$，并将权重 w 发送给服务提供商 S。S 将s 设为边 $e = (u_i, u_j)$的边权重，并将 e 插入边集 E。

DP选项。我们还可以在差分隐私中应用标准拉普拉斯机制，为每个数据孤岛提供 ϵ-差分隐私，从而为每个数据所有者 u_i 获得计数向量 $\tilde{v}_i$ 的噪声版本（因为每个网格中计数的敏感度为 1，相对于数据孤岛中一条记录的变化而言）。然后，可以使用 $\cos(\tilde{v}_i, \tilde{v}_j)$ 为每对 u_i, u_j计算噪声相似度分数。随后将噪声权重 $\tilde{w}$ 用作代理分数，作为两个数据所有者私有输入的空间相似度分数。服务提供商 S 会像非私有版本一样，为每对数据所有者计算噪声分数。然而，我们现在不再使用根据私有输入计算出的精确 权重，而是使用根据噪声计数 $\tilde{v}_i$ 计算出的噪声分数作为每条边的权重。

显然 中，DP选项会在图构建 n

过程中引入大量噪声，而原始权重 w 可能与噪声权重 w'相差很大，具体取决于隐私参数 ϵ 以及拉普拉斯噪声的随机生成过程。此外，网格结构也可能影响误差规模，因为余弦相似度函数会对两个噪声计数进行点积运算，而且误差至少会随网格数量 $|g|$线性增长。从第 7 行到第 9 行，我们首先将计算得到的噪声权重 $\tilde{w}$ 与下界阈值 r_l 进行比较。如果它小于 r_l，我们就直接剪枝删除这条边。然后，我们将其与上界阈值 r_u进行比较。如果 $\tilde{w}$ 大于 r_u，我们就直接将这条边插入图中。

第 12 行展示了临界情况，此时我们会调用计算相似度函数的高开销安全版本。安全计算的详细步骤从略，其主要思路是输入私有输入 D_i 和 D_j，并使用安全原语（包括安全加法、乘法和条件判断），以确保私有输入在整个计算过程中始终保密。

证明。 我们将最小团划分（MCP）问题归约为 DOG问题。由于 MCP问题是 NP难问题，因此 DOG 具有相同的难度。

我们证明，DOG问题存在方差为 $2\lambda/\epsilon^2$ 的解，当且仅当 MCP问题存在具有 λ 个划分的解。

1. MCP $\Longrightarrow$ DOG：如果 MCP 存在具有 λ 个划分的解，我们只需将该最优划分作为 DOG问题的分组返回。由于 MCP 中的划分彼此不相交，因此它们构成 λ 个有效组 g_1，g_2，……，g_λ。由于 MCP 的解返回的是团的划分，因此每个划分本身也是一个团，满足每个组都是 r-团的要求。因此，该解提供了一个包含 λ 个组的有效分组。随后，根据引理 1，我们知道该解为 FPRC问题提供了一个方差为 $2\lambda/\epsilon^2$ 的噪声答案。

2. DOG $\Longrightarrow$ MCP：根据引理1，如果 DOG问题有一个能够提供方差为 $2\lambda/\epsilon^2$的噪声答案的解，我们便知道组数为 λ。由于g_1、……、 g_λ 中的每个组互不相交，且每个组都是一个 r-团，符合定义6的要求，因此分组的解也是 MCP问题的解，它提供了一个由互不相交的团构成的划分，且团数为 λ。

□

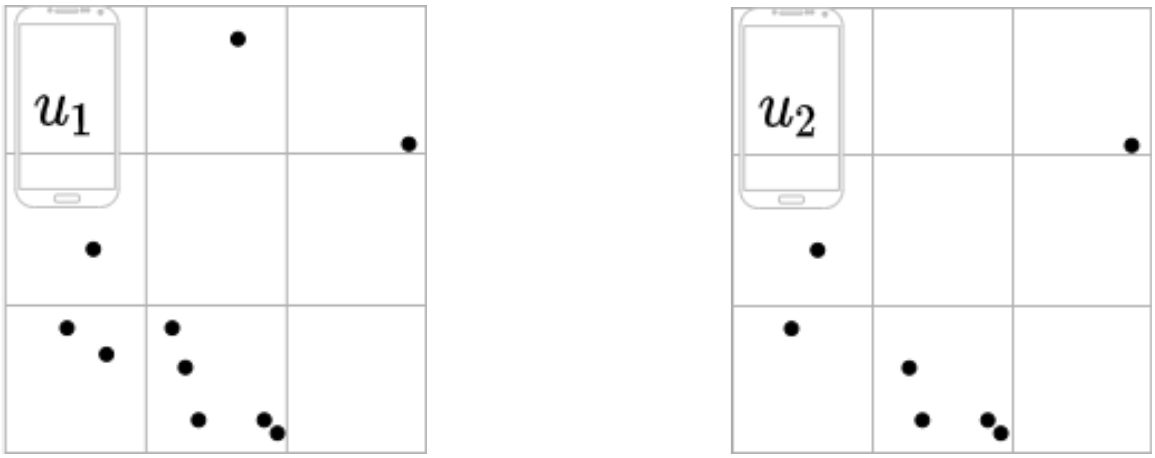
示例4。我们以图*7*为输入，通过该示例演示贪心解。原图 G_s 包含*4*个节点： u_1-u_4。它仅包含*3*条边：(u_1, u_2)、(u_1, u_3)和(u_2, u_3)。补图 $G_c = (V, E_c)$的顶点集与原图相同，为 V，且包含*3*条边：(u_1, u_4)、(u_2, u_4)、(u_3, u_4)；这些边在原图的边集中均不存在 E。

首先，我们找到顶点 u_1 并将其分配到第 *1* 组。已分配$[1] = 1$，以及 $\lambda = 1$ ．

然后，我们查看顶点 u_2。 u_2 在 G_c中只有 *1* 个邻居，即 u_4 ，这是由于边(u_2, u_4)，但 u_4 尚未分配到任何组。因此，最小的可能组标识符是 *1*。所以， u_2 也被分配到第 *1* 组。已分配$[2] = 1$，以及 $\lambda = 1$。

接下来，我们继续处理 u_3。与 u_2类似， u_3 唯一的邻居是 u_4 ，这是由于边(u_3, u_4)。最小的可用组标识是 *1*。因此， u_3 被分配到第 *1* 组。已分配$[3] = 1$，以及 $\lambda = 1$。

最后，我们来看顶点 u_4。其他所有顶点都是 u_4的邻居，并且它们都使用相同的组标识符 $gid = 1$。因此，它使用最小的可用组标识符 $gid = 2$，这是一个新的组标识符。现在，已分配$[1] = 2$和 $\lambda = 2$。返回的结果为 g_1、 g_2，其中$g_1 = \{u_1, u_2, u_3\}$ 和 $g_2 = \{u_4\}$。组数为 $\lambda = 2$，这表明我们找到了 *2* 个组。



(a) 数据孤岛 D_1。 (b) 数据孤岛 D_2。

图8：两个数据孤岛 D_1 和 D_2的示例。

A.3.1 受约束版本

正如我们在 FedGroup 的关键思想中简要讨论过的那样，即使在家庭（组）内部，每个数据所有者仍可能希望获得隐私保护。因此，我们提出了数据所有者分组（DOG）问题的两个变体，并将个人隐私要求纳入考量。

需要注意的是，我们牺牲了图的准确性（与构建 G_s的非隐私版本，即真实图相比），以满足提供经过证明的隐私保证的需求。然而，正如第4节的实验结果所示，我们提出的混合方案力求在效用与隐私的权衡之间实现出色的平衡。

给定置信区间后，可以根据给定的隐私预算 ϵ 和初始阈值 r，通过分析推导出下界和上界阈值 r_l 和 r_u 。在我们的实验中，我们将这两个参数视为超参数，而初步尝试已经取得了令人满意的结果。

接下来，我们证明贪心解的正确性。

定理4. 算法 *2* 返回的组是互不相交的 r-团，并且满足 *DOG* 问题的要求（定义*6*）。

证明. 我们用反证法证明每个组都是 r-团。假设某个组 g 不是团，即，存在 u_i、$u_j \in g$，且在原图 $\tilde{}$ $\tilde{}$ G_s 中两者之间没有边（即，$(u_i$、$u_j) \notin E$）。因此，在 G_s G_c 的补图中，存在这样一条边 $(u_i$、$u_j)$。然而，根据算法，在第 6 行，我们会为给定节点寻找未被其任何邻居使用的最小组标识符，这意味着u_i 和 u_j 不可能被分配到同一组 g。因此，这就导致了矛盾。g_1、g_2、$\ldots, g_\lambda$ 中的每个组都是 r-团。

这些组彼此不相交，因为我们只会将数据所有者 u 分配到一个组中一次，而且分配完成后不再更改。

因此，算法 2 返回的组是 r-团，且彼此不相交，满足 DOG问题（定义 6）中的要求。

□

我们将该解决方案称为贪心 算法，是因为在尝试确定应将数据所有者 u 分配到哪个组时，它总会从 u的邻居中选取最小的可用组标识。如果先前使用过的所有组标识都已被 u的某个邻居占用，那么我们

就增加组总数并使用一个新的组标识符。该解决方案的详细步骤见算法 2。

该算法使用一个名为 assigned 的数组来记录映射关系，以表明一个顶点（数据所有者）是否已被分配。如果第 i 个数据所有者已被分配到组标识符为 k 的组，则 assigned$[i] = k$。如果该数据所有者尚未被分配到任何组，则 assigned$[i] = 0$。

每个组 g_{gid} 都是一个组（以集合实现），其中 gid 是该组的组标识符。例如， g_1 包含第一组中的所有数据所有者。 g_{gid} 包含组标识符为 gid 的组中的所有数据所有者。对于未分配的顶点 u，在第 6 行（伪代码中省略了具体细节），我们会遍历 u 的邻居，并寻找一个尚未被任何邻居使用的最小组标识符。然后，将这个最小组标识符作为顶点 u 的组分配结果。

最后，返回已分配的组 g_1， g_2，……， g_λ。每个组都包含相应的数据所有者，而 λ 表示贪心解得到的组数。

效用分析。 在相似度计算中注入拉普拉斯噪声，不可避免地会降低判断 G_s 中是否存在边的准确性。由于采用了多方计算技术作为补充方法，我们能够确保在两个数据孤岛之间的空间相似度较高时，正确识别边。结果见第4节的实验结果。

定理 5。 算法 *1* 中的混合方案满足定义 *4* 中的所有隐私要求（*R1-R3*）。

证明。 我们先分析各数据孤岛发布的中间结果 D_i，并证明其满足 R2。对于数据所有者 u_i，唯一发布的中间结果是由私有数据孤岛 D_i 构造的噪声计数向量 $\tilde{v_i}$。真实计数向量 v_i 的构造方式如第 3.1 节所示，只需统计网格结构 g 中每个网格内的数据记录数量即可。 v_i 仅为各网格计数构成的向量，长度为 $|g|$。噪声计数向量 $\tilde{v_i}$ 是通过向每个网格添加尺度为 $1/\epsilon$ 的拉普拉斯噪声得到的。数据孤岛 D_i 中一条数据记录的变化，最多会使 v_i 中的一个元素改变，且改变幅度至多为 +/-1（也称为敏感度为 1）。因此，对于发布的中间结果，向噪声计数向量 $\tilde{v_i}$ 注入 Lap($1/\epsilon$)，足以提供ϵ-差分隐私。综上，R2 得以满足。

接下来，我们利用差分隐私的后处理性质和并行组合性质，证明 R1 得到满足。对于算法 1，输出是构建的空间相似性图 G_s，它~存储在服务提供商 S处。此时尚未向终端用户 Q 发布最终结果。因此，为证明 R1 得到满足，我们证明构建的 G_s 满足 ϵ-差分隐私，相对于任意数据孤岛中 D_i的一条记录发生变化的情况。我们聚焦于数据所有者 u_i的特定数据孤岛 D_i，并考虑一个相邻数据库 D_i'，它与 D_i 仅有一条数据记录不同。然后， u_i 向服务提供商 S发布噪声计数向量 $\tilde{v_i}$。如上所述（R2 得到满足）， $\tilde{v_i}$满足 ϵ-差分隐私。接下来，我们关注算法 1。对于第 7 行至第 9 行，仅使用 $\tilde{v_i}$。根据差分隐私的后处理性质，我们知道，向 E（或构建的 G_s）中插入任何边都不会违反 ϵ-差分隐私。对于第 12 行的边界情况，使用了多方计算协议，因此私有数据库严格保密。由此，我们得出结论：对于数据孤岛 D_i，构建的 G_s 满足 ϵ-差分隐私。最后，由于各数据孤岛 D_1、……、 D_m 彼此不相交，在其中任意一个数据孤岛中更改一条记录都不会影响其他数据孤岛的噪声计数，因此，根据差分隐私的并行组合定理，我们知道构建的G_s 满足 $\max_{i=1}^{m} \epsilon = \epsilon$-差分隐私。这里，每个数据孤岛使用相同的隐私要求 ϵ。

由于使用了多方计算协议，显然 R3 得到满足。私有输入仅在第 12 行使用，并被输入多方计算协议，因此数据的机密性得以保障。

□

A.3.2 DOG问题的受约束版本

正如我们在 FedGroup 的核心思想中简要讨论过的那样，即使在家庭（组）内部，每个数据所有者仍可能希望获得隐私保护。因此，我们提出了数据所有者分组（DOG）问题的两个变体，并将个人隐私要求纳入考虑。

在本节中，我们提出数据所有者分组（DOG）问题的两个变体，并将个人隐私要求纳入考虑。

定义7（DOG-Global）。*DOG-Global*问题的输入和输出与定义*6*中的*DOG*问题相同，但增加了一项要求：分配给每个数据所有者的组最多包含 t 个数据所有者，即，组的大小不得超过 t。

该变体称为DOG-Global问题，因为它对所有数据所有者施加了全局约束。每个数据所有者最多只愿意与 t 个其他数据所有者分享其私有数据。

定义8（DOG-Personal）。*DOG-Personal*问题的输入和输出与定义*6*中的*DOG*问题相同，但对每个数据所有者增加了以下要求：每个数据所有者 u 都有个人隐私要求 t_u，并且 u 最多只能与 t_u 个数据所有者分享其私有数据，即，分配给 u 的组的大小不得超过 t_u。

与全局约束要求每个数据所有者共享相同的隐私约束 t 不同，DOG-Personal 变体允许每个数据所有者拥有自己的约束 t_u。

我们修改 DOG问题的贪心解，以应对隐私约束（DOG-Global 和 Personal 两种变体）。

对于 DOG-Global 变体，贪心解的修改方式是：每当我们将数据所有者 u 分配到一个候选组时，就检查该组的大小；如果组的大小达到 t，则不将该数据所有者分配到该组。

对于 Personal 变体，修改方式是：当我们尝试将数据所有者 u分配到组标识符为 i 的候选组时，我们会检查是否满足 $|g_i| \geq t_u$，即组 i 的大小是否大于或等于 t_u。如果满足这一条件，那么该分配会使 g_i 变为 $t_u + 1$，从而违反 u 的个人约束。因此，我们选择编号最小的下一个可用组标识作为数据所有者 u的候选组。

由于篇幅有限，我们省略了修改后贪心解的详细伪代码。解的正确性显而易见，因为受约束版本与原始 DOG 问题的区别仅在于约束部分。

B 附加实验结果

B.1 系统配置

实验在配备 Intel(R) Xeon(R) Gold 6240R CPU @2.40GHz 和 1007G 内存的 CentOS Linux 系统上进行。我们实现了这些方法并开展了实验

使用 Python 和 C。MPC 扩展由 Obliv-C [28] 提供。我们在同一台机器上使用两个进程来模拟客户端-服务器安全多方计算协议。

为了更好地了解我们提出的 FG 方案的表现，我们使用根据数据所有者输入构建的真实空间相似性图来测试查询准确度，并直接在 FG 中执行分组和查询答案聚合。该测试方法称为 FG-C（其中 C 表示清晰/真实图）。对于团枚举变体 FG-Ex，我们也在清晰图上进行测试，并将其命名为 FG-ExC。结果如图3c和图3d所示。显然，由于隐私与效用权衡，FG-C 的表现优于 FG，而 FG-ExC 的表现略优于 FG-Ex，因为使用噪声图（而非原图）会导致平均相对误差和平均绝对误差略高。不过，与 FG-ExC 和 FG-Ex 相比，FG 的平均相对误差和平均绝对误差处于相近水平，这表明我们的贪心启发式方法相比其他替代方案是有效的。

不同下的噪声图与原图对比。 与我们针对不同ϵ的观察结果类似，由于不可避免的效用与隐私的权衡，FG 的平均相对误差和平均绝对误差略差于直接使用真实空间相似性图（图4c）的情况。然而，差距很小，且 FG 始终优于另一种启发式方法 FG-Ex。