UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

JOSÉ SERAFIM DA COSTA FILHO

ANSWERING DIFFERENTIALLY PRIVATE MULTI-DIMENSIONAL QUERIES

FORTALEZA

2025

JOSÉ SERAFIM DA COSTA FILHO

ANSWERING DIFFERENTIALLY PRIVATE MULTI-DIMENSIONAL QUERIES

Thesis submitted to the Programa de Pós-graduação em Ciência da Computação of the Centro de Ciências of the Universidade Federal do Ceará, as a partial requirement for obtaining the title of Doctor in Ciência da Computação. Concentration Area: Ciência da Computação

Advisor: Prof. Dr. Javam de Castro Machado

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

Costa Filho, José Serafim da.
Answering differentially private multi-dimensional queries / José Serafim da Costa Filho. – 2025.
150 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2025.
Orientação: Prof. Dr. Javam Machado.

1. Differential Privacy. 2. Multi-dimensional data. 3. Attribute Correlations. 4. Query Answering. I. Título.

CDD 005

JOSÉ SERAFIM DA COSTA FILHO

ANSWERING DIFFERENTIALLY PRIVATE MULTI-DIMENSIONAL QUERIES

Thesis submitted to the Programa de Pós-graduação em Ciência da Computação of the Centro de Ciências of the Universidade Federal do Ceará, as a partial requirement for obtaining the title of Doctor in Ciência da Computação. Concentration Area: Ciência da Computação

Approved on: 14/08/2025

EXAMINATION BOARD

Prof. Dr. Javam de Castro Machado (Advisor)
Federal University of Ceará (UFC)

Prof. Dr. Cesar Lincoln Cavalcante Mattos
Universidade Federal do Ceará (UFC)

Prof. Dr. Victor Aguiar Evangelista de Farias
Universidade Federal do Ceará (UFC)

Dr. Heber Hwang Arcolezi
Inria - Genoble - FR

Dr. Divesh Srivastava
AT&T Chief Data Office - USA

To my beloved family, whose constant love and enduring patience have been with me every step of the way over the years.

ACKNOWLEDGEMENTS

First and foremost, I give special thanks to God, whose grace and guidance have sustained me throughout this journey.

To my beloved family: my mother Socorro Mota, my father José Serafim, and my siblings Marcelo Henrique and Marinna Maria. Thank you for your unconditional love, constant encouragement, and unwavering support. Your faith in me has been a foundation upon which I've built every step of this path.

To my advisor and mentor, Prof. Javam Machado, I express my deepest gratitude for your guidance, patience, and trust. Your mentorship has been instrumental in shaping the work and the researcher I have become.

I extend my sincere thanks to Profa. Rosélia Machado, for the teachings and invaluable advice shared along the way. Thank you for all the support and encouragement since my very first day at LSBD back in 2012.

A heartfelt thank you to Divesh Srivastava for the exceptional collaboration and insights that enriched this experience. I am also truly grateful for the warm welcome at the AT&T Labs during my period in the US.

I would also like to thank my research colleague Cheryl Brooks, at AT&T Labs, for her significant contributions to this work.

A very special thanks to the members of the examination board Heber Arcolezi, Cesar Lincoln Mattos, and Victor Farias, for generously dedicating their time, sharing their expertise, and offering thoughtful and constructive feedback. Their insights and critical evaluations helped enriching the quality of this work and guiding its final direction.

To my colleagues at the lab Felipe Timbo, Eduardo Rodrigues, Paulo Amora, Daniel Praciano, Iago Chaves, André Luís, Malu Maia, Renan Oliveira, and Antônio Neto, thank you for the camaraderie, collaboration, and shared moments that have made this journey lighter and more enjoyable.

Finally, I am deeply grateful to the Laboratório de Sistemas e Bancos de Dados (LSBD) for all the support provided throughout my PhD. The provided infrastructure, resources, and academic environment have played a key role in enabling this research.

To all of you, thank you from the bottom of my heart.

"Let nothing disturb you, let nothing frighten you. All things are passing; God never changes. Patience obtains all things."

(St. Teresa of Ávila)

ABSTRACT

Providing privacy-preserving answers to multi-dimensional range queries is a critical problem that has attracted significant attention in recent years, given that range queries constitute a fundamental operation in data analysis. However, four principal technical challenges remain: (i) effectively capturing correlations among attributes, (ii) mitigating the curse of dimensionality, (iii) handling large attribute domains, and (iv) accommodating heterogeneous user privacy requirements. Existing methods fail to comprehensively address all these challenges. We build our approach on the idea of using multi-dimensional grids. Specifically, users' data are mapped onto grid structures, which are then perturbed to ensure privacy before being transmitted to an aggregator. The aggregator utilizes the perturbed grid information to estimate the underlying data distribution and subsequently answer range queries. There exists a trade-off in grid granularity: finer grids amplify noise-induced error, whereas coarser grids introduce bias-induced error. To overcome these limitations, we propose a grid construction optimization that considers multiple factors to enhance accuracy. In addition, we build a correlation model to determine how attributes are correlated, enabling the use of fewer, strategically constructed grids, which improves the signal-to-noise ratio. Also, we incorporate a novel optimization procedure that accounts for workload-specific characteristics. This step finds the user-to-grid assignment that minimizes the total expected error. Finally, we combine different differentially private estimators to improve the accuracy when answering multi-dimensional range queries. We validate our approach through extensive experiments on both real-world and synthetic datasets, demonstrating that our method significantly outperforms existing state-of-the-art techniques.

Keywords: differential privacy; multi-dimensional data; attribute correlations; query answering.

RESUMO

Fornecer respostas com garantia de privacidade para consultas por intervalo em múltiplas dimensões é um problema fundamental que tem atraído significativa atenção nos últimos anos, dado que esse tipo de consulta constitui uma operação central em análise de dados. No entanto, quatro principais desafios técnicos permanecem: (i) capturar de forma eficaz as correlações entre atributos, (ii) mitigar a maldição da dimensionalidade, (iii) lidar com domínios de atributos extensos e (iv) acomodar requisitos heterogêneos de privacidade entre os usuários. Métodos existentes falham em abordar todos esses desafios de maneira abrangente. Nossa abordagem baseia-se na ideia de utilizar grades multidimensionais. Especificamente, os dados dos usuários são mapeados em estruturas de grade, que são então perturbadas para garantir privacidade antes de serem transmitidas a um agregador. O agregador utiliza as informações perturbadas das grades para estimar a distribuição subjacente dos dados e, em seguida, responder às consultas por intervalo. Existe um trade-off inerente à granularidade da grade: grades mais finas amplificam o erro induzido pelo ruído, enquanto grades mais grosseiras introduzem erro por viés. Para superar essas limitações, propomos uma otimização na construção das grades que considera múltiplos fatores com o objetivo de melhorar a acurácia. Além disso, desenvolvemos um modelo de correlação para identificar como os atributos se relacionam, permitindo a utilização de um número reduzido de grades estrategicamente construídas, o que melhora a razão sinal-ruído. Também incorporamos um novo procedimento de otimização que leva em conta características específicas da carga de trabalho. Esta etapa determina a atribuição ideal de usuários às grades, minimizando o erro esperado total. Por fim, combinamos diferentes estimadores com garantia diferencial de privacidade para melhorar a precisão na resposta a consultas por intervalo multidimensionais. Validamos nossa abordagem por meio de extensos experimentos em conjuntos de dados reais e sintéticos, demonstrando que nosso método supera significativamente as técnicas mais avançadas disponíveis na literatura.

Palavras-chave: privacidade diferencial; dados multidimensionais; correlações entre atributos; resposta a consultas.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Over the past decade, the rapid advancement of Internet-connected devices, including but not limited to smart home gadgets, Internet of Things (IoT) appliances, wearable technologies, and mobile phones, has significantly transformed the digital landscape. This surge in connectivity has driven the widespread expansion of the mobile Internet, facilitating ubiquitous access to digital services and enabling seamless interactions across various platforms. Consequently, there has been a marked acceleration in the development and deployment of mobile applications, each generating and relying upon vast quantities of user data to function effectively and deliver personalized experiences (CHENG *et al.*, 2017).

In this context, companies have become increasingly motivated to collect, store, and analyze user-generated data to enhance application functionalities, optimize user engagement, and inform strategic business decisions. However, the growing appetite for data has brought serious privacy concerns. The datasets collected through mobile applications and online services frequently contain highly sensitive information, including personal identifiers, behavioral patterns, location data, and usage histories (ZHU *et al.*, 2017; YANG *et al.*, 2017). The exposure of such information, intentional or inadvertent, poses significant risks to individuals' privacy and can lead to reputational and legal consequences for data collectors.

Moreover, the advent of sophisticated data analytics and machine learning techniques has increased these risks. Advanced inference methods can uncover latent attributes and correlations within data, potentially enabling adversaries to reconstruct or re-identify anonymized records (WANG *et al.*, 2020). These threats are not merely theoretical. Numerous real-world incidents underscore the severity of the issue. Prominent data breaches involving major corporations such as Facebook (ISAAC, 2018) and Marriott International (PERLROTH, 2018) have resulted in the unauthorized disclosure of sensitive user information on a massive scale. Such events have amplified public awareness and regulatory scrutiny, underscoring the urgent need for robust privacy-preserving mechanisms in data-driven systems.

To address the growing imperative of safeguarding user privacy, differential privacy (DP) has emerged as a mathematically rigorous and widely utilized framework for ensuring that the inclusion or exclusion of a single individual's data does not significantly affect the outcome of any analysis (DWORK *et al.*, 2016). Differential privacy provides formal guarantees against information leakage, making it a foundational tool for privacy-preserving data analysis. Its practical relevance is demonstrated by its adoption across a range of major technology companies,

including Google (ERLINGSSON *et al.*, 2014), Apple (TEAM, 2017), Microsoft (DING *et al.*, 2017), and Uber (JOHNSON *et al.*, 2018), all of which have incorporated DP mechanisms into their data collection and analytics pipelines.

Traditionally, differential privacy has been implemented under the centralized model, wherein a trusted curator collects raw data from users and subsequently applies random noise to the aggregated data or analysis results to achieve privacy guarantees. While this model is effective under the assumption of a trustworthy data curator, it may be unsuitable for scenarios involving online platforms or crowdsourcing systems, where users may be reluctant to give away the control of their data to a centralized authority. In such cases, the foundational trust assumptions of centralized DP are undermined.

To overcome these limitations, Local Differential Privacy (LDP) has been introduced as a decentralized alternative to the standard centralized approach (KASIVISWANATHAN *et al.*, 2008). Unlike centralized DP, which requires trust in a data curator, LDP empowers users to perturb their data locally on their own devices before transmission. This ensures that each data point is rendered differentially private at the source, independent of any external entity. As a result, the data collector only receives noisy data that provides statistical utility while preserving the anonymity of individual users. By eliminating the need for a trusted third party and aligning with user-centric privacy preferences, LDP has gained traction in a variety of real-world applications, particularly those involving streaming data, telemetry, and user behavior analytics.

In this thesis, we study the problem of answering range queries over multi-dimensional data under the constraints of differential privacy. This task is crucial for privacy-preserving data analysis in real-world applications such as finance, healthcare, and web analytics, where data typically exists in high-dimensional spaces. However, the high dimensionality introduces significant technical challenges, often referred to as the *curse of dimensionality*, which impair both the accuracy of query responses and the scalability of privacy-preserving mechanisms.

Under the LDP model, users apply a randomized algorithm to their data before transmission, ensuring that no trusted server ever observes raw information. A common strategy to preserve utility in such settings involves partitioning users into disjoint groups, with each group assigned to report on a distinct subset of the data or perform a specific task (ARCOLEZI *et al.*, 2022a; WANG *et al.*, 2017; WANG *et al.*, 2019a; ARCOLEZI *et al.*, 2021; WANG *et al.*, 2019b). This partitioning strategy, as demonstrated in recent work such as (FILHO; MACHADO, 2023), yields superior utility compared to approaches that allocate the privacy budget across

multiple dimensions or query types. The core insight here is that concentrating users on fewer tasks enhances the signal-to-noise ratio, enabling more accurate aggregate statistics.

The first major challenge is, therefore, to reduce the problem's effective dimensionality. As the number of attributes increases, the number of possible combinations and interactions grows exponentially, exacerbating the impact of noise added for privacy. Efficiently selecting and materializing a reduced set of dimensions that retains most of the information needed for range queries is essential for scalability. Dimensionality reduction, however, must be done cautiously to avoid losing critical structure in the data.

The second major challenge involves capturing inter-attribute correlations. In high-dimensional datasets, attributes may not be independent. For instance, consider the Lending Club Loan dataset (KAGGLE, 2019), which contains detailed records on loan applicants and outcomes. As illustrated in Figure 1, several attributes, such as *num_op_rev_tl*[1], *num_bc_sats*[2], *open_acc*[3], and *num_rev_accts*[4], exhibit strong mutual correlations. When each attribute is materialized independently, these correlations are ignored, resulting in increased estimation error due to the breakdown of underlying statistical dependencies (YANG *et al.*, 2020). Preserving such correlations is crucial, especially for tasks involving statistical inference, risk modeling, or customer profiling.

Figure 1 – Covariance values of 10 attributes from the Lending Club Loan dataset. Stronger correlations are shown in orange.

	num_op_rev_tl	num_bc_sats	open_acc	num_rev_accts	total_cu_tl	emp_length	addr_state	percent_bc_gt_75	revol_util	bc_util
num_op_rev_tl	1.000000	0.747236	0.834451	0.788120	0.019357	-0.040715	0.006854	-0.124687	-0.203791	-0.124082
num_bc_sats	0.747236	1.000000	0.625605	0.591842	-0.013197	-0.018411	-0.008293	-0.141604	-0.116430	-0.131041
open_acc	0.834451	0.625605	1.000000	0.661352	0.039450	0.003025	0.021402	-0.079743	-0.139537	-0.077506
num_rev_accts	0.788120	0.591842	0.661352	1.000000	0.017745	-0.066585	0.006042	-0.115989	-0.183900	-0.122674
total_cu_tl	0.019357	-0.013197	0.039450	0.017745	1.000000	-0.029452	0.014413	-0.060161	-0.051099	-0.073239
emp_length	-0.040715	-0.018411	0.003025	-0.066585	-0.029452	1.000000	-0.004629	0.011632	0.013016	0.015349
addr_state	0.006854	-0.008293	0.021402	0.006042	0.014413	-0.004629	1.000000	0.012762	0.012212	0.009844
percent_bc_gt_75	-0.124687	-0.141604	-0.079743	-0.115989	-0.060161	0.011632	0.012762	1.000000	0.721795	0.846091
revol_util	-0.203791	-0.116430	-0.139537	-0.183900	-0.051099	0.013016	0.012212	0.721795	1.000000	0.833829
bc_util	-0.124082	-0.131041	-0.077506	-0.122674	-0.073239	0.015349	0.009844	0.846091	0.833829	1.000000

Source: elaborated by the author.

To improve utility, it is a common strategy to incorporate *a priori* knowledge into the

[1] Number of open revolving credit accounts a borrower currently has.
[2] Number of bankcard accounts that are currently in not delinquent status.
[3] The number of active credit accounts recorded in the borrower's credit file.
[4] Total number of revolving credit accounts in the borrower's credit file, whether open or closed.

design of privacy-preserving mechanisms. Prior work has shown that integrating knowledge about expected query workloads, data distributions, or user behaviors can significantly enhance the accuracy of the released statistics while maintaining rigorous privacy guarantees (BUREAU, 2021b; REN *et al.*, 2016; TEAM, 2017; CUNNINGHAM *et al.*, 2021a; ALVIM *et al.*, 2011; LI *et al.*, 2019). Analysts know which queries are most frequent or critical to downstream tasks in many data collection scenarios. For example, if the query workload is concentrated on income or credit behavior attributes, prioritizing these dimensions for higher precision collection is a strategic decision. Less relevant attributes, on the other hand, can tolerate greater noise injection, balancing the overall privacy-utility tradeoff.

Population partitioning under LDP is an effective strategy to avoid privacy budget fragmentation when answering multiple queries. Instead of splitting the budget across queries, the user base is divided into disjoint subgroups, and each subgroup reports only on a specific subset of attributes. However, uniform partitioning, where users are evenly divided across all attributes or queries regardless of their importance, can degrade utility, particularly when the number of users is small or the attribute space is large. In such settings, frequently queried attributes may suffer from inadequate sampling, and the compounded effect of noise can overwhelm the signal. Therefore, the third key challenge lies in integrating workload characteristics into privacy-preserving mechanisms to achieve a user partitioning that optimizes the overall utility.

A fourth challenge emerges from the heterogeneity of privacy preferences and regulatory obligations across user populations. In practical deployments, different users may adhere to various privacy standards depending on regional legislation or personal trust in data collectors. For instance, within the United States, privacy regulations vary by state—California's privacy laws being notably stringent compared to others (BISCHOFF, 2023). In such a heterogeneous privacy landscape, users may belong to distinct *privacy groups*, each characterized by a different level of trust or legal protection. Although hybrid differential privacy models have been explored in prior work (KOHEN; SHEFFET, 2022; AVENT *et al.*, 2017; AVENT *et al.*, 2020; BEIMEL *et al.*, 2020; KHALAF *et al.*, 2024; ACQUISTI; GROSSKLAGS, 2005; ACQUISTI *et al.*, 2015), they have not been systematically studied in the context of range queries under LDP. Addressing this gap, we aim to handle diverse privacy levels within a unified framework. This requires accommodating different noise calibration parameters and redesigning aggregation techniques to ensure equitable utility across user groups, without compromising the strict privacy guarantees afforded to more sensitive users.

1.1 Problem Statement

Consider a dataset D with k attributes $a_1, a_2, ..., a_k$ and let each attribute have a domain $d = d_1, d_2, ..., d_k$. Let N be the total number of users. The i-th user's record is a k-dimensional vector expressed by $v_i = \langle v_i^1, v_i^2, ..., v_i^k \rangle$ where v_i^t means the value of attribute a_t in record v_i. Our thesis tackles the problem of privately answering multi-dimensional range queries, where a multi-dimensional range query is a conjunction of multiple predicates for attributes of interest. A λ-dimensional query q is defined as

$$q = (a_{t_1}, v_{t_1}) \wedge (a_{t_2}, v_{t_2}) \wedge ... \wedge (a_{t_\lambda}, v_{t_\lambda}) \tag{1.1}$$

where v_{t_i} is a range $[l_{t_i}, r_{t_i}]$. Also, $1 \leq t_i \leq k$, and $t_i \neq t_j$ when $i \neq j$. Let $A_q = \{a_{t_i} | 1 \leq i \leq \lambda\}$ represent the set of attributes specified in query q. That means that a query q selects all records whose value of attribute a_{t_i} is within the range $[l_{t_i}, r_{t_i}]$. The answer to query q equals the count of all records that satisfy all conditions divided by the number of users N. Formally, the real answer to query q can be expressed as

$$\tilde{f}_q = \frac{|\{v_i | v_i^t \in v_t, \forall a_t \in A_q\}|}{N} \tag{1.2}$$

Our goal is to design an approach to enable the aggregator to get the answers to range queries in a differentially private manner.

1.2 Thesis Hypothesis

The main hypothesis we address in this thesis is stated as follows:

Main Hypothesis. *By using grids to encode combinations of attributes, mapping multi-dimensional data points to grid cells, it is possible to capture correlations and answer differentially private queries.*

We investigate our central hypothesis under a diverse range of constraints and real-world scenarios, aiming to understand its validity and implications across various practical settings. Our analysis is guided by a structured set of research questions, each designed to study specific aspects of our hypothesis. These questions span both theoretical and empirical dimensions. In particular, we seek to assess how our proposed methods perform under differing assumptions about data distribution, privacy budget allocation, user behavior, and application requirements. Our investigation is structured around the following core research questions:

Research Question 1 (RQ1) *How can we answer an unbounded number of differentially private queries, with range and point constraints, when the entire population comprises LDP users?*

Since all query answers must be estimated under LDP, optimizing the decomposition of multi-dimensional domains is needed to deal with large domains and avoid the curse of dimensionality. A finer-grained decomposition leads to more significant errors due to noise since, under LDP, one has to perturb each cell. The more cells, the higher the total noisy error. However, coarser-grained decompositions will lead to high error due to bias. Baselines (YANG *et al.*, 2020; WANG *et al.*, 2019a) utilize grids and hierarchical decompositions in a one-size-fits-all manner. A strategy that optimizes the construction and processing of grids is needed and investigated in Chapter 4.

Research Question 2 (RQ2) *Given a population consisting of both DP and LDP users and a priori knowledge of the query workload, how can we scalably answer an unbounded number of differentially private range queries?*

One widely used approach in data analysis under local differential privacy (LDP) is to assume that all attributes in a dataset are correlated. This assumption necessitates partitioning the user population into many distinct groups to capture the dependencies between different attribute combinations. However, this strategy introduces a significant trade-off. In the context of LDP, the utility of any analysis, measured by the accuracy of aggregated statistics, declines as the number of users per group decreases. This is because the estimates' variance becomes inversely proportional to the number of users contributing data to each group or grid, meaning that fewer users per group lead to noisier results. Thus, a correlation-aware strategy and a technique that adapts to the workload characteristics are needed. We study these aspects in Chapter 5.

Research Question 3 (RQ3) *Given a population with DP and LDP users and a fixed number of queries, how can we combine the DP and LDP estimators in order to optimize utility?*

In many real-world applications, data is collected from diverse sources with varying trust assumptions. While Differential Privacy (DP) in the central model enables accurate statistical analysis by leveraging access to raw data through a trusted aggregator, Local Differential Privacy (LDP) empowers users to protect their information without requiring trust in any central entity. Relying solely on either model can limit utility. However, combining the strengths of both

paradigms is not trivial. We investigate this problem in Chapter 6.

1.3 Contributions

To address the challenges outlined in earlier sections, we propose three distinct strategies for answering differentially private queries, each designed to operate under different assumptions and query settings. These strategies are developed in response to our central hypothesis and the guiding research questions to explore how privacy-preserving data analysis can be optimized across diverse use cases. Each proposed solution targets a unique combination of privacy models, user compositions, and query requirements. For instance, one strategy is tailored to environments where all users employ local differential privacy and the system must support an unbounded number of queries. Another is designed for hybrid populations composed of both DP and LDP users, leveraging prior knowledge about the query workload to optimize performance. The third focuses on combining estimators from central and local models when the number of queries is fixed, aiming to maximize utility through effective aggregation.

By addressing these varied settings, our three strategies collectively span a broad spectrum of practical scenarios encountered in real-world data analysis systems. Furthermore, this approach allows us to investigate multiple points in the privacy-utility design space. In summary, this thesis makes the following contributions:

1. We propose a new algorithm for grid construction by carefully studying the sources of error when answering frequency queries with point and range constraints. Our strategy design enables the precise capture of data characteristics, resulting in high utility.
2. We study the use of different frequency oracle protocols when collecting data under LDP using grids. We propose to adaptively select the best-performing frequency oracle protocol for each data grid to achieve lower estimation errors.
3. We propose a new framework to answer an unbounded number of range queries capable of scaling to datasets with many attributes under the LDP model. By taking advantage of different privacy expectations among users, it explicitly models and identifies the correlation among attributes to enable the materialization of k-dimensional grids with only strongly correlated attributes.
4. We propose a new optimization stage within our framework that considers common *a priori* characteristics of the types of data analysis analysts are interested in doing. We analyze the different sources of error when answering queries and determine the user distribution

among grids that minimizes the total error of answering such a workload.

5. We propose a variance-aware adaptive strategy to query answering that combines DP and LDP estimators to minimize the total sum of errors when answering a bounded number of range queries.
6. We extensively experiment with our solution under different scenarios with distinct distributions and correlation levels. Also, we study scalability and workload adaptability capabilities. We use synthetic and real datasets from different domains and show the superiority of our approach over existing techniques in the literature.

Published papers:

- COSTA FILHO, J. S.; MACHADO, J. C. FELIP: A local differentially private approach to frequency estimation on multidimensional datasets. *International Conference on Extending Database Technology, 2023*. (FILHO; MACHADO, 2023)
- MARREIRAS NETO, A. A.; DUARTE NETO, E.; COSTA FILHO, J. S.; MACHADO, J. Locally differentially private and consistent frequency estimation of longitudinal data. *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados, 2024*. (MARREIRAS NETO *et al.*, 2024)
- DUARTE NETO, E.; COSTA FILHO, J. S.; MARREIRAS NETO, A. A.; MACHADO, J. C.: ALOG: Adaptive Longitudinal Grids for Geospatial Data using Local Differential Privacy. *International Conference on Extending Database Technology, 2026*. (DUARTE NETO *et al.*, 2026).

Accepted papers (waiting for publication):

- OLIVEIRA R.; COSTA FILHO, J. S.; MACHADO, J.; MONTEIRO. J. M.: Data Dependent Itemset Mining Under Local Differential Privacy. *Anais do XXXX Simpósio Brasileiro de Bancos de Dados, 2025*.

1.4 Thesis Organization

This dissertation is organized as follows. Chapter 2 introduces the foundational background and formal definitions supporting the thesis's technical developments. Chapter 3 reviews the relevant literature, highlighting the core ideas of existing approaches along with their limitations in addressing the challenges mentioned earlier. In Chapter 4, we address the first research question by presenting the *FELIP* strategy, detailing its design and how it supports unbounded query answering under local differential privacy. Guided by the second research

question, Chapter 5 introduces the *PrivMDC* framework, providing a comprehensive explanation of its architecture and the whole pipeline for answering range queries in a heterogeneous privacy setting. Chapter 6 then presents the adaptive estimator *DP-AE* approach, developed to answer the third research question, which explores an optimized strategy to combine central and local estimators to answer a fixed number of range queries. The strategies proposed in Chapters 4, 5, and 6 are empirically validated through extensive experiments, the results of which are presented and analyzed in Chapter 7. Finally, Chapter 8 concludes this thesis by summarizing the main contributions and outlining open problems and directions for future research.

2 BACKGROUND AND DEFINITIONS

This chapter reviews concepts closely related to this thesis, including the privacy definitions, frequency oracle protocols and Bayesian networks.

2.1 Differential Privacy

Re-identification attacks attempts to infer sensitive attributes or identify individuals by leveraging auxiliary information. They have become increasingly sophisticated over the past two decades. These attacks exploit correlations between quasi-identifiers in anonymized datasets and external sources of information. For instance, an attacker may link age, ZIP code, and gender from a publicly released dataset with voter registration lists to uncover the identities of individuals. As data sharing becomes more common across industries such as healthcare, finance, and government, the risk of such privacy breaches grows considerably.

The limitations of traditional privacy-preserving models such as k-anonymity (SWEENEY, 2002), l-diversity (MACHANAVAJJHALA *et al.*, 2006), and t-closeness (LI *et al.*, 2007) have been extensively documented in the literature. While these models aim to mask individual identities by grouping records or maintaining value distributions, they often fail under adversarial inference. Composition attacks and background knowledge attacks have exposed systemic vulnerabilities in these approaches. Notable works such as (GANTA *et al.*, 2008; CORMODE *et al.*, 2010; JIN *et al.*, 2010; WONG *et al.*, 2011) have demonstrated how adversaries can reverse-engineer anonymized datasets or exploit utility-preserving transformations to re-identify individuals or infer private attributes. One of the fundamental problems with traditional privacy models is their reliance on assumptions about the adversary's knowledge and capabilities. These models often provide guarantees only under restricted scenarios and fail to generalize to real-world conditions where adversaries may possess arbitrary or extensive background information. Moreover, they are susceptible to cumulative risks: when multiple datasets are released over time or in combination, the intersection of their contents can severely degrade privacy.

In response to the demonstrated weaknesses of these conventional techniques, a new and principled privacy framework has emerged: Differential Privacy (DP). Introduced in 2006, differential privacy represents a shift from data transformation to probabilistic reasoning about the information leakage incurred by data analysis. Rather than focusing on how data is

presented (as in k-anonymity), differential privacy directly bounds the risk incurred by any single individual's participation in a dataset. Differential privacy does not make assumptions about the adversary's background knowledge. Instead, it provides a mathematically rigorous guarantee: the inclusion or exclusion of any individual's data in a dataset does not significantly change the output distribution of a computation. Thus, even a powerful adversary with access to arbitrary auxiliary information cannot determine with high confidence whether an individual's data was used (DWORK *et al.*, 2016).

Importantly, differential privacy is not a single mechanism or algorithm; it is a framework that defines what it means to protect privacy and allows for the development of mechanisms that meet its definition. These mechanisms inject calibrated randomness into computations to ensure that individual contributions are obfuscated. The level of randomness and thus the trade-off between privacy and accuracy is governed by a parameter known as the *privacy budget*. The differential privacy paradigm has been adopted across a broad range of data analytic tasks, from simple statistical queries to more complex applications in machine learning and artificial intelligence. For example, (ZHU *et al.*, 2022) explores the integration of differential privacy in various AI applications, showing how privacy can be preserved without severely compromising model utility.

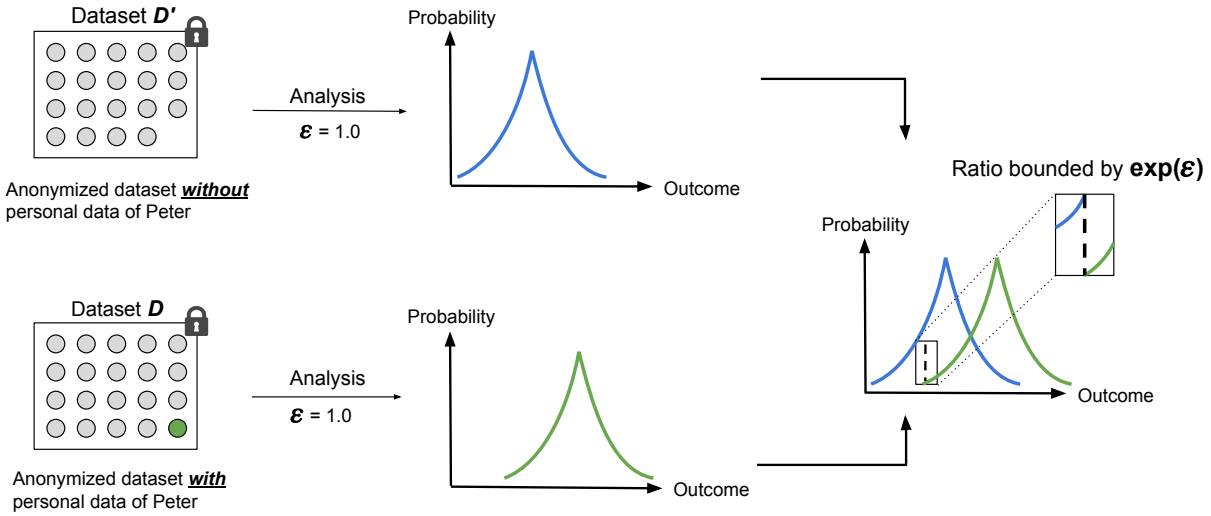
2.2 Differential Privacy Definition

In its original formulation, differential privacy conceptualizes a dataset as a collection of individual records, each corresponding to a single person. The core idea behind differential privacy is to protect these individuals by introducing controlled randomness into the outputs of computations performed on the data, rather than modifying the data itself (DWORK *et al.*, 2016). This approach ensures that the presence or absence of any single individual does not significantly influence the outcome of an analysis.

Let Q be a query function *i.e.*, an analysis or computation, that is to be evaluated on a dataset D, which contains sensitive information about a population of individuals. To provide privacy guarantees, a randomized algorithm $\mathcal{A}$, which satisfy differential privacy, approximates $Q(D)$. The algorithm $\mathcal{A}$ produces outputs that can be close to the true query result $Q(D)$, but with randomness injected in a calibrated manner. The goal is twofold: (i) to maintain utility by making $\mathcal{A}(D)$ as close as possible to $Q(D)$, and (ii) to protect the privacy of all individuals in D by ensuring that the outcome does not reveal whether any specific record was used.

A key concept in differential privacy is that of *neighboring datasets*. Two datasets D and D' are considered neighbors, denoted $D \sim D'$, if they differ in at most one individual's data (*i.e.*, a single tuple). This relation is symmetric, meaning that if $D \sim D'$, then $D' \sim D$ as well. Figure 2 presents an illustrative example of two neighboring datasets: one dataset D' excludes Peter's information, while the neighboring dataset D includes it.

Figure 2 – Example of a private analysis on two neighboring datasets with and without the personal data of Peter.



Source: elaborated by the author.

Differential privacy mandates that for any pair of neighboring datasets, the probability distribution over the possible outputs of $\mathcal{A}$ should be nearly identical, regardless of whether the computation was performed on D or on D'. This concept is referred to as indistinguishability between neighboring datasets. Intuitively, this means that the output of $\mathcal{A}$ does not allow an observer to confidently infer the participation of any individual, including Peter, in the dataset. Formally, let Range($\mathcal{A}$) denote the set of all possible outputs of algorithm $\mathcal{A}$. For instance, if $\mathcal{A}$ returns the count of individuals satisfying a certain predicate, then Range($\mathcal{A}$) corresponds to the set of non-negative integers. The formal definition of differential privacy is given as follows:

Definition 2.1 (ϵ-Differential Privacy (DWORK *et al.*, 2016))**.** A randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy, if for any two neighboring datasets D and D', and for any possible output $O \subseteq$ Range($\mathcal{A}$),

$$Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in O], \tag{2.1}$$

where $Pr[\cdot]$ denotes the probability of an event.

Intuitively, ϵ-DP offers strong privacy protection. If $\mathcal{A}$ satisfies ϵ-DP, one can claim that publishing $\mathcal{A}(D)$ does not violate the privacy of any tuple t in D, because even if one leaves t out of the dataset, in which case the privacy of t can be considered to be protected, one may still publish the same outputs with a similar probability. In practice, however, ϵ-DP can be too strong to satisfy in some scenarios. A commonly used relaxation is to allow a small error probability δ.

Definition 2.2 ((ϵ, δ)-Differential Privacy (DWORK *et al.*, 2006))**.** A randomized algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy, if for any pair of neighboring datasets D and D' and for any $O \subseteq \text{Range}(\mathcal{A})$:

$$Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in O] + \delta \tag{2.2}$$

Here, $\epsilon \geq 0$ bounds the multiplicative privacy loss, while $\delta \geq 0$ allows a small probability of failure of the privacy guarantee. When $\delta = 0$, the definition reduces to pure ϵ-DP, whereas allowing a small $\delta > 0$ provides additional flexibility in designing algorithms that can achieve stronger utility at the cost of a negligible chance of a larger privacy breach. In practice, δ is typically chosen to be very small, often smaller than the inverse of the dataset size, for example $\delta < 1/|D|$.

2.3 The Privacy Budget in Differential Privacy

The parameter ϵ in Definition 2.1, commonly referred to as the privacy budget, plays a central role in determining the strength of the privacy guarantees offered by a differentially private algorithm. As previously discussed, differential privacy requires that the output of an analysis remains nearly indistinguishable, whether or not any particular individual's data is included in the input dataset. The privacy budget $\epsilon \in \mathbb{R}_{>0}$ quantitatively measures this indistinguishability.

A smaller value of ϵ enforces a stronger privacy guarantee, as it tightly bounds the ratio between the probabilities of any output occurring when the algorithm is applied to neighboring datasets. In the limiting case where $\epsilon = 0$, the outputs of the algorithm become completely indistinguishable across neighboring datasets. This corresponds to perfect privacy, in which the algorithm reveals no information about any individual. However, such a setting renders the output effectively useless for any analytical purpose, as the added randomness must entirely obscure the true computation. Conversely, larger values of ϵ allow for less noise to be added, thereby improving data utility at the cost of weaker privacy guarantees. The privacy

budget thus formalizes the privacy-utility trade-off: increasing utility by reducing noise inherently compromises the privacy protection provided to individuals in the dataset.

The choice of ϵ is nontrivial and typically determined based on a combination of policy considerations, empirical evaluation, and stakeholder feedback. It is neither dictated by theory alone nor universally agreed upon. The literature has explored normative and practical recommendations for setting the privacy budget (HSU *et al.*, 2014; LI *et al.*, 2016). Some studies suggest that $\epsilon = 0.1$ offers very strong privacy protection, and values around $\epsilon = 1.0$ are often regarded as acceptable in many practical applications (LEE; CLIFTON, 2011; LI *et al.*, 2016).

Despite these reference points, the process of selecting a suitable ϵ remains a complex and often application-specific task. It requires the input of privacy experts, data practitioners, and domain stakeholders to jointly evaluate the acceptable balance between privacy and accuracy. This challenge is exemplified in the case of the U.S. Census Bureau. For the 2020 Census, the Bureau engaged in extensive consultations with the data user community. After thorough deliberation, the Data Stewardship Executive Policy Committee (DSEP) approved a total privacy budget of $\epsilon = 19.61$ for the final data releases (U.S. Census Bureau, 2021). This value, though significantly larger than what is typically used in academic literature, reflects the practical constraints and diverse demands placed on official statistics.

2.4 Sensitivity

As previously outlined, differential privacy is typically achieved by injecting calibrated noise into the results of data analyses. However, introducing too much noise can significantly degrade data utility by compromising the accuracy of analytical outputs, whereas adding too little noise may result in insufficient privacy protection for individuals in the dataset. The noise added depends on the global sensitivity of a query Q (DWORK *et al.*, 2016), as defined next:

Definition 2.3 (Global Sensitivity (DWORK *et al.*, 2016))**.** The global sensitivity of a query Q is the maximum l_1 distance between the outputs of Q on any two neighboring datasets.

$$\Delta Q = \max_{D,D'} ||Q(D) - Q(D')||_1. \tag{2.3}$$

The concept of global sensitivity plays a foundational role in determining how much noise must be added to a query's output in order to satisfy privacy guarantees. Formally, the global sensitivity of a query function quantifies the maximum possible change in the query result that can arise from the addition or removal of a single individual record in the dataset. This

measure captures the worst-case impact any single participant could have on the outcome of the analysis. Global sensitivity is a property of the query function itself, rather than of the dataset to which it is applied. It is independent of the actual data values and is computed over all possible pairs of neighboring datasets (datasets that differ by one record). As such, it defines the upper bound of influence that any individual could exert on the result, providing the necessary scale for calibrating the magnitude of the noise to be added for differential privacy.

The importance of global sensitivity lies in its direct influence on the trade-off between privacy and utility. When a query function has low sensitivity, only a small amount of noise is needed to obscure the effect of any single individual's data, which in turn allows for high utility in the released results. Conversely, functions with high sensitivity require the injection of greater noise to preserve privacy, which can substantially degrade the accuracy and utility of the output. For certain types of queries, computing the global sensitivity is straightforward. For example, counting queries, which return the number of records satisfying a given condition, have a global sensitivity of 1. This is because adding or removing one record can alter the count by at most one unit, regardless of the dataset's content.

However, the situation becomes more nuanced for other types of queries, such as summation queries. Consider a query that returns the sum of the ages of all individuals in a dataset. In this case, adding a new record increases the output by the age of the new individual. This implies that the sensitivity is not fixed at 1 but depends on the maximum possible value that the age attribute can take. Since the sensitivity must be determined independently of any specific dataset, one must define a plausible upper bound for the domain of the queried attribute. In domains such as age, it is often reasonable to assume an upper limit based on empirical or biological constraints. For example, one could define the global sensitivity of the sum of ages query as 122, which corresponds to the verified maximum human lifespan to date (GIBBS; ZAK, 2023). This upper bound enables the application of differential privacy mechanisms by anchoring the sensitivity to a rational and justifiable constant. Nevertheless, this choice remains somewhat arbitrary, as it is not guaranteed that future individuals will not exceed this age limit. This introduces a layer of uncertainty and reflects the inherent difficulty in specifying global sensitivity in domains where natural or legal limits are not sharply defined or may evolve over time (NEAR; ABUAH, 2023).

In practice, choosing an appropriate sensitivity value requires domain knowledge and careful consideration of the potential values that input data may assume. Failing to account

for worst-case scenarios can compromise privacy, while overly conservative sensitivity bounds may result in unnecessary loss of utility. Hence, sensitivity analysis is a critical step in designing differentially private mechanisms and must be approached with both theoretical rigor and empirical insight.

2.5 Laplace Mechanism

A randomized algorithm is also referred to as a mechanism. Mechanisms are ways of achieving differential privacy. For numeric queries, DP can be achieved by mechanisms such as the Laplace mechanism (DWORK *et al.*, 2016). The Laplace mechanism is a widely used algorithm to satisfy differential privacy. As the name of the mechanism suggests, it relies on the Laplace distribution to generate random values that are added to the true query response. Let x be the noise added to the output of a query function f.

Definition 2.4 (Laplace Distribution)**.** The Laplace distribution (centered at 0) with scale b is the distribution with probability density function:

$$Lap(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right) \tag{2.4}$$

Denote $Lap(b)$ the Laplace distribution with scale b. Then, the Laplace mechanism (DWORK *et al.*, 2016) simply compute f, and perturb the result with noise drawn from the Laplace distribution. The scale of the noise is calibrated by the sensitivity Δf divided by ϵ.

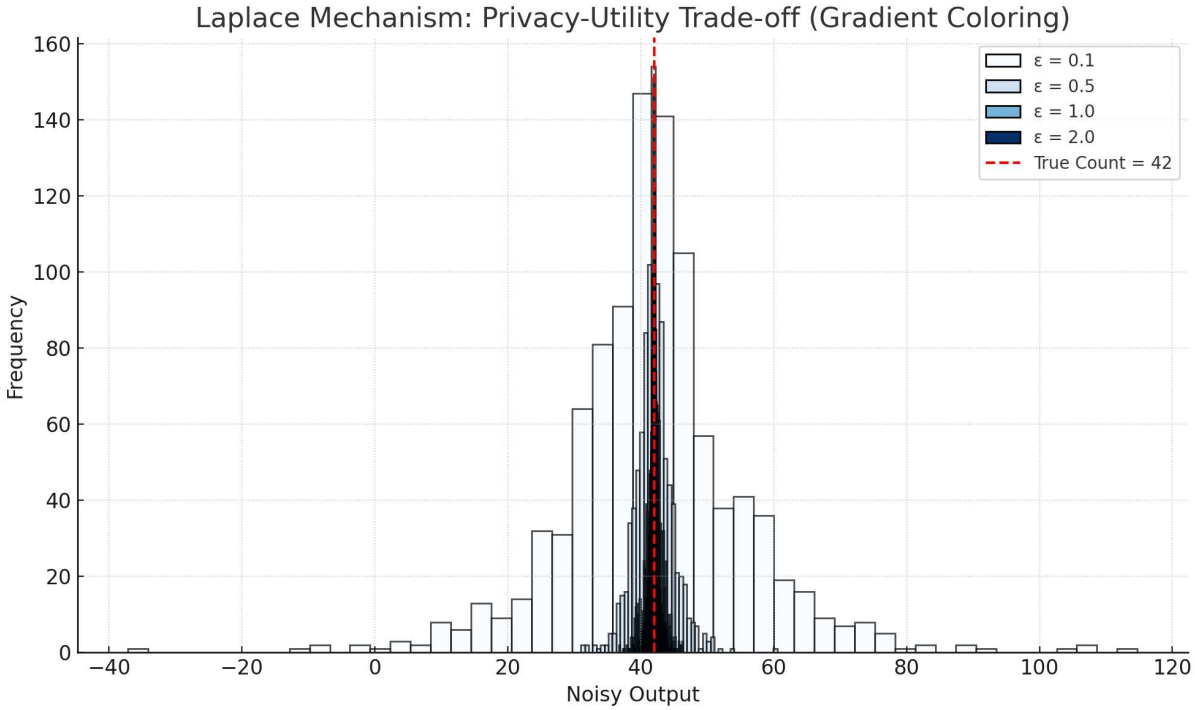
Theorem 2.5 (Laplace Mechanism (DWORK *et al.*, 2016))**.** The Laplace mechanism that adds noise drawn from $Lap(\Delta f/\epsilon)$ satisfies ϵ-DP.

The variance of the specified Laplace distribution is $2(\Delta f/\epsilon)^2$. Intuitively, this makes sense: the larger Δf, the more sensitive the output, requiring more noise to be added to the output to achieve a target level of protection. Similarly, the smaller the privacy parameter, the higher the target level of privacy, again requiring larger amounts of Laplace noise.

To illustrate the effect of differential privacy in practice, consider a simple query that returns the number of individuals in a dataset who satisfy a particular condition, for instance, the number of users who clicked on an ad. Suppose the true count is 42. To make the result differentially private, we apply the Laplace mechanism, which adds noise drawn from the Laplace distribution calibrated to the global sensitivity of the query and a privacy parameter ϵ. In this case, the sensitivity is 1, since adding or removing a single individual changes the count by at

most 1. The amount of noise added depends on the value of ϵ, known as the privacy budget. A smaller ϵ results in stronger privacy protection by introducing more noise, while a larger ϵ provides more accurate results but weaker privacy guarantees. Figure 3 illustrates this trade-off by showing the distribution of outputs from the Laplace mechanism for several values of ϵ.

Figure 3 – Output distributions of the Laplace mechanism for different values of ϵ. The true count is 42. Lower ϵ values yield wider distributions, indicating greater noise and stronger privacy.



Source: elaborated by the author.

As shown in the figure, when $\epsilon = 0.1$, the distribution is very wide, reflecting a high level of noise added to preserve privacy. The outputs are highly variable and may deviate significantly from the true value. As ϵ increases (*e.g.*, to 1.0 or 2.0), the distributions become narrower and more concentrated around 42, indicating improved utility but reduced privacy protection. This highlights the fundamental trade-off at the core of differential privacy: the choice of ϵ must carefully balance the need to protect individual information against the desire to release accurate and useful analytical results.

2.6 Differential Privacy Properties

2.6.1 Post-processing

In post-processing, any function applied on the output of a differentially private algorithm also satisfies differential privacy.

Theorem 2.6 ((DWORK *et al.*, 2016)). Let $\mathcal{A}$ be any randomized algorithm such that $\mathcal{A}(D)$ is ϵ-differentially private, and let f be any function. Then, $f(\mathcal{A}(D))$ also satisfies ϵ-DP.

2.6.2 Sequential Composition.

Any sequence of computations that are individually differentially private also maintain differential privacy when executed in sequence. This holds true even if later computations use the outputs of earlier ones, not just when they are run independently.

Theorem 2.7 (Sequential Composition (MCSHERRY, 2009a)). Let each $\mathcal{A}_i$ provide ϵ_i-differential privacy. A sequence of differentially private algorithms $\mathcal{A}_i(D)$ provides $(\sum \epsilon_i)$-DP.

To illustrate the sequential composition, suppose a hospital maintains a database of 1,000 patients and intends to release two statistics based on this data:

1. The number of patients diagnosed with diabetes.
2. The number of patients with $age > 60$.

To ensure individual privacy, both queries are answered using mechanisms that satisfy differential privacy. The hospital uses the Laplace mechanism to inject noise into each query result. Each query is configured to satisfy $\epsilon = 0.5$-differential privacy. According to the sequential composition theorem, if multiple differentially private mechanisms are applied to the same dataset, the total privacy loss is additive. Therefore, after answering both queries, the total privacy budget consumed is:

$$\epsilon_{\text{total}} = \epsilon_1 + \epsilon_2 = 0.5 + 0.5 = 1.0$$

This indicates that although each query individually satisfies $\epsilon = 0.5$-differential privacy, the cumulative effect of both queries results in an overall privacy cost of $\epsilon = 1.0$.

2.6.3 Parallel Composition.

While the privacy costs of a sequence of queries add up, an improved bound can be achieved when the queries operate on disjoint subsets of the data. Specifically, if the domain of input records is partitioned into disjoint sets, independent of the actual data, and each subset is analyzed separately using differentially private methods, the overall privacy guarantee is determined by the maximum privacy cost among the analyses, rather than their sum.

Theorem 2.8 (Parallel Composition (MCSHERRY, 2009a))**.** If D_i are disjoint subsets of the original database and M_i provides ϵ-differential privacy for each D_i, then the sequence of M_i provides ϵ-differential privacy.

To illustrate the parallel composition, consider a nationwide survey dataset that includes residents from two different cities: City A and City B. A statistical agency wishes to compute the following quantities in a differentially private manner:

1. The average income of residents in City A.
2. The average income of residents in City B.

Assume that the data from City A and City B are *disjoint*, meaning no individual appears in both datasets. The Laplace mechanism is used to release both statistics. Each query is configured to satisfy $\epsilon = 1.0$-differential privacy. Since the analyses are performed on disjoint subsets of the data, the parallel composition theorem states that the overall privacy guarantee for the entire population remains:

$$\epsilon_{\text{total}} = \max\{\epsilon_1, \epsilon_2\} = \max\{1.0, 1.0\} = 1.0$$

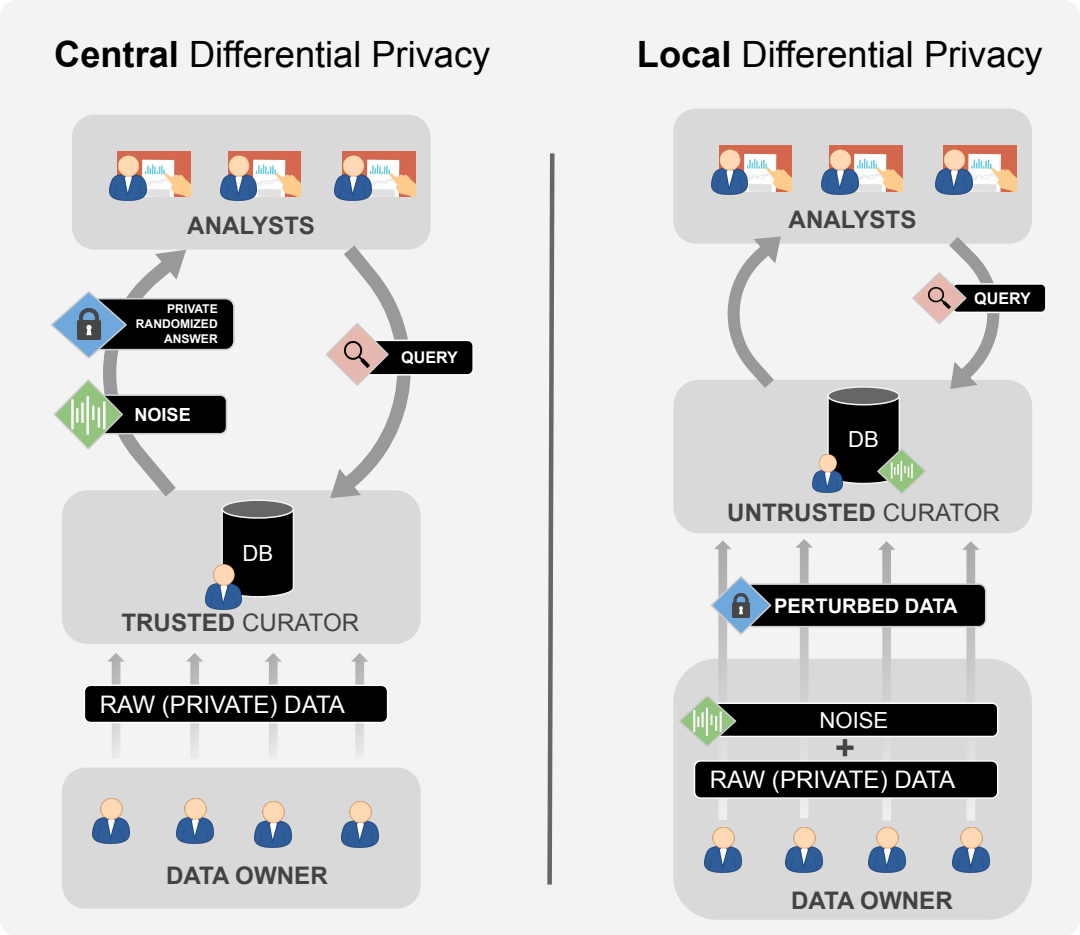
Thus, even though two queries are issued, the overall privacy loss does not exceed $\epsilon = 1.0$. This is because no single individual contributes to more than one of the analyses. Parallel composition allows multiple differentially private computations to be conducted efficiently, provided that the underlying datasets are disjoint. It is especially useful in distributed data collection scenarios, federated learning, or stratified survey analysis, where the population can naturally be segmented (KHALAF *et al.*, 2024).

When designing a differentially private technique, the aforementioned properties provide the flexibility to easily combine several DP steps into a DP mechanism.

2.7 Local Differential Privacy

Differential privacy was originally formulated under the *centralized model*, where a trusted curator collects and stores raw user data, applies a randomized mechanism to ensure privacy, and then releases the noisy results. While this model provides strong privacy guarantees and high utility due to access to unperturbed data, it requires users to trust the central data aggregator with their sensitive information. However, in many real-world scenarios, such trust assumptions do not hold. Users may be unwilling or unable to share their raw data with a centralized authority due to privacy concerns, legal restrictions, or organizational policies (ERLINGSSON *et al.*, 2014). In such contexts, the centralized model becomes impractical, and a more decentralized privacy model is needed.

Figure 4 – Central DP versus Local DP.



Source: elaborated by the author.

Local Differential Privacy (LDP) addresses this limitation by shifting the privacy-preserving mechanism from the data collector to the user side In the local model, each user

perturbs their data locally before transmission, ensuring that no trusted curator is required, as illustrated in Figure 4. As a result, the data collector never sees the original input, only its noisy version. This model aligns well with privacy-preserving data collection in decentralized environments, such as:

- Telemetry collection in mobile devices (e.g., iOS, Android) (DING *et al.*, 2017).
- Browser usage statistics (e.g., Chrome, Firefox) (ERLINGSSON *et al.*, 2014).
- Smart assistants and IoT applications (WANG *et al.*, 2020).
- Federated learning with untrusted central coordination (KHALAF *et al.*, 2024).

Prominent industry applications, such as Apple's iOS differential privacy deployment (NADA, a) and Google's RAPPOR system (ERLINGSSON *et al.*, 2014), have demonstrated the feasibility and scalability of LDP in practice. In both cases, users' data are randomized on-device before being sent to the server, allowing global patterns to be inferred while preserving the privacy of individuals. From a technical perspective, LDP provides stronger privacy guarantees than the central model, as it removes the need to trust any central entity. However, this robustness comes at the cost of increased noise, which typically results in lower utility. Designing mechanisms under LDP thus involves careful trade-offs to minimize utility degradation while maintaining strong local guarantees. Given the increasing concerns over data misuse (PERLROTH, 2018; ISAAC, 2018), regulatory requirements such as GDPR and CCPA, and the need for scalable privacy technologies in consumer-facing systems, local differential privacy has emerged as a compelling and practical solution for trustworthy data analysis.

Under LDP, sensitive value v from each user is encoded by a randomized algorithm Ψ, and its output $\Psi(v)$ is sent to the aggregator, which is responsible for collecting all users' reports. Intuitively, LDP guarantees that, no matter what the value of $\Psi(v)$ is, it is approximately equally as likely to be a result of perturbing v as any other v' differing from v. Therefore, if $\Psi(v)$, instead of v, is collected, the users never reveal their private value v. The user's degree of privacy is controlled by the privacy budget ϵ. More formally,

Definition 2.9 (ϵ-Local Differential Privacy (ERLINGSSON *et al.*, 2014))**.** An algorithm $\Psi(\cdot)$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$, if and only if for any pair of inputs (v, v'), and any set R of possible outputs of Ψ, we have

$$Pr[\Psi(v) \in R] \leq e^{\epsilon} Pr[\Psi(v') \in R] \tag{2.5}$$

To illustrate the concept of local differential privacy, we consider the classic technique

known as randomized response (WARNER, 1965), first introduced in social science surveys to elicit truthful answers to sensitive questions while preserving individual privacy. Consider a small dataset of students in Table 1, along with a sensitive binary attribute indicating whether they have ever sought mental health counseling:

Student ID	Sought Mental Health Counseling
1	Yes
2	No
3	No
4	Yes
5	Yes
6	Yes
7	No
8	Yes

Table 1 – Original dataset with sensitive attribute

The true proportion of "Yes" answers in the dataset is $p = \frac{5}{8} = 0.625$. Next, each student applies the following randomized response procedure:

- Flip a fair coin.
- If heads: report their truthful answer.
- If tails: flip a second fair coin and report "Yes" if heads, "No" if tails.

Assume the randomized responses observed are shown in Table 2:

Student ID	Privatized Response
1	Yes
2	No
3	No
4	No
5	Yes
6	No
7	Yes
8	Yes

Table 2 – Privatized responses using randomized response

The observed proportion of "Yes" responses is $\hat{p}_{\text{obs}} = \frac{4}{8} = 0.5$ Given the randomized response mechanism, the expected "Yes" response is $\mathbb{E}[\text{"Yes"}] = 1/2 \cdot p + 1/4$

Solving for the estimated true proportion:

$$\hat{p} = 2 \cdot \left(\hat{p}_{\text{obs}} - \frac{1}{4} \right) = 2 \cdot (0.5 - 0.25) = 0.5$$

2.8 Frequency Oracles

A *frequency oracle (FO)* protocol can be used to estimate the frequency of any value $v \in D$ under LDP, where D is the domain. A FO consists of two algorithms. The first one is Ψ, which users use locally to perturb their private data. The second one Φ is used by the aggregator to estimate the frequencies based on the perturbed data received. We present below two FOs used in this work.

2.8.1 Generalized Randomized Response (GRR)

Randomized Response (WARNER, 1965) was introduced for binary responses, as exemplified in in the previous examples, but it can easily be generalized to larger domains. In the GRR, each user with private value $v \in D$ sends their true value v with probability p. Otherwise, with probability $1 - p$, the users send a randomly chosen value $v' \in D \setminus v$. Formally, the perturbation algorithm is

$$\forall_{x \in D}\ Pr\left[\Psi_{GRR_{(\epsilon)}}(v) = x\right] = \begin{cases} p = \frac{e^{\epsilon}}{e^{\epsilon}+|D|-1} & \text{if x = v} \\ q = \frac{1-p}{|D|-1} = \frac{1}{e^{\epsilon}+|D|-1} & \text{if x} \neq \text{v} \end{cases}$$

GRR satisfies ϵ-LDP since $p/q = e^{\epsilon}$. From a population of n users, the aggregator receives a vector $\mathbf{x} = \langle x_1, x_2, ...x_n \rangle$ of length $|\mathbf{x}| = n$ where $x_i \in D$ is the reported value of the i-th user. Then, it can estimate the frequency of $v \in D$, which consists of the ratio of users with private value v among all n users. Considering $C(v)$ as the number of times v appears in vector $\mathbf{x}$, the algorithm for estimating the frequency of $v \in D$ is

$$\Phi_{GRR_{(\epsilon)}}(v) := \frac{C(v)/n - q}{p - q} = \frac{\frac{C(v)}{n} - \frac{1}{e^{\epsilon}+|D|-1}}{\frac{e^{\epsilon}-1}{e^{\epsilon}+|D|-1}} \tag{2.6}$$

This is an unbiased estimation of the true count (WANG *et al.*, 2017), and the variance for this estimation is

$$Var\left[\Phi_{GRR_{(\epsilon)}}(v)\right] = \frac{e^{\epsilon} + |D| - 2}{n(e^{\epsilon} - 1)^2} \tag{2.7}$$

Since variance (2.7) is linear in $|D|$, the accuracy of this protocol decreases when the domain size $|D|$ increases.

2.8.2 Optimized Local Hashing (OLH)

The Optimized Local Hashing (WANG *et al.*, 2017) protocol tackles the limitation of large domains observed in GRR by adopting a hash function to map an input value into a smaller domain of size g. It applies randomized response to the hashed value. To minimize the variance, the value of g should be $\lceil e^{\epsilon} + 1 \rceil$ (WANG *et al.*, 2017). The hashing process uses a universal hash function family called $\mathbb{H}$ such that each $H \in \mathbb{H}$ takes as input a value in D and outputs a value in $1...g$. The protocol used to report a value using OLH is $\Psi_{OLH_{(\epsilon)}}(v) := \left\langle H, \Psi_{GRR_{(\epsilon)}}(H(v)) \right\rangle$. Let $\left\langle H^i, x^i \right\rangle$ be the report from the i-th user. To compute the frequency of each $v \in D$, the aggregator first computes $C(v) = |\{j | H^j(v) = x^j\}| = \sum_{j \in n} \mathbb{1}_{\{H^j(v) = x^j\}}$, which is the number of user reports that supports the value v. After that, the aggregator transforms $C(v)$ to its unbiased estimation as follows

$$\Phi_{OLH_{(\epsilon)}}(v) := \frac{C(v)/n - 1/g}{\left(\frac{e^{\epsilon}}{(e^{\epsilon}+g-1)} - 1/g\right)} \tag{2.8}$$

The variance found for that estimation is

$$Var\left[\Phi_{OLH_{(\epsilon)}}(v)\right] = \frac{4e^{\epsilon}}{n(e^{\epsilon} - 1)} \tag{2.9}$$

2.9 Summary - Approaches for DP/LDP Query Answering

There are two common approaches for answering queries under differential privacy (DP): the query-based composition model and the data model generation approach. Each presents distinct trade-offs in terms of flexibility, utility, and computational requirements.

In the query-based composition model, each query is answered directly on the private dataset using a differentially private mechanism (e.g. Laplace Mechanism). The total privacy budget ϵ is split among Q queries using the composition theorem 2.7. This approach supports interactive querying, and yields high accuracy when the number of queries is small. Privacy guarantees are directly derived from DP composition theorem, providing precise control over cumulative privacy loss. However, its main limitation is that the privacy budget is consumed incrementally with each query. As the number of queries increases, the per-query budget becomes smaller, resulting in higher noise and reduced utility. It is important to notice that the budget ϵ may be non-uniformly allocated among queries. However, once the budget is exhausted, no further queries can be answered without compromising the privacy guarantee. This model is

well-suited for scenarios where the number of queries is known and limited in advance (LI *et al.*, 2010; LI; MIKLAU, 2012; YUAN *et al.*, 2012).

In contrast, the data model generation constructs a model using a differentially private algorithm that approximates the statistical properties of the original data. Once the synthetic model is released, an unbounded number of queries can be answered on it without additional privacy loss. This model facilitates sharing data with external analysts without requiring further access to the original private data. Unlike noisy answers to individual queries, synthetic data models supports complex analyses. However, generating high-fidelity synthetic data models that preserves complex distributions is computationally challenging and may lead to degraded utility, especially in high dimensions (ZHANG *et al.*, 2017; ABAY *et al.*, 2019; ZHANG *et al.*, 2016).

In contrast to both the query-based composition and synthetic data models generation approach under central differential privacy, the local differential privacy (LDP) model inherently supports an unbounded number of queries. In LDP, each user applies a randomized mechanism to their own data before transmission, ensuring privacy is preserved independently of the data aggregator. Since the privacy guarantee is enforced at the user level, the aggregator can issue any number of queries without further degrading the privacy of individual users. However, this flexibility comes at the cost of reduced accuracy. Because the noise is injected at the user level, prior to aggregation, the variance of LDP estimators is typically much higher than in central DP, especially when the population size is limited or the query domain is high-dimensional (YANG *et al.*, 2020; FILHO; MACHADO, 2023; WANG *et al.*, 2023; ZHANG *et al.*, 2018). As a result, while LDP allows for unlimited querying, it often requires larger populations or additional structural assumptions to achieve utility comparable to central DP approaches.

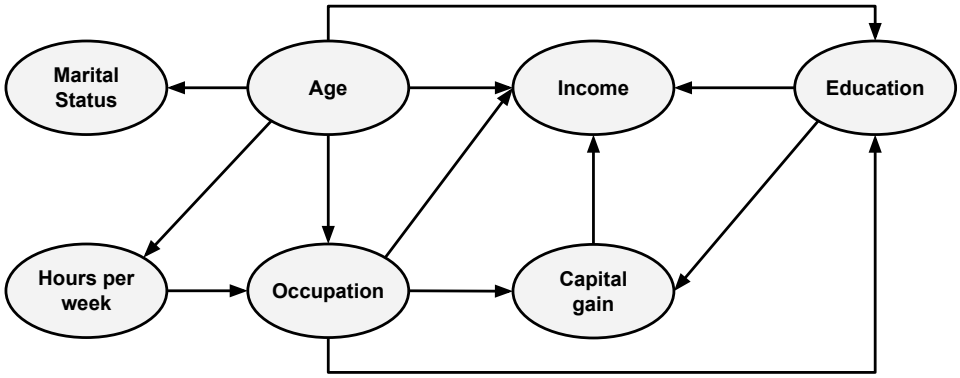
2.10 Bayesian Network

Let A denote the set of attributes in a dataset $\mathcal{D}$. We interpret $\mathcal{D}$ as a joint probability distribution defined over the Cartesian product of the domains of the attributes in A. A Bayesian network provides a compact representation of this joint distribution by encoding conditional independence relationships among the attributes. Formally, a Bayesian network over A is a directed acyclic graph (DAG) with the following characteristics:

1. Each node in the graph corresponds to a distinct attribute in A.
2. Models conditional independence among attributes in A using directed edges.

As an example, Figure 5 shows a Bayesian network over a set A of seven attributes,

Figure 5 – A Bayesian network over 7 attributes.



Source: elaborated by the author.

namely, marital status, age, income, education, hours per week, occupation and capital gain. For any two attributes $X, Y \in A$, there exist three possibilities for the relationship between X and Y:

- **Case 1: Direct Dependence.** There is an edge between X and Y, say, from Y to X. It indicates that for any tuple in $\mathcal{D}$, its distribution on X is determined (in part) by its value on Y. Y is defined as a parent of X, and we refer to the set of all parents of X as its parent set. For instance, we can see that the Income distribution depends on the occupation, age and capital gain.
- **Case 2: Weak Conditional Independence.** There is a path (but no edge) between Y and X. Assume without loss of generality that the path goes from Y to X. Then, X and Y are conditionally independent given X's parent set. For example, there is a path (2-steps) from Age to Capital gain and the parent set of Capital gain is Education and Occupation. This suggests that, given an individual's Education and Occupation, their Capital Gain and Age are conditionally independent.
- **Case 3: Strong Conditional Independence.** There is no path between Y and X. Then, X and Y are conditionally independent given any of X's and Y's parent sets.

Let d denote the size of A. Formally, a Bayesian network $\mathcal{N}$ over A is defined as a set of d attribute-parent (AP) pairs, $(X_1, \Pi_1), \ldots, (X_d, \Pi_d)$, such that:

1. Each X_i is a unique attribute in A;
2. Each Π_i is a subset of the attributes in $A \setminus \{X_i\}$. Then, Π_i is the parent set of X_i in $\mathcal{N}$;
3. For any $1 \leq i < j \leq d$, we have $X_j \notin \Pi_i$, *i.e.*, there is no edge from X_j to X_i in $\mathcal{N}$. This ensures that the network is acyclic (DAG).

We define the *degree* of $\mathcal{N}$ as the maximum size of any parent set Π_i in $\mathcal{N}$. Let $\Pr[A]$ denote the full distribution of tuples in database $\mathcal{D}$. The d AP pairs in $\mathcal{N}$ essentially define a way

to approximate $\Pr[A]$ with d conditional distributions $\Pr[X_1|\Pi_1], \Pr[X_2|\Pi_2], \ldots, \Pr[X_d|\Pi_d]$. Under the assumption that any X_i and any $X_j \notin \Pi_i$ are conditionally independent given Π_i, we have

$$\begin{aligned}\Pr[A] &= \Pr[X_1, X_2, \ldots, X_d] \\ &= \Pr[X_1] \cdot \Pr[X_2|X_1] \cdot \Pr[X_3|X_1, X_2] \cdots \Pr[X_d|X_1, \ldots, X_{d-1}] \\ &= \prod_{i=1}^{d} \Pr[X_i|\Pi_i]. \end{aligned} \tag{2.10}$$

Let $\Pr_N[A] = \prod_{i=1}^{d} \Pr[X_i|\Pi_i]$ be the above approximation of $\Pr[A]$ defined by $\mathcal{N}$. If $\mathcal{N}$ accurately models the conditional independence among the attributes in A, then $\Pr_{\mathcal{N}}[A]$ would be a good approximation of $\Pr[A]$. Additionally, if the Bayesian network $\mathcal{N}$ has a small degree, then the computation of $\Pr_N[A]$ is relatively simple as it requires only d low-dimensional distributions $\Pr[X_1|\Pi_1], \Pr[X_2|\Pi_2], \ldots, \Pr[X_d|\Pi_d]$.

2.11 Kullback–Leibler Divergence

The Kullback–Leibler divergence (PEREZ-CRUZ, 2008) (KLD) quantifies the difference between two probability distributions, denoted P and Q. For discrete distributions over the same support $\mathcal{X}$, it is defined as:

$$D_{\mathrm{KL}}(P \parallel Q) \;=\; \sum_{i \in \mathcal{X}} P(i) \, \log\left(\frac{P(i)}{Q(i)}\right).$$

A fundamental property of the KL divergence is that it is always non-negative:

$$D_{\mathrm{KL}}(P \parallel Q) \;\geq\; 0,$$

with equality if and only if $P = Q$ almost everywhere.

Intuitively, the closer $D_{\mathrm{KL}}(P \parallel Q)$ is to zero, the more similar the distributions P and Q are. As the KLD is moves away from zero, P and Q are more dissimilar (diverging, more distant).

3 RELATED WORK

In this chapter, we describe the related work. It encompasses the DP and LDP methods for answering range queries as well as the techniques that estimate frequencies and marginals since those can also be applied to answer range queries. Moreover, we present the main direct competitors baselines in detail, enumerating how they resolve or not the challenges mentioned in Chapter 1. Additionally, we present Table 3 which summarizes the main features of baselines along with the three strategies developed in this thesis.

3.1 Frequency estimation under LDP

Local Differential Privacy (LDP) was first introduced by (KASIVISWANATHAN *et al.*, 2011). A foundational theoretical analysis of LDP was carried out by (DUCHI *et al.*, 2013), who derived optimal bounds based on information-theoretic principles and proposed protocols that achieve minimax optimality. One of the most basic and fundamental tasks under LDP is frequency estimation, *i.e.*, estimating the distribution of values held by users. Accurate frequency estimation serves as the building block for a wide range of more complex tasks, including classification, histogram release, and joint distribution learning. A notable real-world implementation of frequency estimation under LDP is Google's RAPPOR system (ERLINGSSON *et al.*, 2014), which collects statistics from client devices while ensuring user-level privacy. RAPPOR encodes inputs using Bloom filters (BLOOM, 1970), followed by a randomized response mechanism (WARNER, 1965) to ensure differential privacy guarantees.

A significant body of work has since been devoted to improving frequency oracle mechanisms. (BASSILY; SMITH, 2015) proposed protocols that produce a succinct histogram representation of the data. A succinct histogram is a list of the most frequent items in the data (often called "heavy hitters") along with estimates of their frequencies. (BASSILY *et al.*, 2017) studied local differentially private heavy hitters algorithms achieving optimal or near-optimal worst-case error and running time – TreeHist and Bitstogram. (ACHARYA *et al.*, 2019) proposed the Hadamard Response HR, a local privatization scheme that requires no shared randomness and is symmetric with respect to the users. (YE; BARG, 2018) considered the minimax estimation problem of a discrete distribution with support size k under locally differential privacy and proposed a family of new privatization schemes. The work of (WANG *et al.*, 2017) provides a comparative analysis of a number of different LDP protocol, offering guidelines on when

each protocol is most effective depending on domain size, privacy budget, and expected data distribution. (GURSOY *et al.*, 2022) introduced a Bayesian adversarial model to analyze the behavior of LDP mechanisms under various privacy threat models. To reduce variance, (WANG *et al.*, 2020) proposed an efficient and optimal ϵ-LDP mechanism, wheel mechanism, for set-valued/categorical data distribution estimation. It is domain agnostic (i.e., independent of dimension size and utility optimal. Extensions of frequency estimation to high-dimensional and longitudinal settings have also been explored. (ARCOLEZI *et al.*, 2022b) introduced a new LDP data collection protocol for longitudinal frequency monitoring named LOLOHA with formal privacy guarantees. (ARCOLEZI *et al.*, 2022a) investigated re-identification and attribute inference attacks against LDP protocols for multidimensional data. (ARCOLEZI; GAMBS, 2024) introduced a framework for empirically estimating the privacy loss of locally differentially private mechanisms. (ARCOLEZI; GAMBS, 2025) proposed a general multi-objective optimization framework for refining LDP protocols, enabling the joint optimization of privacy and utility under various adversarial settings. (WANG *et al.*, 2019a) studied mechanisms for multidimensional data collection under LDP. (ARCOLEZI *et al.*, 2022b) extended the analysis of three state-of-the-art LDP protocols for both longitudinal and multidimensional data collections and proposed to randomly sample a single attribute to be sent with the whole privacy budget and adaptively select the optimal frequency oracle protocol.

3.2 Marginal Release under DP

Multi-dimensional range queries are often answered by aggregating information from λ-way marginals. However, releasing high-dimensional marginals while maintaining privacy and utility remains a central challenge. (CORMODE *et al.*, 2018) refined the theoretical foundations of marginal release under LDP by analyzing how transformations can improve accuracy and reduce redundancy. The CALM framework (ZHANG *et al.*, 2018) extended this by adapting consistency enforcement and maximum entropy estimation principles from the PriView method (QARDAJI *et al.*, 2014a). Other methods have focused on estimating joint distributions. (REN *et al.*, 2018) generalized the Expectation-Maximization algorithm for estimating pairwise joint distributions from locally privatized data. More recently, (LIU *et al.*, 2023) introduced a synthetic data generation framework that approximates the joint distribution of the original dataset while satisfying LDP. These approaches typically face utility degradation when the dimensionality or range of queries grows due to cumulative noise.

3.3 Range query under DP

The hierarchical decomposition mechanism was originally proposed by (HAY *et al.*, 2010). Utility is increased by enforcing consistency among the noisy frequencies along the tree. (QARDAJI *et al.*, 2013b) analyzed the different aspects that affect the accuracy of range queries under a hierarchical structure, and propose a way to decide the fanout of the tree structure. (ZHANG *et al.*, 2016) proposed a perturbation mechanism which decides whether a sub-domain on the tree should be split, thus there is no need of a pre-defined threshold on the depth of sub-domain splitting. (PROCOPIUC *et al.*, 2012) proposed spatial indexing methods such as quad trees and kd-trees to provide a private description of the data distribution to answer range queries. (ALNEMARI *et al.*, 2021) proposed a Multi-Attribute DisAssembly Mechanism for answering multidimensional range queries on healthcare data. (QARDAJI *et al.*, 2013a) proposed an adaptive grid method that lays a coarse-grained grid over the dataset, and then further partitions each cell according to its noisy count. Both levels of partitions are then used in answering queries over the dataset. This thesis also uses grids to map users' multidimensional data. However, unlike our work, (QARDAJI *et al.*, 2013a) is developed to work in the central setting of differential privacy, and it works for only two dimensions. Applying the same idea in the local setting is not trivial. Moreover, it proposes to split the privacy budget among cell-splitting phases which could add excessive noise in the local setting.

3.4 Range query under LDP

Range query mechanisms under LDP face more stringent utility constraints due to the absence of a trusted aggregator. (CORMODE *et al.*, 2019) compared two techniques in the 1-D domain: hierarchical structures and wavelet transforms. Their findings show that wavelet transforms excel under high privacy budgets, while hierarchical trees perform better at low privacy levels. (DU *et al.*, 2021) proposed a dynamic structure that adapts its depth based on data characteristics, enabling better noise management. (LI *et al.*, 2020) introduced the Square Wave approach, which leverages basis transformation to reconstruct the distribution of ordinal attributes. Recent work by (WANG; CHENG, 2022) proposed PRISM, which collects a small, data-dependent set of prefix-sums to support multi-dimensional range queries efficiently. This line of research has also broadened to other query types. LDP-based mechanisms have been applied to frequency and mean estimation in key-value datasets (GU *et al.*, 2020; YE *et al.*, 2021a;

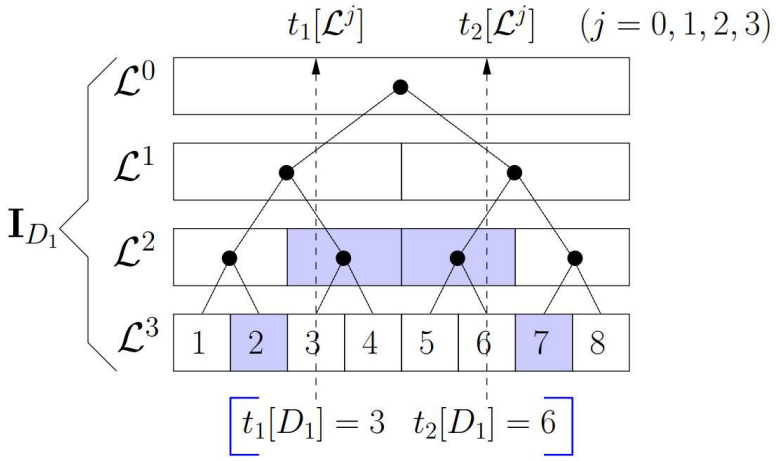
SUN *et al.*, 2019; LI *et al.*, 2022b), linear queries (EDMONDS *et al.*, 2020; BASSILY, 2019), geospatial analysis (HONG *et al.*, 2021; YE *et al.*, 2021b; GU *et al.*, 2019), and time-evolving or streaming datasets (JOSEPH *et al.*, 2018; WANG *et al.*, 2021; REN *et al.*, 2022).

The baselines evaluated in this thesis—HDG (YANG *et al.*, 2020), PrivNUD (WANG *et al.*, 2023), and HIO (WANG *et al.*, 2019a)—offer a range of trade-offs across accuracy, scalability, and dimensionality. Table 3 summarizes the key differences among these baselines and how they handle multi-attribute range queries in the local privacy setting.

3.5 HIO

(WANG *et al.*, 2019a) proposed a hierarchical decomposition approach designed to answer multi-dimensional analytical queries under LDP. Given an ordinal dimension k_i that has domain d, HIO constructs a hierarchical collections of intervals with a *fan-out* b. Each node corresponds to an interval and has b children (except for the leaves), corresponding to b equally sized subintervals ($d = b^h$). The leaves have unit length. There are b^j intervals on level j and each interval covers d/b^j values. Users are partitioned into h groups where h is the height of the tree. Users in level j send their LDP reports of level j using privacy budget ϵ. Figure 6 shows an example of a 1-D domain ($d = 8$) decomposition with *fan-out* $b = 2$. Consider a range query q_1: SELECT COUNT(*) FROM T WHERE $D_1 \in [2, 7]$. To answer q_1, the frequency estimates of subintervals in blue are summed.

Figure 6 – Hierarchy of intervals for 1-D domains

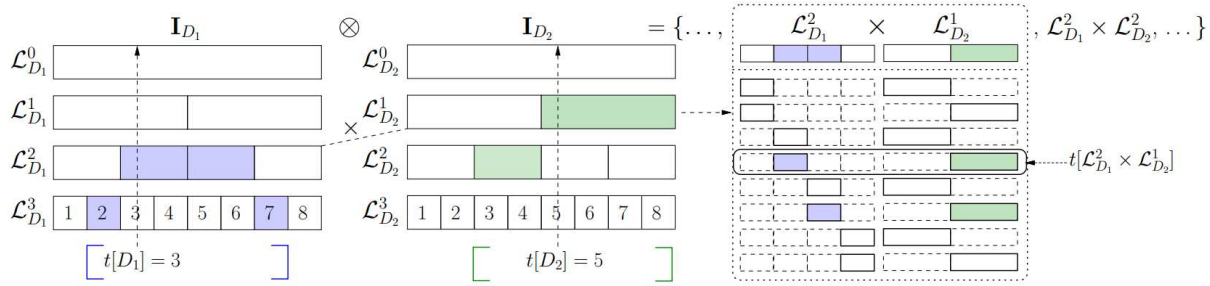


Source: (WANG *et al.*, 2019a)

HIO extends the decomposition of 1-D domains to multiple dimensions by introducing

a multi-dimensional hierarchy of intervals. It defines k-dim level hierarchy with a total of $(h+1)^k$ levels. For the multi-dimensional case, users pick one of the $(h+1)^k$ k-dim level uniform at random and send their LDP reports. HIO uses OLH to get the noisy frequencies of all k-dimensional intervals in every k-dimensional levels. Then, the aggregator expands a query q to a new k-dimensional range query q' that is interested in all dimensions and not only the ones specified in query q. For these other added dimensions, it assigns the complete interval (*i.e.,* covers the entire domain d). Next, for each attribute k in the query q', the aggregator finds the least number of subintervals that can make up its specified interval in q' from its corresponding 1-D hierarchy. In the last step, the aggregator sums all estimated frequencies of all the k-intervals to have the query answer. Figure 7 show an example of 2-dimensional hierarchy of dimensions D_1 and D_2 that take values in $1, 2, ..., 8$. Their individual 1-dim hierarchies I_{D_1} and I_{D_1} are on the left, each with 4 levels. The 2-dim hierarchy is a Cartesian product of the two, with 4×4 2-dim levels. In particular, the 2-dim level, $\mathcal{L}^2_{D_1} \times \mathcal{L}^1_{D_2}$, depicted on the right, is a Cartesian product of two 1-dim interval sets $\mathcal{L}^2_{D_1}$ and $\mathcal{L}^1_{D_2}$, with 4×2 2-dim intervals, each of which is a pair of 1-dim intervals, with one from $\mathcal{L}^2_{D_1}$ and the other from $\mathcal{L}^1_{D_2}$. For example, the 4th one (top-to-bottom) in the figure is [3, 4][5, 8]. Consider a tuple t with $t[D_1] = 3$ and $t[D_2] = 5$. It belongs to the above 2-dim interval as $t[D_1] \in [3, 4]$ and $t[D_2] \in [5, 8]$. Thus, the augmented dimension $t[\mathcal{L}^2_{D_1} \times \mathcal{L}^1_{D_2}] = [3, 4][5, 8]$.

Figure 7 – Hierarchy of intervals for 2-D domains.



Source: (WANG *et al.*, 2019a)

Consider the following 2-D query q_2: SELECT COUNT(*) FROM T WHERE $D_1 \in [2, 7]$ AND $D_2 \in [3, 8]$. As shown in Figure 7, [2, 7] can be partitioned into 4 intervals in I_{D_1} (blue ones), and [3, 8] partitioned into 2 in I_{D_2} (green ones). Thus, q_2 can be decomposed into 4 × 2 disjoint sub-queries that are to be answered using weighted frequency oracles – two are on the 2-dim level $t[\mathcal{L}^2_{D_1} \times \mathcal{L}^1_{D_2}$: one with "$D_1 \in [3, 4]$ AND $D_2 \in [5, 8]$" and one with "$D_1 \in [5, 6]$ AND $D_2 \in [5, 8]$".

Limitations. Since users are divided into $(h+1)k$ groups where $h = \log_b d$. The number of users in each group decreases when the attributes' domain or the number of dimensions increases. That translates to a high amount of noise in the frequencies of k-dim intervals, which results in low utility. Therefore, HIO fails to handle the curse of dimensionality and attributes with large domains.

3.6 HDG and TDG

Two-Dimensional Grid (TDG) (YANG *et al.*, 2020) is a grid-based approach focused on answering multi-dimensional range queries under LDP. TDG's idea is to use binning to partition the 2-D domains of all attribute pairs into 2-D grids that can answer all 2-D range queries and then estimate the answer of a higher dimensional range query from the answers of corresponding 2-D range queries. In TDG, given k attributes, the aggregator first generates all attribute pairs. Then, the aggregator divides m users into $\binom{k}{2}$ groups. Each group corresponds to a pair of attributes. Next, for each attribute pair $\langle a_i, a_j \rangle$ where $1 \leq i < j \leq k$, the aggregator assigns the same granularity (*i.e.,* grid size of one dimension) g_2 to construct a 2-D grid $G(i, j)$ by partitioning the 2-D domain $d \times d$ into $g_2 \times g_2$ 2-D cells of equal size. It assumes that all attributes have the same domain. In particular, each 2-D cell specifies a 2-D subdomain of $d \times d$. Each user reports their private value on one of these grids, and the frequencies of each cell are estimated using OLH. When answering queries using TDG, the aggregator assumes that values in a cell are uniformly distributed, which may incur additional error.

Hybrid-Dimensional Grid (HDG) introduces finer-grained 1-D grids that, in combination with the information of 2-D grids, provide better utility when compared to TDG. In HDG, the aggregator constructs k 1-D grids for the k attributes, respectively. Thus, there are a total of $k + \binom{k}{2}$ grids in HDG, with users being divided into $m = k + \binom{k}{2}$ groups. As in TDG, one user group reports values on one specific grid. The aggregator calculates the granularity g_1, which is the size of all 1-D grids. Each 1-D cell specifies a 1-D subdomain of d. Finally, the aggregator uses OLH to obtain noisy frequencies of every cell of 1-D and 2-D grids. On the aggregator side, negativity is removed from the estimates. Also, since an attribute is related to multiple grids, inconsistencies among estimates are removed. After that, $\lambda - D$ queries are split into $\binom{\lambda}{2}$ related 2-D range queries and then from all answers of these $\binom{\lambda}{2}$ 2-D queries the answer of the $\lambda - D$ is estimated.

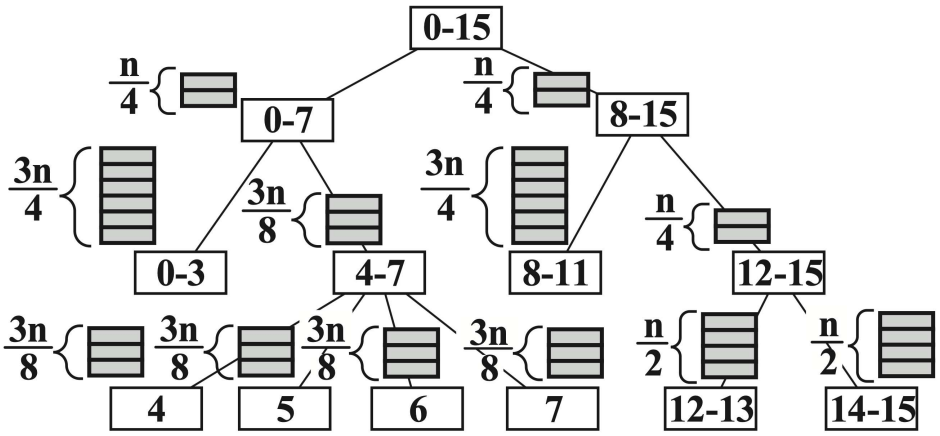
Limitations. We identify three main limitations with TDG and HDG. First, they only support range queries, limiting the analysis one may be interested in performing. Also, it assumes that queries will correspond to half the interval of attributes and that assumption impacts the construction of grids, which limits the utility gains. Additionally, it uses the same size for every grid. Moreover, to ensure that g_1 and g_2 are divisible by domain size d simultaneously, they take the power of two closest to its derived value as the final value. These aspects negatively impact accuracy since the actual size of the grids may be very different from the optimal one. For instance, suppose the optimal granularity is 25 for 1-D grid, the granularity of the actual grid will be 32, which is 20% larger than the optimal value. Also, suppose that the optimal value of a 2-D grid is 11×11, totaling 121 cells; then the actual granularity will be 8×8 totaling 64 cells. Then, the total number of cells will be almost half of what it should be.

3.7 PrivNUD

(WANG *et al.*, 2023) proposed a domain decomposition mechanism for one-dimensional range query under LDP, which decomposes the sub-domains with non-uniform granularities. PrivNUD adopts a heuristic algorithm to iteratively search a better granularity for the domain to be decomposed, in order to improve the performance when answering range queries. The non-uniform decomposition makes the paths on the tree have different lengths. PrivNUD chooses the decomposition granularity by minimizing the sum of error variances brought by answering the effected queries with the decomposed sub-domains. PrivNUD improves accuracy by stopping decomposing the sub-domain with smaller frequency and estimating the frequency distribution on the domain with uniform distribution assumption.

Figure 8 illustrates an example of the hierarchical tree structure produced by the PrivNUD algorithm. In this structure, internal nodes may exhibit varying fanouts, meaning that the corresponding data domains are partitioned with differing levels of granularity. Additionally, the paths from the root to the leaf nodes are of non-uniform lengths, reflecting the flexibility of the decomposition process. Each node is annotated with a gray box indicating the subset of users assigned to it for frequency estimation; the height of the box visually encodes the number of users. Notably, the number of users allocated to nodes can vary significantly, even among nodes at the same level in the hierarchy. This variation results from an adaptive user allocation strategy employed by the aggregator. Specifically, when estimating the frequency of a given node, the

Figure 8 – Example of the non-uniform decomposition mechanism of PrivNUD.



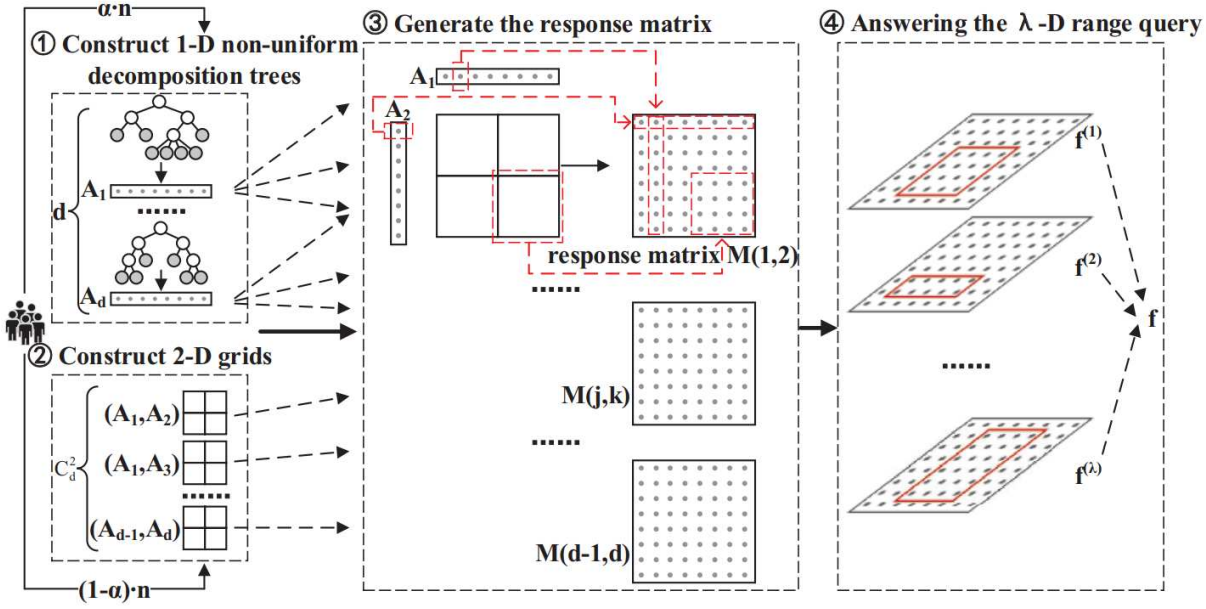
Source: (WANG *et al.*, 2023)

aggregator first determines the path length ℓ from a leaf to the current node, based on the current decomposition granularity. The available user set is then divided into ℓ disjoint subsets, and only one subset is assigned to the node. Because the decomposition granularity is tailored to each domain, the resulting size of the user subset may differ across nodes. This design enables the allocation strategy to adapt to the heterogeneity of domain structures and ensures efficient utilization of user reports across the hierarchy.

Figure 9 shows the workflow used to answer multi-dimensional λ-D range queries, PrivNUD materializes all pair-wise 2-D grids, similar to (YANG *et al.*, 2020). A total of $n\alpha$ users are allocated to estimate frequencies of 1-D domains and to estimate the frequencies of 2-D domains, $(1-\alpha)n$ users are used. The aggregator estimates the answer of high-dimensional queries from answers of associated 2-D queries.

Limitations: PrivNUD materializes a large number of grids since it assumes all attributes are correlated. Regarding the 2-D grids of PrivNUD, it uses the same size for every grid. Moreover, to guarantee that both g_1 (size of 1-D grids) and g_2 (size of 2-D grids), are divisible by the domain size d, the mechanism rounds each to the nearest power of two. While this simplifies the decomposition process and ensures compatibility with the domain, it can substantially deviate from the theoretically optimal grid size. As a result, the final grid configuration may introduce suboptimal partitioning, which negatively affects the overall accuracy of the frequency estimation. Finally, it assumes that each materialized grid has the same importance during query answering.

Figure 9 – PrivNUD flowchart



Source: (WANG *et al.*, 2023)

3.8 Baselines limitations summary

Table 3 compares to all techniques by highlighting the main challenges and features associated with privately answering multi-dimensional queries. It includes the state-of-the-art strategies: HIO, TDG/HDG and PrivNUD. Additionally, it lists our three different approaches named FELIP, PrivMDC and DP-AE presented in this thesis in Chapters 4, 5 and 6, respectively.

Feature / Method	**Avoid the curse of dimensionality**	**Deal with large domains**	**Capture Correlations**	**Multi-dimensional grids/levels** ($\lambda > 2$)	**Optimize user distribution**	**Heterogeneous privacy**	**Adaptive Estimator**
HIO	No	No	Multi-Dimensional	Yes	No	No	No
TDG/HDG	Yes	Yes	All pair-wise	No	No	No	No
PrivNUD	Yes	Yes	All pair-wise	No	No	No	No
FELIP	Yes	Yes	All pair-wise	No	No	No	No
PrivMDC	Yes	Yes	Multi-Dimensional	Yes	Yes	Yes	No
DP-AE	Yes	Yes	Multi-Dimensional	Yes	Yes	Yes	Yes

Table 3 – Baselines comparison and how they deal with the different challenges when answering multi-dimensional range queries.

4 FELIP: A LOCAL DIFFERENTIALLY PRIVATE APPROACH TO FREQUENCY ESTIMATION ON MULTIDIMENSIONAL DATASETS

In this chapter, we tackle the problem of answering an unbounded number of queries, with point and range constraints, where all users do not trust a central curator and only agrees to share a noisy version, under LDP, of their data. We start by presenting an overview of our proposed solution: FELIP. Then, in Section 4.3, we formalize the grid construction algorithm for all types of grids. Next, in Section 4.6, we present the post-processing techniques and the algorithm for estimating answers to high dimensional queries from noisy LDP reports. Finally, we discuss the privacy and expected utility in Section .

4.1 Overview

To tackle the problem of answering an unbounded number of queries, we propose to use two-dimensional grids to map, using binning, two-dimensional domains of all combinations of two data attributes. Once all these grids are constructed, one can answer any higher dimensional queries from them. When dealing with range constraints from a query, there is the possibility that a grid's cell is partially included in the answer. One approach is to assume that the data distribution is uniform, which may incur higher errors. Another approach is to combine the two-dimensional grids with other one-dimensional grids (YANG *et al.*, 2020). The purpose of one-dimensional grids is to capture finer-grained information on the distribution of each attribute. In this chapter, we study both approaches and make further enhancements to each strategy. The total number of grids depends on the number of attributes in the dataset.

To improve the utility, we choose to divide the population of users into groups, and each group will be responsible for collecting information from one grid. Each user reports a perturbed grid under LDP to the aggregator. With all reports, the aggregator estimates the frequency of each grid cell. After that, in the post-processing phase, we remove the negative frequency estimations since it is known that all individual frequencies should be non-negative, and summing all of them, the result should be one. Also, in the post-processing phase, since each attribute is involved in various grids, we make the estimations consistent for each attribute across respective grids, which helps to improve utility.

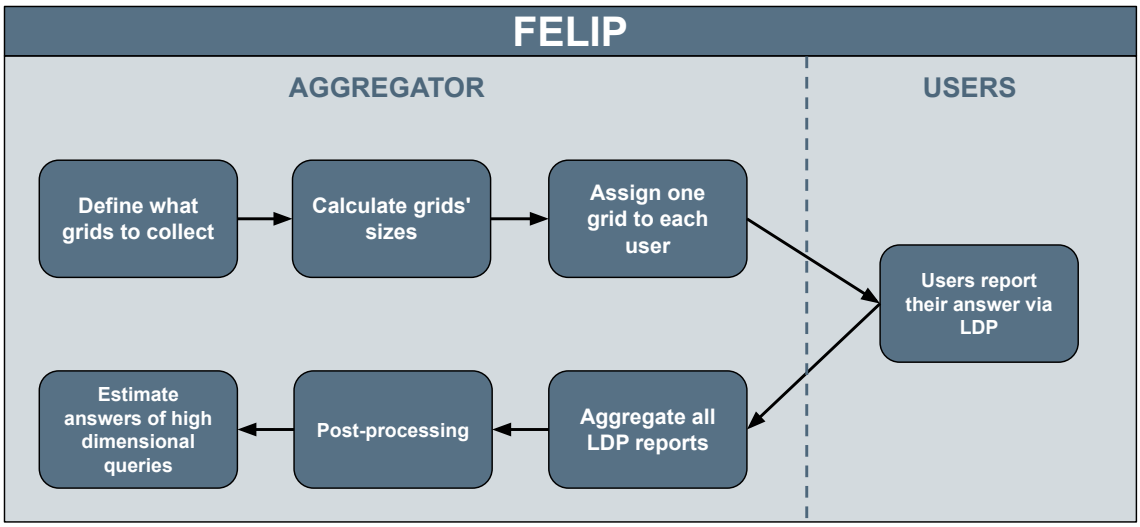
The most challenging task here is determining the best size for each grid. A finer-grained grid leads to more significant errors due to noises since, under LDP, one has to perturb each cell. The more cells, the higher the total noisy error. However, coarser-grained grids will

lead to high error due to bias. Choosing the size of the grids can be viewed as a bias-variance tradeoff. To decide on the exact size of each grid, we thoroughly analyze the error involved when answering a query. The decision will be based on which attributes are in the grid (grids can be categorical, numerical, numerical $\times$ numerical, categorical $\times$ categorical, numerical $\times$ categorical), the query selectivity on each dimension, the number of users, privacy budget and a data distribution property. In this work, we also study different frequency oracle (FO) protocols when estimating the frequencies of grids' cells. Once we have determined the exact size of the grid, we dynamically choose what FO will lead to a better utility for that specific grid. By using 2-D grids, we are able to capture correlation among attributes and avoid the curse of dimensionality. By carefully binning for each grid, we avoid the problem of dealing with large domains.

FELIP maps users answers to 1-D and 2-D grids which are perturbed to satisfy LDP. Figure 10 shows the step-by-step of FELIP. The first task consists of the aggregator determining the number of grids to be estimated and dividing the users into groups where each group is responsible for one grid. In the second step, the aggregator calculates the dimensions' size of each grid, and it can leverage the information it has about the queries' selectivity. Note that the grid construction happens on a per-data collection basis. With one collection, the aggregator can answer all queries. So when choosing the selectivity, the aggregator can use a precise number of one specific query it wants to answer, or it can use the average selectivity of a set of queries. After that, the aggregator sends to each user one grid configuration. On the user side, each user projects their answer in the grid, perturbs it under LDP, and sends it to the aggregator. Next, the aggregator collects all answers and estimates the frequencies for each cell of every grid. After that, the post-processing is done. Finally, the aggregator estimates the answers of all high-dimensional queries.

Within FELIP, we develop two strategies called Optimized Hybrid Grid (OHG) and Optimized Uniform Grid (OUG). Since FELIP collects multiple grids, we argue that the utility can be improved by dividing users into groups and show the proofs for that when using OLH and GRR protocols (Section 4.2). Then, we discuss one of the most important aspects of collecting the data, that is, how we can optimize the construction of LDP grids by minimizing the total error while using OLH and GRR protocols (Section 4.3). Next, we motivate why combining GRR and OLH protocols can benefit our problem, presenting the adaptive frequency oracle (Section **??**). After that, we detail how the aggregator can improve utility by achieving consistency among

Figure 10 – Overview of FELIP.



Source: elaborated by the author.

grids and removing negative estimations in the post-processing stage (Section 4.6). Next, we show the algorithms used for building the response matrix (Section 4.6.3) and estimating the multi-dimensional query from its related sub-queries (Section 4.6.4). Finally, we discuss the privacy guarantees of our approach and specify the different sources of error (Section 4.7).

4.2 Population Partitioning

In our approach, we collect, under LDP, information on m different grids, and each grid has L cells. One strategy is to divide the privacy budget into m parts, and the entire population of n users sends reports for all m grids, using the sequential composition (DWORK *et al.*, 2006; MCSHERRY, 2009b). The other strategy is to divide the population into m groups; each group is responsible for reporting information about one specific grid.

Theorem 4.1 (Population Partitioning)**.** The variance of GRR and OLH is smaller when dividing n users into m groups instead of dividing the privacy budget ϵ into ϵ/m.

Proof. The variance when answering a query dividing the users with GRR is:

$$Var[\Phi_{GRR_{du}}] = m \cdot \frac{e^{\epsilon} + L - 2}{n(e^{\epsilon} - 1)^2}$$

where L is the domain size (*i.e.*, number of cells within a grid). The variance when splitting the budget is:

$$Var[\Phi_{GRR_{d\epsilon}}] = \frac{e^{\epsilon/m} + L - 2}{n(e^{\epsilon/m} - 1)^2}$$

Then, if we do $Var[\Phi_{GRR_{d\epsilon}}] - Var[\Phi_{GRR_{du}}]$, we have

$$
\begin{aligned}
=&\frac{1}{n}\left[\frac{e^{\epsilon/m}+L-2}{(e^{\epsilon/m}-1)^2} - m\frac{e^{\epsilon}+L-2}{(e^{\epsilon}-1)^2}\right] \\
=&\frac{L-2}{n}\left[\frac{1}{e^{\epsilon/m}-1)^2} - \frac{m}{(e^{\epsilon}-1)^2}\right] + \\
&\frac{e^{\epsilon/m}}{n}\left[\frac{1}{e^{\epsilon/m}-1)^2} - \frac{me^{\epsilon-\epsilon/m}}{(e^{\epsilon}-1)^2}\right] \\
=&\frac{(L-2)}{n(e^{\epsilon/m}-1)^2(e^{\epsilon}-1)^2}\left[(e^{\epsilon}-1)^2 - m(e^{\epsilon/m}-1)^2\right] + \\
&\frac{e^{\epsilon/m}}{n(e^{\epsilon/m}-1)^2(e^{\epsilon}-1)^2}\left[(e^{\epsilon}-1)^2 - me^{\epsilon-\epsilon/m}(e^{\epsilon/m}-1)^2\right]
\end{aligned}
$$

Denoting $e^{\epsilon/m}$ as z and since $\epsilon > 0, z > 1$, the two first terms will always be greater than zero and for the two second terms we have that

$$
\begin{aligned}
&(e^{\epsilon}-1)^2 - m(e^{\epsilon/m}-1)^2 \\
=&(z^m-1)^2 - m(z-1)^2 \\
=&(z-1)^2\left[(1+z^2+...+z^{m-1})^2 - m\right] > 0
\end{aligned}
$$

Also, we have that

$$
\begin{aligned}
&(e^{\epsilon}-1)^2 - me^{\epsilon-\epsilon/m}(e^{\epsilon/m}-1)^2 \\
=&(z^m-1)^2 - mz^{m-1}(z-1)^2 \\
=&(z-1)^2\left[(1+z^2+...+z^{m-1})^2 - mz^{m-1}\right] > 0
\end{aligned}
$$

Therefore, $Var[\Phi_{GRR_{d\epsilon}}] - Var[\Phi_{GRR_{du}}] > 0$. That is, the variance of the approach that divides the privacy budget $Var[\Phi_{GRR_{d\epsilon}}]$ is always greater than the variance of the approach that divides users $Var[\Phi_{GRR_{du}}]$. The proof for the OLH protocol is similar to GRR, and we have that $Var[\Phi_{OLH_{d\epsilon}}] - Var[\Phi_{OLH_{du}}] > 0$. That is, the approach of dividing the privacy budget $Var[\Phi_{OLH_{d\epsilon}}]$ is always greater than the approach of dividing users $Var[\Phi_{OLH_{du}}]$. Then, we can conclude that for either protocol, OLH and GRR, dividing users will achieve better utility than dividing the budget. □

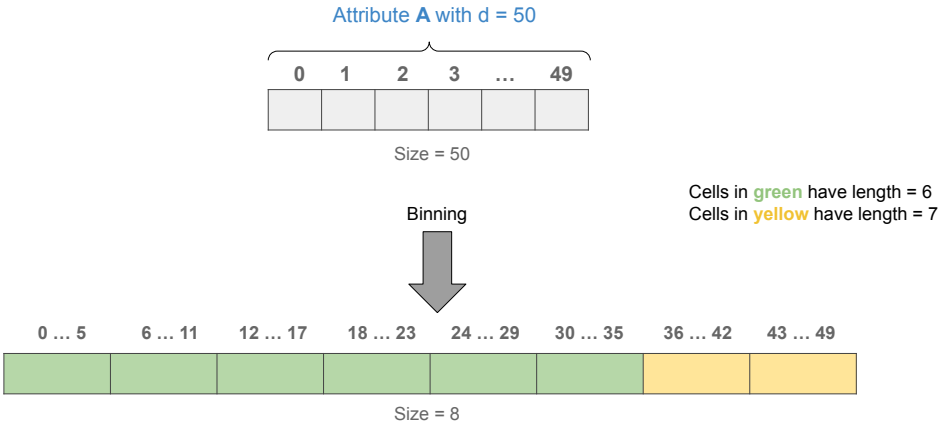
4.3 Constructing Grids with OLH and GRR

This section discusses how grids are constructed for OUG and OHG while using the OLH and GRR protocols. We design our strategies to handle numerical and categorical

attributes with different domain sizes. Also, we incorporate knowledge of the queries' selectivity. By considering these aspects, we can calculate more accurate grid sizes, which translates to a higher utility by balancing the bias-variance tradeoff well. The construction of grids varies in complexity depending on the grid type. It can be O(1) for 1-D grids, or it can involve solving an equation system numerically, which can be done with several different algorithms. In this work, we use the bisection method in all scenarios, and it has converged quickly in all of them.

In OUG, given k attributes, a total of $\binom{k}{2}$ pairs of attributes are generated, representing all the possible attribute combinations. The n users' population is randomly divided into $\binom{k}{2}$ groups. The size of a grid is represented by l_x and l_y, which refers to the size of the grid along the x and y axis, respectively. For each pair of attributes $\langle a_i, a_j \rangle$ where $1 \leq i < j \leq k$, a 2-D grid $G(i, j)$ is constructed partitioning the 2-D domain $[d_i]$ x $[d_j]$ into $l_x \times l_y$ 2-D cells.

Figure 11 – Domain reduction of 1-D dimensions. FELIP allows for cells to have different sizes so grids have precisely the optimal size.

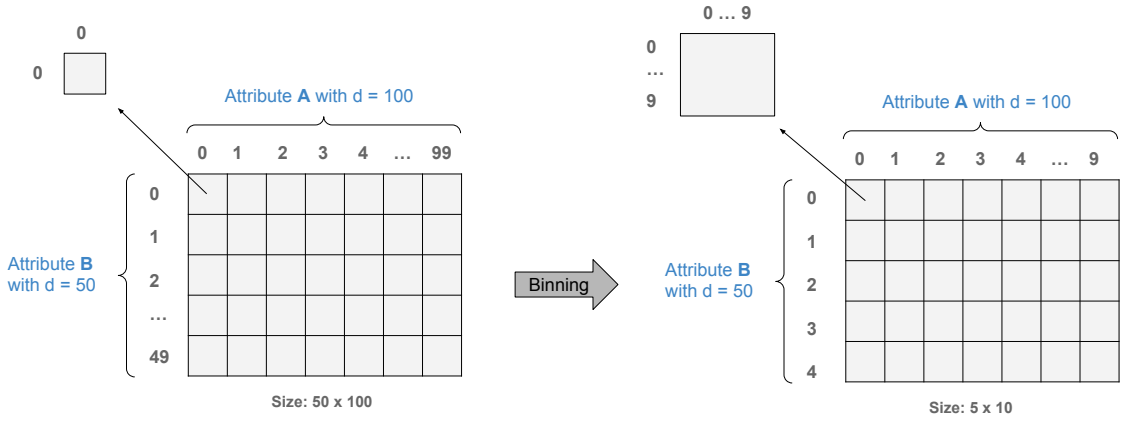


Source: elaborated by the author.

Figure 11 shows the process of using binning to reduce the domain size of 1 dimensional grids. Attribute A has domain $d = 50$ and, in the example, it is found that specific grid should have 8 cells ($l_x = 8$). Important to note that certain cells may cover a larger sub-interval than others in order to be constructed with the precise optimal size found $l_x = 8$. Figure 12 shows a similar process, but for 2-D domains being formatted to a 2-D grid. In the figure a $[d_a = 100]$ x $[d_b = 50]$ is mapped into grid of size $l_x = 10 \times l_5$. By using binning, FELIP avoids large errors when answering range queries for instance, since only the frequency of a small number of cells will have to be added to the estimation.

In the OHG strategy, 2-D and 1-D grids are constructed. 1-D grids are constructed

Figure 12 – 2-D grids are constructed to represent 2-D domains. By using binning, FELIP solves the challenge of dealing with large domains.



Source: elaborated by the author.

for numerical attributes only. Later, we utilize this information to refine the final answer grid (Section 4.6.3). As in OUG, 2-D grids are constructed for each pair of attributes. From a total of k attributes, k_n represents the number of numerical attributes. The population of n users is divided into $k_n + \binom{k}{2}$ groups, and each group is responsible for one grid. 2-D grids are constructed the same way as in OUG. For each numerical attribute $a_i (1 \leq i \leq k)$, 1-D grid $G(i)$ is constructed with size l_x. Finally, users are asked to report which cell of their assigned 2-D or 1-D grid their private value is in using the AFO protocol.

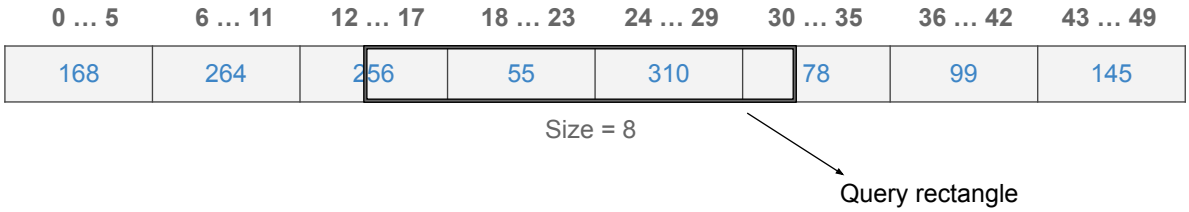
While constructing grids, we have to consider the effect of non-uniformity error in the numerical dimensions. The non-uniformity error accounts for the difference caused by cells that intersect the query rectangle but are not contained in it. Since we do not have access to the actual data distribution, we need to estimate how many data points are in the intersected cells. A common approach assumes that the data points are uniformly distributed in each cell (QARDAJI *et al.*, 2013a; YANG *et al.*, 2020). The magnitude of this error in any intersected cell, in general, depends on the number of data points in that cell, and is bounded by it. Thus, the finer the grid's granularity, the lower the non-uniformity error. When computing the size of the grids, we approximate the non-uniformity error and control the degree of this error by using constants α_1 and α_2 for 1-D and 2-D grids, respectively.

4.4 Calculating the size of grids for OLH and GRR

4.4.1 Numerical 1-D Grids

To construct 1-D numerical grids, we need to estimate the error of a typical 1-D query on such grid type. Figure 14 illustrates this process. The 1-D grid shown has a total of 8 cells. We can see that 2 cells are totally included in the query rectangle and 2 cells are partially included. Once we estimate the total error of a general query, we can minimize the error with respect to the size (*i.e.* number of cells) of the grid l_x.

Figure 13 – Example of answering a count query on 1-D numerical grid.



Source: elaborated by the author.

We consider that the ratio of interval to the attribute's domain size *i.e.,* query selectivity is r_x and the number of cells in the grid is l_x. The squared noise and sampling error for OLH is

$$(l_x \cdot r_x) \cdot \frac{4me^{\epsilon}}{n(e^{\epsilon}-1)^2} = \frac{4l_x r_x m e^{\epsilon}}{n(e^{\epsilon}-1)^2}$$

For GRR is

$$(l_x \cdot r_x) \cdot \frac{m(e^{\epsilon}+l_x-2)}{n(e^{\epsilon}-1)^2} = \frac{l_x r_x m(e^{\epsilon}+l_x-2)}{n(e^{\epsilon}-1)^2}$$

The non-uniformity error is $\left(\frac{\alpha_1}{l_x}\right)$. The sum of the two errors for OLH is

$$\left(\frac{\alpha_1}{l_x}\right)^2 + \frac{4l_x r_x m e^{\epsilon}}{n(e^{\epsilon}-1)^2} \tag{4.1}$$

and for GRR is

$$\left(\frac{\alpha_1}{l_x}\right)^2 + \frac{l_x r_x m(e^{\epsilon}+l_x-2)}{n(e^{\epsilon}-1)^2} \tag{4.2}$$

We can minimized the sum by taking the partial derivative with respect to l_x. For OLH, we have

$$\frac{\partial\left[\left(\frac{\alpha_1}{l_x}\right)^2+\frac{4l_x r_x m e^\epsilon}{n(e^\epsilon-1)^2}\right]}{\partial l_x} = \frac{4e^\epsilon m r_x}{n(e^\epsilon-1)^2} - \frac{2\alpha_1^2}{l_x^3} = 0$$

$$l_{xOLH} = \sqrt[3]{\frac{n\alpha_1^2\cdot(e^\epsilon-1)^2}{2mr_x e^\epsilon}} \tag{4.3}$$

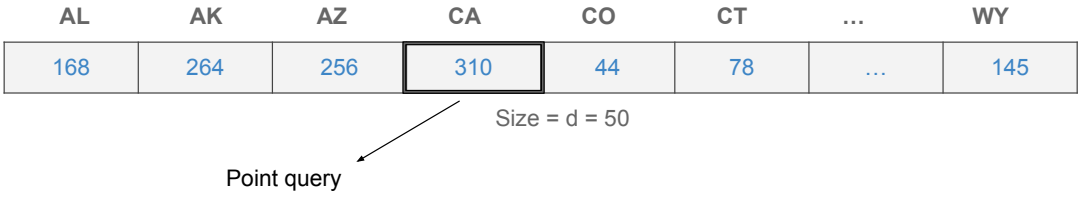
and for GRR, we take the partial derivative of Equation 4.2 with respect to l_x, we have

$$\frac{\partial\left[(4.2)\right]}{\partial l_x} = -\frac{2\alpha_1}{l_x^3} + \frac{l_x r_x m}{n(e^\epsilon-1)^2} + \frac{(l_x+e^\epsilon-2)ms}{n(e^\epsilon-1)^2} \tag{4.4}$$

Then, we find the roots of the Equation 4.4 and determine $l_{x_{GRR}}$. Once we have $l_{x_{OLH}}$ and $l_{x_{GRR}}$, we determine the size of the grid (*i.e.* number of cells).

4.4.2 Categorical 1-D Grids

Figure 14 – Example of answering a count query on 1-D categorical grid. Each category represent a state. Total size of the grid is 50 cells.



Source: elaborated by the author.

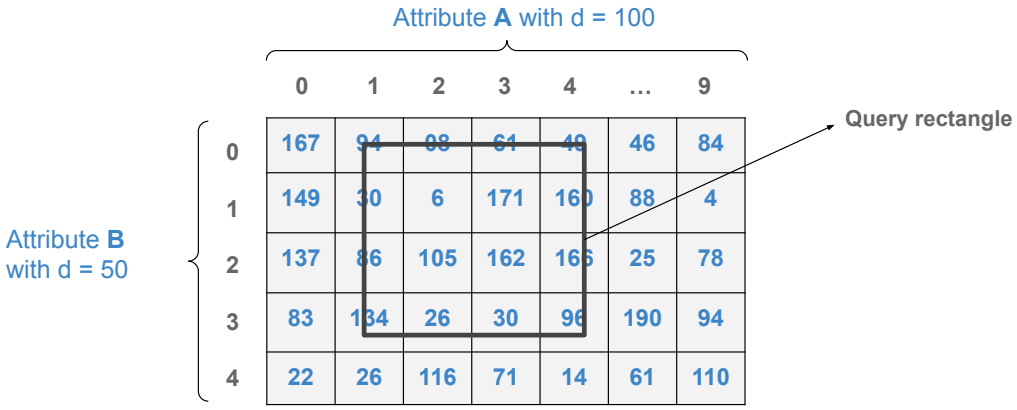
The size of the grid is equal to the domain of the attribute $l_{x_{OLH}} = l_{x_{GRR}} = d$.

4.4.3 Numerical × Numerical 2-D Grids

The process of defining the grid size of 2-D numerical grids follows a similar process as in 1-D. Figure 15 shows a 2-D query over a grid AxB that has size $5x10$.

We consider that the ratio of interval to the attribute's domain size *i.e.*, query selectivity is r_x and r_y along the x and y axis respectively. The number of cells included in the

Figure 15 – Example of answering a count query on 2-D [numerical x numerical] grid.



Source: elaborated by the author.

query rectangle is $l_x r_x \cdot l_y r_y$. Thus, the squared noise and sampling error for OLH is

$$l_x r_x l_y r_y \cdot \frac{4me^\epsilon}{n(e^\epsilon - 1)^2} = \frac{4l_x r_x l_y r_y m e^\epsilon}{n(e^\epsilon - 1)^2} \tag{4.5}$$

and for GRR is

$$(l_x r_x \cdot l_y r_y) \cdot \frac{m(e^\epsilon + l_x l_y - 2)}{n(e^\epsilon - 1)^2} = \frac{l_x r_x l_y r_y m(e^\epsilon + l_x l_y - 2)}{n(e^\epsilon - 1)^2} \tag{4.6}$$

The number of cells included in the four edges of the query rectangle is $2(l_x r_x + l_y r_y)$. The expected sum of frequencies of values included in these cells is $2(l_x r_x + l_y r_y) \cdot \frac{1}{l_x \times l_y} = \frac{2(l_x r_x + l_y r_y)}{l_x l_y}$. The non-uniformity error is $\left(\frac{2\alpha_2(l_x r_x + l_y r_y)}{l_x l_y}\right)^2$. The sum of the two errors for OLH is

$$\left(\frac{2\alpha_2(l_x r_x + l_y r_y)}{l_x l_y}\right)^2 + \frac{4l_x r_x l_y r_y m e^\epsilon}{n(e^\epsilon - 1)^2} \tag{4.7}$$

and for GRR is

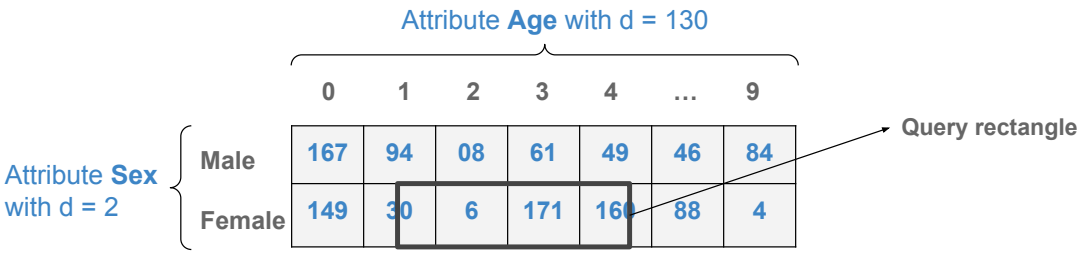
$$\left(\frac{2\alpha_2(l_x r_x + l_y r_y)}{l_x l_y}\right)^2 + \frac{l_x r_x l_y r_y m(e^\epsilon + l_x l_y - 2)}{n(e^\epsilon - 1)^2} \tag{4.8}$$

Taking the partial derivative of Equations 4.7 and 4.8 with respect to l_x and l_y, we get a system of polynomial equations for each one. Then, we solve them to find the values l_x and l_y for each protocol. The grid's size for GRR and OLH is $l_{x_{GRR}} \times l_{y_{GRR}}$ and $l_{x_{OLH}} \times l_{y_{OLH}}$ for GRR and OLH respectively.

4.4.4 Categorical x Numerical grid 2-D Grids

For 2-D grids with one categorical attribute, the dimension with the categorical values l_y will have the size of the attribute's domain. Then, the task is to calculate the length

Figure 16 – Example of answering a count query on 2-D [categorical x numerical] grid.



Source: elaborated by the author.

of the other dimension l_x. We consider that the ratio of the interval to the numerical attribute's domain size and the categorical attribute's domain size is r_x and r_y respectively. Assuming that there are $l_x r_x \cdot l_y r_y$ cells in the query. The squared noise and sampling error for OLH is Equation 4.5 and for GRR is Equation 4.6. However, in this type of grid, the number of cells in the border of the query rectangle is $2 l_y r_y$. The expected sum of frequencies of values included in these cells is $2 l_y r_y \cdot \frac{1}{l_x \times l_y} = \frac{2 r_y}{l_x}$. Then the squared error from non-uniformity is $\left(\frac{2\alpha_2 r_y}{l_x}\right)^2$. The sum of the two errors for OLH is

$$\left(\frac{2\alpha_2 r_y}{l_x}\right)^2 + \frac{4 l_x r_x l_y r_y m e^{\epsilon}}{n(e^{\epsilon} - 1)^2} \tag{4.9}$$

and for GRR is

$$\left(\frac{2\alpha_2 r_y}{l_x}\right)^2 + \frac{l_x r_x l_y r_y m(e^{\epsilon} + l_x l_y - 2)}{n(e^{\epsilon} - 1)^2} \tag{4.10}$$

Taking the partial derivative with respect to l_x, we find the value of l_{xOLH} and l_{xGRR}.

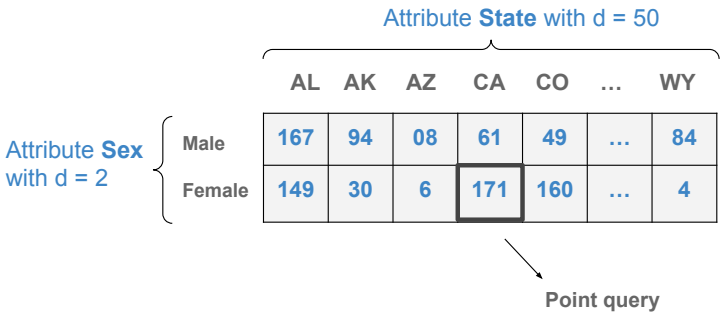
4.4.5 Categorical × Categorical 2-D Grids

The size of grids with 2 categorical attributes a_i, a_j corresponds to the size of the product of the attributes' domains $l_{x_{GRR}} = l_{x_{OLH}} = d_i$ and $l_{y_{GRR}} = l_{y_{OLH}} = d_j$.

4.5 Adaptive Frequency Oracle

After we calculate each grid size using the GRR strategy and OLH strategy (Section 4.3), we need to decide which protocol to choose. We propose selecting the protocol with the lowest variance for reporting each specific grid. In the AFO, we make use of two frequency

Figure 17 – Example of answering a count query on 2-D [categorical x categorical] grid.



Source: elaborated by the author.

oracle protocols: GRR and OLH. the variance of AFO is the following:

$$
\begin{aligned}
Var[\Phi_{AFO_{(\epsilon)}}] &= min\Big(Var[\Phi_{GRR_{(\epsilon)}}], Var[\Phi_{OLH_{(\epsilon)}}]\Big) \\
Var[\Phi_{AFO_{(\epsilon)}}] &= min\left(\frac{e^{\epsilon}+L-2}{(e^{\epsilon}-1)^2}, \frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}\right)\cdot\frac{m}{n}
\end{aligned}
\tag{4.11}
$$

4.6 Post-Processing

Answering queries using the information in the grids obtained from the adaptive frequency oracle is an estimation problem. The utility can be improved in the post-processing phase by removing negative estimations and making grids that have attributes in common consistent.

4.6.1 Removing Negative Estimations

When using the AFO, it is common to have negative estimations for many values in the domain. Moreover, the sum of all frequencies may be different from one. To minimize the error in estimations in the post-processing step, one can make every negative estimation have a value of zero and make all frequencies sum up to 1. Algorithm 1 shows how we can remove all negative estimations values and make the sum of all positive frequencies 1. The algorithm's input is the grid's estimations in the form of an estimation vector $\tilde{f}$. First, we make every negative estimation have a value of 0 (line 5). Then, we sum all remaining estimates and take the average difference by dividing by the number of positive estimates (line 6 and 7). Next, we add the difference value to every positive estimate (line 10). We repeat the whole process until all values in the output estimation vector $\tilde{f}'_v$ are non-negative, and all the frequencies sum up to 1. We call this algorithm for each estimated grid.

Algorithm 1: Algorithm for removing negative estimations

Input : Estimation vector $\tilde{f_v}$
Output Estimation vector $\tilde{f_v'}$
:

$\tilde{f_v'} \leftarrow \tilde{f_v}$;
while *any negative value in* $\tilde{f_v'}$ *or* $\sum \tilde{f_v'} > n$ **do**
  for $i \leftarrow 0$ **to** $|\tilde{f_v}|$ **do**
    if $\tilde{f_v'}[i] < 0$ **then**
      $\tilde{f_v'}$[i] $\leftarrow 0$;
    end
  end
  sum $\leftarrow \sum_{j=0}^{j=|\tilde{f_v}|} \tilde{f_v'}(j)$;
  diff $\leftarrow \frac{n-sum}{|\tilde{f_v}|}$;
  for $i \leftarrow 0$ **to** $|\tilde{f_v}|$ **do**
    if $\tilde{f_v'}[i] > 0$ **then**
      $\tilde{f_v'}$[i] $\leftarrow \tilde{f_v'}$[i] + diff;
    end
  end
end
return $\tilde{f_v'}$;

4.6.2 Consistency Step

When different grids have some attributes in common, those attributes are estimated multiple times. The utility will increase if these estimates are utilized together (ZHANG *et al.*, 2018; YANG *et al.*, 2020). We apply consistency techniques similar to (HAY *et al.*, 2010). The consistency of estimates among grids can be achieved with Algorithm 2. Each attribute a is related to w grids in total, which includes one 1-D grid and $w-1$ 2-D grids. Let $G_{(a,w)}$ be the w-th grid that is related to attribute a and assume each grid has a set of cells $L_{G_{(a,w)}}$. We define $S_{G_{(a,w)}}(i)$ to be the sum of frequencies of $G_{(a,w)}$'s cells that are in a subdomain value $D_{G_{(a,w)}}(i)$ of attribute a. For example, suppose an attribute a with domain $d = 50$ is related to a 1-D grid $G_{(a,0)}$, which has 5 cells ($|L_{G_{(a,0)}}| = 5$). $L_{G_{(a,0)}}(0)$ represents the first cell of this 1-D grid, and it defines a subdomain value $[0, 9]$ (range) or 0 (categorical value). The goal is to make all $S_{G_{(a,w)}}(j)$ consistent. we compute their weighted average as $S_{(a,i)} = \sum_{j=1}^{w} \theta_j \cdot S_{G_{(a,w)}}(i)$, where θ_j is the weight of $S_{G_{(a,w)}}(i)$. The values of the weights θ impact the variance of $S_{(a,i)}$. To minimize the variance of $S_{(a,i)}$

$$Var[S_{(a,i)}] = \sum_{j=1}^{w} \theta_j^2 \cdot Var[S_{G_{(a,w)}}(i)] = \sum_{j=1}^{w} \theta_j^2 \cdot |L_{G_{(a,w)}}(j)| \cdot Var_0$$

where Var_0 is the basic variance for estimating a single cell. $|L_{G_{(a,w)}}(j)|$ for 2D grids equals the length of grids $G_{(a,w)}$ in the axis of attribute a and for 1D grids equals the length of the corresponding 1D grid of a divided by the length of grids $G_{(a,w)}$ in the axis of attribute a. Based on the analysis in (ZHANG *et al.*, 2018), we calculate in Line 4 each grid's weight

$$\theta_j = \frac{\frac{1}{L_{G_{(a,w)}}(j)|}}{\sum_{i=j}^{w} \frac{1}{|L_{G_{(a,w)}}(j)|}}$$

Next, for each subdomain value in the grid $G_{(a,w)}$ (Line 5), we calculate (Line 8) the optimal weighted average

$$S_{(a,i)} = \frac{\sum_{i=j}^{w} \frac{1}{|L_{G_{(a,w)}}(j)|} \cdot S_{G_{(a,w)}}(i)}{\sum_{j=1}^{w} \frac{1}{|L_{G_{(a,w)}}(j)|}}$$

The next step (Line 11) is to make every sum of frequencies $S_{G_{(a,w)}}(i)$ have value $S_{(a,i)}$. To achieve that, we need to update its frequency as

$$S_{G_{(a,w)}}(i) = S_{G_{(a,w)}}(i) + \left(S_{(a,i)} - S_{G_{(a,w)}}(i)\right)/|L_{G_{(a,w)}}(i)|$$

It is essential to mention that a later consistency step does not invalidate the consistency resulting from the previous steps (QARDAJI *et al.*, 2014b). Since a consistency step may introduce negative estimations, Algorithm 1 has to be called after the consistency algorithm finishes. Removing negative estimations as the last step in post-processing.

Note that applying the consistency step may incur negativity and vice versa. Thus in the post-process, we interchangeably invoke these two steps multiple times. Since we need to ensure non-negativity for the response matrix generation (Section 4.6.3), we end the post-process with the non-negativity step. While the last step may again introduce inconsistency, it tends to be very small.

4.6.3 Response Matrix

The response matrix generation phase uses 1-D and 2-D grids to build a matrix for each pair of attributes. The resulting matrices will be used during the phase of answering queries. Figure 18 illustrate this process. On the left, we have estimated grids (A, B and AxB) that are using binning, meaning each cell spans a sub-interval of the domain. On the right, the response grid consists of cells with unit-sized granularity, which can be used to answer any query over AxB. This idea was also explored in (YANG *et al.*, 2020). However, in our case, the selection

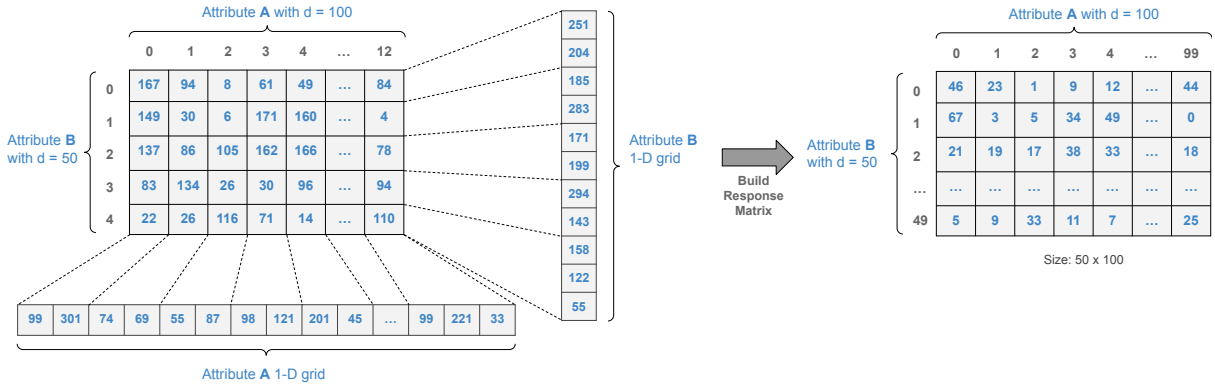
Algorithm 2: Consistency Algorithm

1 Initialize vector θ with w zeros;
2 **for** $a \leftarrow 0$ **to** $|A|$ **do**
3 **for** $j \leftarrow 0$ **to** w **do**
4 $\theta_j = \frac{\frac{1}{|L_{G_{(a,w)}}(j)|}}{\sum_{j=0}^{w} \frac{1}{|L_{G_{(a,w)}}(j)|}}$
5 **end**
6 **for** $i \leftarrow 0$ **to** $|D_{G_{(a,w)}}|$ **do**
7 $S_{(a,i)} \leftarrow 0$;
8 **for** $j \leftarrow 0$ **to** w **do**
9 $S_{(a,i)} \leftarrow S_{(a,i)} + \theta_j \cdot S_{G_{(a,w)}}(i)$
10 **end**
11 **for** $j \leftarrow 0$ **to** w **do**
12 **for** *each cell* $c \in L_{G_{(a,w)}}(j)$ **do**
13 $f_c \leftarrow f_c + S_{(a,i)}$
14 **end**
15 **end**
16 **end**
17 **end**

of the set of related grids Γ is different since we have grids with categorical dimensions. Each pair of attributes (a_i, a_j) has a corresponding response matrix $M(i, j)$ of size $d_i \times d_j$, where the element $M_{(i,j)}[x, y]$ represents the estimated frequency of 2-D value (x, y) in the $[d_i] \times [d_j]$ 2-D domain of the attribute pair (a_i, a_j). If a_i and a_j are categorical attributes, the estimated grid $G(i, j)$ is already the response matrix $M(i, j)$. Otherwise, we need to build $M_{(i,j)}$ by invoking an efficient estimation method called Weighted Update (ARORA *et al.*, 2012; HARDT *et al.*, 2012). If a_i and a_j are numerical, we build $M(i, j)$ from its related grids $\Gamma = \{G(i), G(j), G(i, j)\}$ which represent the grids of $\{a_i, a_j, (a_i, a_j)\}$, respectively. Note that, If one of the attributes is categorical, say a_i, we build $M(i, j)$ using only the grids $\{G(j), G(i, j)\}$, which corresponds to $\Gamma = \{a_j, (a_i, a_j)\}$, respectively.

Algorithm 7 shows the step-by-step of building a response matrix. First, select the set, denoted by $\delta(c)$, of all 2-D (x, y) values that contribute to the frequency f_c of cell c from $G \in \Gamma$. For example, suppose that the pair of numerical attributes (a_i, a_j) have domains $d_i = 100, d_j = 50$. Its corresponding grid $G(i, j)$ has size 10×5 cells. Each cell c defines a partition $[l_i, r_i]$ of d_i and $[l_j, r_j]$ of d_j. The first cell, in the position [0, 0], from $G(i, j)$, for instance, may define the subdomain $[0, 9] \times [0, 9]$. Then, in line 4, we would collect all 2-D values in the subdomain $[0, 9] \times [0, 9]$ *i.e.,* values that contribute to the frequency of c. Once we have all the values $\delta(c)$, the elements in the response matrix are updated (Lines 8-10). Algorithm

Figure 18 – Example of the construction of response matrices by combining 2-D grids with their respective 1-Ds grids. On the left, grids are using binning *i.e.* cells cover a sub-interval of the domain. On the right, we see a response grid which has unit sized cells.



Source: elaborated by the author.

Algorithm 3: Building Response Matrix

Input : Set of related grids Γ, domain sizes d_i, d_j
Output : Response matrix $M_{(i,j)}$

Initialize all $d_i \times d_j$ elements in matrix $M_{(i,j)}$ as $\frac{1}{d_i \cdot d_j}$ **while** *not convergence* **do**
  for *each grid* $G \in \Gamma$ **do**
    for *each cell* $c \in G$ **do**
      Find the set of 2-D values $\delta(c)$ from c;
      $S = 0$;
      for *each 2-D value* $(x, y) \in \delta(c)$ **do**
        $S = S + M_{(i,j)}[x, y]$;
      end
      if $S \neq 0$ **then**
        for *each 2-D value* $(x, y) \in \delta(c)$ **do**
          $M_{(i,j)}[x, y] \leftarrow \frac{M_{(i,j)}[x,y]}{S} \cdot f_c$;
        end
      end
    end
  end
end
return $M_{(i,j)}$;

7 is set to converge when the sum of the changes of all elements in $M_{(i,j)}$ after each update process is smaller than a certain threshold. It is shown in (YANG *et al.*, 2020) that the threshold should be smaller than $\frac{1}{n}$.

4.6.4 λ-D Query Estimation

To estimate the answer of a high dimensional query we leverage the idea of answering low dimensional 2-D queries and use those values to estimate the answer of λ-D query. This idea has been successfully employed in different works (QARDAJI *et al.*, 2014b; ZHANG *et al.*, 2018; YANG *et al.*, 2020; DU *et al.*, 2021). To estimate the answer f_q of a $\lambda - D$ of a query

$$q = (a_{t_1}, o_{t_1}, v_{t_1}) \wedge (a_{t_2}, o_{t_2}, v_{t_2}) \wedge ... \wedge (a_{t_\lambda}, o_{t_\lambda}, v_{t_\lambda})$$

where $o_{t_i} \in \{=, in, between\}$ specifies an operator, v_{t_i} is either a set of size $0 < |v_{t_i}| \leq d_{t_1}$ of categorical values or a range $[l_{t_i}, r_{t_i}]$. $A_q = \{a_{t_\psi} | 1 \leq \psi \leq \lambda\}$. The first step is splitting q into $\binom{\lambda}{2}$ associated 2-D queries

$$\left\{q_{(i,j)} = (a_i, o_i, v_i) \wedge (a_j, o_j, v_j) | a_i, a_j \in A_q\right\}$$

and then get their answers $\left\{f_q^{i,j} | a_i, a_j \in A_q\right\}$. Next, one should use the $\binom{\lambda}{2}$ corresponding 2-D queries' answers to estimate f_q. Algorithm 8 shows in detail how estimating the answer of a λ-D query q works. The input of the algorithm is the answers of $\binom{\lambda}{2}$ associated 2-D queries. As output, it returns a estimated answer vector z. In particular, the vector z consists of 2^λ elements that are in one-to-one correspondence with the answers of $2^\lambda \lambda$-D queries in $Q(q) = \wedge_t (a_t, o_t, v_t) \vee (a_t, o_t, v_t') | a_t \in A_q$, where v_t' is the complement of v_t on the domain of a_t. In Algorithm 8, for each $f_q^{(i,j)} \in \left\{f_q^{(i,j)} | a_i, a_j \in A_q\right\}$, the aggregator performs the following update process on z. The aggregator first finds the set of $\lambda - D$ queries $Q(q)^{(i,j)}$ corresponding to the 2-D query $q^{(i,j)}$, which means that $Q(q)^{(i,j)}$ consists of all those $\lambda - D$ queries whose answers can contribute to $f_q^{(i,j)}$. In particular, $Q(q)^{(i,j)}$ contains $2^{\lambda-2}$ λ-D queries from $Q(q)^{(i,j)}$ and is defined as $\left\{\wedge_t (a_t, o_t, v_t) \vee (a_t, o_t, v_t') | a_t \in A_q / \{a_i, a_j\}\right\}$. Then, the aggregator calculates the sum Y of $z[q']$ for all $q' \in Q(q)^{(i,j)}$, where $z[q']$ is the element corresponding to the answer of q'. Next, the aggregator uses $f_q^{(i,j)}$ to update the elements in z as Lines 6-8. This process is repeated until convergence. The estimated answer f_q of the λ-D query q equals its corresponding element in z, *i.e.,* $z[q]$. Algorithm 8 is set to converge when the sum of the changes of all elements in $z[q]$ after each update process is smaller than a certain threshold. It is shown in (YANG *et al.*, 2020) that the threshold should be smaller than $\frac{1}{n}$.

Algorithm 4: Estimating Answer of $\lambda - D$ Query

Input : Associated 2-D queries' answers $\left\{f_{q^{(i,j)}} | a_i, a_j \in A_q\right\}$
Output Estimated answer vector **z**
:

Initialize all 2^λ elements in the vector z as $\frac{1}{2^\lambda}$;
while *not convergence* **do**
  for *each* $f_{q^{(i,j)}} \in \left\{f_{q^{(i,j)}} | a_i, a_j \in A_q\right\}$ **do**
    Find the set of queries $Q(q)^{(i,j)}$ corresponding to $q^{(i,j)}$;
    Calculate $Y = \sum_{q' \in Q(q)^{(i,j)}} z[q']$;
    if $Y \neq 0$ **then**
      for *each query* $q' \in Q(q)^{(i,j)}$ **do**
        $z[q'] \leftarrow \frac{z[q']}{Y} \cdot f_{q^{(i,j)}}$;
      end
    end
  end
end
return z;

4.7 Privacy and Error Analysis

FELIP satisfies ϵ-LDP because all the information collected from each user goes through either OLH or GRR with privacy budget ϵ.

The error analysis has to consider all sources of error. They are the following: noise error, sampling error, non-uniformity error, and estimation error.

Noise error. Due to the use of LDP frequency oracles, we have the noise error since, to satisfy ϵ-LDP, an error is added to each cell. These noises are independently generated noise and have the same standard deviation. When summing the noisy frequency of cells to answer a query, the noise error is the sum of the corresponding noises. To generate these independent noises, zero-mean random variables are used; thus, the noises cancel each other out to a certain degree. We know that, for independent random variables X and Y, the variance of their sum is the sum of their variances. Hence, the more partitioned the domain, the more cells are included in a query, and the larger the noise error is.

Sampling Error. In our strategy, we select a fraction n/m of the total number of users to answer the frequency of each cell. As the fraction of users may have a different distribution from the global (*i.e.,* the group of all users), this error must be considered. Let $\bar{f}_v$ represent the true frequency, and f_v is the estimated one. D_n is a fraction of dataset D and m is the number of

groups that the total n users are divided. The expected squared error for estimating one value is

$$\begin{aligned}\mathbb{E}\left[\left(f_v(D_n)-\bar{f}_v\right)^2\right] &= \mathbb{E}\left[\left(\left(f_v(D_n)-\bar{f}_v(D_n)\right)+\left(\bar{f}_v(D_n)-\bar{f}_v\right)\right)^2\right] \\ &= \mathbb{E}\left[\left(f_v(D_n)-\bar{f}_v(D_n)\right)^2\right]+\mathbb{E}\left[\left(\bar{f}_v(D_n)-\bar{f}_v\right)^2\right] \\ &+ 2\mathbb{E}\left[\left(f_v(D_n)-\bar{f}_v(D_n)\right)\cdot\left(\bar{f}_v(D_n)-\bar{f}_v\right)\right] \end{aligned} \tag{4.12}$$

The expected squared noise and sampling error is basically the first part of Equation 4.12 since the third part of the equation equals zero and the second part is a constant whose value is much smaller when compared to the first part. In OLH, $p = 1/2$ and $p' = 1/(e^\epsilon + 1)$. Thus, the noise and sampling error is expressed as

$$\mathbb{E}\left[\left(f_v(D_n)-\bar{f}_v(D_n)\right)^2\right] = \frac{4me^\epsilon}{n(e^\epsilon-1)^2}+\frac{m}{n}\cdot\bar{f}_v$$

and GRR with $p = e^\epsilon/e^\epsilon + d - 1$ and $p' = 1/(e^\epsilon + d - 1)$, we have

$$\mathbb{E}\left[\left(f_v(D_n)-\bar{f}_v(D_n)\right)^2\right] = \frac{m(e^\epsilon+L-2)}{n(e^\epsilon-1)^2}+\frac{m}{n}\cdot\bar{f}_v$$

Ignoring the small factor $\frac{m}{n}\cdot\bar{f}_v$, for the OLH protocol, the error is dominated by

$$\mathbb{E}\left[f_v\right] = \frac{4me^\epsilon}{n(e^\epsilon-1)^2} \tag{4.13}$$

and for the GRR protocol by

$$\mathbb{E}\left[f_v\right] = \frac{m(e^\epsilon+L-2)}{n(e^\epsilon-1)^2} \tag{4.14}$$

Non-Uniformity Error. This error occurs when we have numerical attributes in the grid. In numerical dimensions, we may have the query rectangle intersecting cells. Since we do not have access to the actual data distribution inside the cell, we have to compute the approximate non-uniformity error. More details are in Section 4.3).

Estimation Error. When estimating the answer of a λ-dimensional query where $\lambda > 2$ from the associated answers of 2-D range queries, estimation error will occur. Since the estimation error is dataset dependent, there is an exact way of estimating it (YANG *et al.*, 2020).

4.8 Discussion

FELIP and TDG/HDG (YANG *et al.*, 2020) propose the use of grids to answer multi-dimensional queries. However, there are several differences between the two, and they are summarized as follows:

- Unlike TDG/HDG, the focus of FELIP is to answer queries with point and range constraints.
- TDG/HDG is designed around OLH protocol. FELIP analyzes GRR and OLH. Moreover, it proposes a framework to adaptively choose the protocol that offers the best utility on a per-grid strategy.
- Unlike TDG/HDG, FELIP considers attributes with different domain sizes. Moreover, in TDG/HDG grids' dimensions are not optimal since they have to be divisible by domain size (detailed explanation in Section 3.6). We overcome that limitation by enabling cells to have different sizes within a grid.
- In TDG/HDG, all 1-D grids share the same size g_1. Similarly, 2-D grids all have the same size g_2. FELIP calculates the size of each grid individually while considering each grid's unique aspects.
- In TDG/HDG, the grids assume that the query selectivity is always 50%. In FELIP, the aggregator can utilize the knowledge it has about queries selectivity during the construction of grids.

5 PRIVMDC: LEVERAGING MULTI-DIMENSIONAL CORRELATIONS TO ANSWER DIFFERENTIALLY PRIVATE RANGE QUERIES

In this chapter, we tackle the problem of answering an unbounded number of range queries, where the population is composed of users that have different privacy requirements. Additionally, we also investigate how to incorporate *a priori* knowledge about the workload to improve utility. We introduce and formalize the PrivMDC framework in Section 5.2, where the core components of the PrivMDC framework are detailed, including data partitioning, grid construction, and estimation techniques tailored to hybrid privacy requirements. Section 5.3 focuses on post-processing techniques applied to the privatized data to enforce consistency and improve utility. The full end-to-end algorithm is presented in Section 5.4. Section 5.5 specifies the threat model and the trust assumptions under which PrivMDC operates. Finally, Section 5.6 provides formal privacy guarantees.

5.1 Overview and Motivation

We present PrivMDC, a scalable differentially private approach to answer multi-dimensional range queries. PrivMDC introduces a new framework for this task in scenarios where users have different privacy expectations. Specifically, one group, the DP group, of u_{dp} users trusts the curator with their true data, while another group of u_{ldp} users, the LDP group, is more privacy-conscious and prefers to share a noisy version of their data. We focus on the natural case where $u_{dp} \ll u_{ldp}$ (KOHEN; SHEFFET, 2022; BEIMEL *et al.*, 2020; AVENT *et al.*, 2017; AVENT *et al.*, 2020). Our work tackles the scalability limitation of previous techniques (YANG *et al.*, 2020; WANG *et al.*, 2019a; WANG *et al.*, 2023) by building a Bayesian network that models the correlations. Our work allows the aggregator to understand which attributes are correlated or not. Therefore, we avoid the collection of all possible combinations of attributes avoiding the curse of dimensionality.

As the number of attributes increases, it becomes increasingly challenging to maintain meaningful utility using traditional techniques. In many cases, these prior methods fail to deliver sufficient accuracy or insight, rendering them impractical for large-scale or high-dimensional data analysis tasks. To illustrate the scalability challenge, consider the Lending Club Loan dataset (KAGGLE, 2019), which contains 151 attributes. If every pairwise combination of attributes were to be materialized as a 2-dimensional grid, as in our baselines (YANG *et al.*, 2020; WANG *et al.*, 2019a; WANG *et al.*, 2023), this would result in a total of 11,325 such grids. Managing

and processing such a large number of grids is computationally expensive and difficult to scale, especially when operating under the constraints of LDP. This exponential growth in complexity makes scalability a pressing concern when applying LDP algorithms to high-dimensional data.

To selectively collect the right groups of attributes, our work also introduces higher k-dimensional grids that group k strongly correlated attributes. For instance, if our correlation model finds that the four attributes [*num_op_rev_tl, num_bc_sats, open_acc, num_rev_accts*], as shown in Figure 1, are strongly correlated, we build a 4-D grid in order to preserve those correlations. We further enhance the accuracy of k-dimensional grids by combining them with fined-grained 1-D grids. By explicitly capturing high-dimensional correlations, our method significantly decreases the number of grids that must be materialized. As a result, each grid accumulates reports from a greater number of users, thereby enhancing utility. This improvement becomes more pronounced as the dimensionality of the dataset increases. Also, we propose to incorporate *a priori* knowledge about the workload and minimize the total error incurred when answering queries by finding an optimized user population partitioning. This is achieved by carefully analyzing each grid configuration and assessing the various types of errors that may arise, including noise, sampling and non-uniformity error. By considering these sources of error, our method seeks to optimize population partitioning which preserves both privacy and utility across various datasets and workloads.

In this context, we identify real-world use cases in which our proposed technique can be effectively deployed to enhance utility across different data analysis tasks. The first use case involves private multi-dimensional data analysis (FILHO; MACHADO, 2023; YANG *et al.*, 2020; WANG *et al.*, 2019a). It is not uncommon for such beta testers to trust the central server. Many companies rely on a group of beta testers with whom they have higher levels of mutual trust (MICROSOFT, 2017; AVENT *et al.*, 2020; KHALAF *et al.*, 2024; KOHEN; SHEFFET, 2022; LI *et al.*, 2022a). For instance, Mozilla and Google inform beta testers of Firefox and Chromium, respectively, that telemetry data is automatically collected during testing phases (AVENT *et al.*, 2017). In such scenarios, data from trusted beta testers can be used to model attribute correlations under DP, which subsequently guide the collection of data from the broader user population, under LDP, using production (release) software. The second use case addresses Approximate Query Processing (AQP) in private federated settings over horizontally partitioned data. Here, multiple data providers collaborate to analyze distributed data (LAOUIR; IMINE, 2025; KHALAF *et al.*, 2024), with some providers being trusted by users. These trusted entities

can model correlations under central DP using a trusted data subset, and the resulting models can be used to optimize data collection from a larger population under LDP. Additionally, prior knowledge of the query workload is often leveraged to further improve utility in AQP (LAOUIR; IMINE, 2025; LI *et al.*, 2014a), a strategy also incorporated in our framework.

5.2 The framework of PrivMDC

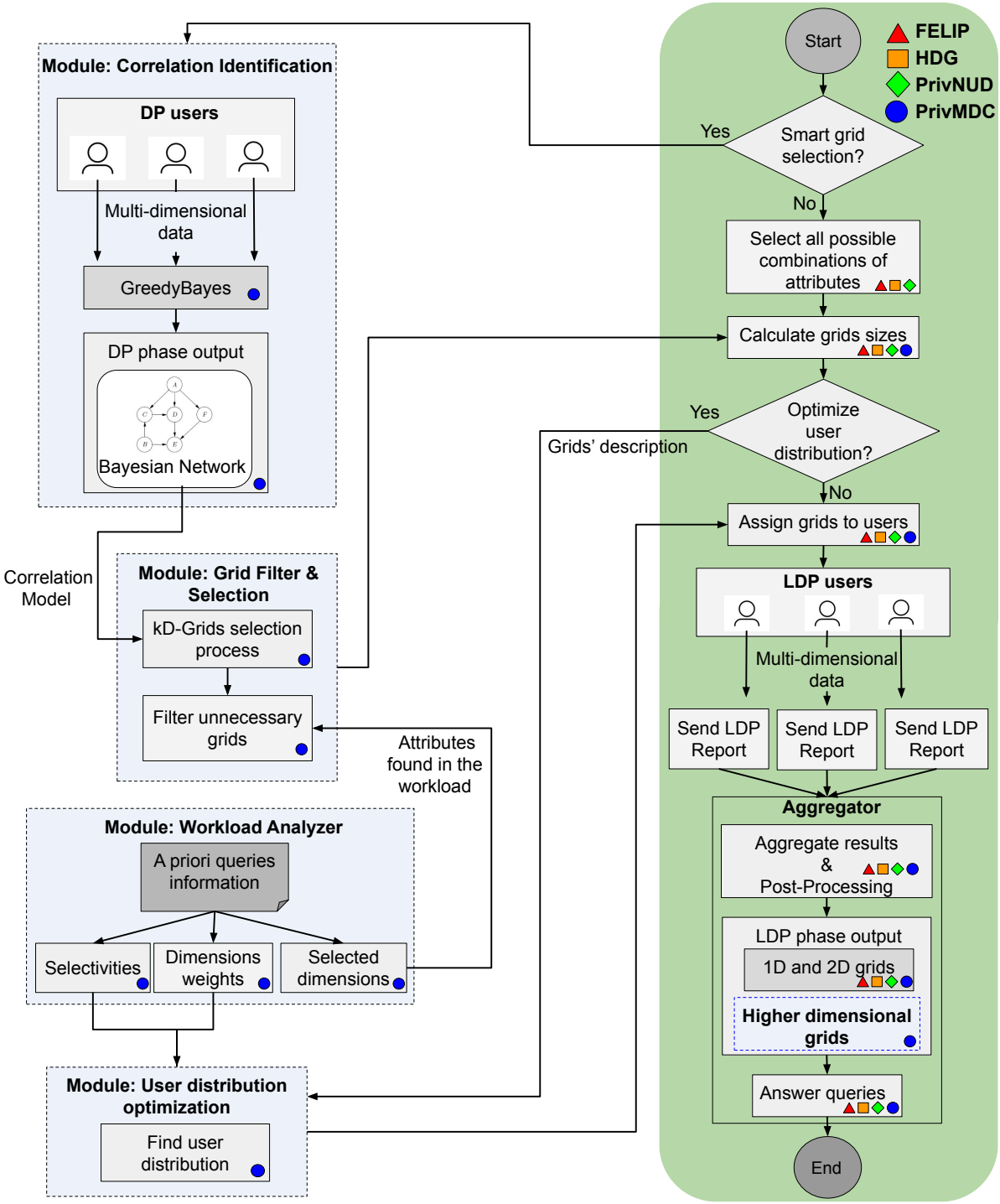
PrivMDC is designed to accommodate users with different privacy expectations. This type of heterogeneous setting has also been explored in prior research (KOHEN; SHEFFET, 2022; AVENT *et al.*, 2017; AVENT *et al.*, 2020; BEIMEL *et al.*, 2020; KHALAF *et al.*, 2024; ACQUISTI; GROSSKLAGS, 2005; ACQUISTI *et al.*, 2015). More specifically, one group, referred to as the DP group, is comfortable sharing their true data with the curator, while another group, known as the LDP group, prioritizes privacy and opts to share a modified, noisy version of their data. The step-by-step process of PrivMDC is shown in Figure 19. The proposed framework begins by constructing a Bayesian network from users in the DP group, ensuring ϵ-DP, to capture data correlations and guide the construction of k-dimensional grids. Utilizing prior knowledge of the analysts' query workload, the framework extracts workload parameters. Subsequently, it defines specific grids and their sizes based on the identified correlations and workload parameters. To improve accuracy for frequently queried grids, users in the LDP group are partitioned into disjoint sets, each assigned to one grid. Users submit ϵ-LDP reports to the aggregator, which estimates cell frequencies and performs post-processing to answer all queries. Below, we describe each step of the proposed framework in detail.

5.2.1 Correlation Identification

The correlation model enables us to collect only the group of strongly correlated attributes, reducing the amount of data collected and enhancing efficiency in the subsequent stages of the framework as exemplified in Figure 20. We propose to build to construct a k-degree Bayesian network $\mathcal{N}$ over the attributes A, using an ϵ-DP method. More precisely, we employ the *GreedyBayes* (ZHANG *et al.*, 2017) technique with privacy budget ϵ to construct $\mathcal{N}$.

Algorithm 1 shows each step of this process. Specifically, we first calculate the mutual information of each pair $[a, p] \in \Omega$ (Lines 6-8). After that, we sample an pair from Ω, such that the sampling probability of any pair $[a, p]$ is proportional to $exp(I(a, p)/2\Delta)$, where Δ

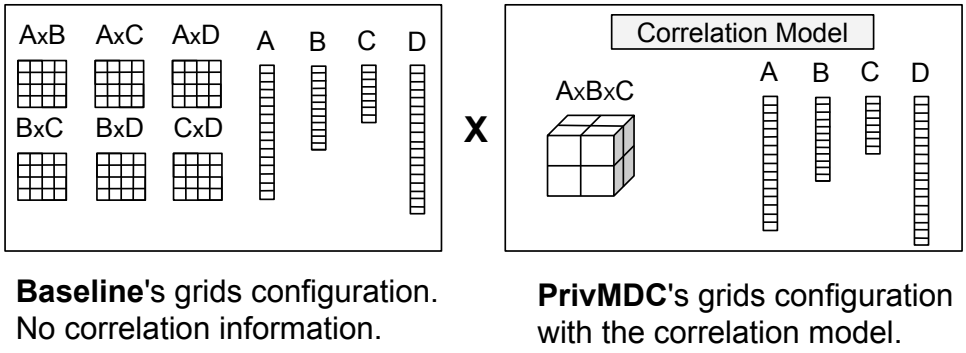
Figure 19 – The complete framework of PrivMDC. The basic steps to answer multi-dimensional range queries in the literature are shown in green. The boxes in blue represent new steps proposed by this work.



Source: elaborated by the author.

is a scaling factor. We set $\Delta = (|A| - 1)S(I)/\epsilon$, where $S(I)$ represents the sensitivity of $I(a, p)$.

Figure 20 – PrivMDC constructs a correlation model by leveraging GreedyBayes and k-dimensional grids are configured following the model specifications. High-dimensional grids are constructed for strong correlated attributes and 1-D grids are constructed for each attribute, which includes the independent ones.



Source: elaborated by the author.

This is calculated as follows:

$$S(I(a,p)) = \begin{cases} \frac{1}{n}log(n) + \frac{n-1}{n}log\left(\frac{n}{n-1}\right), & \text{if } a \text{ or } p \text{ is binary;} \\ \frac{2}{n}log(\frac{n+1}{2}) + \frac{n-1}{n}log\left(\frac{n+1}{n-1}\right), & \text{otherwise,} \end{cases}$$

where n is the number of users in the DP group. Finally, the sampled pair $[a_i, p_i]$ is added to the network $\mathcal{N}$ (Line 10).

It is important to mention that the group of users participating in this step and those composing the LDP group dataset are disjoint. It is to be observed that the data distribution in the DP group may differ from the LDP group. The larger the difference, the less valuable the correlation model will be when utilized to enhance the data collection under LDP. This aspect is experimentally explored in Section 7.2 where we use sample users in a manner that results in distributions that are increasingly different (measured by the Kullback-Leibler (KL) divergence (PEREZ-CRUZ, 2008)). Also, we experiment with a scenario where all users are LDP users. As discussed later, our experimental results suggest our approach is effective up to a moderate amount of divergence.

5.2.2 Workload Analyzer

It is common to know a priori some aspects of the type of queries the analysts are interested in answering. Various techniques have been proposed (LI *et al.*, 2010; LI; MIKLAU, 2012; LI *et al.*, 2014b; LI *et al.*, 2020; YUAN *et al.*, 2012; YUAN *et al.*, 2015; MCKENNA *et al.*, 2020; MCKENNA *et al.*, 2018) but not with the goal of answering an unbounded number of queries. They focused on optimizing the utility for a specific set of queries. In our case, we focus

Algorithm 5: Build Correlation Model

Input : DP Group dataset D_{dp}, Network's Degree k, Privacy Budget ϵ, Set of Attributes A

Output Bayesian network $\mathcal{N}$

:

1 initialize $\mathcal{N} = \emptyset$ and $V = \emptyset$;
2 randomly select an attribute a_1 from A;
3 add $[a_1, \emptyset]$ to $\mathcal{N}$; add a_1 to V;
4 **for** $i \leftarrow 2$ **to** $|A|$ **do**
5 initialize $\Omega = \emptyset$;
6 **for** *each* $a \in A \setminus V$ **do**
7 **for** *each* $p \in \binom{V}{k}$ **do**
8 add $[[a, p], I(a, p)]$ to Ω;
9 **end**
10 **end**
11 select a key pair $[a_i, p_i]$ from Ω, using the exponential mechanism with privacy budget ϵ;
12 add $[a_i, p_i]$ to $\mathcal{N}$; add a_i to V;
13 **end**
14 **return** $\mathcal{N}$;

on incorporating broader workload characteristics and designing a strategy that can be used to answer an unbounded number of range queries. Data dimensions that are frequently queried by analysts or systems are typically critical to generate insights that directly influence strategic decisions. Identifying these dimensions usually involves both quantitative analysis and domain-specific knowledge (GARFINKEL *et al.*, 2021; STEINKE; ULLMAN, 2014; CUNNINGHAM *et al.*, 2021b; LI *et al.*, 2016). PrivMDC enables reducing the noise level for these high-utility dimensions, so analysts can achieve more accurate aggregate insights.

5.2.3 Grid Filter & Selection

With the correlation model and the information from the Workload Analyzer, the Grid Filter & Selection module is responsible for defining which grids should be materialized. It requires no privacy budget consumption since it does not touch users' data. Algorithm 6 details the complete process. Let $\mathcal{N}_b = ([a_0, P_0], [a_1, P_1], ..., [a_k, P_k])$ be a b-degree Bayesian network composed of a set of pairs of attribute a_i and corresponding parent set P_i. For each $[a_i, P_i]$, we extract all $e_i = \{[a_i, p_j] | \forall p_j \in P_i\}$, with $0 \leq |P_i| \leq b$, to form set of all connections E (line 3). Next, we compute all cliques in E and add them as a grid to the set of grids $\mathcal{G}$. Each clique becomes a grid in line 5. Then, for each attribute in the dataset, we add a 1-D

Algorithm 6: Grid Filter & Selection

Input : Dimension' weights W and Bayesian network $\mathcal{N}$, Attributes in the dataset A
Output Grids $\mathcal{G}$
:

1 **for** $i \leftarrow 0$ **to** $|\mathcal{N}| - 1$ **do**
2 $[a, P] \leftarrow \mathcal{N}_i$;
3 **for** $j \leftarrow 0$ **to** $|P| - 1$ **do** add $[a, P_j]$ to E;
4 **end**
5 C = Compute cliques in E;
6 **for** $i \leftarrow 0$ **to** $|C| - 1$ **do** add C_i to $\mathcal{G}$;
7 **for** $i \leftarrow 0$ **to** $|A| - 1$ **do** add A_i to $\mathcal{G}$;
8 **for** $i \leftarrow 0$ **to** $|\mathcal{G}| - 1$ **do**
9 **for** $j \leftarrow 0$ **to** $|\mathcal{G}_i| - 1$ **do**
10 **if** $W[\mathcal{G}_{i,j}] == 0$ **then** remove $\mathcal{G}_{i,j}$ from $\mathcal{G}_i$;
11 **end**
12 **end**
13 **return** $\mathcal{G}$;

grid (line 6) which may be used to answer queries directly and are used in combination with multi-dimensional grids to construct a response matrix later (Algorithm 7). Finally, in line 9, we remove unnecessary dimensions from each grid that are not utilized and return the set of grids $\mathcal{G}$ that should be materialized. In the case where we do not have a correlation model (*i.e.*, the entire population is in the LDP group), PrivMDC offers limited advantages in low-dimensional datasets, as it tends to materialize a similar number of grids as baseline methods. In such cases, baselines with more flexible one-dimensional decompositions may perform better. This observation is supported by Figure 35, which shows no clear leader in utility when the number of attributes is small.

5.2.4 Grid size calculation

Once the set of grids to be materialized $\mathcal{G}$ is determined, the aggregator computes the size of each grid without accessing any user data. Cells are grouped into bins to facilitate efficient range query processing through a minimal number of sub-intervals, which precisely cover the query range and thereby reduce the amount of noise added to the final result. The number of cells per grid is chosen based on the privacy budget ϵ, balancing the trade-off between noise and bias: too many cells introduce excessive noise, while too few result in higher bias. To optimize grid size, the total expected error, which includes the sum of the noise error, sampling error, and non-uniformity error is approximated and minimized. The higher the privacy budget ϵ, the

more cells a grid has. This notion is also explored in (YANG *et al.*, 2020; FILHO; MACHADO, 2023; WANG *et al.*, 2023). We use size guidelines and derivations from FELIP (Chapter 4) to determine the size of 1-D and 2-D grids. The number of cells in 1-D grids is determined as

$$l_x = \sqrt[3]{\frac{n\alpha_1^2 \cdot (e^\epsilon - 1)^2}{2|\mathcal{G}|r_x e^\epsilon}}$$

The parameter α works as a hyperparameter to estimate how uniform data is distributed within a grid cell. r_i represents the query selectivity along the ith axis of the grid. The size of 2-D grids are calculated by taking the partial derivative of the following expression with respect to l_x and l_y, which represent the number of cells in each axis

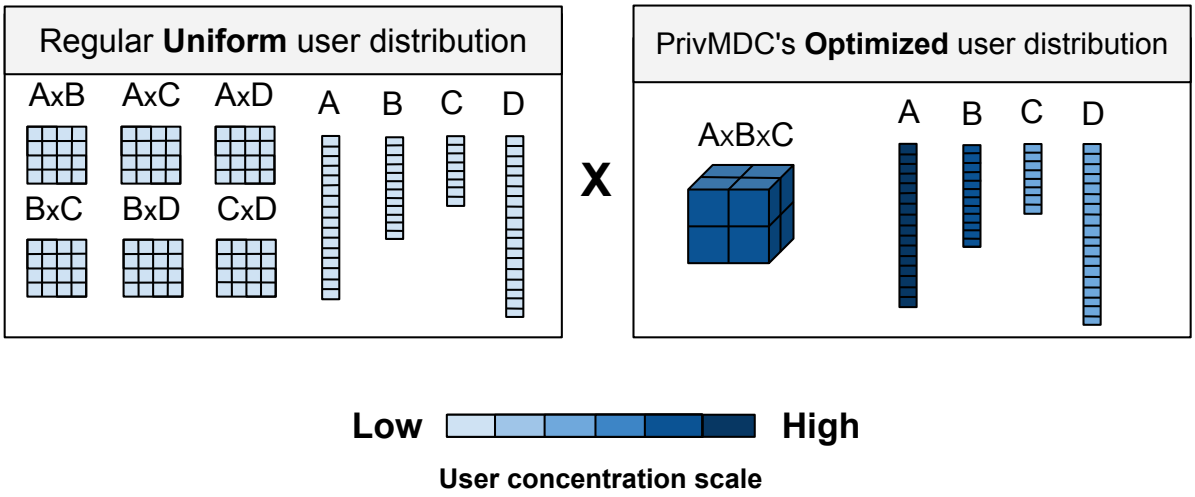
$$\left(\frac{2\alpha_2(l_x r_x + l_y r_y)}{l_x l_y}\right)^2 + \frac{4 l_x r_x l_y r_y |\mathcal{G}| e^\epsilon}{n(e^\epsilon - 1)^2}$$

For $k \geq 3$ grids, we make the total number of cells be close to the number of cells of 1-D grids which may include having dimensions with fewer cells than others. The intuition is that we want to keep the same noise/signal ratio per grid cell. For example, if 1-D grids with $\epsilon = 2$ have 64 cells, 3-D grids would be formatted as $4 \times 4 \times 4$, 4-D grids would be formatted as $3 \times 3 \times 3 \times 2$, and so on. Also, the length of a dimension is limited by the size of the domain. If the domain of an attribute is not divisible by the length of its respective dimension, we allow sub-intervals to have different sizes. Our approach uses a non-data dependent approach to select grid sizes for consistency with previous research on range queries (*e.g.*, HIO (WANG *et al.*, 2019a), HDG (YANG *et al.*, 2020), PrivNUD (WANG *et al.*, 2023), FELIP (Chapter 4), all of which are compared in the evaluation Section 7.2).

5.2.5 User distribution optimization

Prior work employing 1-D and 2-D grids (YANG *et al.*, 2020; WANG *et al.*, 2023) and FELIP (Chapter 4) typically assigns users to disjoint groups of equal size, resulting in uniform utility across all grids. However, this uniform allocation may be suboptimal, as some dimensions and grids are more frequently queried and thus more important for analysis. To enhance overall utility, it is beneficial to allocate more users to grids with higher query frequency. To this end, we introduce the User Distribution Optimization module, which determines the user-to-grid allocation that minimizes the total estimation variance, as illustrated in Figure 21. This phase takes as input the weights W and average selectivities S of dimensions obtained from the Workload Analyzer, along with the configuration of the grids to be materialized.

Figure 21 – The optimized user distribution strategy proposed by PrivMDC yields two key effects: it results in a higher concentration of users per grid compared to baseline methods, and it ensures that grids queried more frequently receive a greater number of user reports.



Source: elaborated by the author.

To do this optimization step, we need to calculate the error when answering queries utilizing the OLH and GRR. Let the average query selectivity of dimension i be denoted by s_i, and let l_i represent the length of dimension i within a grid. When answering a query, we deal with four kinds of errors: noise error, sampling error, non-uniformity error and estimation error. Under the LDP constraint, independently generated noise with equal standard deviation is added to each grid cell. Query answering involves summing the noisy frequencies of the cells within the query region, and the total noise error corresponds to the variance of these summed noises. Because the noise is generated independently with zero mean, the total variance is additive. When we use cells' frequency estimations from a grid that was reported by a subset of users, we need to take into consideration that the data distribution of that subset of users may not follow the distribution of the total set of users, hence, we have the sampling error. Noise and sampling errors are quantified together and represent the first term of Equations 5.1 and 5.2.

Non-uniformity error arises when query rectangles partially intersect cells without fully containing them. To estimate the number of data points in these intersected cells, it is commonly assumed that the data is uniformly distributed. However, deviations from uniformity introduce non-uniformity error. The magnitude of this error in a cell is bounded by the number of data points it contains, and it generally decreases with finer grid granularity. In the local setting, since the true intra-cell distribution is unknown, the non-uniformity error is estimated by counting the number of intersected cells as shown in the second term of Equations 5.1 and 5.2.

The parameter α quantifies the degree of uniformity in the data distribution within a grid cell. As the true distribution is inaccessible, α is empirically determined using synthetic datasets. The sum of errors for OLH is

$$Var_g = \prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{4e^{\epsilon}}{n_i(e^{\epsilon}-1)^2} + \left(\frac{2\alpha\left(\sum_{i=1}^{\lambda} l_i s_i\right)}{\prod_{i=1}^{\lambda} l_i}\right)^2 \tag{5.1}$$

and for GRR is

$$Var_g = \prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{e^{\epsilon} + \left(\prod_{i=1}^{\lambda} l_i\right) - 2}{n_i(e^{\epsilon}-1)^2} + \left(\frac{2\alpha\left(\sum_{i=1}^{\lambda} l_i s_i\right)}{\prod_{i=1}^{\lambda} l_i}\right)^2 \tag{5.2}$$

We choose the best FO protocol for each individual grid based on the number of cells in it. For a set of G grids that we want to materialize, we find the number of users per grid n_i, with the sum of all n_i being at most the total number of users N in the dataset D, that minimizes the total error when answering all queries. Each grid has a weight w_i that is computed based on the information from the Workload Analyzer module (Section 5.2.2). All grids' weights are further normalized. Formally, we have a multi-variable optimization problem as follows:

$$\min_{n_i} \quad \sum_{i=1}^{G} {w_i}^2 \cdot Var_{g_i} \quad \text{s.t.} \quad \sum_{i=1}^{G} n_i = N \tag{5.3}$$

In practice, we use a Differential Evolution (DE) (STORN; PRICE, 1997) technique to solve this optimization problem. It is important to mention that this optimization step done by the aggregator does not touch users' data thus it does not consume privacy budget.

5.2.6 Adaptive Frequency Oracle (AFO)

Given the number of cells L in each grid, we decide which frequency oracle protocol is most suited to estimate them. We choose to select the LDP protocol with the lowest variance for reporting each specific grid. We employ the OLH and GRR frequency oracles protocols which variances derivations are found in (WANG *et al.*, 2017). Typically, grids with a small number of cells are reported using GRR, and for the ones with a larger number of cells, the OLH protocol will be used. The number of users allocated for reporting a grid is n, and the variance of AFO is

$$Var[\Phi_{AFO_{(\epsilon)}}] = min\left(\frac{e^{\epsilon}+L-2}{(e^{\epsilon}-1)^2}, \frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}\right) \cdot \frac{1}{n}$$

Each user perturbs their respective grid answer by introducing LDP noise using GRR or OLH algorithm with privacy budget ϵ. Then, they send their LDP reports to the aggregator.

5.3 Aggregation & Post-processing

Once the aggregator have received all LDP reports, frequency estimations are calculated for every cell in all $\mathcal{G}$ grids. From this point forward, any post-processing can be done to improve utility without consumption of privacy budget (DWORK *et al.*, 2006). Answering queries using the information in the grids obtained from a frequency oracle is an estimation problem.

5.3.1 Removing Negative Estimations

When using the FOs, OLH and GRR, it is common to encounter negative estimations for numerous values within the given domain. Additionally, the sum of frequencies of all values in the domain may not be equal to 1. Utility can be improved if we adjust those estimations (WANG *et al.*, 2019b). More precisely, given a vector $\tilde{f}$, which stores the estimations of cells within a grid, we find δ such that $\sum_{v \in D} max(\tilde{f}_v + \delta, 0) = 1$. Then the estimation for each value v is $f'_v = max(\tilde{f}_v + \delta, 0)$. While there is any negative values in $\tilde{f}$, we make negative estimations equal to 0 and correct the values of the positive estimates by adding the average difference. These steps are executed for every materialized grid.

5.3.2 Consistency among Grids

When different grids have some attributes in common, those attributes are estimated multiple times. As seen in (ZHANG *et al.*, 2018), the utility will increase if these estimates are utilized together. Each attribute a appears in b K-D grids in total. Let g_b^a be the z-th grid that is related to attribute a and $L_{g_b^a}$ be the number of cells in grid g_b^a. We define $S_{g_b^a}(i)$ to be the sum of estimates of g_b^a's cells that are in a subdomain value $I_{g_b^a}(i)$ of attribute a. For instance, suppose an attribute a with domain $d_a = 30$ is related to a 1-D grid g_0^a, which has 6 cells ($|L_{g_0^a}| = 6$). $L_{g_0^a}(0)$ represents the first cell of this 1-D grid, and it defines a subdomain value $[0, 4]$ (range). The overall process consists of making all $S_{g_b^a}(i)$ consistent. To achieve that we need to compute their weighted average as $S_{(a,i)} = \sum_{j=1}^{b} \theta_j \cdot S_{g_b^a}(i)$. Based on (ZHANG *et al.*, 2018), we derive the optimal weight θ_j for each $S_{g_b^a}(i)$ that minimizes the variance of $S_{(a,i)}$ as

$$\theta_j \leftarrow \left(\frac{1}{|L_{g_b^a}(j)|}\right) / \left(\sum_{j=0}^{b} \frac{1}{|L_{g_b^a}(j)|}\right)$$

The intuition is to put higher weights on grids that have more accurate estimations. Next, for each subdomain value in grid g_b^a, we calculate the optimal weighted average $S_{(a,i)} \leftarrow S_{(a,i)} + \theta_j \cdot S_{g_b^a}(i)$. Then, the cells' estimates are updated, making the sum of all frequency estimates $S_{g_b^a}(i)$ equal to $S_{(a,i)}$. After ensuring grid consistency, any resulting negative frequency estimates are removed.

5.3.3 Response Matrix

The response matrix generation phase combines the coarse granular k-D ($k \geq 2$) and fine granular 1-D grids. The goal is to create a response matrix that balances the estimations of these two types of grids to build a more accurate grid. This process is illustrated in Figure 22. A response matrix covers the entire domain of the attributes involved and each cell has unit size. These matrices are used during the query answering stage. Algorithm 7 shows the entire process of building a response matrix using the Weighted Update estimation method (ARORA *et al.*, 2012; HARDT *et al.*, 2012). It starts by initializing all elements in a matrix M_Γ that have related grids Γ. For instance, if we have a 3-D grid $[a, b, c]$, we combine this grid with 1-D grids $[a]$, $[b]$, and $[c]$ to build a $[a, b, c]$ response matrix with $\Gamma = ([a, b, c], [a], [b], [c])$. In line 6, we find the set $\delta(g_j)$ of all K-D values that have coordinates (x) that contribute to the frequency estimate of cell g_j from grid g and sum all of them in line 9. For instance, let $d_a = 100$, $d_b = 60$ and $d_c = 40$ and its corresponding grid $[a, b, c]$ has size $4 \times 4 \times 4$ cells. Each cell g_j defines a 3-D partition with coordinates vector (x). The first cell from $g(a, b, c)$, for instance, may define the subdomain $[0, 24] \times [0, 14] \times [0, 10]$. Then, in line 4, we would collect all 3-D values in the subdomain $[0, 24] \times [0, 14] \times [0, 10]$ *i.e.,* values that contribute to the estimate frequency of cell g_j. With the sum, we can update the response matrix in line 12. Convergence occurs when threshold $t < 1/n$.

5.3.4 λ-D Query Estimation

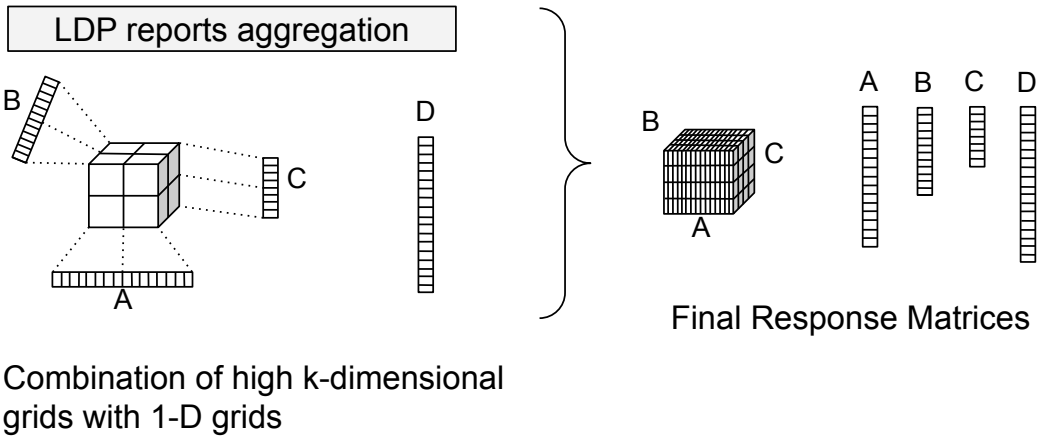
The query answering process involves using response matrices to estimate multi-dimensional queries. Combining different grids to estimate an answer introduces additional error. In PrivMDC, with fewer and higher dimensional grids, this issue is minimized as shown in Figure 23. A query q is decomposed into T associated queries and getting their respective answers. The first step of estimating the response f_q of a λ-dimensional $q = (a_{t_1}, v_{t_1}) \wedge (a_{t_2}, v_{t_2}) \wedge \ldots \wedge (a_{t_\lambda}, v_{t_\lambda})$ query involves decomposing q into T associated queries and getting their respective answers. We find T, by selecting among $\mathcal{G}$, the estimated grids that can contribute to answer q. For

Algorithm 7: Build the k-dimensional response matrix

Input : Related grids Γ, attributes' domain sizes D
Output : Response matrix M_Γ

Initialize all $\bigtimes_{i=1}^{|D|} D_i$ elements in matrix M_Γ as $\frac{1}{\prod_{i=1}^{|D|} D_i}$
while *not convergence* **do**
  for $i \leftarrow 0$ **to** $|\Gamma| - 1$ **do**
    grid $g \leftarrow \Gamma_i$;
    for $j \leftarrow 0$ **to** $|g| - 1$ **do**
      Find the set of K-D values $\delta(g_j)$ from g_j; $S \leftarrow 0$;
      for *each K-D value* $(x) \in \delta(g_j)$ **do**
        $S \leftarrow S + M_\Gamma[(x)]$;
      end
      if $S \neq 0$ **then**
        for *each K-D value* $(x) \in \delta(g_j)$ **do**
          $M_\Gamma[(x)] \leftarrow \frac{M_\Gamma[(x)]}{S} \cdot \tilde{f}_{g_j}$;
        end
      end
    end
  end
end
return M_Γ;

Figure 22 – High-dimensional grids, such as ABC, are combined with A, B and C to create a response matrix where cells have unit size. In this example, D is identified as an independent attribute.



Source: elaborated by the author.

example, let $A_q = [1, 2, 3, 4]$ be the set of attributes involved in query q and suppose we have grids $\mathcal{G} = ([1, 2, 3], [1], [2], [3], [4])$. In this case, grids $[1, 2, 3]$ and $[4]$ are going to be used to get the T associated queries answers. These T answers are then used to estimate f_q. In the case of answering 1-D range queries or when a certain attribute in q is found not to be correlated with

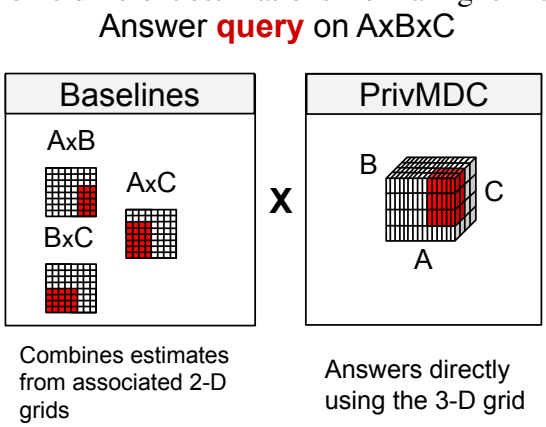
any other one, 1-D grids are used. In this case, the query window might intersect one or two cells when estimating the answer. When that happens, we assume that the distribution of data points within that cell is uniform. Algorithm 8 shows the steps of answering a λ-D query. It receives as input a set of response matrices corresponding to grids $\mathcal{G}$ and a query q. It outputs an estimated answer vector z that contains 2^λ elements. For each $f_{q^t} \in \left\{ f_{q^t} \mid t \in A_q \right\}$, the algorithm updates z iteratively. The aggregator finds the set of λ-D queries $Q(q)^t$ related to the query q_t, meaning $Q(q)^t$ includes all λ-D queries whose answers contribute to f_q^t (line 5). Next, the aggregator sums the values of $z[q']$ for all $q' \in Q(q)^t$, where $z[q']$ corresponds to the answer of q' (line 10). The answer f_q^t is then used to update z as shown in line 13. This process repeats until convergence. The estimated answer f_q for the λ-D query q is the element in z corresponding to q, *i.e.*, $z[q]$. Algorithm 8 converges when the sum of changes across all elements in z after each iteration falls below threshold $t < 1/n$, as demonstrated in (YANG *et al.*, 2020).

Algorithm 8: Answer λ-D range query

Input : The estimations of all grids $\mathcal{G}$ and query q
Output Estimated answer vector **z**
:

1 $TG = []$;
2 **for** *each grid* $g \in \mathcal{G}$ **do**
3 **if** $|A_g \cap Aq| > 0$ **then** add $A_g \cap Aq$ to TG;
4 **end**
5 $T \leftarrow$ AssociatedAnswers(TG, q);
6 Initialize all 2^λ elements in z as $\frac{1}{2^\lambda}$;
7 **while** *not convergence* **do**
8 **for** *each* $f_{q^t} \in T$ **do**
9 Find the set of queries $Q(q)^t$ corresponding to q^t;
10 Calculate $Y = \sum_{q' \in Q(q)^t} z[q']$;
11 **if** $Y \neq 0$ **then**
12 **for** *each query* $q' \in Q(q)^t$ **do**
13 $z[q'] \leftarrow \frac{z[q']}{Y} \cdot f_{q^t}$;
14 **end**
15 **end**
16 **end**
17 **end**
18 **return** z;

Figure 23 – With higher dimensional grids, PrivMDC can answer queries directly instead of having to combine different estimations from a higher number of associated grids.



Source: elaborated by the author.

5.4 Aggregator and Client Algorithms

The end-to-end algorithm of PrivMDC executed by the aggregator is presented in Algorithm 9. It starts by building the correlation model using data from the DP group (Line 2). Next, it extracts the workload characteristics and selects the correct grids that should be materialized (Lines 3 and 4). Then, the optimized user distribution is performed in Line 6. Using Algorithm 10, users perturb their private value using the protocol (Lines 3 and 4) that provides the lowest variance according to the AFO. Continuing in Algorithm 9, line 7, the aggregator combines all reports and materializes all the grids, estimating the frequency of each cell (Lines 9 and 10). Next, all the post-processing actions are taken in Steps 11, 12 and 13. Finally, using the response matrices, the aggregator answers all the queries.

5.5 Threat Model and Trust Assumptions

The population in this work consists of users who may or may not trust a third-party entity. Users are divided into two groups: the DP group, which consents to sharing raw data with a trusted curator, and the LDP group, which only shares a noisy version of their data. For the DP group, we adopt the standard strong threat model in differential privacy (LIU *et al.*, 2018; KULYNYCH *et al.*, 2024). The curator is fully trusted to collect, store, and process unperturbed data, correctly implement the DP algorithm, and prevent data leakage. The adversary is assumed to have access to the outputs of the DP mechanism *i.e.*, the private Bayesian network $\mathcal{N}$ and may possess extensive auxiliary information, including all data except that of one individual (CORMODE, 2011; GKOUNTOUNA *et al.*, 2022), with goals such as membership or attribute

Algorithm 9: PrivMDC - Aggregator Algorithm

Input : Privacy budget ϵ, Domain of attributes D, Set of attributes in the dataset A

Build correlation model with Algorithm 1;
Build workload parameters W, S, T; `// Section 5.2.2`
Select grids $\mathcal{G}$ with Algorithm 2;
Calculate sizes of grids in $\mathcal{G}$; `// Section 5.2.4`
Find LDP Group user distribution; `// Section 5.2.5`
`// Each user in the LDP group receives one grid configuration and executes their Client program (Algorithm 10) locally`
Aggregates all LDP reports of all $\mathcal{G}$ grids;
for *each grid* $g \in \mathcal{G}$ *and cell* $g^i \in g$ **do**
    if $size(g) > 3e^{\epsilon} + 2$ **then** $g^i = \Phi_{OLH(\epsilon)}(subinterval(g^i))$;
    else $g^i = \Phi_{GRR(\epsilon)}(subinterval(g^i))$;
end
Remove negative estimations from $\mathcal{G}$;
Make grids in $\mathcal{G}$ consistent; `// Section 5.3.2`
Build response matrices with Algorithm 3;
Estimate the answer of all queries using Algorithm 8;

Algorithm 10: PrivMDC - Client Algorithm

Input : Grid configuration g, Privacy budget ϵ
Output LDP report g'_v
:

user maps their private value v to grid g resulting in private grid value g_v;
if $size(g) > 3e^{\epsilon} + 2$ **then** $g'_v = \Psi_{OLH(\epsilon)}(g_v)$;
else $g'_v = \Psi_{GRR(\epsilon)}(g_v)$;
return g'_v;

inference.

For the LDP group, we follow assumptions aligned with prior works (REN *et al.*, 2018; HE *et al.*, 2025; CHEN *et al.*, 2022; GAO; ZHOU, 2024; WU *et al.*, 2020; ZHANG *et al.*, 2023; ZHANG *et al.*, 2018; ZHANG *et al.*, 2024), considering both the aggregator and users to be honest-but-curious (PAVERD *et al.*, 2014). Entities follow protocols specifications honestly, but may seek to infer private information. They share common public knowledge, including the privacy-preserving FO protocols, hash functions, attribute domains, and grid configurations. Grid selection is derived from an ϵ-DP correlation model (Algorithm 9), ensuring no privacy leakage if adversaries observe the grids. Also, we assume collusion or malicious user behavior such as data poisoning is not present (**??**CHEU *et al.*, 2021).

5.6 Privacy guarantees

In our setting, n users are classified in two different disjoint groups, (the DP and LDP group), based on their level of trust in a third-party. Next, we demonstrate that PrivMDC satisfies privacy requirements based on users' trust levels. More specifically, ϵ-DP for users in the DP group, and ϵ-LDP for users in the LDP group.

Proposition 5.1. PrivMDC satisfies ϵ-differential privacy for users in the DP group.

Proof. The DP group dataset is only used in step 1 of Algorithm 9, where Algorithm 1 is executed and requires direct access to the DP group data in order to build the correlation model. More specifically, in Algorithm 1, there is only one place where we interact directly with the DP group data, namely, the greedy selection of an attribute-parent pair a_i, p_i in each iteration of the algorithm (Line 10). Such selection is done by using the exponential mechanism (MCSHERRY; TALWAR, 2007) using the mutual information $I(a, p)$ as score function and privacy budget ϵ. No further access to raw DP group data is required in Algorithm 9, and all subsequent processing operates solely on the privatized correlation model $\mathcal{N}$. □

Proposition 5.2. PrivMDC satisfies ϵ-local differential privacy for users in the LDP group.

Proof. In Algorithm 9, PrivMDC does not touch LDP group dataset until step 7, where each user is asked to execute their PrivMDC client Algorithm 10 locally. In Algorithm 10, users perturb their private mapped value g_v either using Ψ_{OLH}, in step 3, or using Ψ_{GRR}, in step 4, with privacy budget ϵ. Both Ψ_{OLH} and Ψ_{GRR} are proved to satisfy ϵ-LDP (Section 5.2.6). The only value that leaves each user's device is the noisy value g'_v. In steps 8-10 of Algorithm 9, it estimates the frequency of each cell using the appropriate protocol choice. Note that this estimation is made on aggregated noisy LDP reports only. Thus, no privacy is leaked. Finally, the steps 11 through 14 are post-processing steps, which do not violate users' privacy (DWORK *et al.*, 2006). □

5.7 Discussion

The data distribution in the DP group can differ from the distribution in the LDP group. In order to have an indication of the error bounds, the querier can compare the data distribution from the DP group, which is estimated under central DP, to the data distribution from the LDP group which is estimated under local DP. Querying data of users in the DP group, under central DP, requires splitting the privacy budget ϵ among queries which deteriorates utility.

Thus, even if the DP group represents a random sample of users from the overall user population, answering an unbounded number of queries using the DP group is not practical due to large errors. We evaluate that scenario in Section 7.2. PrivMDC can also be used in an incremental adaptive setting that considers that query workload and correlations in data changes over time. That can be done by partitioning the users over time and using data from more recent users to handle changes in correlations and workload over time.

6 DP-AE: DIFFERENTIALLY PRIVATE ADAPTIVE ESTIMATOR

In this chapter, we introduce the DP-AE strategy, an adaptive estimator designed for a fixed-size workload of differentially private multi-dimensional range queries in the context of a hybrid user population. We begin by motivating the need for such a method, and then describe the DP-AE algorithm in detail in Section 6.2. We study and formalize the utility of standalone estimators in Section 6.3. Section 6.4 describes the query answering optimization problem and presents the algorithm to solve it. Finally, in Section 6.5, we investigate privacy amplification aspects.

6.1 Motivation

In any large-scale data collection effort, it may be impractical to assume a monolithic privacy preference across all participants. Individuals possess diverse technical backgrounds, personal values, and perceptions of risk, leading to a spectrum of privacy requirements (KHALAF *et al.*, 2024; AVENT *et al.*, 2020). A one-size-fits-all approach to privacy preservation often fails, either by providing insufficient protection for the most cautious users or by being overly restrictive and thus unnecessarily degrading data utility for the less concerned (BEIMEL *et al.*, 2020). An important factor that models an individual's privacy stance is their level of trust in the central data curator. This trust may be influenced by the curator's reputation, security practices, and the sensitivity of the data in question (AVENT *et al.*, 2017). This notion of trust naturally splits the population into at least two main groups, each aligning with a distinct differential privacy model: DP group and LDP group.

The previously discussed techniques, FELIP (Chapter 4) and PrivMDC (Chapter 5), are designed to support answering an unbounded number of queries. In this chapter, we shift our focus on designing an approach that addresses all the challenges on multi-dimensional range queries as discussed in Chapter 1, but with the goal of answering a fixed number of queries. Given that the population of users is composed of DP (DP group) and LDP users (LDP group), we can utilize both groups to estimate the answers of range queries over the total of the population. We argue that estimators relying exclusively on either the central DP data or the LDP data are fundamentally suboptimal, as they fail to efficiently manage the inherent trade-off between sampling error and privacy-induced error. In contrast, we propose to leverage the complementary strengths of both data sources to minimize the total Mean Squared Error

(MSE) and thus maximize statistical utility.

We identify several scenarios where a combination of estimators may be leverage to answer private queries.

- Public Health Surveillance: A national health agency aims to track the prevalence of a specific influenza strain. A few dozen major hospitals and labs agree to share exact, anonymized patient records, which are protected under a central DP model. Concurrently, a public health mobile application allows hundreds of thousands of citizens to self-report symptoms under LDP. To get an accurate, real-time national prevalence estimate, the agency must combine the hospital data (DP group) with the noisy crowdsourced data (LDP group).
- Smart City Mobility Analytics: A municipal transport authority wants to understand commuter flow to optimize public transit. It collects high-fidelity, second-by-second GPS data from its fleet of buses and trams, which it analyzes under central DP. Simultaneously, it obtains LDP-protected location data from a third-party navigation app used by thousands of private car drivers. To estimate the number of people within a specific downtown corridor during rush hour (a spatial range query), a hybrid model can combine estimates from from both data sources.
- Economic Activity Monitoring: A central bank wants to gauge consumer spending. It obtains detailed, transactional data from a few consenting financial institutions, protecting it with central DP. It also aggregates, LDP-protected data on spending categories from mobile payment apps. To estimate total spending in the retail sector, the bank must combine estimates from the reliable but limited institutional data with the noisy app data to produce a robust economic indicator.

6.2 DP-AE: Adaptive Differentially Private Estimator

The DP-AE is an adaptive approach to answer differentially private multi-dimensional range queries. It is designed to operate in heterogeneous privacy environments, accommodating users with differing privacy expectations. In particular, the population is composed of two distinct groups: users who consent to differential privacy (DP) guarantees at the aggregator level, and those who require local differential privacy (LDP) guarantees, applying randomization directly on their devices. Such heterogeneous privacy settings have been the subject of increasing interest in recent years and have been explored in several prior works (KOHEN; SHEFFET,

2022; AVENT *et al.*, 2017; AVENT *et al.*, 2020; BEIMEL *et al.*, 2020; KHALAF *et al.*, 2024; ACQUISTI; GROSSKLAGS, 2005; ACQUISTI *et al.*, 2015). DP-AE is developed on top of the PrivMDC framework as shown on Figure 24, as both methods share the same privacy models and assumptions, differing only on the query answering objective. While PrivMDC (Chapter 5) focus on answering an unbounded number of queries, DP-AE's goal is to find the best strategy to answer a fixed size set of queries by combining DP and LDP estimates.

Figure 24 illustrates the full architecture of the DP-AE approach, which consists of several coordinated steps. The process begins with the construction of a Bayesian network using data contributed by the DP group. This network is learned under ϵ-DP guarantees and is used to capture statistical dependencies among attributes. These correlations serve two purposes: (i) to guide the construction of k-dimensional grids that define the query space, and (ii) to inform the data partitioning strategy for the LDP group.

In parallel, the framework incorporates prior knowledge about the analysts' query workload. From this information, it extracts workload parameters such as attribute weights. These parameters are then used, together with the correlation structure obtained from the Bayesian network, to define the collection of specific grids and determine their respective dimensionalities.

To ensure high utility in answering queries over these grids, the LDP group is partitioned into disjoint subsets, each assigned to a distinct grid. Users in each subset report their data under ϵ-LDP guarantees, submitting privatized responses that correspond to their assigned grid. The aggregator collects these noisy reports, estimates the cell frequencies using standard LDP estimation techniques, and applies post-processing procedures: removal of negative estimations and consistency among grids. Finally, the aggregator invokes the DP-AE's adaptive estimator to answer a fixed set of queries. The estimator adaptively selects among the best estimator to use based on the different stages of query answering.

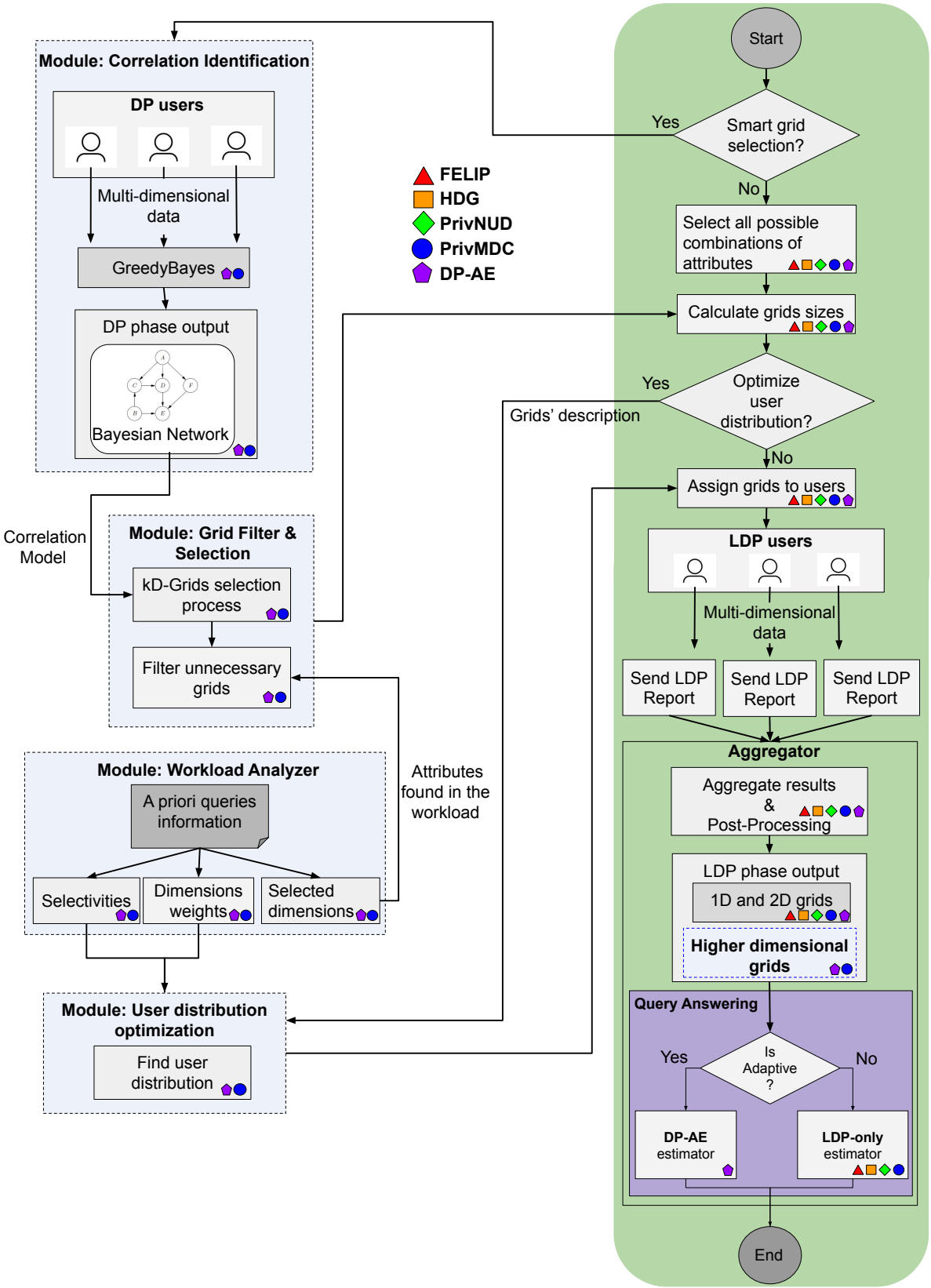
Since DP-AE is built on top of PrivMDC's framework, they share all the steps shown in Figure 24 but the purple box. Thus, for the remainder of this chapter, we focus on presenting and formalizing the adaptive estimator.

6.3 Standalone Estimators - Utility Analysis

Let us consider a dataset D of N individuals, whose data is partitioned into two disjoint sets: DP group D_{DP} and LDP group D_{LDP}. The total population size is $N = N_{\text{DP}} + N_{\text{LDP}}$.

- DP group D_{DP}: A set of N_{DP} individuals who trust a central curator. Their exact data is

Figure 24 – We build DP-AE on top of the PrivMDC framework. The basic steps to answer multi-dimensional range queries in the literature are shown in green. The purple box represent the new proposed adaptive query answering stage.



Source: elaborated by the author.

collected and subsequently protected by a central mechanism $\mathcal{M}_{\text{DP}}$ satisfying ϵ-Differential Privacy. The DP group participates in two distinct tasks, resulting in the division of its total privacy budget as $\epsilon = \epsilon_{\text{cm}} + \epsilon_{\text{qa}}$. The portion ϵ_{cm} is allocated to modeling correlations among attributes, while ϵ_{qa} is reserved for answering queries.

- LDP group D_{LDP}: A set of N_{LDP} individuals who do not trust the curator. Each individual $i \in D_{\text{LDP}}$ applies a local randomizer $\mathcal{R}_i$ to their data v_i, producing a noisy version $\tilde{v}_i$, before transmission. Each $\mathcal{R}_i$ satisfies ϵ-Local Differential Privacy. Data from the LDP group will be used exclusively for query estimation.

We want to estimate the answers of a set of queries Q. More specifically, for each query $q \in Q$, we want to estimate the number of users $q(D)$ in the population that satisfies a conjunction of predicates. An optimal estimator is one that minimizes the Mean Squared Error (MSE), which decomposes into squared bias and variance. Next, we analyze and formalize the utility of the DP and LDP estimators individually. Following this, in Section 6.4, we present a formal framework for combining the two estimators and formulate the corresponding optimization problem, which is solved using the procedure described in Algorithm 20.

6.3.1 Central DP-Only Estimator

Let $\hat{q}$ be the true answer count by querying D_{DP}. To privatize the answer of $\hat{q}$, we apply the Laplace mechanism $\hat{q}_{\text{dp}} = \hat{q} + \eta_{N_{\text{DP}}}$. The Laplace mechanism adds noise $\eta_{N_{\text{DP}}} \sim \text{Lap}(b)$. This means that $\eta_{N_{\text{DP}}}$ is a random variable drawn from the Laplace distribution with scale parameter b. In the case of estimating the answers of count queries, the sensitivity of the query is $\Delta q = 1$. Then the noise is drawn from $\eta_{N_{\text{DP}}} \sim \text{Lap}\left(\frac{1}{\epsilon}\right)$, where ϵ is the privacy budget. The Laplace noise $\eta_{N_{\text{DP}}}$ added to satisfy DP has zero mean and variance

$$\text{Var}(\eta_{N_{\text{DP}}}) = 2b^2 = \left(\frac{2}{\epsilon^2}\right) \tag{6.1}$$

Next, we define the expected error of estimating $q(D)$. By querying the DP group dataset D_{DP} of N_{DP} individuals we obtain the noisy estimate $\hat{q}_{\text{DP}}(D_{\text{DP}})$, thus we determine the mean square error of this estimation as $\text{MSE}(\hat{q}_{\text{DP}}(D_{\text{DP}})) = \mathbb{E}\left[(\hat{q}_{\text{DP}}(D_{\text{DP}}) - q(D))^2\right]$. Expanding it, we have

$$\begin{aligned}\text{MSE} =& \mathbb{E}\left[(\hat{q}_{\text{DP}}(D_{\text{DP}}) - q(D) + \eta_{N_{\text{DP}}})^2\right] \\ =& \mathbb{E}[(\hat{q}_{\text{DP}}(D_{\text{DP}}) - q(D))^2] + \mathbb{E}[\eta_{N_{\text{DP}}}^2] + 2\mathbb{E}[(\hat{q}_{\text{DP}}(D_{\text{DP}}) - q(D))\eta_{N_{\text{DP}}}]\end{aligned}$$

Since $\hat{q}_{\mathrm{DP}}(D_{\mathrm{DP}})$ and $\eta_{N_{\mathrm{DP}}}$ are independent, and $\mathbb{E}[\eta_{N_{\mathrm{DP}}}] = 0$, the cross term vanishes:

$$\mathrm{MSE}(\hat{q}_{\mathrm{DP}}(D_{\mathrm{DP}})) = \underbrace{\mathbb{E}[(\hat{q}_{\mathrm{DP}}(D_{\mathrm{DP}}) - q(D))^2]}_{\text{Sampling Error}} + \underbrace{\mathbb{E}[\eta^2_{N_{\mathrm{DP}}}]}_{\text{DP Noise Error}}$$

Considering we want to answer Q queries, we divide the total privacy budget for query answering ϵ_{qa} into $|Q|$ partitions. Plugging the Laplace noise variance we have

$$\mathrm{MSE}(\hat{q}_{\mathrm{DP}}) = \mathbb{E}[(\hat{q}_{\mathrm{DP}}(D_{\mathrm{DP}}) - q(D))^2] + \left(\frac{2}{(\epsilon_{\mathrm{qa}}/|Q|)^2}\right)$$

The first term reflects statistical sampling error from using the DP group dataset. The second term quantifies error introduced by the Laplace mechanism to satisfy ϵ-DP. Sampling error is incurred by using the frequency estimates of the DP group dataset D_{DP} to represent those obtained from the entire population D, since D_{DP} may have different distribution from the global D one. Since, we don't have access to the population D distribution, we approximate the expected error of the DP estimator considering only the induced privacy error. We use Var^*_{DP} do denote the approximation of Var_{DP}.

$$Var^*_{\mathrm{DP}} = \left(\frac{2(|Q|^2)}{\epsilon^2_{\mathrm{qa}}}\right) \tag{6.2}$$

In summary, we can expect the utility from the DP estimator to increase as ϵ_{qa} increase. Also, the more queries we want to answer the larger is the variance.

6.3.2 Local LDP-Only Estimator

Using only the LDP data D_{LDP}, the curator aggregates the noisy responses from the N_{LDP} individuals to form an estimate $\hat{q}_{\mathrm{LDP}}$ for each query $q \in Q$. Users in the LDP group are divided into $|\mathcal{G}|$ groups of n_i users, where $N_{\mathrm{LDP}} = \sum_{i=1}^{|\mathcal{G}|} n_i$, in order to estimate all the grids necessary to answer all queries. The LDP estimator has four kinds of errors: noise error, sampling error, non-uniformity error, and estimation error. The noise error is due to the use of LDP frequency oracles. To satisfy LDP, one adds, to each cell, an independently generated noise, and these noises have the same standard deviation. The sampling error is incurred by using cells' frequencies obtained from a user group to represent those obtained from the entire LDP group N_{LDP}, since the user group may have different distribution from the global one. Those two sources of error can be quantified together. To estimate λ-dimensional query, we consider that

the ratio of interval to the attribute's domain size *i.e.,* query selectivity is r_i along $i \in \lambda$. Also, consider l_i to be the number of cells in dimension $i \in \lambda$. The number of cells included in the query rectangle is $\prod_{i=1}^{\lambda}(l_i s_i)$. Thus, the squared noise and sampling error for OLH is

$$\prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{4e^{\epsilon}}{n_i(e^{\epsilon}-1)^2}$$

and for the GRR protocol is

$$\prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{(e^{\epsilon}+L-2)}{n_i(e^{\epsilon}-1)^2}$$

where L is the number of cells in a λ-dimensional grid and n_i is the number of users from P_{LDP} allocated to estimating the corresponding λ-dimensional grid.

The non-uniformity error arises from cells that partially overlap with the query rectangle but are not fully contained within it. For such cells, the number of data points falling within the query region must be estimated under the assumption of a uniform distribution across the cell. When the actual data distribution deviates from uniformity, this approximation introduces error. The magnitude of the non-uniformity error in any intersected cell generally depends on, and is bounded by, the number of data points in that cell. Consequently, increasing the granularity of the partition *i.e.*, using finer cells reduces the non-uniformity error. We can approximate the non-uniformity error by

$$\left(\frac{\alpha\left(\sum_{i=1}^{\lambda} l_i s_i\right)}{\prod_{i=1}^{\lambda} l_i}\right)^2 \tag{6.3}$$

Adding the squared noise, sampling and non-uniformity error, we have the following estimate for the OLH protocol

$$Var^*_{\text{OLH}} = \prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{4e^{\epsilon}}{n_i(e^{\epsilon}-1)^2} + \left(\frac{\alpha\left(\sum_{i=1}^{\lambda} l_i s_i\right)}{\prod_{i=1}^{\lambda} l_i}\right)^2 \tag{6.4}$$

and for the GRR protocol we have

$$Var^*_{\text{GRR}} = \prod_{i=1}^{\lambda}(l_i s_i) \cdot \frac{(e^{\epsilon}+L-2)}{n_i(e^{\epsilon}-1)^2} + \left(\frac{\alpha\left(\sum_{i=1}^{\lambda} l_i s_i\right)}{\prod_{i=1}^{\lambda} l_i}\right)^2 \tag{6.5}$$

Finally, when estimating the answer to a λ-dimensional range query with $\lambda > 2$ we may use the corresponding answers of β-dimensional range queries ($\lambda \geq \beta$), estimation error is

inevitably introduced. This error is inherently dependent on the underlying dataset, making it analytically not practical to express with a general formula.

We use Var^* do denote the approximation of Var. The variances Var^*_{OLH} and Var^*_{GRR} do not include sampling error from using a subset of the population data, in this case D_{LDP}, to estimate a query on the entire population (D). However, since we don't have access to the entire dataset D distribution, we approximate the expected error of the LDP estimator as expressed in Var^*_{OLH} and Var^*_{GRR}.

6.4 Adaptive Estimator

An adaptive estimation strategy can effectively address the limitations inherent in using either central or local differential privacy estimators in isolation. The core objective is to minimize the total expected error incurred when answering a finite set of queries Q. This requires balancing the trade-offs between utility and privacy constraints characteristic of the two paradigms.

It is well-established that mechanisms based on local differential privacy tend to introduce a significantly higher level of noise than their central DP counterparts, leading to larger estimation errors for individual queries (ERLINGSSON *et al.*, 2014). This is because noise must be added at the level of each individual record in LDP to achieve privacy without trust in a central curator. In contrast, central DP mechanisms can inject calibrated noise once at the aggregate level, resulting in higher utility for a given privacy budget.

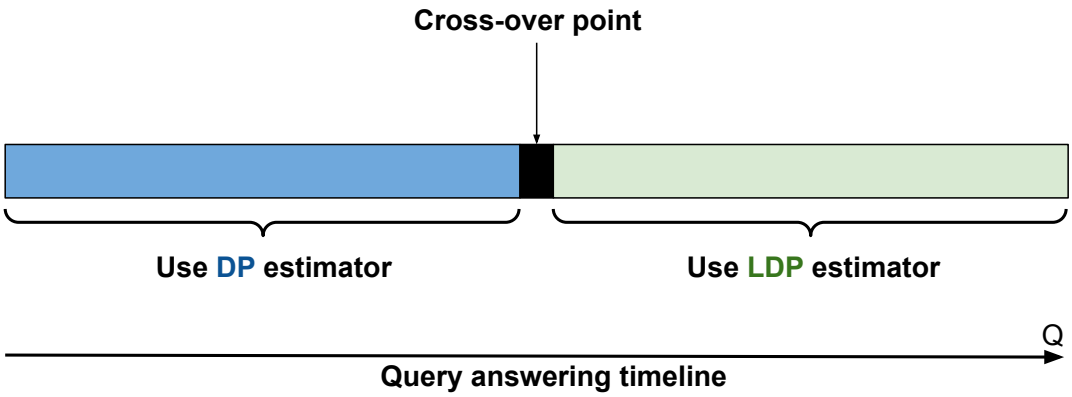
However, this advantage comes with an important limitation: in the central DP model, the total privacy loss must be bounded by a fixed privacy budget ϵ_{qa}. Since the privacy cost accumulates with each additional query, only a limited number of queries can be answered with acceptable utility before the privacy budget is exhausted.

On the other hand, LDP mechanisms enable per-user privacy guarantees that are independent of the number of queries issued to the server. This allows an unbounded number of queries to be answered without further degrading any individual's privacy, provided the users' reports are collected once with randomized encoding. This scalability in the number of queries is a key strength of the LDP framework.

To capitalize on the strengths of both approaches, we propose *AE-DP*, an adaptive estimator for answering differentially private range queries. Figure 25 illustrates the query answering process. For the first c_{p} (cross-over point) queries, the mechanism employs the central

DP estimator, using the available privacy budget ϵ_{qa} to ensure high utility responses. For the remaining $|Q| - c_p$ queries, the system switches to the local DP estimator, utilizing the previously collected LDP-encoded user reports, thus incurring no additional privacy cost.

Figure 25 – The query answering timeline goes from left to right. The first subset of queries are answered using the DP estimator (with DP group data). Then, after c_p queries, the LDP estimator (with LDP group data) is used for answering the remainder number of queries.



Source: elaborated by the author.

The choice of the cross-over point c_p, which determines how many queries are answered using the DP estimator before switching to the LDP estimator, plays a critical role in balancing utility across the workload. If c_p is set too small, the first few queries will benefit from the low-variance estimates provided by the DP mechanism (Equation 6.2), resulting in high utility for those initial queries. However, this strategy forces the remaining, potentially larger portion of the workload to be answered using the LDP estimator, which generally introduces higher noise due to its per-user randomization. As a consequence, the overall cumulative error may increase despite the initial gains.

Conversely, if c_p is chosen too large, a greater portion of the workload will rely on the DP estimator, which results in dividing the privacy budget into large number of partitions. In this case, the quality of the DP estimates degrade significantly, and the variance can become comparable to, or even exceed, that of the LDP estimator, thereby negating the expected advantages of using the DP group for query answering. Thus, determining an appropriate value for c_p requires carefully balancing the number of high-utility DP queries against the rising variance due to privacy budget constraints, while minimizing reliance on the inherently noisier LDP estimates.

Given these considerations, we formulate the construction of the adaptive estimator as an optimization problem. The objective is to determine the optimal cross-over point c_p that

minimizes the total expected error incurred when answering a fixed-size workload consisting of $|Q|$ queries. Formally,

$$\begin{aligned} \min_{c_{\mathrm{p}}} \quad & c_{\mathrm{p}} \cdot Var^{*}_{DP}(\epsilon_{\mathrm{qa}}/c_{\mathrm{p}}) + (|Q| - c_{\mathrm{p}}) \cdot Var^{*}_{LDP}(\epsilon) \\ \text{s.t.} \quad & c_{\mathrm{p}} \leq |Q| \end{aligned}$$

The Var^{*}_{DP} is expressed in Equation 6.2 and the Var^{*}_{LDP} is estimated by using Equation 6.4, since most grids tends to use the OLH protocol (Section 5.2.6). If c_{p}.

Algorithm 20 presents an adaptive strategy for estimating the answers to a finite set of range queries by combining the DP and LDP estimators. The goal is to minimize the total expected error when answering the query set $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$, given a central privacy budget ϵ_{qa} and a local privacy budget ϵ.

Algorithm 11: Adaptive Estimator for Differentially Private Range Queries

Input: Query set $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$, local privacy budget ϵ, central privacy budget ϵ_{qa}, threshold c_{p}, user reports with local DP
Output: Estimated answers $\{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{|Q|}\}$

Initialize empty answer set $\mathcal{A} \leftarrow \emptyset$;
$min\_cost \leftarrow 0$;
$c_{\mathrm{p}} \leftarrow 0$;
for $i \leftarrow 1$ **to** $|Q|$ **do**
    $cost_i \leftarrow i \cdot Var^{*}_{\mathrm{DP}}(\epsilon_{\mathrm{qa}}/i) + (|Q| - i) \cdot Var^{*}_{\mathrm{LDP}}(\epsilon)$;
    if $cost_i < min\_cost$ **then**
        $min\_cost \leftarrow cost_i$;
        $c_{\mathrm{p}} \leftarrow i$;
    end
end
Set per-query central budget $\epsilon_q \leftarrow \epsilon_{\mathrm{qa}}/c_{\mathrm{p}}$;
for $i \leftarrow 1$ **to** $|Q|$ **do**
    if $i \leq c_p$ **then**
        `// Use Central DP Estimator`
        Compute $\hat{a}_i \leftarrow q_i(\mathrm{D}_{\mathrm{DP}}) + \mathrm{Lap}(1/\epsilon_q)$;
    else
        `// Use Local DP Estimator`
        Estimate $\hat{a}_i$ using the materialized grids; `// Algorithm 8`
    end
    Append $\hat{a}_i$ to $\mathcal{A}$;
end
return $\mathcal{A}$;

The algorithm begins by initializing an empty answer set $\mathcal{A}$ (line 1), and setting the initial minimum cost and optimal allocation threshold c_{p} to zero (lines 2-3). The value c_{p} represents the number of queries that will be answered using the central DP estimator. In lines 4-8, the algorithm iterates over all possible values of $i \in \{1, \ldots, |Q|\}$ to evaluate the total expected error. For each split point i, it computes the total cost as the sum of the variances of the central DP mechanism (with per-query budget ϵ_{qa}/i) and the LDP mechanism (with fixed budget ϵ), weighted by the number of queries in each group (line 5). If the computed cost is lower than the current minimum, it updates the minimum and records the corresponding threshold c_{p} (lines 6-8). After determining the optimal value of c_{p}, the algorithm assigns a per-query central privacy budget $\epsilon_q = \epsilon_{\text{qa}}/c_{\text{p}}$ (line 11), assuming uniform allocation across the central DP queries. In the second phase (lines 12-18), the algorithm processes each query in Q:

- For queries indexed by $i \leq c_{\text{p}}$, the algorithm uses the central DP estimator (line 14). This involves evaluating the query on a differentially private representation of the dataset, denoted P_{DP}, and adding Laplace noise scaled to the sensitivity Δf and the per-query budget ϵ_q.
- For queries where $i > c_{\text{p}}$, the local DP estimator is used (lines 16). These estimates are computed by using the materialized grids.

Each estimated answer is appended to the result set $\mathcal{A}$ (line 18), and the final output is returned in line 20.

6.5 Privacy Amplification by subsampling

In this section, we study the approach of extending the number of queries using the DP group through subsampling. We formalize the expected utility of employing such procedure into our adaptive technique. In recent years, several studies have examined the extent to which sampling contributes to enhanced privacy guarantees when the privatized output is computed using only a subset of the original data (BALLE *et al.*, 2020; ABADI *et al.*, 2016; KASIVISWANATHAN *et al.*, 2011; BUN *et al.*, 2015; WANG *et al.*, 2016; LI *et al.*, 2012; HU *et al.*, 2022). Intuitively, it is plausible that the use of a sample, rather than the full dataset, increases the level of privacy protection, as it introduces uncertainty regarding the inclusion of any particular individual in the sample. This notion has been long recognized by statistical agencies, which have historically disseminated public use microdata samples—often comprising only 1 to 5 percent of the census population—such as those provided through IPUMS International. The

underlying rationale is that sampling inherently provides a substantial degree of privacy, thereby permitting a relaxation of additional protective mechanisms relative to what would be required for full-population releases.

In the heterogeneous privacy settings where a portion of the population contributes data under central DP and the remainder under LDP, it is essential to efficiently allocate the available query answering privacy budget ϵ_{qa} to maximize utility. In this context, we study the use of privacy amplification by subsampling, a well-established result in differential privacy theory that states that applying a DP mechanism to a randomly selected subset of individuals amplifies the effective privacy guarantees for each participant (HU *et al.*, 2022).

Leveraging this property, one could partition the population of DP users into disjoint batches and apply the DP mechanism to each batch sequentially. Because of privacy amplification, each batch can tolerate a relatively small privacy cost per query, thus potentially enabling more queries to be answered under central DP before the total privacy budget is exhausted. Each batch of users have ϵ_{qa} to spend on query answering. Once the cumulative budget for DP users is consumed, the system can switch to the LDP estimator, which offer stronger individual guarantees but typically incur higher noise as discussed earlier.

An intuitive estimator for the frequency count using only the DP group is to compute the private count on the sub-sample N_S and scale it up:

$$\hat{q}_{\text{DP}} = \mathcal{M}_{\text{DP}}(q(S(D_{\text{DP}}))) \cdot \frac{N}{N_S}$$

The error of this estimator is driven by two distinct sources. The first one is the privacy-induced variance where the central mechanism $\mathcal{M}_{DP}$, in our case the Laplace mechanism, introduces noise to satisfy DP. The variance of this noise is proportional to the query's global sensitivity squared. For a count query, the sensitivity is 1, so $Var(\text{noise}) \propto 1/\epsilon^2$. This variance is constant with respect to the dataset size N_{DP}. The second source of error is the sampling error that accounts for the extrapolation from a sample of size N_{DP} to the full population N.

Subsampling-based algorithms are frequently encountered in learning tasks (ABADI *et al.*, 2016). The variance of the privatized output is determined by two principal sources: the sampling variance and the variance arising from noise infusion. Within the class of differential privacy mechanisms based on additive noise, such as the Laplace mechanism, accuracy improvements can only be realized when the reduction in variance due to noise addition exceeds the increase in sampling variance. Although the sampling variance typically decreases

inversely with the sample size for most estimators, the dependence of the noise, related variance on the sample size is more intricate and does not exhibit such a straightforward inverse relationship.

Theorem 6.1 (Privacy Amplification (HU *et al.*, 2022))**.** Suppose that a randomized function $\mathcal{F}$ gives (ϵ, δ)-DP for the query $q(D_N)$. Then, $\mathcal{A}$ satisfies (ϵ_n, δ_n)-DP for the estimator $q(S(D_N))$ where $\epsilon_n = log(1 + n/N(e^{\epsilon} - 1))$ and $\delta_n = (n/N)\delta$, that is, for fixed values of $\epsilon > 0$ and $\delta \geq 0$ and for all $\omega \in \Omega$

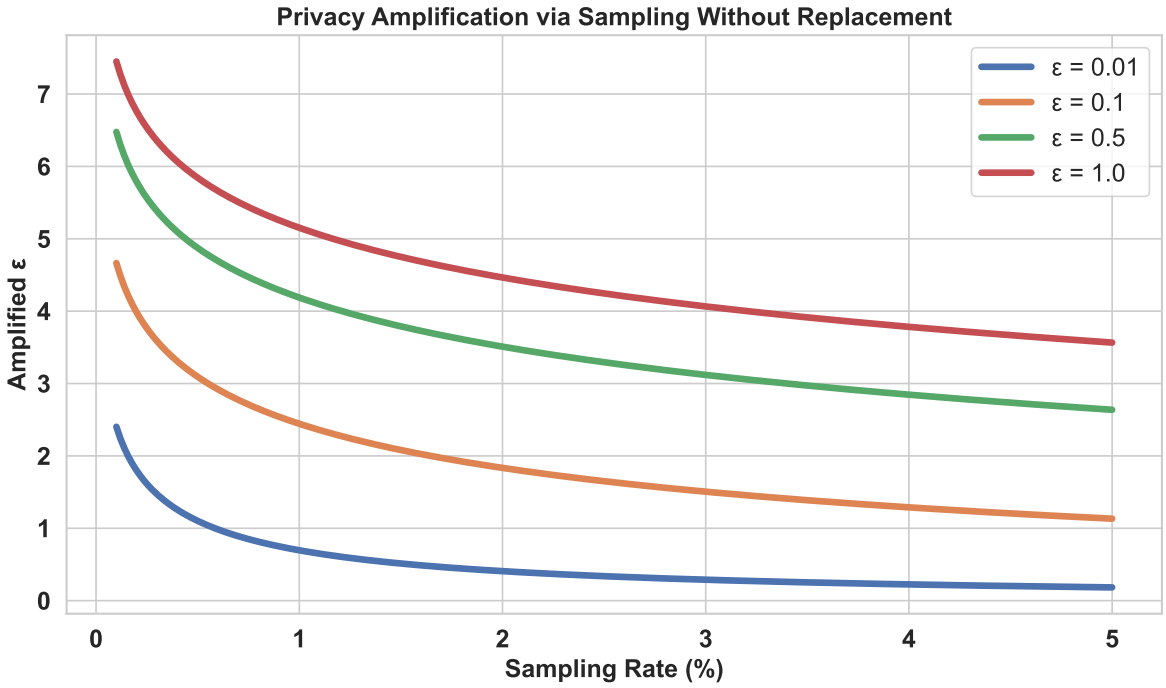
$$Pr[\mathcal{A}(q(S(D_N))) \leq \omega] \leq (1 + \tfrac{n}{N}(e^{\epsilon} - 1)))Pr[\mathcal{A}(q(S(D'_N))) \leq \omega] + \tfrac{n}{N}\delta$$

for any $D_N = d_1, ..., d_N$ and $D'_N = d'_1, ..., d'_N$ differing on at most one pair of variables $d_i \neq d'_i$ for any $d_i, d'_i \in \mathcal{D}$.

Theorem 6.1 demonstrates that the privacy parameters associated with releasing $\mathcal{A}(q(S(D_N)))$ are smaller than those for $\mathcal{A}(q(D_N))$, indicating that the simple random sampling mechanism S amplifies the level of data protection. This result is particularly relevant in two data collection contexts. The most immediate application arises in sample survey settings, where a statistical agency selects a random sample $S(D_N)$ from the population dataset D_N in order to estimate a finite population query $q(S(D_N))$. When the agency targets a specific privacy guarantee, expressed through fixed values (ϵ, δ), the actual privatization mechanism is applied to $q(S(D_N))$, not to $q(D_N)$. Consequently, the effective privacy parameters for the sample-based query will typically be more favorable (i.e., smaller) than (ϵ, δ). In particular, under simple random sampling without replacement, Theorem 6.1 establishes that applying a privacy mechanism with parameters $\epsilon_n = \log(1 + \frac{N}{n}(e^{\epsilon} - 1))$ and $\delta_n = \frac{N}{n}\delta$ to $q(S(D_N))$ suffices to achieve the original privacy guarantee (ϵ, δ). This is advantageous, as it implies that less noise must be added to meet the desired level of privacy protection.

Figure 26 illustrates the behavior of the amplified privacy parameter ϵ_n following Theorem 6.1. This formulation highlights how smaller sampling rates (i.e., more aggressive subsampling) result in substantially higher amplified ϵ_n values. The curves correspond to different values of the original privacy parameter $\epsilon \in \{0.01, 0.1, 0.5, 1.0\}$. As evident from the plot, the degree of amplification increases nonlinearly as the sampling rate decreases. This underscores the potential for privacy amplification through subsampling. For instance, for a 1% sampling rate and $\epsilon = 0.5$, we obtain an amplified privacy budget $\epsilon_n = 4.187$.

Figure 26 – Amplified ϵ_n for different sampling rates.



Source: elaborated by the author.

6.5.1 Accuracy of sampling using Laplace mechanism

When using subsampling as part of a privacy-preserving data release mechanism, the total error in the privatized output arises from two primary sources: the variability introduced by subsampling and the perturbation induced by the differential privacy mechanism. Note that in the setting of our adaptive estimator DP-AE, dataset D_N used in Theorem 6.1 is represented by the DP group data D_{DP}. Estimating the answer of query q using the a sample $S(D_{\text{DP}})$ of D_{DP} has the following mean square error:

$$\begin{aligned}\text{MSE}(\hat{q}_{\text{DP}}(S(D_{\text{DP}}))) =& \mathbb{E}[(\hat{q}_{\text{DP}}(S(D_{\text{DP}})) - q(D))^2] \\ =& \underbrace{\mathbb{E}[(\hat{q}_{\text{DP}}(S(D_{\text{DP}})) - q(D))^2]}_{\text{Sampling Error}} + \underbrace{\mathbb{E}[\eta^2_{n_{\text{DP}}}]}_{\text{DP Noise Error}}\end{aligned}$$

When using the Laplace mechanism, we have $\mathcal{A}(q(S(D_{\text{DP}}))) = q(S(D_{\text{DP}})) + \eta_n$, where $\eta_{n_{\text{DP}}} \sim Laplace(\Delta^G_n / \epsilon_{\text{qa}_n})$ and Δ^G_n are the global sensitivity of the output in the sample and ϵ_n is the amplified privacy budget to answer one query given the sampling rate of n/N_{DP}.

We denote $Var_{\mathrm{DP}_{\mathrm{amp}}}$ as the variance of the estimate using subsampling.

$$\begin{aligned} Var_{\mathrm{DP}_{\mathrm{amp}}} =& \mathrm{MSE}(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}}))) \\ &= \mathbb{E}[(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}})) - q(D))^2] + Var(\eta_{n_{\mathrm{DP}}}) \\ &= \mathbb{E}[(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}})) - q(D))^2] + \frac{2(\Delta_n^G)}{\epsilon_n^2} \end{aligned}$$

where $\mathbb{E}[(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}})) - q(D))^2]$ is the sampling variance of the estimate and $\frac{2(\Delta_n^G)}{\epsilon_n^2}$ is the variance of the Laplace noise. A key challenge lies in the fact that the underlying data distribution of the dataset D is unknown, which prevent us of determining the sampling variance. As a result, it becomes difficult to ascertain under which conditions subsampling will yield improvements in estimation accuracy.

In order for the variance of querying a sample of the DP group the entire DP group be smaller or equal to the variance of querying the entire DP group data *i.e.*, $Var_{\mathrm{DP}_{\mathrm{amp}}} \leq Var^*_{\mathrm{DP}}$, the additional sampling variance from subsampling D_{DP} always needs to be smaller than the variance of the noise added to protect the population parameter. In many cases, this might already imply that the sampling rate cannot be too small to avoid large sampling variances. However, when the sampling rate is large, the reduction in variance due to noise infusion, relative to applying the mechanism on the full population, is minimal. Consequently, it remains essential to account for the contribution of noise-induced variance even when the estimate is based on a sample. Since we are estimating count queries, sensitivity is independent of the sampling rate ($\Delta_N^G = \Delta_n^G = 1$) *i.e.,* the global sensitivity of querying the entire dataset D_{DP} is equal to the one when query the sample $S(D_{\mathrm{DP}})$. Under the setting of our adaptive estimator of count queries, we can obtain an upper bound for the sampling variance as

$$
\begin{aligned}
Var_{\mathrm{DP}_{\mathrm{amp}}} &\leq Var^{*}_{\mathrm{DP}} \\
\mathbb{E}[(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}})) - q(D))^2] + Var(\eta_{n_{\mathrm{DP}}}) &\leq Var^{*}_{\mathrm{DP}} \\
\mathbb{E}[(\hat{q}_{\mathrm{DP}}(S(D_{\mathrm{DP}})) - q(D))^2] &\leq Var^{*}_{\mathrm{DP}} - Var(\eta_{n_{\mathrm{DP}}}) \\
&\leq \left(1 - \frac{Var(\eta_{n_{\mathrm{DP}}})}{Var^{*}_{\mathrm{DP}}}\right) Var^{*}_{\mathrm{DP}} \\
&\leq \left(1 - \frac{2(\Delta^{G}_{n}/\epsilon_n)^2)}{2(\Delta^{G}_{N}/\epsilon)^2)}\right) Var^{*}_{\mathrm{DP}} \\
&\leq \left(1 - \frac{\epsilon^2}{\epsilon_n^2}\right) Var^{*}_{\mathrm{DP}}
\end{aligned}
$$

Given the upper bound of the sampling variance, we can approximate the variance of answering a query using subsampling under the DP-AE estimator as

$$
Var^{*}_{\mathrm{DP}_{\mathrm{amp}}} \leq \left(1 - \frac{\epsilon^2}{\epsilon_n^2}\right) Var^{*}_{\mathrm{DP}} + \frac{2}{\epsilon_n^2} \tag{6.6}
$$

The privacy budget partition allocated for answering a query in the DP-AE strategy is $\epsilon_q = \frac{\epsilon_{\mathrm{qa}}}{c_{\mathrm{p}}}$. With the amplified privacy protection by subsampling without replacement, ϵ_n will be used for each query and is computed as

$$
\epsilon_n = \log(1 + b(e^{\epsilon_q} - 1))
$$

where $b = \frac{|D_{\mathrm{DP}}|}{|S(D_{\mathrm{DP}})|}$ represent the number of batches *i.e.*, the number of disjoint samples.

When employing subsampling without replacement as a mechanism for privacy amplification in differentially private data analysis, the choice of b introduces a fundamental trade-off between privacy protection and statistical accuracy. On one hand, lower sampling rates offer stronger privacy amplification. Subsampling reduces the effective privacy loss, as the probability that any individual is included in the sample decreases. This results in smaller amplified privacy parameters and, consequently, allows for the addition of less noise to achieve a given level of privacy protection.

On the other hand, reducing the sampling rate increases the sampling variance of the estimator. Since the estimate is based on a smaller subset of the data, the statistical efficiency deteriorates, leading to higher estimation error due to reduced representativeness of the sample.

For many estimators, the sampling variance is inversely proportional to the sample size, implying that aggressive subsampling may compromise accuracy.

6.6 Discussion

Our adaptive DP-AE estimator integrates the advantages of both central and local differential privacy estimators by systematically analyzing the sources of error encountered when answering range queries using data from the DP and LDP groups. A critical observation is that the data distribution in the DP group may deviate from that in the LDP group, which has direct implications for the standalone estimators, thus impacting the DP-AE estimator.

The MSE expressions for the DP and LDP estimators, as provided in Equations 6.2, 6.4, and 6.6, are approximations. Consequently, the cross-over point c_{p}, which governs the DP-AE estimator, should also be regarded as an approximation. To obtain tighter bounds for the DP-AE estimator, the querier can compare the distribution estimated from the DP group with the distribution estimated from the LDP group.

Furthermore, the use of subsampling presents an opportunity to extend the applicability of the DP estimator by exploiting the privacy amplification effect. However, potential gains in utility from subsampling are not guaranteed and depend on the population variance, for instance. Analytical results we found for the variance introduced by subsampling provide only an upper bound on the sampling variance. These results typically assume that the global sensitivity of the query output is independent of the database size. In scenarios where sensitivity decreases with the size of the dataset, the variance ratio $Var(\eta_{n_{\mathrm{DP}}})/Var_{\mathrm{DP}}$ increases, yielding a tighter bound on the total error. In our problem, we find that we can never expect to see accuracy gains whenever

$$Var(q(S(D_N))) \geq 2\left(\frac{1}{\epsilon^2} - \frac{1}{\epsilon_n^2}\right) \tag{6.7}$$

7 EXPERIMENTAL EVALUATION

In this chapter, we evaluate our three strategies to query answering under different workloads and datasets. We experiment FELIP, PrivMDC and DP-AE in sections 7.1, 7.2 and 7.3, respectively.

Error metric. Unless stated otherwise, we use the Mean Absolute Error (MAE) to measure the accuracy of estimated answers in all scenarios explored in this chapter. Given a set Q queries, it is computed as $MAE = \frac{1}{|Q|}\sum_{q\in Q}|f_q - \bar{f_q}|/N$, where f_q and $\bar{f_q}$ are the estimated and true answers of query q, respectively. N is the number of users in the dataset. This metric is chosen due to its adoption in the most relevant baseline methods (YANG *et al.*, 2020; WANG *et al.*, 2019a), ensuring consistency and comparability across experimental evaluations.

7.1 FELIP - Evaluation

We implemented FELIP in Python 3.10. All the experiments were conducted on a server with Ubuntu 20.04, Intel Core i7-7820X 3.60GHz, and 128GB memory.

Datasets. We experiment each scenario with four different datasets, two synthetic and two real-world data from different kinds of applications.

- Uniform: Synthetic dataset with all attribute values sampled uniformly. The values in the attributes' domain have roughly the same frequency.
- Normal: Synthetic dataset with attribute values taken from the normal distribution. The distribution is set to cover all the domains of each attribute. The mean is the middle value of the attribute's domain. In this dataset, we experiment with how the strategies behave with more skewed data distribution.
- Ipums (RUGGLES *et al.*, 2022): Dataset consists of US census microdata samples from 2014 to 2018. It has information about individuals, such as age, income, and education. We sample 10 million user records with ten attributes (5 categorical and 5 numerical) with different distributions.
- Loan (KAGGLE, 2019): Dataset from the Lending Club, a peer-to-peer lending company. The dataset includes information about accepted loans and attributes such as credit scores, income, and interest rates. In this dataset, we sample 2 million (from a total of 2.2 million) user records with ten attributes (5 categorical and 5 numerical) with different distributions.

7.1.1 Evaluation Scenarios

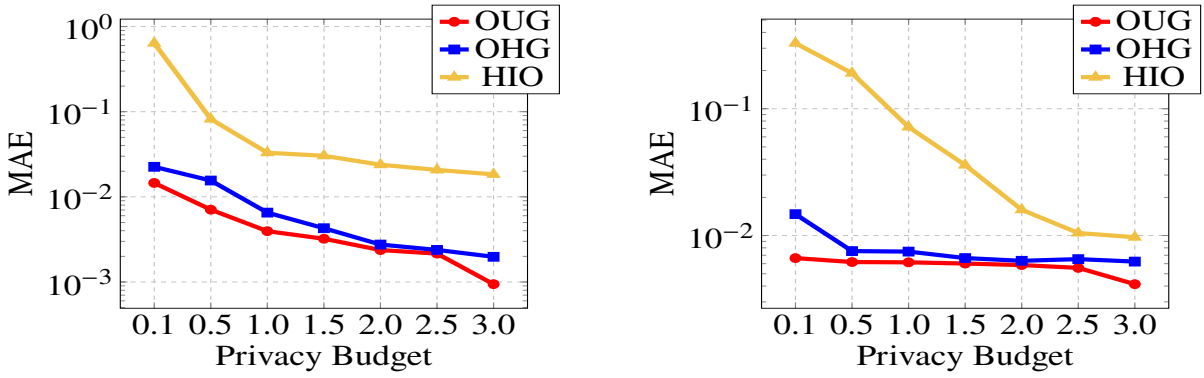
We vary the privacy budget ϵ, the query selectivity s, the number of attributes $|A|$, the domain d of attributes, the query dimension λ, and the number of users n. To experiment with different numbers of users, we generate multiple test datasets from the two synthetic datasets with the number of users ranging from 100k to 1000k. For evaluation varying different numbers of attributes and domain sizes, we generate multiple versions of these four datasets with the number of attributes ranging from 3 to 10, their domain sizes range from 2^4 to 2^{10}.

To the best of our knowledge, the most recent work to support queries with range and point constraints in the local setting, which is the focus of our work, is HIO (WANG *et al.*, 2019a); thus we compare extensively to HIO. We set the branching factor $b = 4$. We set $\alpha_1 = 0.7$ and $\alpha_2 = 0.03$. To evaluate the performance of our approach, we randomly generate a set Q of λ-D queries and calculate their MAE. We generate queries with different selectivity, denoted by s, which means the ratio of the specified interval to the domain size for each queried numerical attribute. In all subsequent experiments, unless explicitly stated, the privacy budget has default value $\epsilon = 1.0$, the query selectivity $s = 0.5$, the number of attributes $d = 6$, the domain of each numerical attribute $d = 100$, the domain of each categorical attribute $|A| = 6$, the number of users $n = 10^6$, the query dimensions $\lambda = 2, 4$ and $|Q| = 10$.

7.1.2 Privacy budget

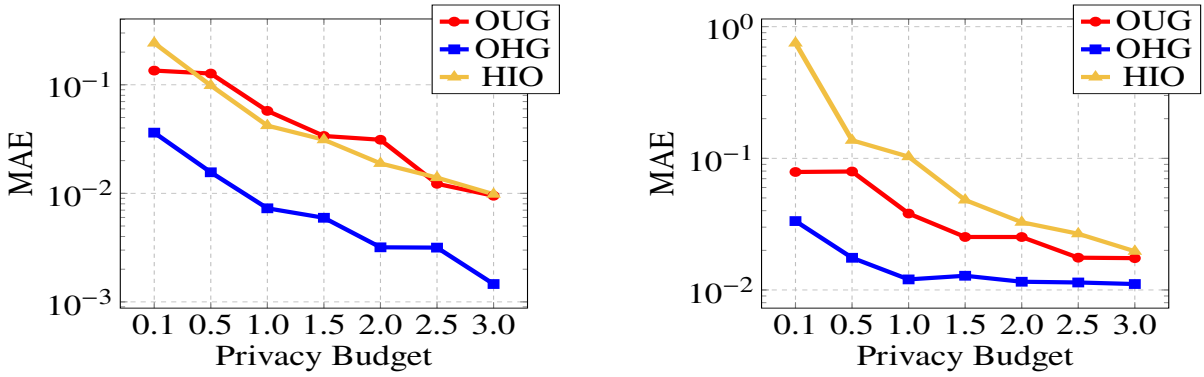
Figure 27 shows the results when varying the privacy budget ϵ. The strategy OUG performs better, as expected, in the uniform dataset, achieving better results than even OHG, especially for larger ϵ. In all other datasets, OHG is superior. Due to its refined estimation, which uses auxiliary 1-D grids, it obtains the best utility among all approaches. HIO had the larger MAE in all datasets.

Figure 27 presents the experimental results under varying values of the privacy budget parameter ϵ. As anticipated, the OUG strategy achieves the best performance in the Uniform dataset, outperforming even the more sophisticated OHG approach. This is primarily due to its alignment with the uniformity assumption underlying the method, which allows it to exploit the data distribution more effectively. In contrast, across all other datasets, OHG consistently demonstrates superior utility. This can be attributed to its refined estimation procedure, which leverages auxiliary one-dimensional grids to better the underlying distribution of the data. The



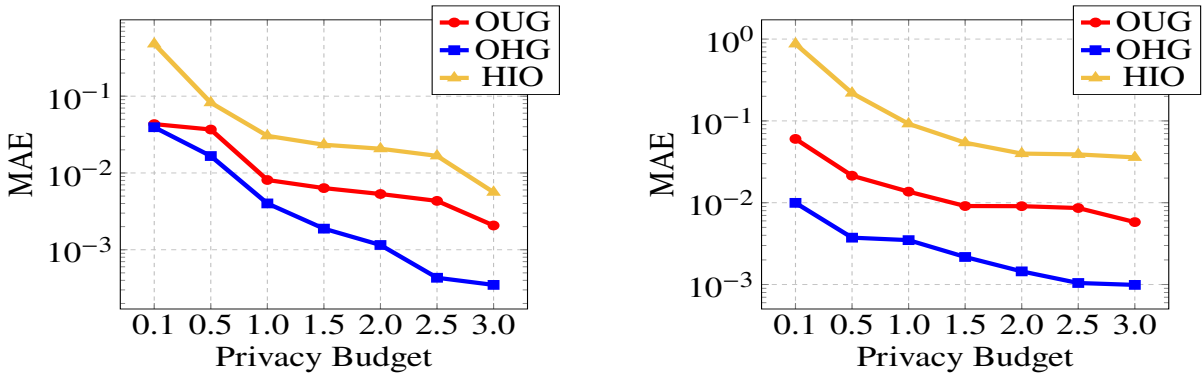
(a) Uniform $\lambda = 2$

(b) Uniform $\lambda = 4$



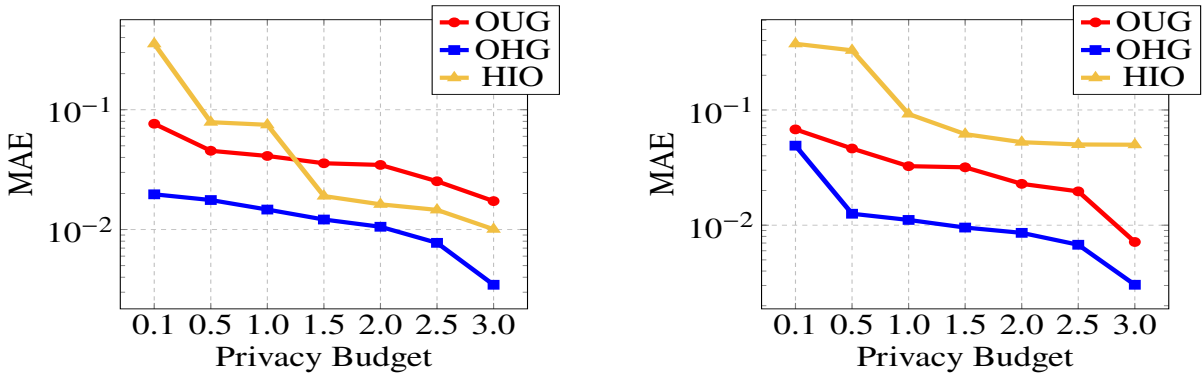
(c) Normal $\lambda = 2$

(d) Normal $\lambda = 4$



(e) Ipums $\lambda = 2$

(f) Ipums $\lambda = 4$



(g) Loan $\lambda = 2$

(h) Loan $\lambda = 4$

Figure 27 – Results on different datasets varying the privacy budget ϵ.

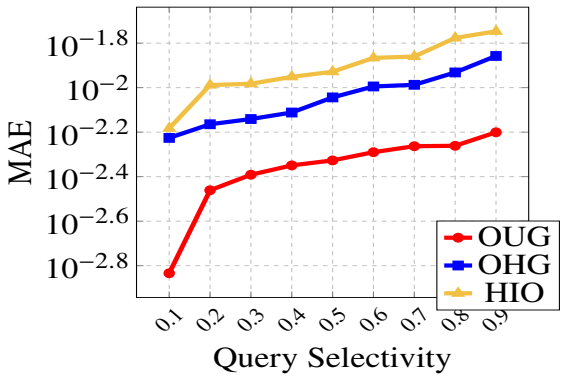
HIO method yields the highest Mean Absolute Error (MAE) across all evaluated datasets. This is primarily due to its use of a large number of sub-intervals, which causes the privacy budget to be distributed over a greater number of partitions, thereby increasing the amount of noise added to each estimate.

7.1.3 Query Selectivity

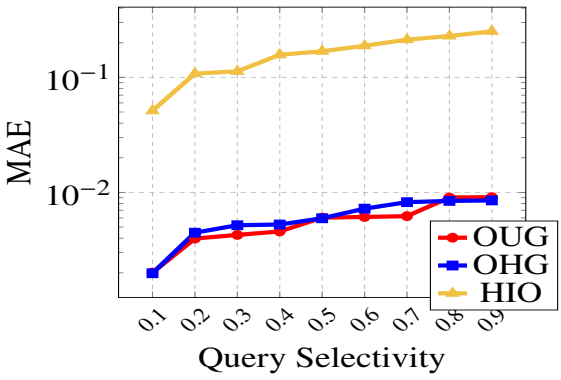
In this experiment, we investigate the impact of query selectivity on utility. Specifically, we vary the selectivity of the query by adjusting the fraction of the domain it targets, beginning with queries that cover 10% of the domain and progressively increasing up to 90%. The results are presented in Figure 28. As the selectivity increases, *i.e.* as the query becomes broader, the number of domain cells included in the answer also grows. This effect is particularly evident in the OHG and OUG strategies, where the query response aggregates over a larger set of cells. Since local differential privacy mechanisms inject noise independently into each cell to ensure privacy, the cumulative effect of this noise increases with the number of cells. Consequently, the overall estimation error rises as the query becomes less selective. Across all selectivity levels, both OHG and OUG consistently outperform HIO, which exhibits higher MAE. Notably, OUG shows superior utility on uniform datasets, particularly when the domain granularity parameter is set to $\lambda = 2$, reflecting its advantage in settings with homogeneous distributions. In contrast, for all other scenarios involving varying selectivity, OHG delivers the most accurate results, benefiting from its finer-grained estimation approach.

7.1.4 Domain size

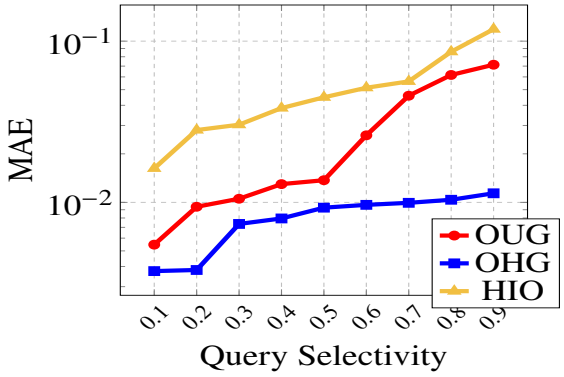
Figure 29 presents the results of varying the domain size of all attributes involved in the query. In this experiment, numerical attributes are scaled from 25 up to 1600 distinct values, while categorical attributes range from 2 to 8 categories. This variation allows us to evaluate the robustness of each method with respect to increasing domain complexity.The results indicate that the performance of both OUG and OHG remains relatively stable as the domain size increases. While minor fluctuations in error can be observed, no consistent upward or downward trend is evident across datasets, suggesting that these methods are largely insensitive to domain size within the tested range. This robustness can be attributed to their ability to adaptively allocate privacy budget and structure their estimations efficiently across varying domain sizes. In contrast, the HIO method exhibits a clear and substantial increase in error as the domain size



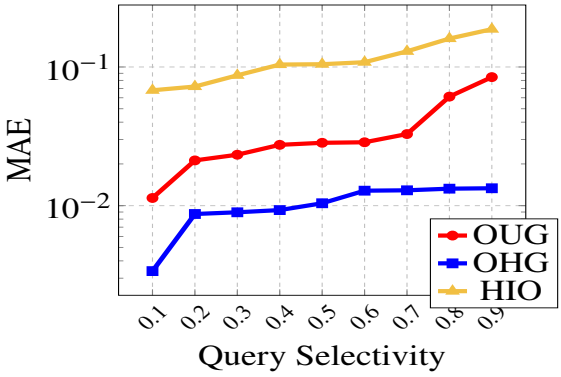
(a) Uniform $\lambda = 2$



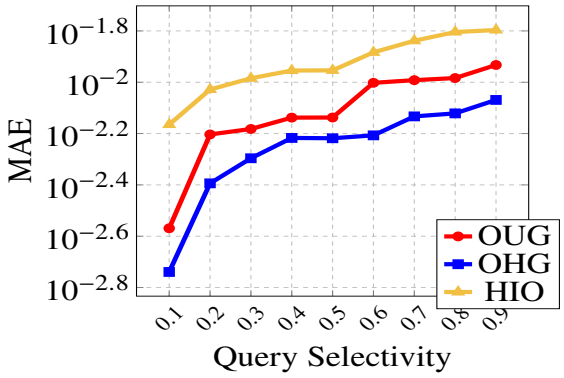
(b) Uniform $\lambda = 4$



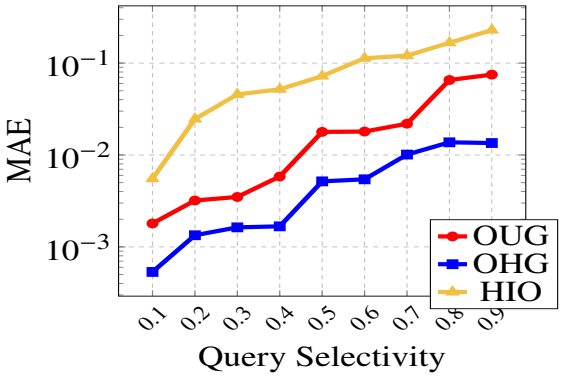
(c) Normal $\lambda = 2$



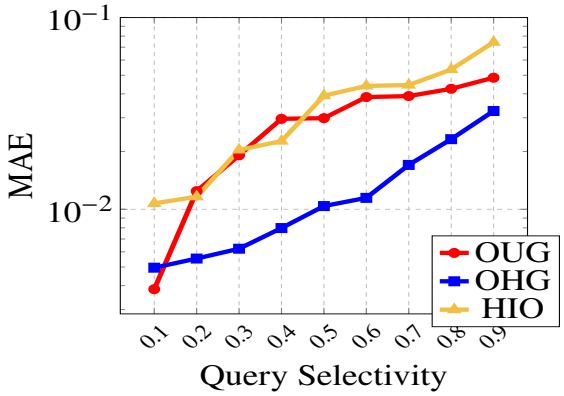
(d) Normal $\lambda = 4$



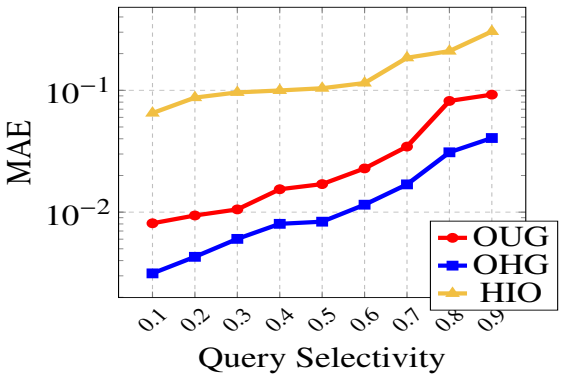
(e) Ipums $\lambda = 2$



(f) Ipums $\lambda = 4$

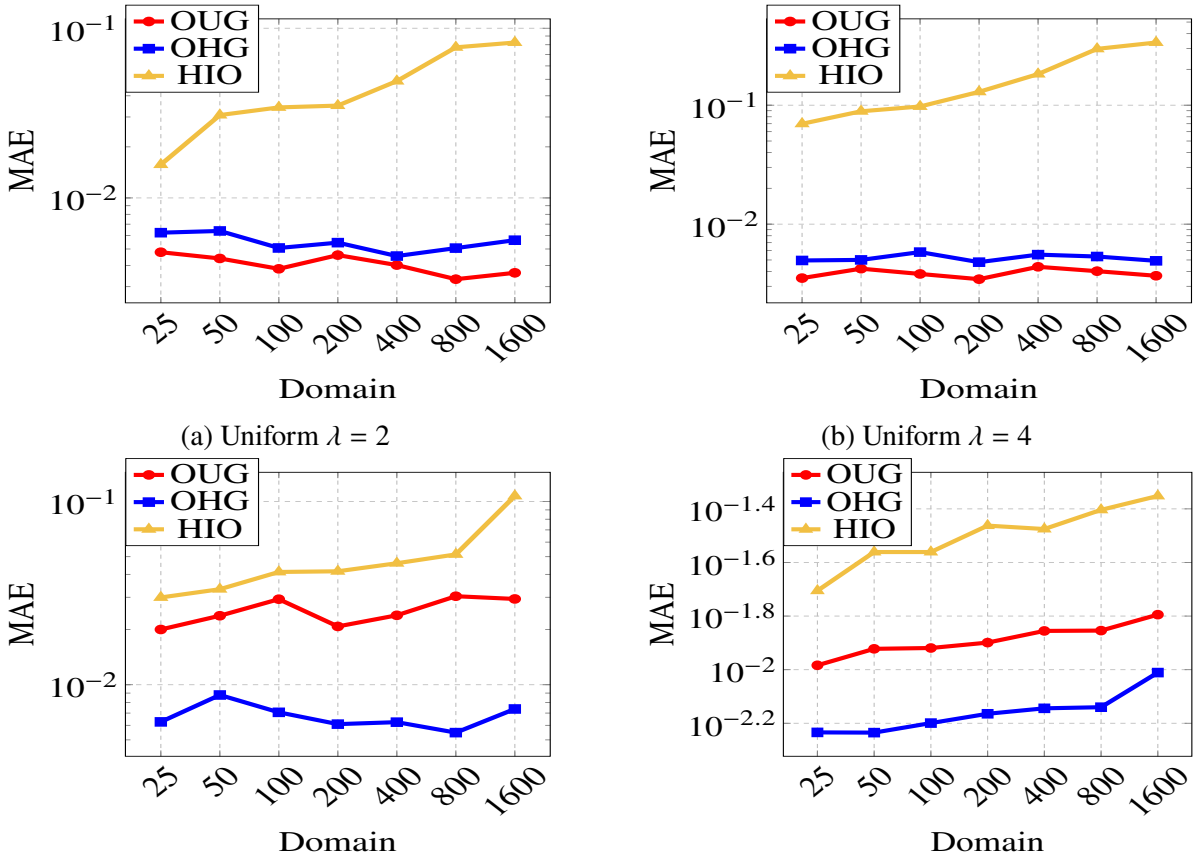


(g) Loan $\lambda = 2$



(h) Loan $\lambda = 4$

Figure 28 – Results on different datasets varying the query selectivity s.



(a) Uniform $\lambda = 2$

(b) Uniform $\lambda = 4$

(c) Normal $\lambda = 2$

(d) Normal $\lambda = 4$

Figure 29 – Results on different datasets varying the attributes' domain d.

grows. This behavior is due to the privacy budget that must be distributed over a greater number of sub-intervals, thereby increasing the amount of noise introduced per estimate and degrading overall utility.

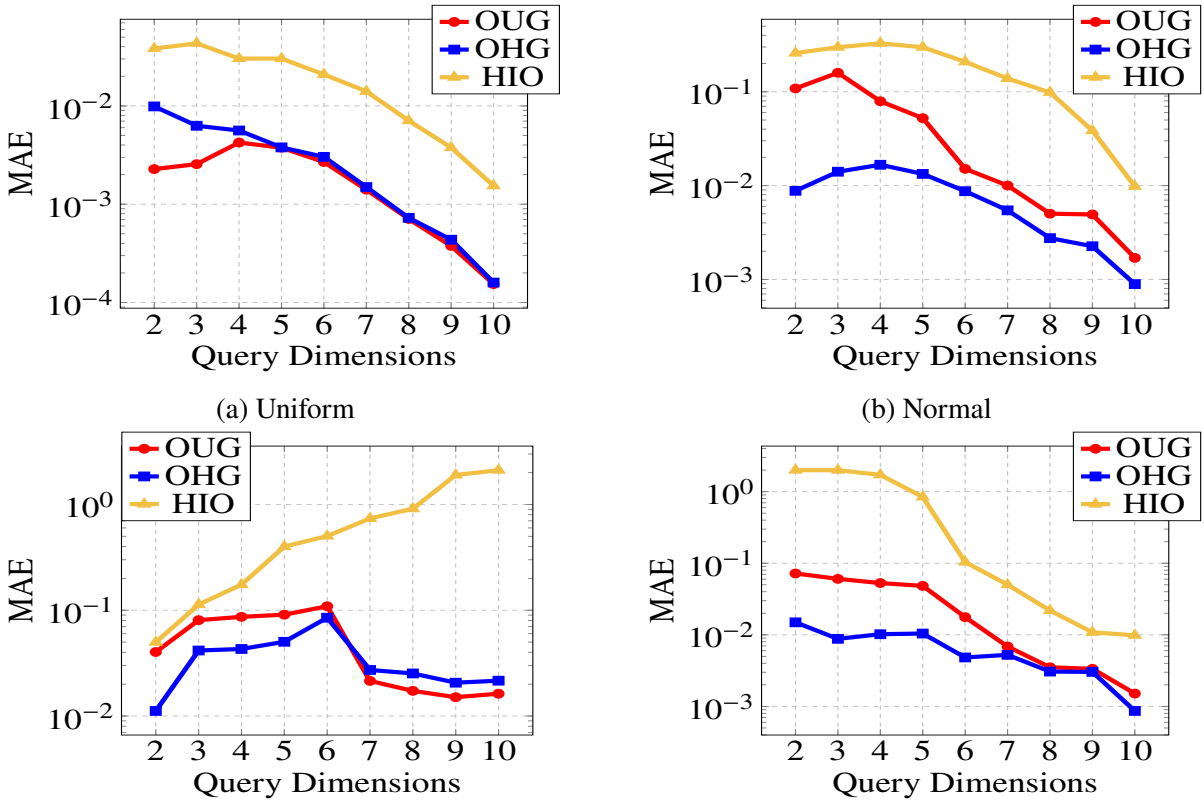
Across all domain configurations, OHG consistently outperforms other methods, with the exception of the *Uniform* dataset. In this case, OUG demonstrates a slight advantage, benefiting from its assumption of uniformity, which aligns well with the underlying data distribution. Overall, these results highlight the scalability and accuracy of OHG and OUG under increasing domain complexity, in contrast to the performance degradation observed in HIO.

7.1.5 Query Dimension

Figure 30 illustrates the behavior of the estimation error as the number of query dimensions increases. We evaluate query workloads ranging from 2 to 10 dimensions, following the same experimental protocol adopted in prior work (YANG *et al.*, 2020). This setup allows for a consistent comparison with established baselines while also enabling the study of how high-dimensional conjunctions impact utility under local differential privacy. As the number of

dimensions in the query increases, the predicate conjunction becomes more restrictive, resulting in fewer users satisfying all conditions simultaneously. Consequently, the true frequency of the query answer tends to decrease, approaching zero. This trend is observed uniformly across datasets. In addition, due to the application of post-processing steps (such as non-negativity and consistency constraints), the estimated frequencies also converge toward zero under high dimensionality. Since both the true value and the privatized estimate become small, the absolute error diminishes, and thus the MAE tends to decrease for very high-dimensional queries.

A more nuanced behavior is observed in the IPUMS dataset, as shown in Figure 30c. In this case, the error initially increases as the number of dimensions grows from 2 to 6. This is attributed to the fact that, within this range, there remains a non-negligible number of users who satisfy the query predicates, leading to moderate true counts. However, beyond 6 dimensions (*i.e.*, for 7 to 10 predicates), the query becomes extremely selective, yielding very small true frequencies. At this point, both the true and estimated values approach zero, and the error starts to decrease accordingly.



(a) Uniform

(b) Normal

(c) Ipums

(d) Loan

Figure 30 – Results on different datasets varying the number of query dimensions λ.

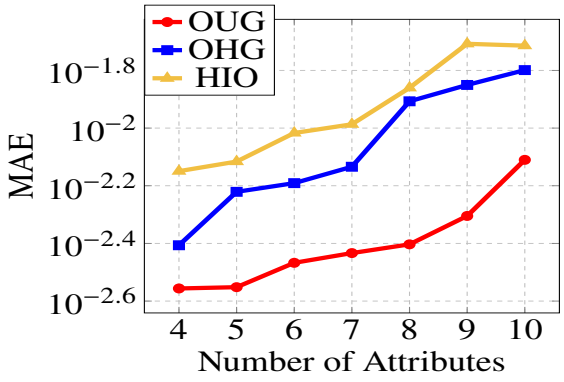
7.1.6 Number of attributes

Figure 31 presents the results obtained when varying the number of attributes in the dataset, ranging from 4 to 10. This experiment is designed to evaluate how the dimensionality of the data ,*i.e.* the number of attributes collected from each user, impacts the utility of LDP techniques. Across all strategies, we observe a clear trend: the estimation error increases as the number of attributes grows. This degradation in utility can be attributed to the combinatorial explosion in the domain size. As the number of attributes increases, the number of possible attribute combinations grows exponentially, causing the total population to be distributed across an increasingly large number of groups. Consequently, the expected number of users per group declines, reducing the statistical reliability of frequency estimates and increasing the relative impact of noise.

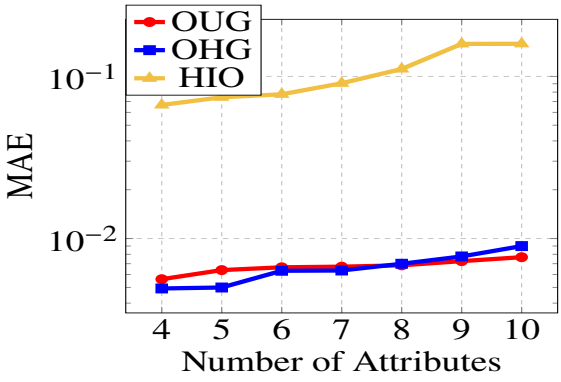
Among the evaluated methods, HIO consistently exhibits the highest MAE across all configurations. This is largely due to the fact that the number of users reporting each sub-interval decreases rapidly when the number of dimensions increases. In contrast, OUG continues to perform well on datasets with uniform distributions, consistent with earlier experiments. OHG, however, achieves the best overall performance on all non-uniform datasets. Its adaptive grid-based approach enables it to more effectively capture the underlying structure of high-dimensional data, allocating privacy budget more efficiently and improving utility even in sparse domains.

7.1.7 Number of Users

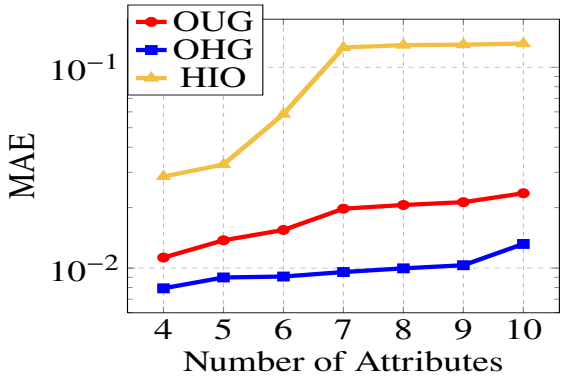
Figure 32 reports the results of varying the total population size n, which corresponds to the number of users participating in the data collection process. In the Loan dataset, the population size is varied from 10,000 to 1,000,000, while in all other datasets, n ranges from 100,000 to 10,000,000. This experiment is designed to assess how the scale of user participation affects the utility of different local differential privacy (LDP) strategies. As expected, increasing the population size results in a consistent improvement in utility across all methods. A larger population leads to a higher number of users per group (i.e., grid). This, in turn, reduces the variance in frequency estimates and mitigates the relative impact of noise introduced by the privacy mechanism. Among the evaluated methods, OUG demonstrates superior performance on the uniformly distributed dataset. In contrast, HIO consistently exhibits lower utility, with substantially MAE values compared to the optimized grid-based approaches. OHG achieves the



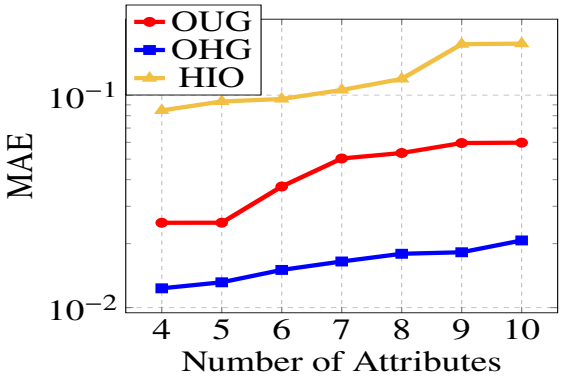
(a) Uniform $\lambda = 2$



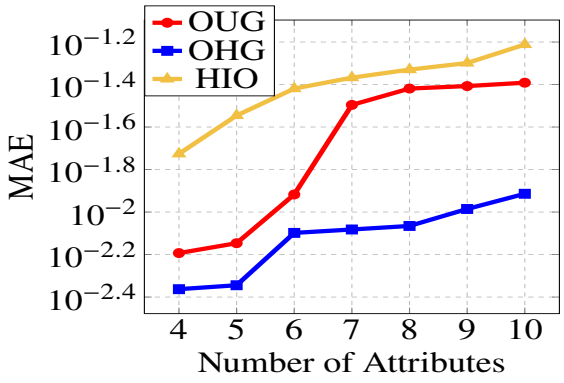
(b) Uniform $\lambda = 4$



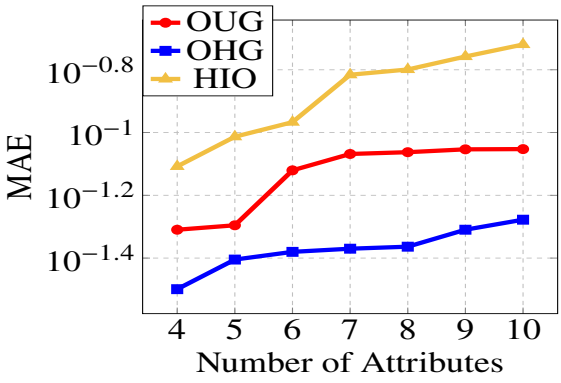
(c) Normal $\lambda = 2$



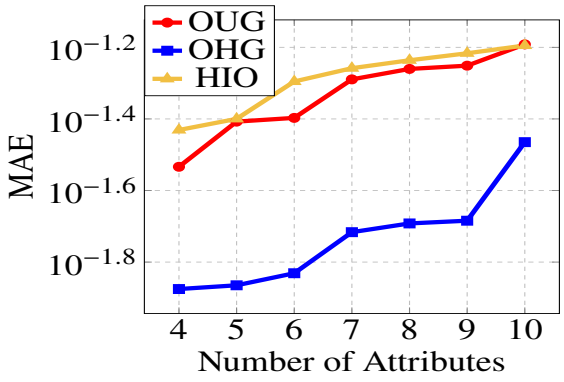
(d) Normal $\lambda = 4$



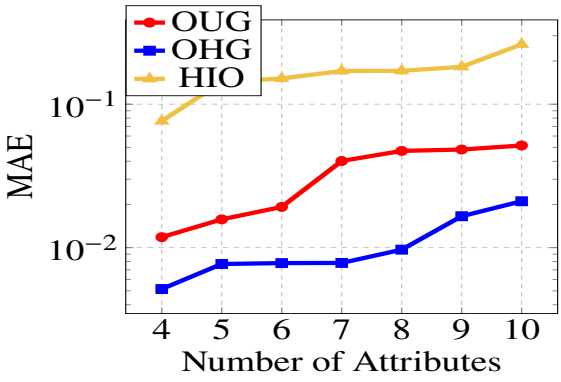
(e) Ipums $\lambda = 2$



(f) Ipums $\lambda = 4$

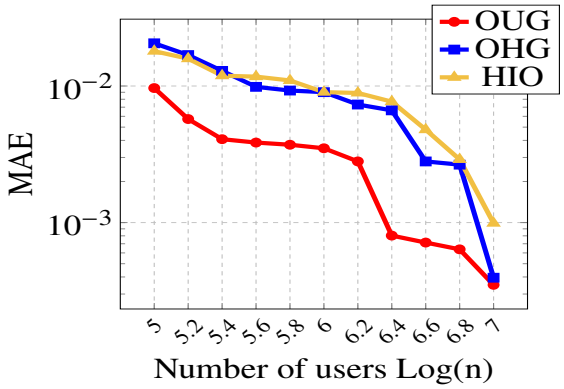


(g) Loan $\lambda = 2$

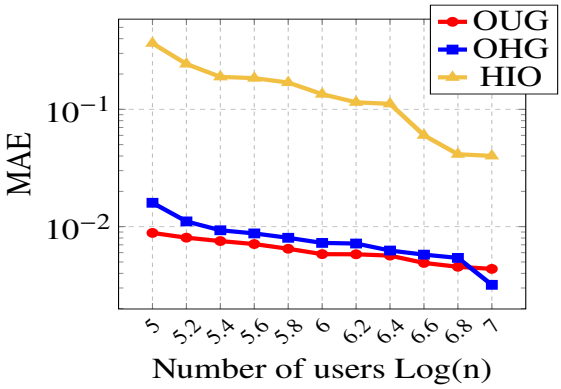


(h) Loan $\lambda = 4$

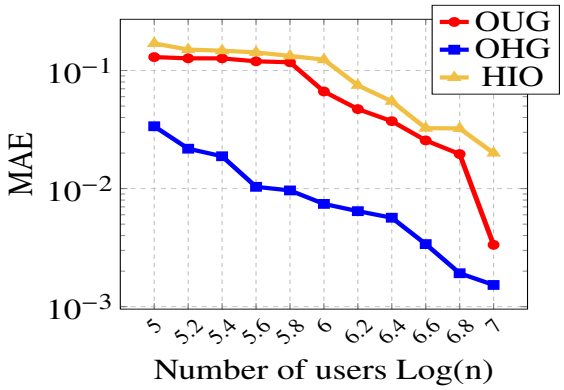
Figure 31 – Results on different datasets varying the number of attributes.



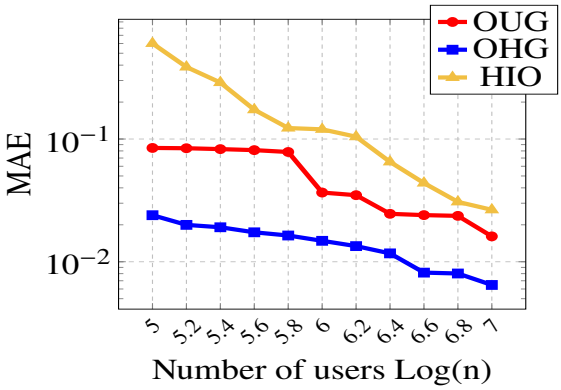
(a) Uniform $\lambda = 2$



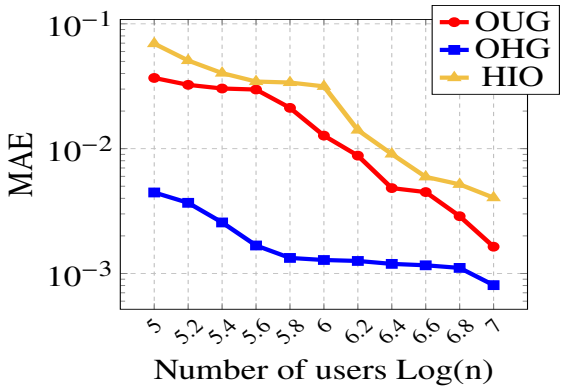
(b) Uniform $\lambda = 4$



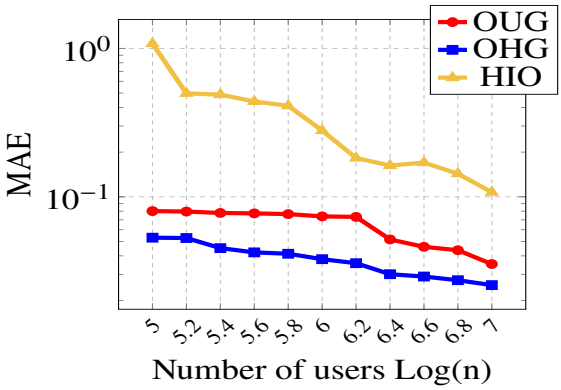
(c) Normal $\lambda = 2$



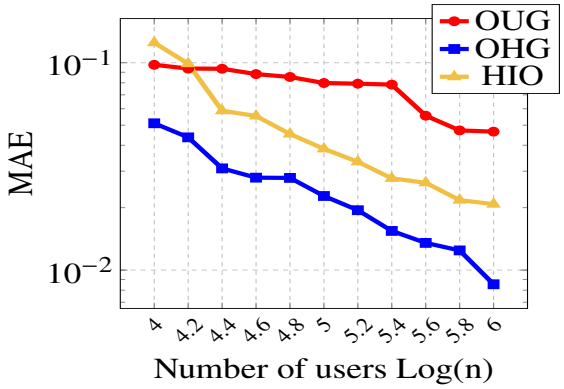
(d) Normal $\lambda = 4$



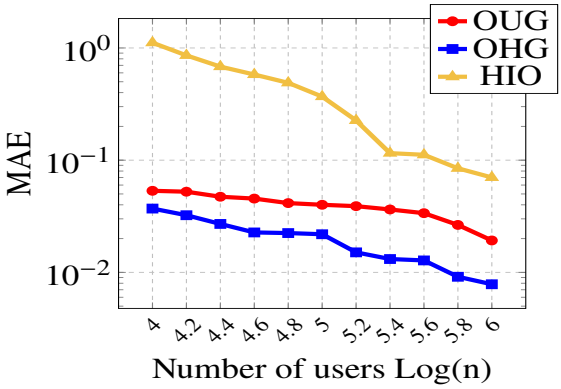
(e) Ipums $\lambda = 2$



(f) Ipums $\lambda = 4$



(g) Loan $\lambda = 2$



(h) Loan $\lambda = 4$

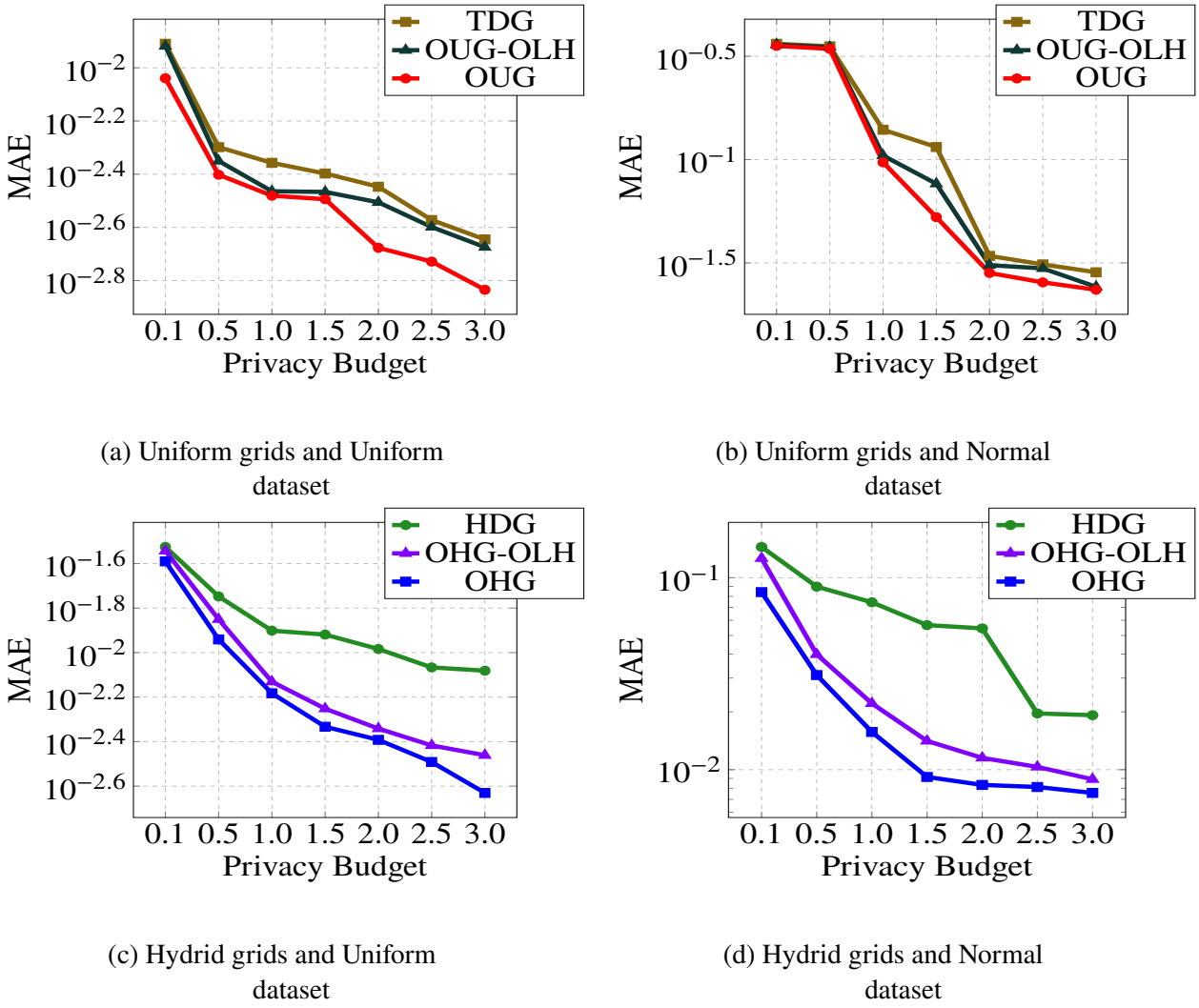
Figure 32 – Results on different datasets varying the number of users Log(n).

best overall performance across all datasets.

7.1.8 Adaptive Protocol Evaluation

In this experiment section, unlike Section 7.1.1, we consider queries with range constraints only, even though FELIP is designed to be a broader solution. We compare our solution to TDG/HDG (YANG *et al.*, 2020). We also evaluate if the adaptive protocol can reduce error in this scenario. So we compare our strategy with and without the adaptive protocol to the baselines TDG and HDG. We call OUG-OLH and OHG-OLH our optimized strategies that use the OLH protocol only. OUG and OHG are the proposed strategies with the adaptive protocol. TDG/HDG is designed around the OLH protocol. In this evaluation, we set the parameter values the same for all strategies. We set $\alpha_1 = 0.7$, $\alpha_2 = 0.03$, the number of users to 1000000, the query selectivity to 0.5, domain size to 100 for all attributes, query dimension to 3, and both datasets (normal, uniform) have six attributes.

Figure 33 (a) and (b) shows the results when using the uniform grid strategies. In both datasets, we notice that even without the adaptive protocol, OUG-OLH still performs better than TDG. That is explained by the fact that we construct more accurate sized grids than TDG. Also, we verify that the best uniform grid strategy is OUG, which chooses the best protocol for each grid. We can see that the non-uniformity error is an essential aspect since all uniform grid strategies have a more significant error on the normal dataset when compared to the uniform dataset. Figure 33 (c) and (d) shows the results when comparing the hybrid grid strategies. In this experiment, we note a more significant difference between our proposed approach OHG and the baseline HDG in both uniform and normal datasets. That is because, in hybrid grids, we are constructing more accurate sized 2-D grids and optimized 1-D grids which significantly impact the overall utility. The OHG-OLH still has higher utility when compared to HDG, but OHG that has the adaptive protocol achieves the best result among all strategies.



(a) Uniform grids and Uniform dataset

(b) Uniform grids and Normal dataset

(c) Hydrid grids and Uniform dataset

(d) Hydrid grids and Normal dataset

Figure 33 – Results from queries with range constraints only when varying the privacy budget.

7.2 *PrivMDC* - Evaluation

We evaluate PrivMDC approach across various settings and datasets, comparing it to competitor methods. Experiments were implemented in Python 3.10, and run on an Ubuntu 20.04 server with an Intel Core i7-7820X and 128GB RAM.

Datasets. We use four real datasets with varying user counts, attribute domain sizes, and correlation levels. Adult and Bfive have lower attribute covariance, while Ipums and Loan exhibit higher correlations. These datasets were also used in prior studies (WANG *et al.*, 2023; FILHO; MACHADO, 2023; YANG *et al.*, 2020; WANG *et al.*, 2019a).

- Adult (BECKER; KOHAVI, 1996): The dataset is from the UCI ML repository with about 33 thousand records.
- Bfive (KAGGLE, 2019): The dataset is collected by an interactive online test platform that describes the time spent on each problem in milliseconds, with around 800 thousand

records.

- Ipums (RUGGLES *et al.*, 2022): The dataset consists of US census microdata samples from 2014 to 2018. It has information about individuals, such as age, income, and education. Our sample has almost 1 million user records.
- Loan (KAGGLE, 2019): The dataset is from the Lending Club, a peer-to-peer lending company. The dataset includes information about accepted loans and attributes such as credit scores, income, and interest rates. In this dataset, we sample 2.2 million user records.

7.2.1 Competitors & Evaluation Scenarios

We compare our approach to prior work including FELIP (FILHO; MACHADO, 2023), HDG (YANG *et al.*, 2020) and PrivNUD (WANG *et al.*, 2023). As in (YANG *et al.*, 2020), We configure HDG with $\alpha_1 = 0.7$ and $\alpha_2 = 0.03$. HIO has been shown to perform worse than FELIP and HDG, so we omitted it due to high MAEs to highlight differences among stronger approaches. To evaluate our approach, we generate 100 random queries per scenario with varying dimensions and selectivity, then compute their MAE. Selected dimensions follow a normal distribution with mean $|A|/2$ and a standard deviation ensuring all appear at least once. Selectivity s represents the interval-to-domain ratio for numerical attributes. Unless stated otherwise, the DP group is sampled from the LDP group using the *Default* KL-divergence level (Table 4). Following prior work (KOHEN; SHEFFET, 2022; AVENT *et al.*, 2017; AVENT *et al.*, 2020; BEIMEL *et al.*, 2020) where $u_{dp} \ll u_{ldp}$, the DP group is set to 1% of the total population. We use the Mean Absolute Error (MAE) to measure the accuracy of estimated answers. The error is shown in the y axis which is in the log scale in all experiments.

7.2.2 Privacy budget

Figure 34 illustrates the performance of our approach and its competitors on four real datasets, considering 12 attributes and various privacy budgets. We vary the privacy budget ϵ between 0.1 to 2.0 and find the MAE. We choose to experiment with ϵ values under 1.0 for evaluation purposes only. In many real-world applications, ϵ values are typically larger (BUREAU, 2021a; RESEARCH, 2023). As expected, the higher the ϵ the lower the estimation errors in all strategies. PrivMDC was able to identify the strongly correlated attributes, especially for larger values of ϵ. Therefore, PrivMDC was able to materialize fewer grids in all datasets. Also, by optimizing the user distribution, PrivMDC successfully allocated more users to the

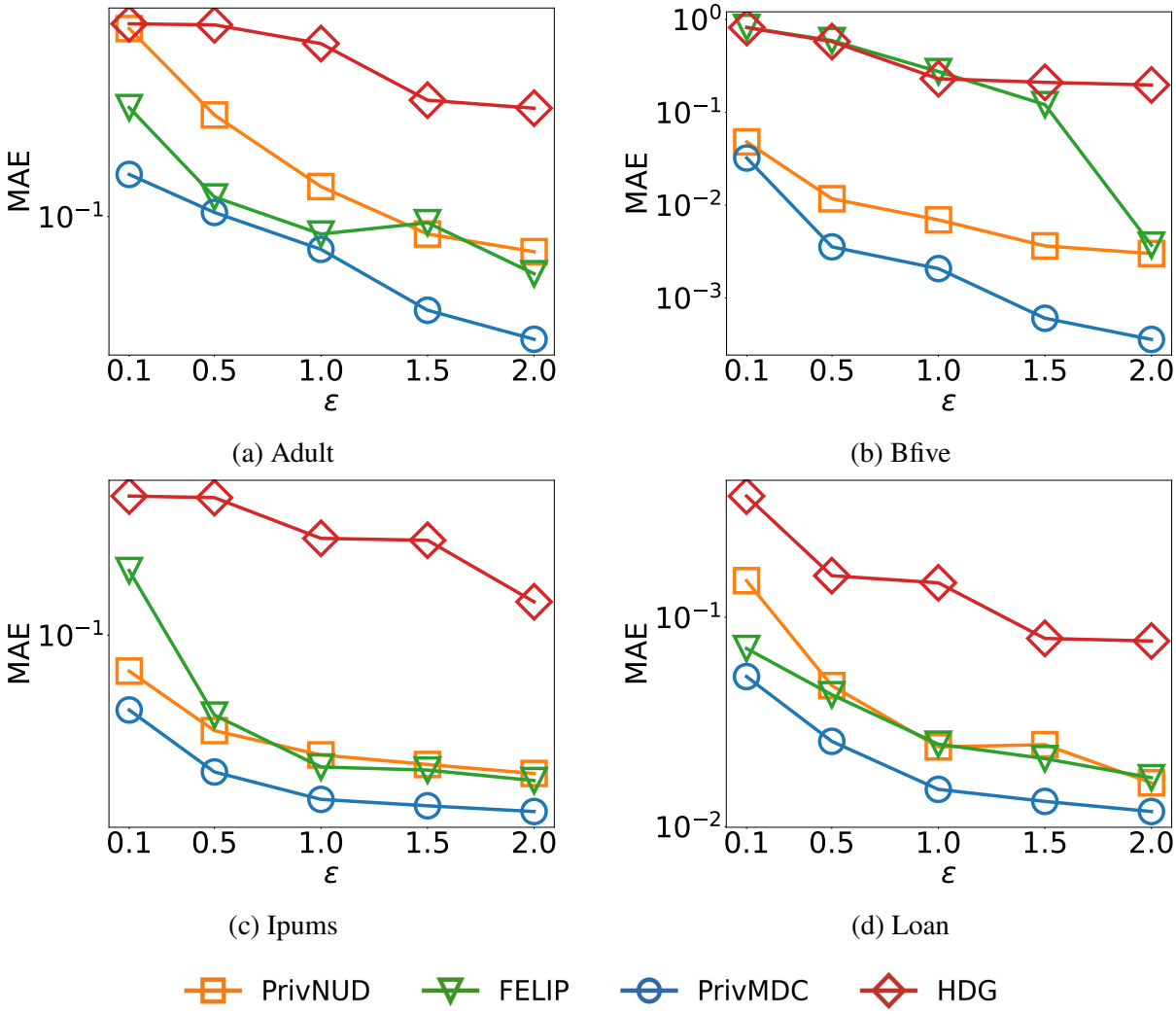


Figure 34 – Result MAE on real datasets containing 12 attributes for varying privacy budget

more important grids. That phenomenon is more pronounced in the Adult and Bfive datasets for privacy budgets 1.5 and 2.0. For instance, in the Adult dataset with $\epsilon = 2.0$, our approach materialized only 26 grids (2 4-D, 3 3-D, 9 2-D, and 12 1-D) instead of 78 (66 2-D and 12 1-D) as all the competitors, achieving 2x better utility compared to the best competitor. The utility of a grid for answering queries increases with the number of users reporting it. We noticed that sometimes FELIP's error curve changes suddenly and that is because it makes more adjustments to the grid's sizes with different ranges of ϵ. In terms of time performance, our algorithm takes on average 34, 47, 49 and 67 seconds to execute on the Adult, Bfive, Ipums and Loan datasets, respectively.

7.2.3 Scalability

This experiment evaluates PrivMDC without its Correlation Identification module, enabling direct comparison with baseline methods that do not incorporate correlation models (FILHO; MACHADO, 2023; YANG *et al.*, 2020; WANG *et al.*, 2023). For fairness, only 99% of the total population is used. The utility of all strategies is evaluated as the number of attributes in the dataset increases, up to the dataset-specific maximum. A set of 100 random queries is generated, and the privacy budget is fixed at $\epsilon = 2.0$ for all datasets. As shown in Figure 35, the estimation error increases with the number of attributes across all methods, due to the need to materialize $\binom{k}{2} + k$ grids. PrivMDC consistently outperforms the baselines, with its utility advantage growing as the dimensionality increases. This is attributed to its ability to optimize user distribution by allocating more users to frequently queried, high-utility grids. In PrivMDC, 1-D grids receive more user reports, as they contribute to both uncorrelated attribute estimation and enhancement of multi-dimensional grids via response matrices. However, in low-dimensional datasets (*e.g.*, with 3 attributes), baselines such as PrivNUD may perform better due to more efficient 1-D decompositions and stronger signal. In addition, with fewer dimensions, PrivMDC tends to build a similar number of grids as the baselines, reducing its utility advantage.

7.2.4 Workloads

In this section, we study how PrivMDC can deal with workloads with different characteristics. This affects the configuration of grids and the user distribution among groups. We added another competitor called *PrivMDC no opt* that is the PrivMDC with the Workload Analyzer and User Distribution Optimization disabled. Without any a priori information about the workload, we have to adopt the uniform user distribution. We observed that turning off the proposed optimization significantly increased the error when querying the Adult dataset. Since this dataset contains fewer than 33k users, the grids frequently used for query responses had a limited number of users contributing data. In datasets with more users, the error grows but in a smaller scale.

In the first workload, 60% of queries are 3-D, while 40% have random dimensions. Applied to the Adult dataset (Figure 36a), PrivMDC achieved the best results, outperforming the closest competitor by up to 59%. Despite the dataset's smaller user count, PrivMDC materialized fewer grids (25 on average), minimizing error. PrivNUD also outperformed FELIP and HDG for

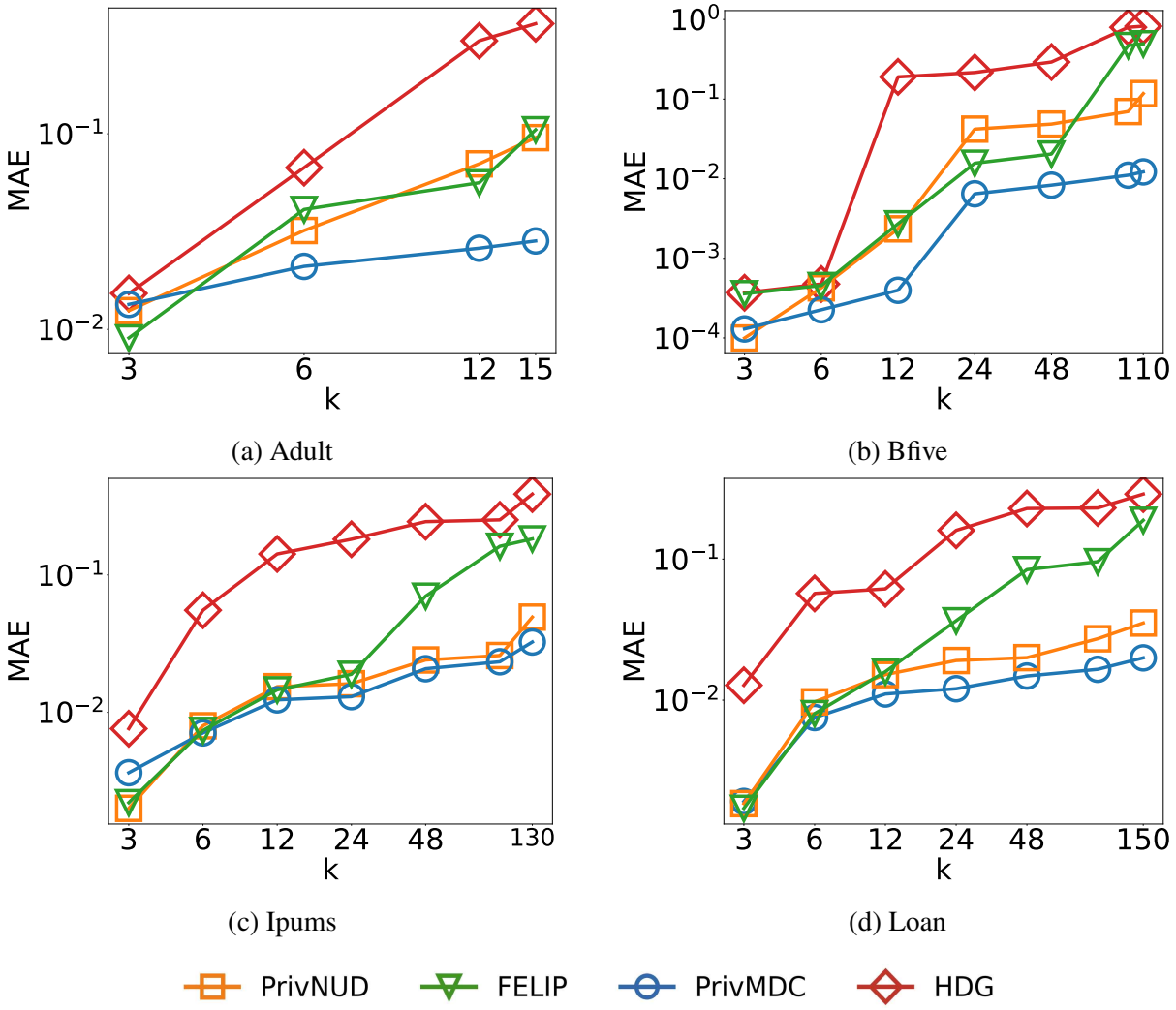


Figure 35 – Result MAE on real datasets while increasing number of attributes k per dataset. Fixed privacy budget at $\epsilon = 2.0$ and disabled the Correlation Identification Module.

larger ϵ, as 1-D grid accuracy dominated in higher-dimensional queries. In the second workload, 70% of queries are 6-D, while the remaining 30% are random λ-D queries with $\lambda \leq 5$. Figure 36b shows the results on the Bfive dataset, where PrivMDC achieved the best utility across all privacy budgets. In 6-D queries, a larger number of grids are queried when estimating the answer. By minimizing materialized grids (averaging fewer than 26), PrivMDC improved accuracy, whereas other approaches required 78 grids with higher noise ratio. In the third workload, 80% of queries are 1-D, and 20% are random λ-D ($\lambda \geq 2$). Figure 36c shows results on the Ipums dataset. PrivMDC allocated, on average, 752,000 of 907,895 users across 12 1-D grids, minimizing error and achieving up to 2x ($\epsilon = 0.1$) utility gain over competitors. PrivNUD performed well due to its accurate non-uniform 1-D decomposition. FELIP and HDG had lower utility, with HDG showing higher MAE due to non-uniformity error and fewer users reporting 1-D grids. In the fourth workload, 90% of queries include 4 correlated attributes in the Loan dataset, while the

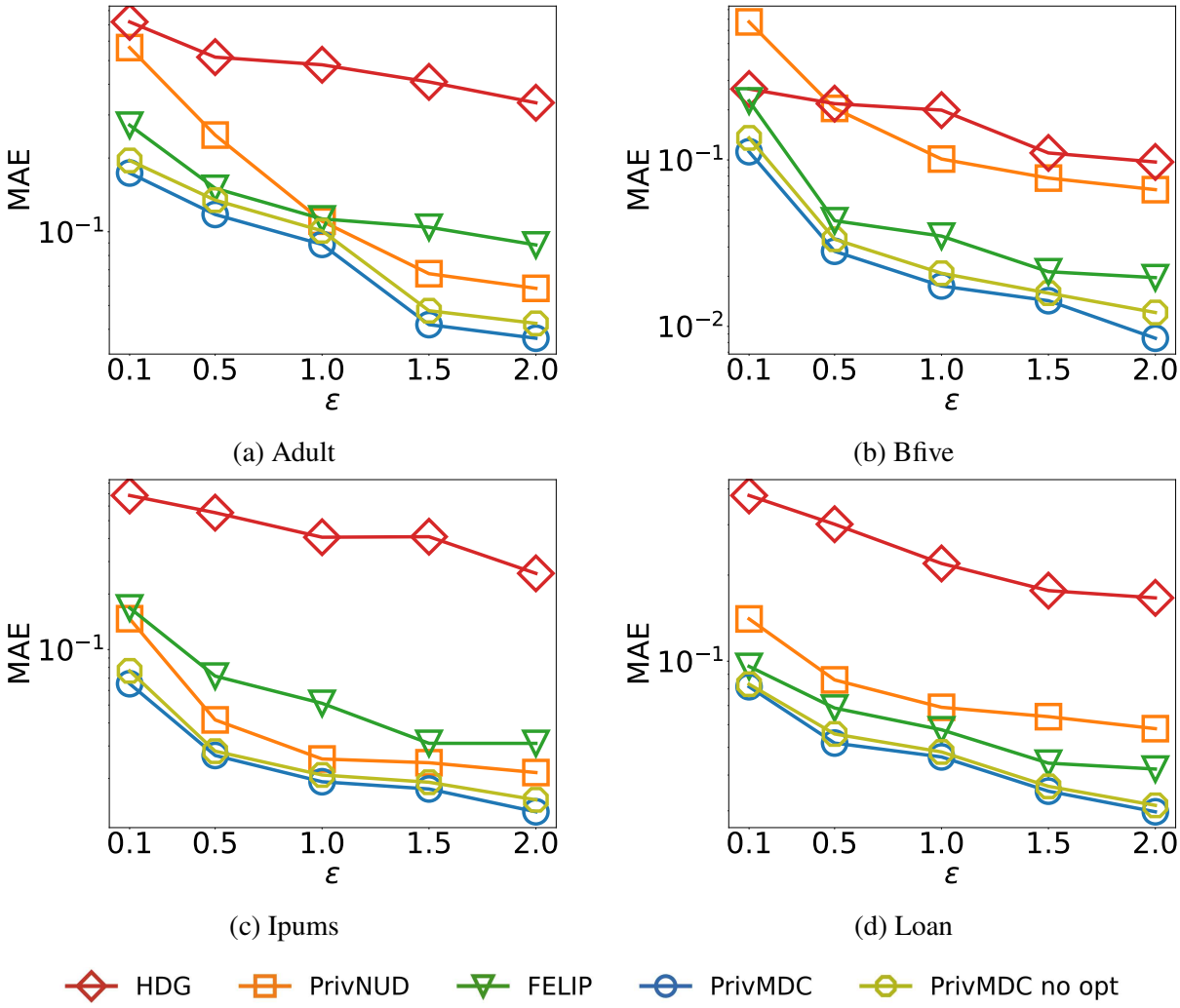


Figure 36 – Result MAE for four different workloads varying the privacy budget on real datasets with 12 attributes.

remaining 10% are λ-D ($\lambda \in [1, 12] \setminus \{4\}$). Figure 36d shows that PrivMDC outperformed all strategies, achieving up to 40% gains ($\epsilon = 2.0$) over the best baseline. It effectively identified correlations and allocated more users to high-demand grids, especially for larger $\epsilon = 2.0$. FELIP outperformed PrivNUD and both significantly surpassed HDG.

7.2.5 Size of DP group

We analyze how DP group size impacts utility across four datasets. The DP group of users share true data with a curator to build a correlation model under ϵ-differential privacy. We test DP group sizes of 0.5%, 1%, 2%, 5% and 10% of the dataset, fixing $\epsilon = 2.0$ and maintaining KL-divergence roughly at *Default* levels (Table 4). Figure 37 shows accuracy results for datasets with 12 attributes each. Across all datasets, larger DP groups reduce MAE, with Ipums and Bfive

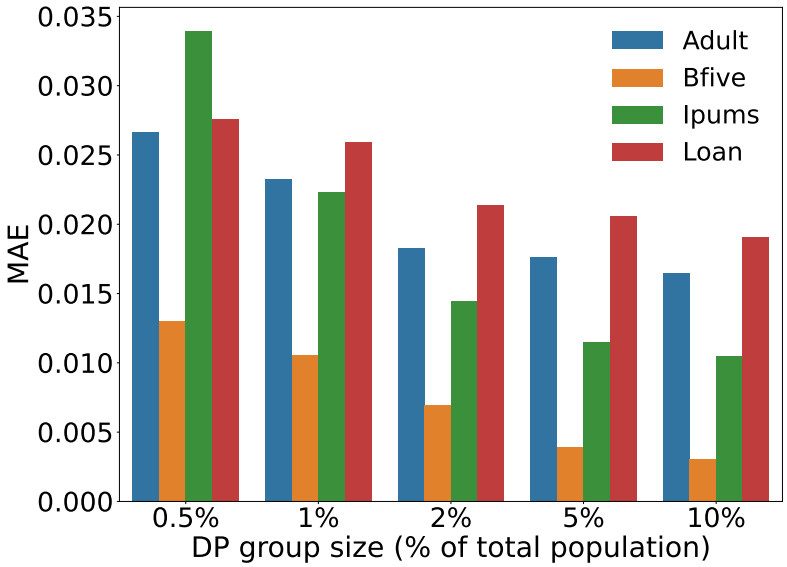


Figure 37 – Result MAE of PrivMDC on all datasets of 12 attributes by varying the DP group size and fixing $\epsilon = 2.0$.

seeing MAE drop by half from 0.5% to 2%. This highlights the strong impact of DP group size on correlation model quality. A larger DP group incurs less noise added in the selection of parent nodes during the Bayesian network construction. Therefore, it improves the correlation model accuracy which, in turn, enhances grid selection and estimation under LDP.

7.2.6 DP and LDP groups' data distribution

In this experiment, we assess the impact of distributional discrepancies between the data used in the DP group and the LDP group on the overall utility of our approach. Specifically, our goal is to understand how differences in statistical properties between these two groups affect the performance of the correlation model, which is built by the DP group and subsequently used to support query estimation for the LDP group. When the DP group distribution deviates significantly from that of the LDP group, the correlation model may capture misleading dependencies that do not generalize to the LDP data. As a result, the quality of the estimates deteriorates, since the model is no longer representative of the population being queried. To systematically quantify the degree of distributional divergence between the DP and LDP groups, we employ the Kullback-Leibler (KL) divergence metric (PEREZ-CRUZ, 2008). KL divergence (in Section 2.11) measures the discrepancy between two probability distributions and is widely used in information theory and statistics. A higher KL divergence value reflects greater dissimilarity, implying that the correlation model trained on the DP group may be increasingly misaligned with the underlying structure of the LDP group.

To construct DP and LDP groups with controlled and measurable divergence in real datasets, we use a k-means clustering-based sampling strategy. Each cluster is assigned a weight, and we manipulate these weights to form samples with varying statistical characteristics. By increasing the representation of samples from clusters farther from the global mean, we introduce skewness, effectively increasing the KL divergence between the sampled groups. This allows us to simulate realistic distribution shifts between DP and LDP groups. Table 4 reports the corresponding KL divergence values for these three configurations across four real-world datasets.

Samples / Dataset	Default	KL-d Mid	KL-d High
Adult	3.766	6.594	13.196
Bfive	2.524	5.280	10.179
Ipums	2.456	5.461	10.061
Loan	2.725	5.313	11.148

Table 4 – Values of KL-divergence for each level and dataset

We define three levels of divergence:

- KL-d High: A sample with maximum observed divergence, obtained by selecting a highly non-uniform cluster weighting that results in the largest KL divergence among all tested configurations.
- KL-d Mid: A distribution with moderate divergence, designed to have approximately half the KL divergence of the most extreme configuration.
- KL-d Default: A baseline sample distribution, designed to have approximately half the KL divergence of the mid configuration.

Figure 38 presents the results, showing the mean absolute error (MAE) of our approach and other baselines under increasing levels of divergence. Across all datasets, we observe that uniform random sampling of the DP group *PrivMDC kld=0* consistently leads to the lowest error, confirming the importance of representativeness in the correlation model's training data. As the KL divergence increases, indicating a growing mismatch between the DP and LDP groups, the error increases accordingly. This trend validates the intuition that building a model on a distribution that does not reflect the target data introduces systematic estimation bias. Nevertheless, even under the most adversarial configuration *KL-d High*, where the DP group's distribution diverges substantially from the LDP group's, our method *PrivMDC* demonstrates competitive results. In several cases, it still outperforms baseline methods that do not rely on

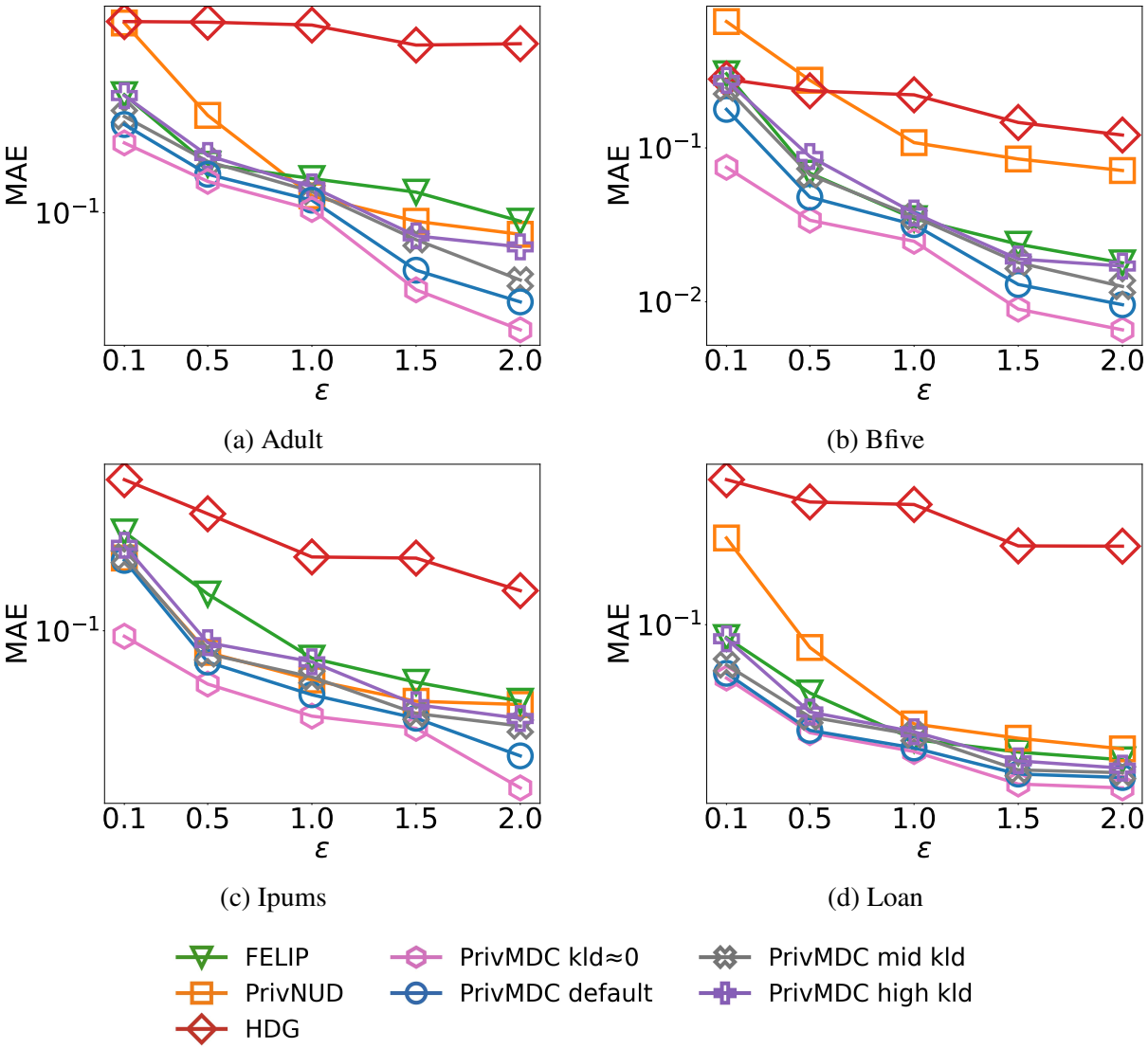


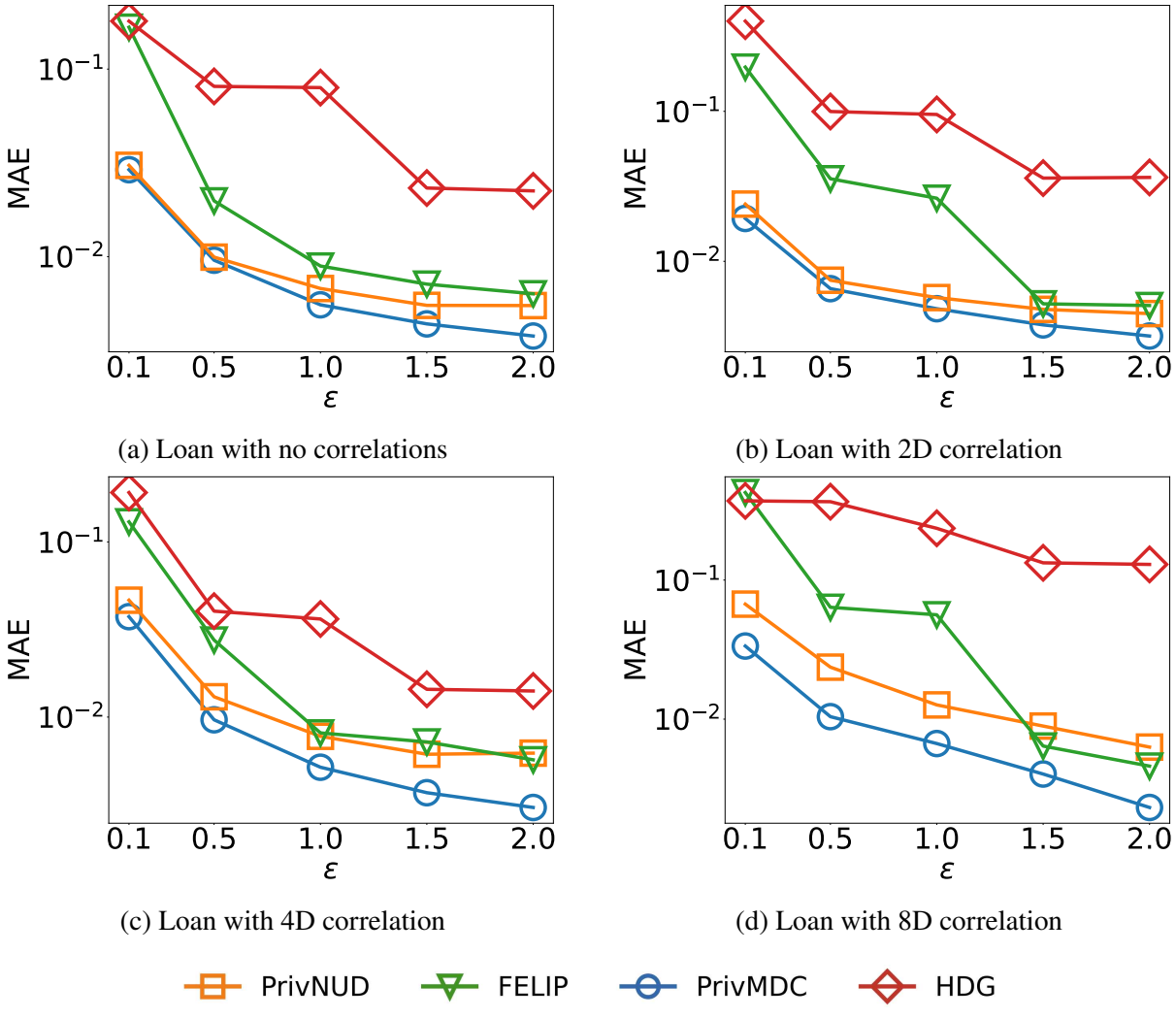
Figure 38 – Result MAE on real datasets with 12 attributes when increasing the KL-divergence among the DP and LDP groups.

correlation modeling. This suggests that while distributional mismatch does degrade utility, the framework remains effective, particularly when the divergence is moderate or the correlation structure is partially preserved.

7.2.7 Different Distributions and Correlations

This section evaluates PrivMDC's performance under varying data distributions and levels of multidimensional correlation using four configurations of the Loan dataset, each containing 12 real attributes. The configurations include: (1) no correlations, (2) 2 correlated and 10 independent attributes, (3) 4 correlated and 8 independent attributes, and (4) 8 correlated and 4 independent attributes. Results in Figure 39 show that PrivMDC consistently outperforms

baselines by leveraging high-dimensional grids to enhance utility, particularly in more correlated settings, as illustrated in Figures 39c and 39d. PrivMDC effectively identifies correlated attribute groups, especially at higher ϵ. However, in scenarios with lower ϵ or more complex correlations (*e.g.*, the 8D case in Figure 39d), it does not detect the full correlation structure but often captures accurate subgroups, maintaining substantial utility. In configurations with weak or no correlations, 1D frequency estimation dominates, making simpler methods like PrivNUD more competitive at low ϵ values. Additionally, in the first two configurations, PrivMDC's detection of non-existent correlations limited its utility gains.



(a) Loan with no correlations

(b) Loan with 2D correlation

(c) Loan with 4D correlation

(d) Loan with 8D correlation

Figure 39 – Result MAE on configurations of the Loan dataset with a group of 0, 2, 4 and 8 correlated attributes.

7.2.8 Naive Bayes attack

In this section, we investigate the extreme case where an attacker possesses all attributes (true data) from users but one sensitive one, and they have gotten access to the LDP reports from PrivMDC. Cormode (CORMODE, 2011) proposed an attack based on a Naïve Bayes (NB) classifier that predicts the value of a sensitive attribute of a tuple with *quasi-identifying* values. We simulate the situation where the attacker wants to learn the *occupation*, *csn10*, *famsize* and *addr_state* from the Adult, Bfive, Ipums and Loan datasets, respectively. A classifier was implemented to generate conditional probabilities given the results of count queries. In the DP setting, the count queries were answered by adding geometric noise based on ϵ. In the LDP setting, the count queries were answered using the materialized grids of PrivMDC using the ϵ-LDP reports.

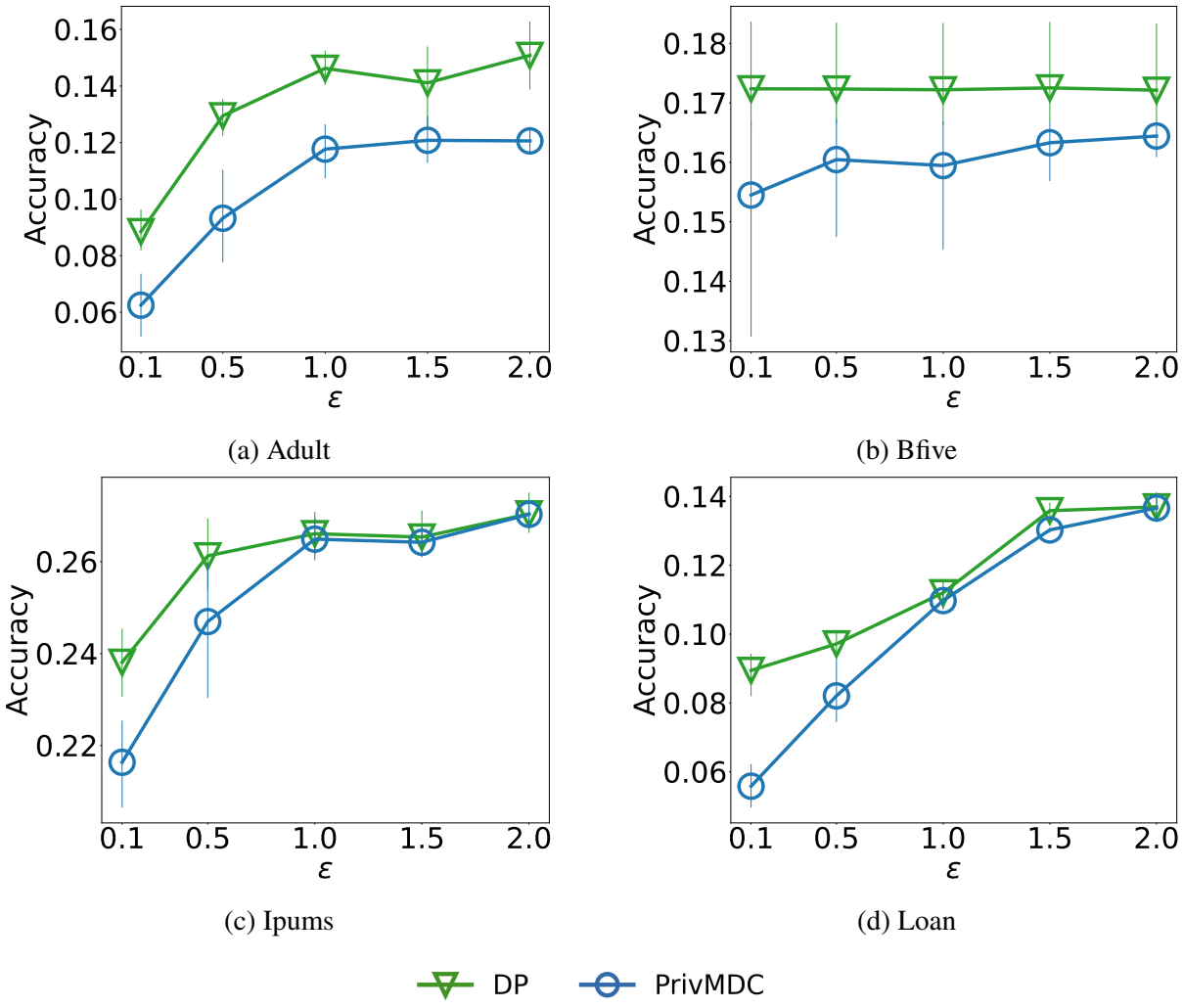


Figure 40 – Accuracy of the learning attack using the Naive Bayes classifier.

As in (CORMODE, 2011), we assessed the classifier's accuracy by sequentially supplying the quasi-identifiers of each input tuple and calculating the proportion of cases where the correct sensitive attribute was predicted. Figure 40 shows the accuracy results for each value of ϵ. Both classifiers, using DP and PrivMDC, achieved similar accuracy values. In most of the situations and especially for larger values of ϵ, the classifier predicted the most common value of attributes which occurs 13%, 35%, 28% and 14% in the Adult, Bfive, Ipums and Loan datasets, respectively. Only for the Bfive dataset, which has domain equal to 6 in all attributes, the accuracy is roughly the same for all privacy budget values. All the other datasets smaller shows a more significant reduction in accuracy for smaller values of ϵ.

7.3 *DP-AE* - Evaluation

In this section, we evaluate the adaptive approach DP-AE proposed in Chapter 6. We design a series of experiments to validate our variance-aware adaptive query answering strategy when applied to the four different datasets:

- Adult (BECKER; KOHAVI, 1996): The dataset is from the UCI ML repository with about 33 thousand records.
- Bfive (KAGGLE, 2019): The dataset is collected by an interactive online test platform that describes the time spent on each problem in milliseconds, with around 800 thousand records.
- Ipums (RUGGLES *et al.*, 2022): The dataset consists of US census microdata samples from 2014 to 2018. It has information about individuals, such as age, income, and education. Our sample has almost 1 million user records.
- Loan (KAGGLE, 2019): The dataset is from the Lending Club, a peer-to-peer lending company. The dataset includes information about accepted loans and attributes such as credit scores, income, and interest rates. In this dataset, we sample 2.2 million user records.

7.3.1 Error metrics

To evaluate the cross-over point c_{p} optimization, we report the Normalized Cumulative Error (NCE) in Section 7.3.3. Let $Q = \{q_1, q_2, \ldots, q_T\}$ denote a workload of T queries. For each query $q_t \in Q$, let $\hat{q}_t$ be the estimated answer and q_t the true answer. The NCE is computed as follows:

$$\text{NCE} = \frac{1}{\sum_{t=1}^{T} |q_t|} \sum_{t=1}^{T} |\hat{q}_t - q_t|. \tag{7.1}$$

This metric captures the aggregate error across all queries in the workload, normalized by the total magnitude of the true answers. For the other experiments in this section, we report the average MAE as in previous experiments.

7.3.2 Competitors & Evaluation Scenarios

We validate the cross-over point c_{p} approximation of our adaptive approach in Section 7.3.3. In Section 7.3.4, we compare our adaptive approach DP-AE to PrivMDC (Chapter 5) and also to prior work including FELIP (FILHO; MACHADO, 2023), HDG (YANG *et al.*, 2020) and PrivNUD (WANG *et al.*, 2023). As in (YANG *et al.*, 2020), We configure HDG with $\alpha_1 = 0.7$ and $\alpha_2 = 0.03$. HIO has been shown to perform worse than FELIP and HDG, so we omitted it due to high MAEs to highlight differences among stronger approaches. To evaluate our approach, we generate random queries varying dimensions and selectivity, then compute their MAE. Selected dimensions follow a normal distribution with mean $|A|/2$ and a standard deviation ensuring all appear at least once. Selectivity s represents the interval-to-domain ratio for numerical attributes. Unless stated otherwise, the DP group is sampled from the LDP group using the *Default* KL-divergence level (Table 4). Following prior work (KOHEN; SHEFFET, 2022; AVENT *et al.*, 2017; AVENT *et al.*, 2020; BEIMEL *et al.*, 2020) where $u_{dp} \ll u_{ldp}$, the DP group is set to 10% of the total population. We use the Mean Absolute Error (MAE) to measure the accuracy of estimated answers. The error is shown in the y axis which is in the log scale in all experiments.

7.3.3 Cross-over point

In this experiment, we assess the effectiveness of our proposed adaptive estimator DP-AE in identifying the optimal cross-over point c_{p}, which determines when to switch from using the central differential privacy (DP) estimator to the local differential privacy (LDP) estimator during the query answering process. The goal is to minimize the total expected error over a finite query workload by leveraging the high utility of the central DP estimator for a limited number of queries and the scalability of LDP for the remainder. Figure41 presents the normalized cumulative error (NCE) as a function of query index, capturing the accumulation of

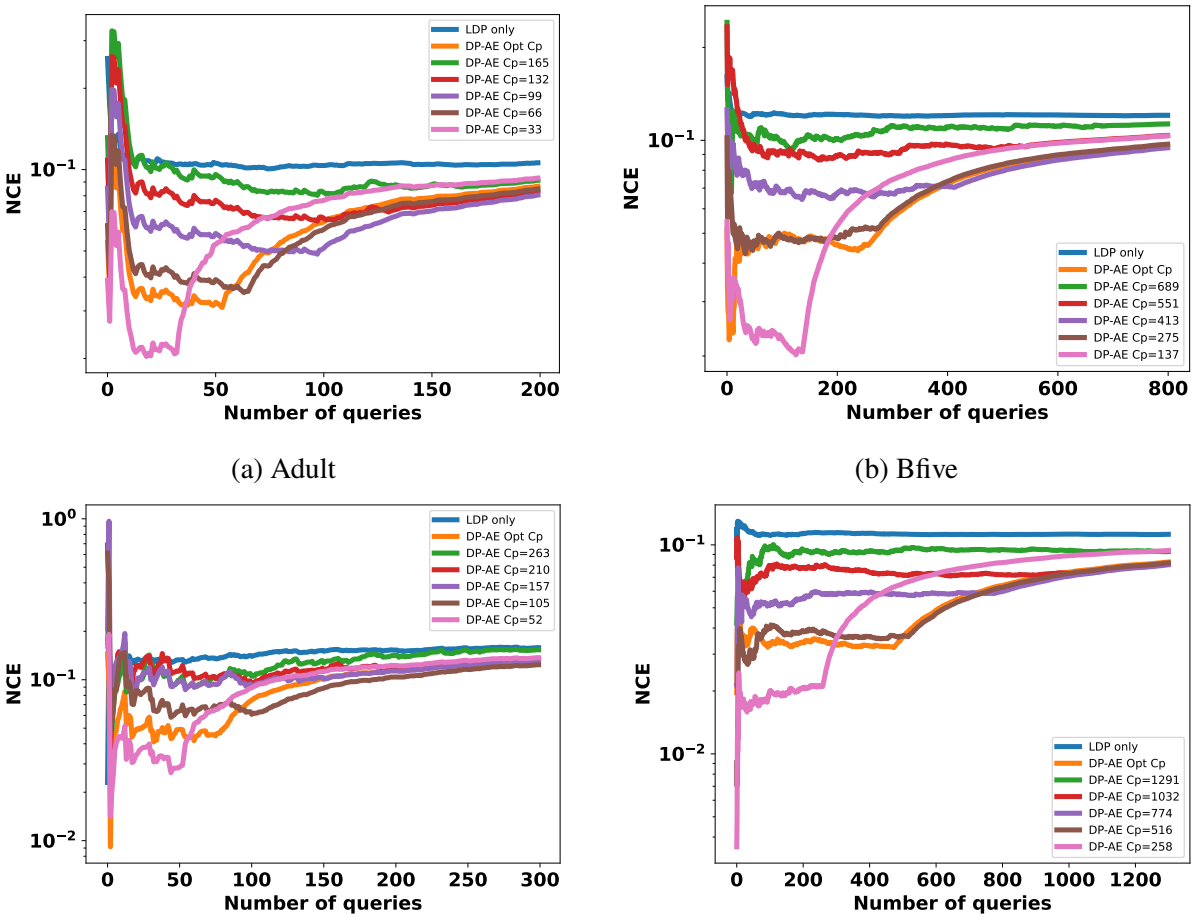
squared error as successive queries are answered. A lower NCE indicates higher overall accuracy in approximating the query workload. The area under this curve reflects the total cumulative error incurred throughout the process, making it a key metric for evaluating the performance of each strategy.

To provide comparative context, we include several baseline configurations where the crossover point c_{p} is fixed to pre-defined values. These baselines illustrate how different allocations between central DP and LDP estimators affect the error trajectory. The blue line represents the performance of a strategy that uses only the LDP estimator across all queries. As expected, the cumulative error increases approximately linearly, reflecting the relatively high and constant per-query noise introduced by LDP.

The orange line corresponds to the DP-AE strategy with an adaptively optimized crossover point, computed based on the estimated variance of both mechanisms (see Algorithm 20). This approach dynamically selects the value of c_{p} that minimizes a cost function combining both the variance of the Laplace mechanism and the expected variance of the OUE estimator under LDP.

The results demonstrate that the *DP-AE opt* strategy achieves a significantly lower cumulative error early in the query sequence, thereby minimizing the area under the curve. This confirms the efficacy of the adaptive estimator in exploiting the limited central DP budget for the first subset of queries. Over time, as the number of queries increases and the central budget is exhausted, the estimator transitions to LDP, resulting in a cumulative error trajectory that gradually aligns with the LDP-only baseline.

An important observation is that all strategies, regardless of the choice of c_{p}, eventually converge to the LDP error trajectory. This is because, beyond a certain number of queries, central DP can no longer be applied without violating the privacy budget, and the estimators must fall back on local mechanisms with higher noise. The difference, therefore, lies in how effectively the initial queries are handled, which has a lasting impact on the overall error profile. For instance, selecting a value of c_{p} that is too small (e.g., the pink curve) results in very low error for the initial queries; however, it necessitates an early transition to the LDP estimator. Conversely, choosing a value of c_{p} that is too large (e.g., the green curve) yields per-query variance that is comparable to or even exceeds that of the LDP estimator.



(a) Adult (b) Bfive

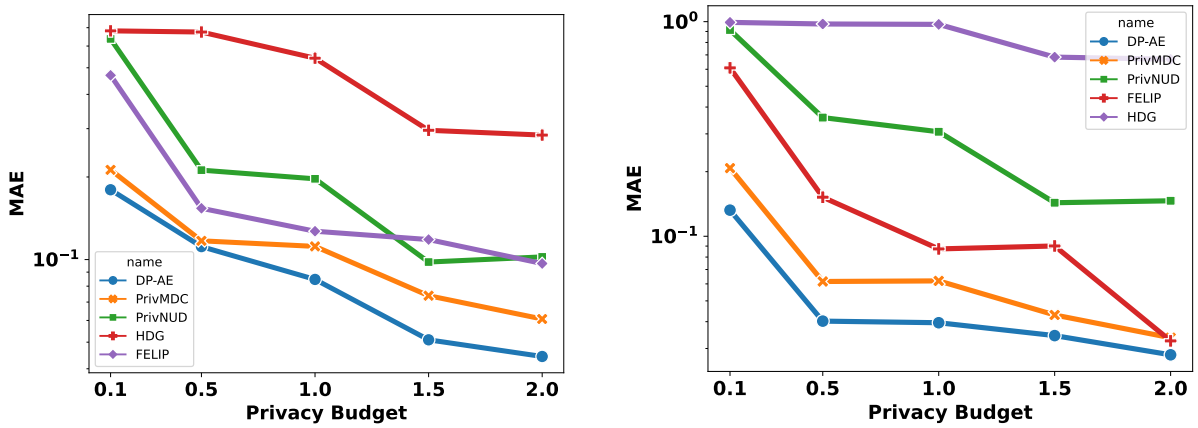
(c) Ipums (d) Loan

Figure 41 – Normalized Cumulative Error when answering a series of queries. Privacy budget is fixed at $\epsilon = 2.0$.

7.3.4 Privacy budget

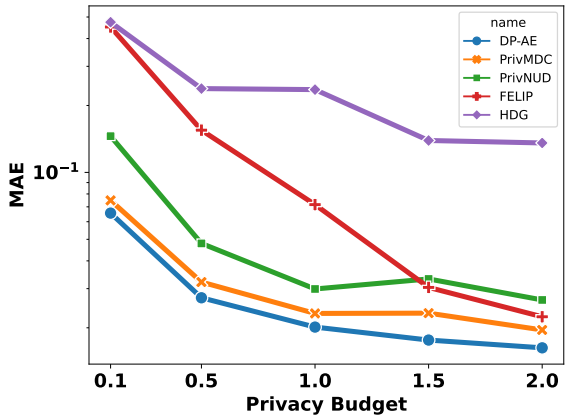
In this experiment, we evaluate the effectiveness of our optimized adaptive estimation strategy, DP-AE, by comparing its utility to that of several state-of-the-art baselines under varying privacy budgets. The goal is to assess how well each method balances privacy and utility as the privacy parameter ϵ increases. The competing methods include PrivMDC (detailed in Chapter 5), PrivNUD (WANG *et al.*, 2023), FELIP (FILHO; MACHADO, 2023), and HDG (YANG *et al.*, 2020). All methods are evaluated using the mean absolute error (MAE) metric across multiple datasets, while the privacy budget ϵ is varied from 0.1 to 2.0. The results are summarized in Figure 42.

The figure demonstrates that *DP-AE*, shown in blue, consistently outperforms all other approaches in terms of accuracy across all datasets and across the entire range of privacy budgets. The superior performance of *DP-AE* can be attributed to its adaptive crossover strategy.

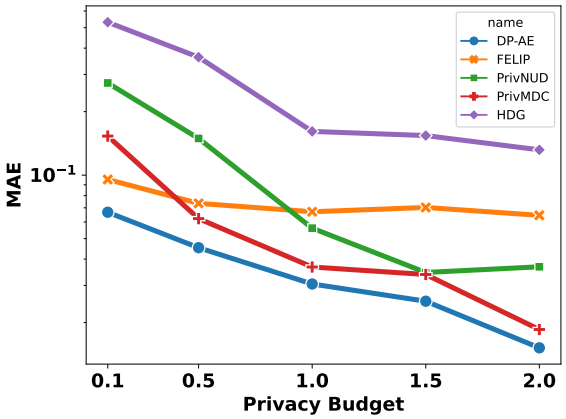


(a) Adult

(b) Bfive



(c) Ipums



(d) Loan

Figure 42 – Mean absolute error when varying the privacy budget.

It was able to adapt and approximate the optimal cross-over point effectively for all different values of privacy budgets. That enabled *DP-AE* to take advantage of the DP estimator, by planning well the use of the query answering privacy budget ϵ_{qa}, for the early queries and LDP estimator for the later ones.

Among the competitors, *PrivMDC* achieved the second-best overall performance. Although *PrivMDC* leverages correlations between attributes to enhance utility, it relies exclusively on the *LDP model* for query answering. While this allows it to scale to an arbitrary number of queries, the per-query error remains relatively high, particularly in the low-query regime where the DP estimator would offer significantly better utility. This limitation becomes evident in the results, where PrivMDC consistently underperforms compared to *DP-AE*. Finally, HDG consistently exhibits the highest MAE across all tested configurations.

7.3.5 Privacy Budget Allocation

In this experiment, we investigate how the allocation of the DP group budget, between two core components of the DP-AE framework *correlation modeling* and *query answering*, affects overall utility. Specifically, we evaluate the impact of different configurations of budget split, where the total DP group's budget is divided as $\epsilon = \epsilon_{\text{cm}} + \epsilon_{\text{qa}}$, with ϵ_{cm} dedicated to building the *correlation model*, and ϵ_{qa} reserved for answering queries using the DP estimator.

This allocation involves a fundamental trade-off. A larger allocation to ϵ_{qa} enables more queries to be answered with high accuracy using the central DP estimator in the early stages of the query workload. Since DP estimator mechanism typically achieve better accuracy per query compared to LDP, especially under small workloads, prioritizing ϵ_{qa} can be beneficial when the total number of queries is small. However, this comes at the cost of reducing the accuracy of the correlation model, which is constructed using a smaller ϵ_{cm}. A noisy or inaccurate correlation model impairs the effectiveness of the LDP estimator in later stages, as it relies on the correlation model to choose grid materializations for answering the remaining queries under LDP.

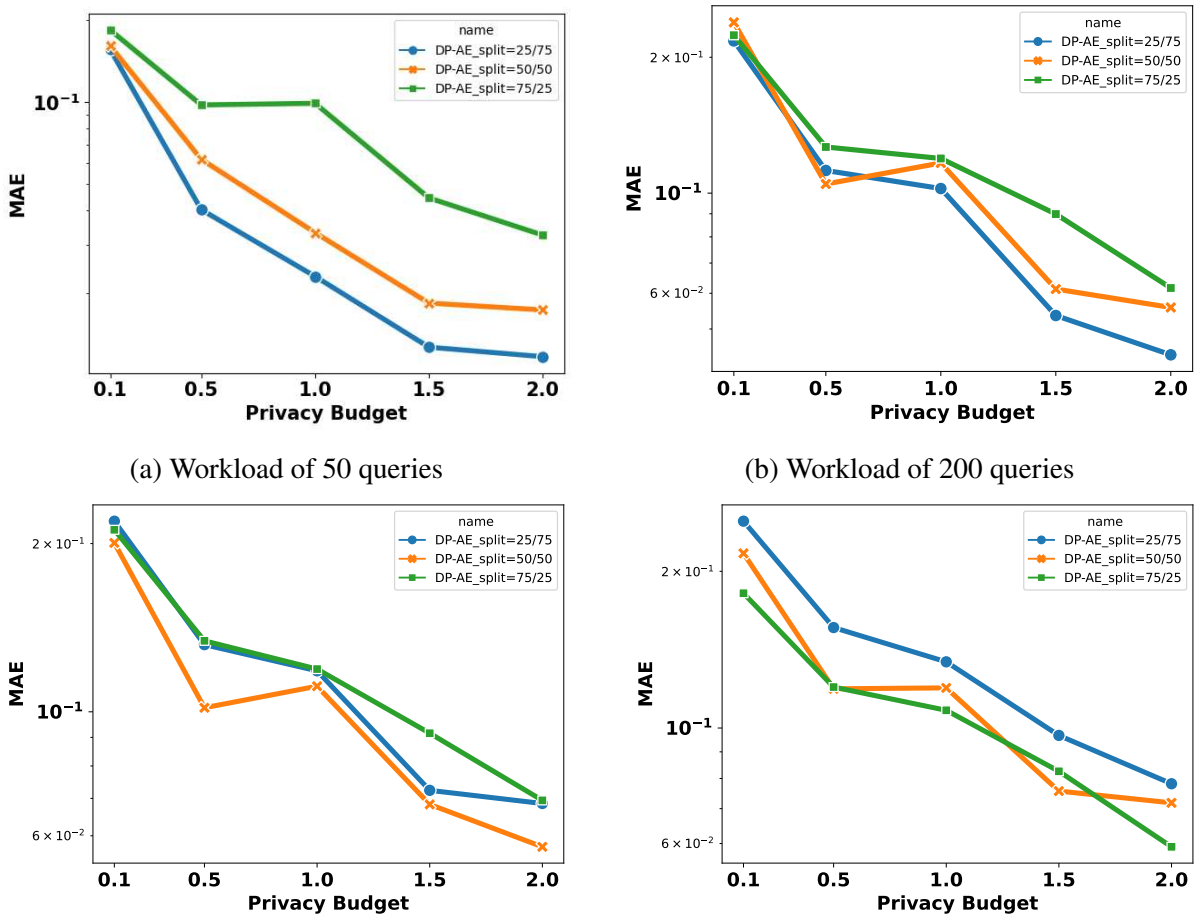
Conversely, allocating a larger fraction of ϵ to ϵ_{cm} improves the fidelity of the correlation model. This leads to better-informed grid materialization decisions and, consequently, improved LDP-based query answering, particularly in the *mid-to-long term* where DP estimator budget is depleted and the LDP estimator is responsible for handling the remainder of the workload. This configuration is especially advantageous when dealing with larger workloads, where the majority of queries will necessarily be answered under the LDP regime.

To systematically explore this trade-off, we evaluate three budget split configurations:

- *DP-AE_split=75/25*: allocates 75% of the central DP budget to the correlation model (ϵ_{cm}) and 25% to query answering (ϵ_{qa}). This favors long-term utility under LDP (indirectly, since LDP estimator benefits from accurate correlation model).
- *DP-AE_split=50/50*: an equal budget split between modeling and answering, representing a balanced strategy designed for general use.
- *DP-AE_split=25/75*: allocates the majority of the budget to query answering, prioritizing short-term utility under central DP.

We apply these configurations across four different query workloads consisting of 50, 200, 1000, and 5000 queries, using the Adult dataset as the evaluation benchmark. The results, shown in Figure 43, report the mean absolute error (MAE).

The observed the following trends:



(a) Workload of 50 queries

(b) Workload of 200 queries

(c) Workload of 1000 queries

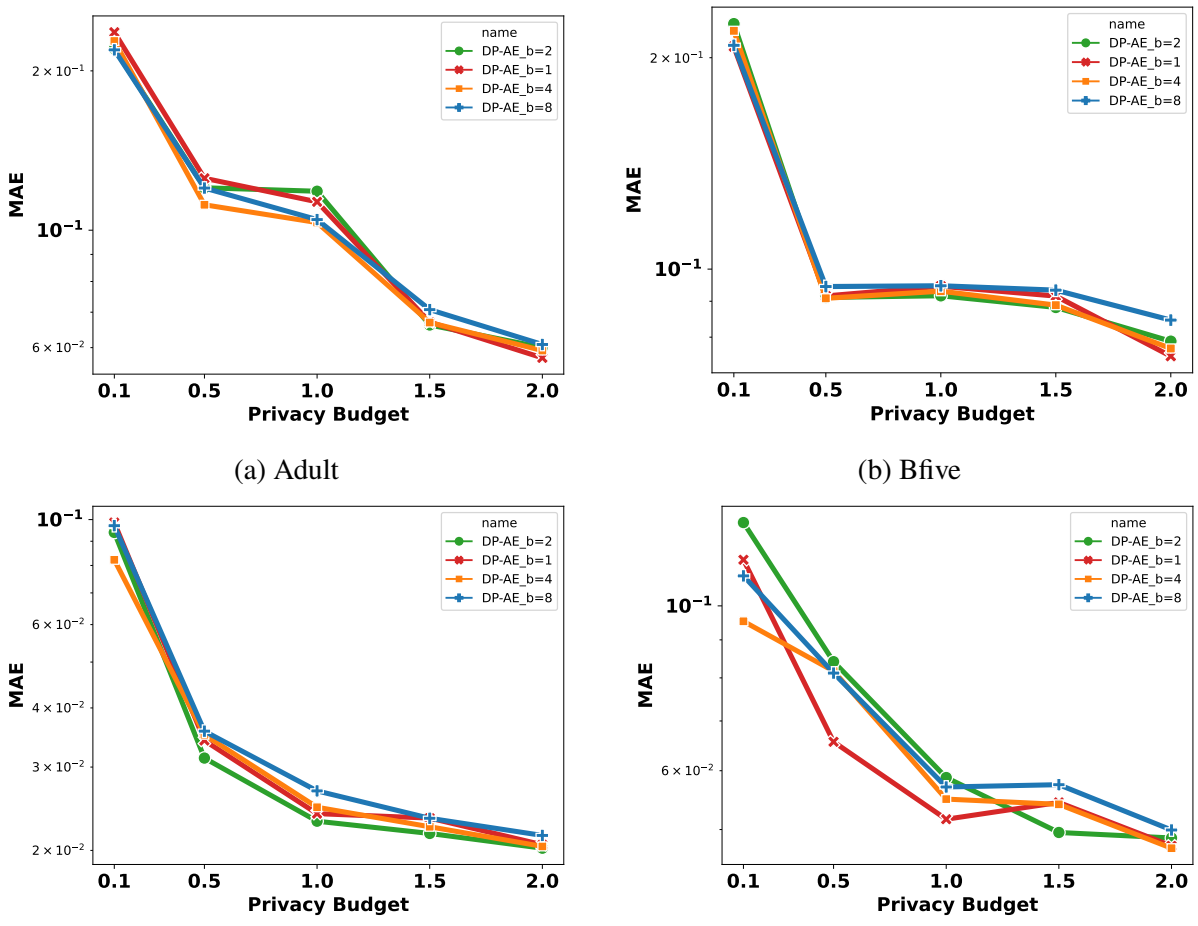
(d) Workload of 5000 queries

Figure 43 – Mean absolute error measured for different privacy budget split $\epsilon = \epsilon_{\text{cm}} + \epsilon_{\text{qa}}$ configurations under workloads with different number of queries.

- For small workloads (*e.g.*, 50 queries), the configuration *DP-AE_split=25/75* achieves the lowest MAE, especially when the total privacy budget ϵ is relatively large. This is because the central DP estimator is used to answer a larger fraction of the queries with high utility, and the limited reliance on the correlation model mitigates the impact of its reduced accuracy.
- As the number of queries increases, the utility of the *DP-AE_split=75/25* configuration improves. Since more queries in these scenarios (*e.g.*, 1000, 5000 queries) must be answered using the LDP estimator, the benefit of a high-quality correlation model becomes more pronounced, and the configuration outperforms others in longer workloads.
- The balanced split *DP-AE_split=50/50* provides consistently good performance across all settings. While not optimal in every specific scenario, it offers robust utility across diverse workloads and privacy levels.

7.3.6 *Privacy Amplification by Subsampling*

In this experiment, we evaluate the impact of subsampling on the utility of DP-AE. We issued the same workload as in previous experiment (Section 7.3.4). The DP group size is always 10% of the dataset. During the query answering stage, we divide the DP group into b (*batches*) disjoint subgroups. These subgroups are drawn uniformly at random without replacement from the DP group. Each batch is then independently responsible for answering a subset of c_p/b queries, and the total privacy budget ϵ_{qa} is utilized per batch using the Laplace mechanism. This batching strategy introduces a trade-off between privacy amplification and statistical accuracy. As the number of batches b increases, each user's probability of being included in any specific batch decreases, which in turn enhances privacy guarantees through the privacy amplification effect. However, dividing the DP group into more subgroups also results in fewer users per batch, thereby increasing the sampling variance and reducing the reliability of the estimates.



(a) Adult

(b) Bfive

(c) Ipums

(d) Loan

Figure 44 – Mean absolute error when using subsampling from the DP group.

Figure 44 displays the results for configurations using $b \in \{1, 2, 4, 8\}$ batches across multiple datasets. For the IPUMS, Adult, and Bfive datasets, the results indicate that the error remains relatively stable across different values of b. This suggests that in these datasets, the increased sampling variance introduced by batching is largely offset by the reduction in noise due to amplified privacy. In contrast, the Loan dataset exhibits higher variance in estimation error across the batching configurations, though no consistent trend emerges as ϵ varies. There is no single batching configuration that clearly outperforms the others across all privacy budgets.

8 CONCLUSIONS

In this chapter, we present a comprehensive overview of the findings obtained through our research. We have studied the problem of answering range queries by frequency estimation in differentially private manner. Particularly, Section 8.1 summarizes the achieved results based on the main research questions introduced in Chapter 1. Finally, open problems and future work directions are provided in Section 8.2.

8.1 Summary of Results

As discussed in Chapter 1, the main research questions that guided our work were:

1. How can we answer an unbounded number of differentially private queries, with range and point constraints, when the entire population is composed of LDP users?
2. Given a population with DP and LDP users and a priori knowledge about the workload, how can we answer an unbounded number of differentially private range queries?
3. Given a population with DP and LDP users and a fixed number of queries, how can we combine the DP and LDP estimators in order to optimize utility?

In the following, we discuss the achieved results for each research goal.

8.1.1 How can we answer an unbounded number of differentially private queries, with range and point constraints, when the entire population is composed of LDP users?

In Chapter 4, we studied the frequency estimation problem, under LDP. We proposed methods to construct grids to support queries with range and point constraints. We carefully analyzed the different factors that impact the estimation error when using 1-D and 2-D grids. We proposed optimizing the grids' construction, enabling users to report their private data by mapping the domain of the attributes using binning. Also, based on each grid characteristic, we proposed to adaptively select the frequency oracle protocol that will offer the best accuracy. Our approach achieved high utility across a wide range of scenarios (Section 7.1).

8.1.2 Given a population consisting of both DP and LDP users, along with a priori knowledge of the query workload, how can we scalably answer an unbounded number of differentially private range queries?

To answer this question, we introduced, in Chapter 5, a novel framework called PrivMDC. We leveraged the availability of DP users to model correlations among attributes to selectively collect only the most relevant grids. We also proposed the use of K-dimensional grids to represent multi-dimensional correlations. Furthermore, we presented an approach to optimize user distribution during the LDP phase, aiming to minimize the utility loss associated with uniform population partitioning. We achieved superior performance in utility including datasets with large number of attributes as demonstrated in Section 7.2.

8.1.3 Given a population with DP and LDP users and a fixed number of queries, how can we combine the DP and LDP estimators in order to optimize utility?

To address the setting of a bounded number of queries, we developed a variance-aware estimator that approximates the optimal strategy for answering a finite workload. Our approach investigates the use of the DP user group to model attribute correlations and support accurate query answering. In particular, we introduced the adaptive estimator DP-AE, detailed in Chapter 6, which integrates information from both DP and LDP estimators. As demonstrated in Section 7.3, DP-AE effectively combines the strengths of the two estimators, yielding lower normalized cumulative error over a finite sequence of queries. The estimator prioritizes high-utility answers in the early stages using the DP model, while subsequent queries are handled by the LDP estimator, ensuring both accuracy and privacy over the query sequence.

8.2 Future Work

There are several promising directions for extending the contributions of this thesis. One immediate area of interest is the refinement of data decomposition strategies to mitigate the adverse effects of noisy estimates in sparsely populated regions. Specifically, when data is partitioned into cells with low true counts, the noise introduced by LDP mechanisms can overwhelm the signal, leading to high estimation error. In one-dimensional cases, our current approach addresses non-uniformity error by assuming approximate fixed-size grid cells. This uniform discretization can misrepresent data with skewed or clustered distributions. As a result,

a natural extension is to explore adaptive one-dimensional decomposition strategies in which the size and number of cells vary according to the underlying data distribution. Such adaptive binning could reduce both noise and non-uniformity error simultaneously.

An important future challenge lies in developing low-dimensional grid approximations that can effectively support query answering in streaming data settings. Our current work assumes a static data collection setting; however, in many practical scenarios, data often arrives continuously. Investigating how pre-constructed grids, especially in lower-dimensional projections, can be incrementally updated, in a correlation-aware manner, and reused for answering streaming queries is an open challenge. A more challenging and unexplored research direction is the detection and exploitation of attribute correlations under Local Differential Privacy. While our current decomposition-based correlation detection operates in the central DP model, extending this capability to the local model remains an interesting problem.

We also recognize the need for an automated approach to privacy budget allocation. In the current framework (Chapter 6), the budget is manually divided between correlation modeling and query answering. Automating this partitioning based on query workload, data sparsity, and sensitivity could yield substantial improvements in utility. Lastly, the streaming setting introduces new challenges and opportunities for adaptive query estimation. As new users contribute data over time, it becomes important to design adaptive estimators that can refine their outputs incrementally without violating privacy constraints.

Together, these directions represent valuable extensions to the current work studied in this thesis, and addressing them would advance the state of the art in privacy-preserving multi-dimensional data analysis.

REFERENCES

APPLE. **Learning Iconic Scenes with Differential Privacy**. 2023. Available at: https://machinelearning.apple.com/research/scenes-differential-privacy. Accessed 15 Oct. 2025.

ABADI, M.; CHU, A.; GOODFELLOW, I.; MCMAHAN, H. B.; MIRONOV, I.; TALWAR, K.; ZHANG, L. Deep learning with differential privacy. **Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security**, Association for Computing Machinery, New York, NY, USA, p. 308–318, 2016.

ABAY, N. C.; ZHOU, Y.; KANTARCIOGLU, M.; THURAISINGHAM, B.; SWEENEY, L. Privacy preserving synthetic data release using deep learning. **Machine Learning and Knowledge Discovery in Databases**, Springer International Publishing, Cham, p. 510–526, 2019.

ACHARYA, J.; SUN, Z.; ZHANG, H. Hadamard response: Estimating distributions privately, efficiently, and with little communication. **The 22nd International Conference on Artificial Intelligence and Statistics**. Naha, Okinawa, Japan, v. 89, p. 1120–1129, 2019.

ACQUISTI, A.; BRANDIMARTE, L.; LOEWENSTEIN, G. Privacy and human behavior in the age of information. **Science**, v. 347, n. 6221, p. 509–514, 2015.

ACQUISTI, A.; GROSSKLAGS, J. Privacy and rationality in individual decision making. **IEEE Security & Privacy**, v. 3, n. 1, p. 26–33, 2005.

ALNEMARI, A.; RAJ, R. K.; ROMANOWSKI, C. J.; MISHRA, S. Interactive range queries for healthcare data under differential privacy. **IEEE 9th International Conference on Healthcare Informatics (ICHI)**, IEEE, Victoria, BC, Canada, p. 228–237, 2021.

ALVIM, M. S.; ANDRÉS, M. E.; CHATZIKOKOLAKIS, K.; DEGANO, P.; PALAMIDESSI, C. Differential privacy: On the trade-off between utility and information leakage. **Formal Aspects of Security and Trust - 8th International Workshop, FAST**, Springer, Leuven, Belgium, v. 7140, p. 39–54, 2011.

ARCOLEZI, H. H.; COUCHOT, J.-F.; Al Bouna, B.; XIAO, X. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. **Digital Communications and Networks**, v. 10, n. 2, p. 369–379, 2022.

ARCOLEZI, H. H.; COUCHOT, J.-F.; Al Bouna, B.; XIAO, X. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. **Digital Communications and Networks**, 2022.

ARCOLEZI, H. H.; COUCHOT, J.-F.; BOUNA, B. A.; XIAO, X. Random sampling plus fake data: Multidimensional frequency estimates with local differential privacy. **Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021)**, ACM, Queensland, Australia (Virtual), p. 47–57, nov 2021.

ARCOLEZI, H. H.; GAMBS, S. Revealing the true cost of locally differentially private protocols: An auditing perspective. **Proceedings on Privacy Enhancing Technologies**, v. 2024, n. 4, p. 123–141, 2024.

ARCOLEZI, H. H.; GAMBS, S. Revisiting Locally Differentially P r ivate Protocols: Towards Better Trade-offs in Privacy, Utility, and Attack Resistance. **CoRR**, abs/2503.01482, 2025.

ARCOLEZI, H. H.; GAMBS, S.; COUCHOT, J.-F.; PALAMIDESSI, C.On the Risks of Collecting Multidimensional Data Under Local Differential Privacy. **CoRR**, abs/2209.01684, 2022.

ARCOLEZI, H. H.; PINZóN, C.; PALAMIDESSI, C.; GAMBS, S. Frequency Estimation of Evolving Data Under Local Differential Privacy. **CoRR**, abs/2210.00262, 2022.

ARORA, S.; HAZAN, E.; KALE, S. The multiplicative weights update method: a meta-algorithm and applications. **Theory Comput.**, v. 8, n. 1, p. 121–164, 2012.

AVENT, B.; DUBEY, Y.; KOROLOVA, A. The power of the hybrid model for mean estimation. **Proceedings on Privacy Enhancing Technologies**, v. 2020, p. 48–68, oct. 2020.

AVENT, B.; KOROLOVA, A.; ZEBER, D.; HOVDEN, T.; LIVSHITS, B. BLENDER: Enabling local search with a hybrid differential p rivacy m odel. **2 6th USENIX Security Symposium (USENIX Security 17)**, USENIX Association, Vancouver, BC, p. 747–764, 2017.

BALLE, B.; BARTHE, G.; GABOARDI, M. Privacy profiles and amplification by subsampling. **Journal of Privacy and Confidentiality**, v. 1 0, n . 1 , 2020.

BASSILY, R. Linear queries estimation with local differential p rivacy. **The 22nd International Conference on Artificial Intelligence and Statistics**, P M LR, Naha, Okinawa, Japan, p. 721–729, 2019.

BASSILY, R.; NISSIM, K.; STEMMER, U.; THAKURTA, A. Practical locally private heavy hitters. **Proceedings of the 31st International Conference on Neural Information Processing Systems**, Curran Associates Inc., Red Hook, NY, USA, 2017.

BASSILY, R.; SMITH, A. Local, private, efficient protocols for succinct histograms. **Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing**, Association for Computing Machinery, New York, NY, USA, 2015.

BECKER, B.; KOHAVI, R. **Adult**. 1996. UCI Machine Learning Repository.

BEIMEL, A.; KOROLOVA, A.; NISSIM, K.; SHEFFET, O.; STEMMER, U. The power of synergy in differential privacy: C ombining a s mall c urator w ith l ocal randomizers. **1st Conference on Information-Theoretic Cryptography, ITC**, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Boston, MA, USA, v. 163, 2020.

BISCHOFF, P. **Internet Privacy Laws by State:** which US states best protect privacy online? 2023. Available at: https://www.comparitech.com/blog/vpn-privacy/which-us-states-best-protect-online-privacy. Accessed 15 Oct. 2025.

BLOOM, B. H. Space/time trade-offs in hash coding with allowable errors. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, 1970.

BUN, M.; NISSIM, K.; STEMMER, U.; VADHAN, S. Differentially private release and learning of threshold functions. **56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)**, IEEE, Berkeley, CA, USA, p. 634–649, 2015.

BUREAU, U. C. **Census Bureau Sets Key Parameters to Protect Privacy in 2020 Census Results**. 2021. Available at: https://www.census.gov/newsroom/press-releases/2021/ 2020-census-key-parameters.html. Accessed 15 Oct. 2025.

BUREAU, U. C. **Disclosure Avoidance for the 2020 Census:** An Introduction. 2021. Available at: https://www.census.gov/library/ publications/2021/decennial/ 2020-census-disclosure-avoidance-handbook.html. Accessed 15 Oct. 2025.

CHEN, Q.; WANG, H.; WANG, Z.; CHEN, J.; YAN, H.; LIN, X. Lldp: A layer-wise local differential privacy in federated learning. **IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)**, IEEE, Wuhan, China, p. 631–637, 2022.

CHENG, X.; FANG, L.; YANG, L.; CUI, S. Mobile big data: The fuel for data-driven wireless. **IEEE Internet of Things Journal**, v. 4, n. 5, p. 1489–1516, 2017.

CHEU, A.; SMITH, A.; ULLMAN, J. Manipulation attacks in local differential privacy. **IEEE Symposium on Security and Privacy (SP)**, IEEE, San Francisco, CA, USA, p. 883–900, 2021.

CORMODE, G. Personal privacy vs population privacy: learning to attack anonymization. **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011**, ACM, San Diego, CA, USA, p. 1253–1261, 2011.

CORMODE, G.; KULKARNI, T.; SRIVASTAVA, D. Marginal release under local differential privacy. **Proceedings of the 2018 International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, 2018.

CORMODE, G.; KULKARNI, T.; SRIVASTAVA, D. Answering range queries under local differential privacy. **Proc. VLDB Endow.**, VLDB Endowment, 2019.

CORMODE, G.; SRIVASTAVA, D.; LI, N.; LI, T. Minimizing minimality and maximizing utility: Analyzing method-based attacks on anonymized data. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 3, n. 1-2, p. 1045–1056, 2010.

CUNNINGHAM, T.; CORMODE, G.; FERHATOSMANOGLU, H.; SRIVASTAVA, D. Real-world trajectory sharing with local differential p rivacy. **P roc. V LDB E ndow.**, VLDB Endowment, 2021.

CUNNINGHAM, T.; CORMODE, G.; FERHATOSMANOGLU, H.; SRIVASTAVA, D. Real-world trajectory sharing with local differential privacy. **Proc. VLDB Endow.**, v. 14, n. 11, p. 2283–2295, 2021.

DING, B.; KULKARNI, J.; YEKHANIN, S. Collecting telemetry data privately. **Proceedings of the 31st International Conference on Neural Information Processing Systems**, Curran Associates Inc., Red Hook, NY, USA, p. 3574–3583, 2017.

DU, L.; ZHANG, Z.; BAI, S.; LIU, C.; JI, S.; CHENG, P.; CHEN, J. Ahead: Adaptive hierarchical decomposition for range query under local differential privacy. **Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security**, Association for Computing Machinery, New York, NY, USA, p. 1266–1288, 2021.

DUARTE NETO, E. R.; COSTA FILHO, J. S.; MARREIRAS NETO, A. A.; MACHADO, J. C. ALOG: adaptive longitudinal grids for geospatial data using local differential privacy. **Proceedings 29th International Conference on Extending Database Technology, EDBT**, Tampere, Finland, p. 1–14, 2026.

DUCHI, J. C.; JORDAN, M. I.; WAINWRIGHT, M. J. Local privacy and statistical minimax rates. **2013 IEEE 54th Annual Symposium on Foundations of Computer Science**, 2013.

DWORK, C.; KENTHAPADI, K.; MCSHERRY, F.; MIRONOV, I.; NAOR, M. Our data, ourselves: Privacy via distributed noise generation. **24th Annual International Conference on the Theory and Applications of Cryptographic Techniques**, Springer, St. Petersburg, Russia, p. 486–503, 2006.

DWORK, C.; MCSHERRY, F.; NISSIM, K.; SMITH, A. D. Calibrating noise to sensitivity in private data analysis. **J. Priv. Confidentiality**, v. 7, n. 3, p. 17–51, 2016.

EDMONDS, A.; NIKOLOV, A.; ULLMAN, J. The power of factorization mechanisms in local and central differential privacy. **Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing**, Association for Computing Machinery, New York, NY, USA, 2020.

ERLINGSSON, U.; PIHUR, V.; KOROLOVA, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. **Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security**, Association for Computing Machinery, New York, NY, USA, 2014.

FILHO, J. S. da C.; MACHADO, J. C. FELIP: A local differentially private approach to frequency estimation on multidimensional datasets. **Proceedings 26th International Conference on Extending Database Technology, EDBT**, OpenProceedings.org, Ioannina, Greece, p. 671–683, 2023.

GANTA, S. R.; KASIVISWANATHAN, S. P.; SMITH, A. Composition attacks and auxiliary information in data privacy. **Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)**, ACM, Las Vegas, Nevada, USA, p. 265–273, 2008.

GAO, W.; ZHOU, S. Privacy-preserving for dynamic real-time published data streams based on local differential privacy. **IEEE Internet of Things Journal**, v. 11, n. 8, p. 13551–13562, 2024.

GARFINKEL, S. L.; ABOWD, J. M.; POWAZEK, S. Advancing differential privacy: Where we are now and future directions. **Harvard Data Science Review**, 2021.

GIBBS, P.; ZAK, N. **A Review of Longevity Validations up to May 2023**. 2023.

GKOUNTOUNA, O.; DOKA, K.; XUE, M.; CAO, J.; KARRAS, P. One-off disclosure control by heterogeneous generalization. **31st USENIX Security Symposium (USENIX Security 22)**, USENIX Association, Boston, MA, p. 3363–3377, 2022.

GU, X.; LI, M.; CAO, Y.; XIONG, L. Supporting both range queries and frequency estimation with local differential privacy. 2019 **IEEE Conference on Communications and Network Security (CNS)**, IEEE, Washington, DC, USA, p. 124–132, 2019.

GU, X.; LI, M.; CHENG, Y.; XIONG, L.; CAO, Y. PCKV: Locally differentially private correlated Key-Value data collection with optimized utility. **29th USENIX Security Symposium**, USENIX Association, Boston, MA, USA, p. 967–984, 2020.

GURSOY, M. E.; LIU, L.; CHOW, K.-H.; TRUEX, S.; WEI, W. An adversarial approach to protocol analysis and selection in local differential p rivacy. **IEEE T ransactions o n Information Forensics and Security**, v. 17, p. 1785–1799, 2022.

HARDT, M.; LIGETT, K.; MCSHERRY, F. A simple and practical algorithm for differentially private data release. **Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2**, Curran Associates Inc., Red Hook, NY, USA, p. 2339–2347, 2012.

HAY, M.; RASTOGI, V.; MIKLAU, G.; SUCIU, D. Boosting the accuracy of differentially private histograms through consistency. **Proc. VLDB Endow.**, VLDB Endowment, v. 3, n. 1–2, p. 1021–1032, Sept. 2010.

HE, Y.; WANG, M.; DENG, X.; YANG, P.; XUE, Q.; YANG, L. T. Personalized local differential privacy for multi-dimensional range queries over mobile user data. **IEEE Transactions on Mobile Computing**, p. 1–15, 2025.

HONG, D.; JUNG, W.; SHIM, K. Collecting geospatial data with local differential privacy for personalized services. **IEEE 37th International Conference on Data Engineering (ICDE)**, IEEE, Chania, Greece, p. 2237–2242, 2021.

HSU, J.; GABOARDI, M.; HAEBERLEN, A.; KHANNA, S.; NARAYAN, A.; PIERCE, B. C.; ROTH, A. Differential p r i v a c y : An economic method f or choosing epsilon. **IEEE 27th Computer Security Foundations Symposium (CSF)**, IEEE, Vienna, Austria, p. 398–410, 2014.

HU, J.; DRECHSLER, J.; KIM, H. J. Accuracy gains from privacy amplification through sampling for differential p r i vacy. **J o u r n a l o f S u r v e y S t a t i s t i c s a n d M e t h o d o l o g y**, Oxford University Press, v. 10, n. 3, p. 688–719, 2022.

ISAAC, S. F. M. **Facebook Security Breach Exposes Accounts of 50 Million Users.** 2018. Available at: https://www.nytimes.com/2018/09/28/technology/facebook-hack-data-breach.html. Accessed 15 Oct. 2025.

JIN, X.; ZHANG, N.; DAS, G. Algorithm-safe privacy-preserving data publishing. **Proceedings of the 13th International Conference on Extending Database Technology (EDBT)**, ACM, Lausanne, Switzerland, p. 633–644, 2010.

JOHNSON, N.; NEAR, J. P.; SONG, D. Towards practical differential privacy for sql queries. **VLDB Endowment**, v. 11, n. 5, 2018.

JOSEPH, M.; ROTH, A.; ULLMAN, J.; WAGGONER, B. Local differential privacy for evolving data. **Proceedings of the 32nd International Conference on Neural Information Processing Systems**, Curran Associates Inc., Red Hook, NY, USA, p. 2381–2390, 2018.

KAGGLE. **All Lending Club loan data**. 2019. Available at: https://www.kaggle.com/datasets/wordsforthewise/lending-club. Accessed 15 Oct. 2025.

KAGGLE. **Big five personality test.** 2019. Available at : https://www.kaggle.com/datasets/tunguz/big-five-personality-test. Accessed 15 Oct. 2025.

KASIVISWANATHAN, S. P.; LEE, H. K.; NISSIM, K.; RASKHODNIKOVA, S.; SMITH, A. D. What can we learn privately? **49th Annual IEEE Symposium on Foundations of Computer Science, FOCS**, IEEE Computer Society, Philadelphia, PA, USA, p. 531–540, 2008.

KASIVISWANATHAN, S. P.; LEE, H. K.; NISSIM, K.; RASKHODNIKOVA, S.; SMITH, A. D. What can we learn privately? **SIAM J. Comput.**, v. 40, n. 3, p. 793–826, 2011.

KHALAF, O. I.; S.R, A.; ALGBURI, S.; S, A.; SELVARAJ, D.; SHARIF, M. S.; ELMEDANY, W. Federated learning with hybrid differential privacy for secure and reliable cross-iot platform knowledge sharing. **IEEE SECURITY AND PRIVACY**, 2024.

KOHEN, R.; SHEFFET, O. Transfer learning in differential privacy's hybrid-model. **International Conference on Machine Learning, ICML**, PMLR, Baltimore, Maryland, USA, v. 162, p. 11413–11429, 2022.

KULYNYCH, B.; GÓMEZ, J. F.; KAISSIS, G.; CALMON, F. P.; TRONCOSO, C. Attack-aware noise calibration for differential privacy. **Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems, NeurIPS**, NeurIPS, Vancouver, BC, Canada, 2024.

LAOUIR, A. E.; IMINE, A. Private approximate query over horizontal data federation. **Proceedings of the 28th International Conference on Extending Database Technology (EDBT)**, OpenProceedings, Barcelona, Spain, p. 132–144, 2025.

LEE, J.; CLIFTON, C. How much is enough? choosing for differential privacy. **Information Security, 14th International Conference, ISC**, Springer, Xi'an, China, v. 7001, p. 325–340, 2011.

LI, C.; HAY, M.; MIKLAU, G.; WANG, Y. A data- and workload-aware algorithm for range queries under differential privacy. **Proc. VLDB Endow.**, VLDB Endowment, v. 7, n. 5, p. 341–352, 2014.

LI, C.; HAY, M.; MIKLAU, G.; WANG, Y. A data- and workload-aware algorithm for range queries under differential privacy. **Proc. VLDB Endow.**, VLDB Endowment, 2014.

LI, C.; HAY, M.; RASTOGI, V.; MIKLAU, G.; MCGREGOR, A. Optimizing linear counting queries under differential privacy. **Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS**, ACM, Indianapolis, Indiana, USA, p. 123–134, 2010.

LI, C.; MIKLAU, G. An adaptive mechanism for accurate query answering under differential privacy. **Proc. VLDB Endow.**, v. 5, n. 6, p. 514–525, 2012.

LI, H.; REN, H.; LIANG, X.; HE, S.; DAI, Y. Privacy-enhanced and multifunctional health data aggregation under differential privacy guarantees. **Sensors**, p. 1015, 2016.

LI, J.; MA, H.; WU, G.; ZHANG, Y.; MA, B.; HUI, Z.; ZHANG, L.; ZHU, B. A workload division differential privacy algorithm to improve the accuracy for linear computations. **Computational Science - ICCS 2020 - 20th International Conference**, Springer, Amsterdam, The Netherlands, v. 12138, p. 439–452, 2020.

LI, N.; LI, T.; VENKATASUBRAMANIAN, S. t-closeness: Privacy beyond k-anonymity and l-diversity. **Proceedings of the 23rd International Conference on Data Engineering, ICDE**, IEEE Computer Society, Istanbul, Turkey, p. 106–115, 2007.

LI, N.; LYU, M.; SU, D.; YANG, W. **Differential Privacy: From Theory to Practice**. Morgan & Claypool Publishers, 2016. 1–138 p.

LI, N.; QARDAJI, W.; SU, D. On sampling, anonymization, and differential p rivacy or, k-anonymization meets differential p rivacy. **Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security**, Association for Computing Machinery, New York, NY, USA, p. 32–33, 2012.

LI, T.; ZAHEER, M.; REDDI, S. J.; SMITH, V. Private adaptive optimization with side information. **International Conference on Machine Learning, ICML**, PMLR, Baltimore, Maryland, USA, v. 162, p. 13086–13105, 2022.

LI, X.; YAN, H.; ZHENG, G.; LI, H.; LI, F. Key-value data collection with distribution estimation under local differential privacy. **Security and Communication Networks**, 2022.

LI, Y.; REN, X.; YANG, S.; YANG, X. Impact of prior knowledge and data correlation on privacy leakage: A unified analysis. **IEEE Trans. Inf. Forensics Secur.**, v. 14, n. 9, p. 2342–2357, 2019.

LI, Z.; WANG, T.; LOPUHAä-ZWAKENBERG, M.; LI, N.; SKORIC, B. Estimating numerical distributions under local differential p rivacy. **Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, 2020.

LIU, G.; TANG, P.; HU, C.; JIN, C.; GUO, S. Multi-dimensional data publishing with local differential privacy. **Proceedings 26th International Conference on Extending Database Technology, EDBT**, OpenProceedings.org, Ioannina, Greece, p. 183–194, 2023.

LIU, H.; WU, Z.; ZHOU, Y.; PENG, C.; TIAN, F.; LU, L. Privacy-preserving monotonicity of differential privacy mechanisms. **Applied Sciences**, v. 8, n. 11, 2018.

MACHANAVAJJHALA, A.; GEHRKE, J.; KIFER, D.; VENKITASUBRAMANIAM, M. l-diversity: Privacy beyond k-anonymity. **Proceedings of the 22nd International Conference on Data Engineering, ICDE**, IEEE Computer Society, Atlanta, GA, USA, p. 24, 2006.

MARREIRAS NETO, A. A.; Duarte Neto, E. R.; Costa Filho, J. S.; MACHADO, J. Locally differentially private and consistent frequency estimation of longitudinal data. **Anais do XXXIX Simpósio Brasileiro de Bancos de Dados**, SBC, Porto Alegre, RS, Brasil, p. 367–380, 2024.

MCKENNA, R.; MAITY, R. K.; MAZUMDAR, A.; MIKLAU, G. A workload-adaptive mechanism for linear queries under local differential privacy. **Proc. VLDB Endow.**, VLDB Endowment, 2020.

MCKENNA, R.; MIKLAU, G.; HAY, M.; MACHANAVAJJHALA, A. Optimizing error of high-dimensional statistical queries under differential privacy. **Proc. VLDB Endow.**, 2018.

MCSHERRY, F.; TALWAR, K. Mechanism design via differential privacy. **48th Annual IEEE Symposium on Foundations of Computer Science**, FOCS 2007, Providence, RI, USA, p. 94–103, 2007.

MCSHERRY, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. **Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, p. 19–30, 2009.

MCSHERRY, F. D. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. **Commun**, New York, NY, USA, 2009.

MICROSOFT. **Windows Insider Program Agreement**. 2017. Available at: https://insider.windows.com/en-us/program-agreement. Accessed 15 Oct. 2025.

NEAR, J. P.; ABUAH, C. **Programming Differential Privacy**. 2023. Available at: https://uvm-plaid.github.io/programming-dp. Accessed 15 May. 2023.

PAVERD, A.; MARTIN, A.; BROWN, I. **Modelling and Automatically Analysing Privacy Properties for Honest-but-Curious Adversaries**, 2014.

PEREZ-CRUZ, F. Kullback-leibler divergence estimation of continuous distributions. **2008 IEEE International Symposium on Information Theory**, 2008.

PERLROTH, A. T. A. S. N. **Marriott Hacking Exposes Data of Up to 500 Million Guests**. 2018. Available at: https://www.nytimes.com/2018/11/30/business/marriott-data-breach.html. Accessed 15 Oct. 2025.

PROCOPIUC, C.; YU, T.; SHEN, E.; SRIVASTAVA, D.; CORMODE, G. Differentially private spatial decompositions. **2013 IEEE 29th International Conference on Data Engineering (ICDE)**, IEEE Computer Society, Los Alamitos, CA, USA, 2012.

QARDAJI, W.; YANG, W.; LI, N. Differentially private grids for geospatial data. **2013 IEEE 29th International Conference on Data Engineering (ICDE)**, p. 757–768, 2013.

QARDAJI, W.; YANG, W.; LI, N. Understanding hierarchical methods for differentially private histograms. **Proc. VLDB Endow.**, VLDB Endowment, 2013.

QARDAJI, W.; YANG, W.; LI, N. Priview: Practical differentially private release of marginal contingency tables. **Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, 2014.

QARDAJI, W.; YANG, W.; LI, N. Priview: Practical differentially private release of marginal contingency tables. **Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, 2014.

REN, H.; LI, H.; LIANG, X.; HE, S.; DAI, Y.; ZHAO, L. Privacy-enhanced and multifunctional health data aggregation under differential privacy guarantees. **Sensors**, v. 16, n. 9, p. 1463, 2016.

REN, X.; SHI, L.; YU, W.; YANG, S.; ZHAO, C.; XU, Z. Ldp-ids: Local differential privacy for infinite data streams. **Proceedings of the 2022 International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, p. 1064–1077, 2022.

REN, X.; YU, C.-M.; YU, W.; YANG, S.; YANG, X.; MCCANN, J. A.; YU, P. S. LoPub : High-dimensional crowdsourced data publication with local differential privacy. **IEEE Transactions on Information Forensics and Security**, v. 13, n. 9, p. 2151–2166, 2018.

RESEARCH, A. M. L. **Learning Iconic Scenes with Differential Privacy**. 2023. https://machinelearning.apple.com/research/scenes-differential-privacy. Accessed 15 Oct. 2025.

RUGGLES, S.; FLOOD, S.; GOEKEN, R.; SCHOUWEILER, M.; SOBEK, M. **IPUMS USA:** Version 12.0. Minneapolis, MN, USA: IPUMS, 2022.

STEINKE, T.; ULLMAN, J. On the 'semantics' of differential privacy: A bayesian formulation. **Journal of Privacy and Confidentiality**, 2014.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **J. of Global Optimization**, Kluwer Academic Publishers, USA, 1997.

SUN, L.; ZHAO, J.; YE, X.; FENG, S.; WANG, T.; BAI, T. Conditional analysis for key-value data with local differential privacy. **CoRR**, abs/1907.05014, 2019.

SWEENEY, L. k-anonymity: a model for protecting privacy. **World Scientific Publishing Co.,** Inc., USA, v. 10, n. 5, p. 557–570, Oct. 2002.

TEAM, A. D. P. **Learning with privacy at scale.** 2017. Available at: https://machinelearning.apple.com/research/learning-with-privacy-at-scale. Accessed 15 Oct. 2025.

U.S. Census Bureau. **Key Parameters Set to Protect Privacy in 2020 Census Results**. 2021. Press release. Press Release Number: CB21-CN.42. Available at: https://www.census.gov/newsroom/press-releases/2021/2020-census-key-parameters.html. Accessed 15 Oct. 2025.

WANG, N.; WANG, Y.; WANG, Z.; NIE, J.; WEI, Z.; TANG, P.; GU, Y.; YU, G. Privnud: Effective range query processing under local differential privacy. **39th IEEE International Conference on Data Engineering, ICDE**, IEEE, Anaheim, CA, USA, p. 2660–2672, 2023.

WANG, N.; XIAO, X.; YANG, Y.; ZHAO, J.; HUI, S. C.; SHIN, H.; SHIN, J.; YU, G. Collecting and analyzing multidimensional data with local differential privacy. **2019 IEEE 35th International Conference on Data Engineering (ICDE)**, p. 638–649, 2019.

WANG, N.; XIAO, X.; YANG, Y.; ZHAO, J.; HUI, S. C.; SHIN, H.; SHIN, J.; YU, G. Collecting and analyzing multidimensional data with local differential privacy. **35th IEEE International Conference on Data Engineering, ICDE**, IEEE, Macao, China, p. 638–649, 2019.

WANG, S.; QIAN, Y.; DU, J.; YANG, W.; HUANG, L.; XU, H. Set-valued data publication with local privacy: Tight error bounds and efficient mechanisms. **Proc. VLDB Endow.**, VLDB Endowment, 2020.

WANG, T.; BLOCKI, J.; LI, N.; JHA, S. Locally differentially private protocols for frequency estimation. **26th USENIX Security Symposium**, USENIX Association, Vancouver, BC, 2017.

WANG, T.; CHEN, J. Q.; ZHANG, Z.; SU, D.; CHENG, Y.; LI, Z.; LI, N.; JHA, S. Continuous release of data streams under both centralized and local differential privacy. **Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security**, Association for Computing Machinery, New York, NY, USA, 2021.

WANG, T.; DING, B.; ZHOU, J.; HONG, C.; HUANG, Z.; LI, N.; JHA, S. Answering multi-dimensional analytical queries under local differential privacy. **Proceedings of the 2019 International Conference on Management of Data**, ACM, New York, p. 159–176, 2019.

WANG, T.; LI, Z.; LI, N.; LOPUHAÄ-ZWAKENBERG, M.; SKORIC, B. Consistent and accurate frequency oracles under local differential privacy. **CoRR**, abs/1905.08320, 2019.

WANG, T.; ZHANG, X.; FENG, J.; YANG, X. A comprehensive survey on local differential privacy toward data statistics and analysis. **Sensors**, v. 20, n. 24, 2020.

WANG, Y.; CHENG, X. PRISM: prefix-sum based r ange queries p rocessing method under local differential privacy. **38th IEEE International Conference on Data Engineering, ICDE**, IEEE, Kuala Lumpur, Malaysia, p. 433–445, 2022.

WANG, Y.-X.; LEI, J.; FIENBERG, S. E. Learning with differential p r i vacy: stability, learnability and the sufficiency and necessity of erm principle. **J. Mach. Learn. Res.**, v. 17, n. 1, p. 6353–6392, 2016.

WARNER, S. L. Randomized response: A survey technique for eliminating evasive answer bias. **Journal of the American Statistical Association**, [American Statistical Association, Taylor Francis, Ltd.], n. 309, p. 63–69, 1965.

WONG, R. C.-W.; FU, A. W.-C.; WANG, K.; YU, P. S.; PEI, J. Can the utility of anonymized data be used for privacy breaches? **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM, v. 5, n. 3, p. 1–24, 2011.

WU, N.; PENG, C.; NIU, K. A privacy-preserving game model for local differential privacy by using information-theoretic approach. **IEEE Access**, v. 8, p. 216741–216751, 2020.

YANG, J.; WANG, T.; LI, N.; CHENG, X.; SU, S. Answering multi-dimensional range queries under local differential privacy. **Proc. VLDB Endow.**, VLDB Endowment, v. 14, n. 3, 2020.

YANG, Y.; WU, L.; YIN, G.; LI, L.; ZHAO, H. A survey on security and privacy issues in internet-of-things. **IEEE Internet of Things Journal**, v. 4, n. 5, p. 1250–1258, 2017.

YE, M.; BARG, A. Optimal schemes for discrete distribution estimation under locally differential privacy. **IEEE Trans. Inf. Theory**, v. 64, n. 8, p. 5662–5676, 2018.

YE, Q.; HU, H.; MENG, X.; ZHENG, H.; HUANG, K.; FANG, C.; SHI, J. Privkvm*: Revisiting key-value statistics estimation with local differential privacy. **IEEE Transactions on Dependable and Secure Computing**, 2021.

YE, Y.; ZHANG, M.; FENG, D. Collecting spatial data under local differential privacy. **17th International Conference on Mobility, Sensing and Networking, MSN**, IEEE, Exeter, United Kingdom, p. 120–127, 2021.

YUAN, G.; ZHANG, Z.; WINSLETT, M.; XIAO, X.; YANG, Y.; HAO, Z. Low-rank mechanism: optimizing batch queries under differential privacy. **VLDB Endowment**, 2012.

YUAN, G.; ZHANG, Z.; WINSLETT, M.; XIAO, X.; YANG, Y.; HAO, Z. Optimizing batch linear queries under exact and approximate differential privacy. **ACM Trans. Database Syst.**, Association for Computing Machinery, 2015.

ZHANG, E. Z.; GUAN, Y.; YU, Y.; LU, R.; ZHANG, H. An efficient range sum query scheme under local differential privacy. **IEEE International Conference on Communications**, ICC, IEEE, Denver, CO, USA, p. 2701–2706, 2024.

ZHANG, J.; CORMODE, G.; PROCOPIUC, C. M.; SRIVASTAVA, D.; XIAO, X. Privbayes: Private data release via bayesian networks. **ACM Trans. Database Syst.**, Association for Computing Machinery, New York, NY, USA, v. 42, n. 4, 2017.

ZHANG, J.; XIAO, X.; XIE, X. Privtree: A differentially private algorithm for hierarchical decompositions. **Proceedings of the 2016 International Conference on Management of Data**, Association for Computing Machinery, New York, NY, USA, 2016.

ZHANG, X.; WANG, J.; LI, H.; GUO, Y.; PEI, Q.; LI, P.; PAN, M. Data-driven caching with users' local differential privacy in information-centric networks. **IEEE Global Communications Conference, GLOBECOM**, IEEE, Abu Dhabi, United Arab Emirates, p. 1–6, 2018.

ZHANG, Z.; MA, X.; MA, J. Local differential privacy based membership-privacy-preserving federated learning for deep-learning-driven remote sensing. **Remote Sensing**, v. 15, n. 20, 2023.

ZHANG, Z.; WANG, T.; LI, N.; HE, S.; CHEN, J. Calm: Consistent adaptive local marginal for marginal release under local differential privacy. **Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security**, New York, NY, USA, p. 212–229, 2018.

ZHU, T.; LI, G.; ZHOU, W.; YU, P. S. Differentially private data publishing and analysis: A survey. **IEEE Transactions on Knowledge and Data Engineering**, v. 29, n. 8, p. 1619–1638, 2017.

ZHU, T.; YE, D.; WANG, W.; ZHOU, W.; PHILIP, S. Y. More than privacy: Applying differential privacy in key areas of artificial intelligence. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 34, n. 6, p. 2824–2843, 2022.