Bipartite Randomized Response Mechanism for Local Differential Privacy*

Abstract—With the increasing importance of data privacy, Local Differential Privacy (LDP) has recently become a strong measure of privacy for protecting each user's privacy from data analysts without relying on a trusted third party. In many cases, both data providers and data analysts hope to maximize the utility of released data. In this paper, we study the fundamental trade-off formulated as a constrained optimization problem: maximizing data utility subject to the constraint of LDP budgets. In particular, the Generalized Randomized Response (GRR) treats all discrete data equally except for the true data. For this, we introduce an adaptive LDP mechanism called Bipartite Randomized Response (BRR), which solves the above privacy-utility maximization problem from the global standpoint. We prove that for any utility function and any privacy level, solving the maximization problem is equivalent to confirming how many high-utility data to be treated equally as the true data on release probability, the outcome of which gives the optimal randomized response. Further, solving this linear program can be computationally cheap in theory. Several examples of utility functions defined by distance metrics and applications in decision trees and deep learning are presented. The results of various experiments show that our BRR significantly outperforms the state-of-the-art LDP mechanisms of both continuous and distributed types.

Index Terms—local differential privacy, randomized response, data utility

I. INTRODUCTION

In recent years, as concerns over data privacy have intensified, differential privacy (DP) has emerged as a vital approach for protecting individual information [1]. DP, first introduced by Dwork in 2006 [2], addresses the issue of privacy leakage caused by minor changes in the data source. By employing rigorous mathematical proofs, DP ensures that the output information remains influenced by any single record only within a specified threshold, thus preventing third parties from inferring sensitive information based on output variations. In DP, users provide real data to a central server, which acts as a trusted entity to safeguard privacy. In contrast, Local Differential Privacy (LDP) allows users to send data with added noise directly to the central server. Both the true data of users and the perturbed data they upload belong to the same answer domain, ensuring that individual privacy is protected.

One foundational mechanism that inspired research in local privacy-preserving techniques, including differential privacy, is the Randomized Response (RR) mechanism introduced by Warner [3]. RR was specifically designed to protect the privacy of survey respondents by introducing uncertainty into their answers. Respondents answer sensitive Boolean questions

Authors: Shun Zhang, Hai Zhu, Zhili Chen, Neal N. Xiong
Identify applicable funding agency here. If none, delete this.

Fig. 1. Operational Process of the BRR Mechanism: assigning publishing probabilities based on similarity to true values

by following a predefined probabilistic procedure. Instead of always answering truthfully, respondents flip a biased coin: they provide their truthful answer with a fixed probability p and respond wrongly with probability *1-p*. This obfuscation ensures that an individual's true response remains hidden while enabling accurate aggregate statistics to be computed across a population.

While RR is effective for protecting binary data, its applicability is limited to scenarios where the answer domain consists of only two values. To overcome this limitation, the Generalized Randomized Response (GRR) mechanism was proposed. GRR extends the RR mechanism to accommodate larger answer domains with N possible values. Under GRR, the data publisher selects and reports the true value with a fixed probability p, while assigning an equal probability to each of the remaining *N-1* values with probability $\frac{1-p}{N-1}$. This extension retains the privacy-preserving properties of RR while broadening its applicability to use cases involving multi-valued categorical data, such as user preferences, survey responses, or demographic attributes.

However, it is important to note that in most scenarios, the data within the answer domain often exhibit a certain degree of similarity, which is not accounted for by the GRR mechanism. To address this limitation, the Bipartite Randomized Response (BRR) mechanism, proposed in this paper, incorporates the similarity between the true value and other values in the answer domain. Building on the GRR framework, BRR adjusts the probabilities assigned to each possible value based on their similarity to the true value, rather than adhering strictly to

Fig. 2. Probability assignment in the BRR mechanism compared to GRR

the binary relationship between *p* and *1-p*. This concept is illustrated in Figure 1.

For instance, when querying the highest score in a class's final assessment, the GRR mechanism would publish the actual highest score x (e.g., 99) with a probability p, while assigning an equal probability q to all other scores in the class. This approach, however, introduces bias because scores close to the highest score (e.g., 98, 97, 96) are treated the same as scores significantly lower than the highest (e.g., 50 or 40). The BRR mechanism resolves this issue by redistributing the probabilities. In BRR, scores such as 98, 97, and 96, which are close to the true value, are assigned a modified probability $p*$, while all other scores are assigned a new probability $q*$. Notably, $p*$ and $q*$ satisfy the relationships $p*<p, q*<q$, and $p*>q*$.

This redistribution ensures that the probabilities for the true value and values significantly different from the true value are reduced, while the probabilities for values close to the true value are increased. When the privacy budget and the answer threshold d remain constant, this adjustment strengthens privacy protection while potentially enhancing the efficiency of data queries. Consequently, BRR offers a more refined probability allocation that theoretically improves query performance under stricter privacy guarantees.

Figure 2 illustrates the operational process of the BRR mechanism. Let X denote the space of true values and Y the space of published values, where Y has the same response domain as X. When perturbing a specific x_i, BRR assigns higher publishing probabilities to the m values most similar to x_i, based on the degree of similarity to other values. It is important to note that this similarity is not necessarily reciprocal. For instance, when x_i is the true value, it has a set of similar answers Y_m, and suppose y_i belongs to this set. However, when y_i is the true value, it also has a set of similar answers based on y_i, but x_i may not belong to that set. By adjusting the publishing probabilities, BRR increases the likelihood of publishing data that is closer to the true value while decreasing the likelihood for data that is farther from the true value. This reallocation mechanism renders the publishing probabilities more rational.

The main contributions of this article are summarized below:

- This paper takes into account data utility to allocate publication probabilities for the first time. We introduce an adaptive LDP mechanism called Bipartite Randomized Response (BRR) that builds upon the GRR mechanism by incorporating the notion of data similarity. It reallocates the release probabilities of the data in the answer domain based on the true value, thereby avoiding the issue where values close to the true value are assigned the same release probability as vastly different values.
- The BRR mechanism enhances data query utility. This optimization theoretically improves the utility of data queries under the same privacy budget and answer domain conditions, providing more useful data release outcomes.
- Under the framework of differential privacy, the BRR mechanism achieves stronger privacy protection by optimizing the allocation of release probabilities. The introduction of $p*$ and $q*$ ensures that the release probabilities of the true value and values close to it are higher, while reducing the probabilities for erroneous values. This balance enhances privacy protection while simultaneously improving data utility.

Roadmap. The second section introduces the related work. In the third section, requisite foundational knowledge is elucidated. The fourth section expounds in detail on the BRR mechanism proposed herein. The fifth section presents some comparative experiments between BRR and other response mechanisms. The sixth section offers a comprehensive conclusion of this paper.

II. RELATED WORK

A. Advancements in LDP Mechanisms

Warner's RR method pioneered privacy protection and laid the groundwork for the later development of local differential privacy (LDP) theory. (2006) [2] were the first to systematically introduce the concept of differential privacy, theorizing the noise-adding mechanism based on sensitivity. Holohan et al. (2017) [4] provided an optimal mechanism for Warner's original RR technique, achieving the best application of differential privacy in randomized response surveys. Dwork et al. These foundational theories set the stage for subsequent advancements in both LDP mechanisms and applications.

With advances in technology, Kairouz et al. (2016) [5] proposed the k-ary randomized response (k-RR, also known as GRR or CRR, and referred to as GRR in this paper), which extended traditional binary response to satisfy differential privacy in multi-value data environments. This mechanism achieved the optimal balance of privacy and utility under convex utility functions, forming the theoretical basis for the BRR mechanism in this study. Building on GRR, Arcolezi et al. (2024) [6] introduced a series of enhanced LDP protocols, such as L-GRR, OUE (Optimized Unary Encoding), and SUE (Symmetric Unary Encoding), further improving the applicability of privacy protection in longitudinal data collection. Makhlouf et al. (2024) [7] discussed the fairness issues of LDP under multiple sensitive attributes, highlighting the potential of the GRR mechanism in ensuring fairness.

Duchi et al. (2013) [8] studied the privacy-accuracy trade-offs in data tasks such as mean estimation, establishing a theoretical foundation for the application of differential privacy. Li et al. (2017) [9] reviewed the ϵ-differential privacy framework and its applications, while Chen et al. (2016) [10] delved into the utility and complexity of personalized local differential privacy (PLDP), expanding the applicability of personalized privacy protection. Song et al. (2020) [11] further proposed the personalized randomized response (PRR) mechanism, enabling LDP to adapt more flexibly to individual differences by setting different privacy requirements for distinct data values. Wang et al. (2017) [12] proposed a generalizable aggregation framework to optimize protocol parameters through a simplified aggregation algorithm, improving the utility of data collection. Zhang et al. (2018) [13] developed the CALM method, which selectively collects subset attributes to reduce noise impact, significantly enhancing the accuracy of marginal statistics in high-dimensional data.

B. Federated Learning with LDP

In the field of federated learning (FL), Truex et al. (2020) [14] proposed the LDP-Fed system, which combines LDP with FL for privacy protection in distributed data, particularly suited for high-dimensional data scenarios. Zhao et al. (2020) [15] developed the LDP-FedSGD algorithm, which protects model privacy by adding noise during gradient upload, addressing some shortcomings of traditional differential privacy in federated learning. LDP-FedSGD not only protects the privacy of individual gradients but also maintains high utility in the trained models, making it a versatile solution for privacy-preserving federated learning.

Meanwhile, Wei et al.(2021) [16] proposed the UDP model for mobile scenarios. The UDP model enables users to dynamically adjust their privacy budget based on individual privacy preferences or application requirements, allowing for a more personalized trade-off between privacy and model accuracy. This adaptability is particularly valuable in scenarios where user devices have varying computational capacities and privacy needs, making the UDP model a practical solution for real-world federated learning applications.

C. Applications and Challenges in Data Mining and Machine Learning

In data mining, Bai et al. (2017) [17] and Fletcher, Islam et al. (2019) [18] combined differential privacy with decision tree models, proposing a multilayer DP model based on Markov Chain Monte Carlo to effectively balance privacy protection and model performance. To optimize LDP's frequency estimation performance, Fang et al. (2023) [19] proposed a convolution-based frequency estimation method that reduces LDP noise through deconvolution, significantly enhancing frequency statistics accuracy. Yang et al. (2024) [20] highlighted the importance and feasibility of LDP in protecting user privacy.

Many studies have proposed different differential privacy techniques to address the challenges posed by data sensitivity. Abadi et al. (2016) [21] proposed a method for training deep neural networks with differential privacy, providing privacy protection for machine learning on sensitive datasets. Bassily and Smith (2015) [22] proposed an efficient LDP protocol for accurately counting high-frequency items. Studies by Smith et al. (2017) [23] and Wang et al. (2018) [24] focused on optimizing the interactive rounds and sample complexity of LDP, proposing methods to reduce computational costs and improve accuracy. Yan et al. (2023) [25] achieved privacy protection for location data by optimizing randomized response with Hilbert curves, enhancing privacy effectiveness for geographic data. Wu et al. (2020) [26] investigated noise injection to protect privacy in distributed machine learning in multi-data-owner environments. Gutiérrez et al. (2024) [27] introduced a privacy-preserving framework for fog computing environments while maintaining model accuracy. Cao et al. (2021) [28] investigated data poisoning attacks on LDP protocols and proposed defense methods. Teng et al. (2021) [29] proposed LDP algorithm capable of providing higher accuracy for handling numerical and categorical data.

Through an in-depth analysis of existing studies, Wang et al. (2023) [30] summarized the applications and challenges of DP in deep learning, including the trade-offs between where noise is introduced, accuracy, and privacy, and the types of DP noise added. Gursoy et al. (2019) [31] proposed the Cohesive Local Differential Privacy (CLDP) mechanism, which concentrates privacy protection on data collection for smaller groups, improving applicability in network security for small-scale data. Niu et al. (2021) [32] introduced the AdaPDP mechanism, which dynamically adjusts noise distribution based on user privacy needs, enhancing LDP adaptability to diverse privacy demands. In high-dimensional data and IoT contexts, Arachchige et al. (2020) [33] improved privacy protection in convolutional neural network training with the LATENT mechanism, providing an efficient privacy solution for IoT devices. Arcolezi et al.(2021) [34] studied multidimensional data frequency estimation under LDP and proposed a multidimensional LDP scheme to address the combined privacy budget challenge.

Rachel et al. (2021) [35] conducted a user survey to explore

Symbol	Definition
X	the space of true valuest
Y	the space of published values, identical to the space of X
N	the publish domain, $N = |Y|$
Y_m	the set of m terms most similar to a specific true value
ϵ	privacy budget
p	probability of publishing the true item in RR or GRR
q	probability of publishing non-true item in RR or GRR
m	the unified count of high-weighted terms
$p*$	probability of publishing the true and close item in BRR
$q*$	probability of publishing non-true and non-close item in BRR
i	index of the current item in the utility sequence
j	index used to traverse all terms
k	index of the real term currently being processed
λ_i	similarity value of the i-th item with respect to the real item
$\lambda_i^{(k)}$	similarity of the i-th item in the k-th event
s_i	similarity weight of the i-th item, initialized as $s_1 = e^\epsilon$
$s_i^{(k)}$	similarity weight of the i-th item in the k-th event
w_i	normalized probability weight for the i-th item, $\Sigma_i w_i = 1$
$w_i^{(k)}$	normalized weight of the i-th item in the k-th event
Q	the total utility function defined as $Q = \Sigma_i \lambda_i w_i$
$\pi(k)$	the prior probability distribution of the k-th occurrence
$Q^{(k)}$	the utility function of the k-th local event, $Q^{(k)} = \Sigma_i \lambda_i^{(k)} w_i^{(k)}$
Q_g	the global utility function, defined as $Q_g = \Sigma_k \frac{\Sigma_i \lambda_i^{(k)} s_i}{\Sigma_j s_j}$

users' perceptions and expectations of differential privacy, pointing out the importance of improving data utility while protecting privacy. Kairouz, Bonawitz, and Ramage (2016) [36] presented new mechanisms, including hashed kRR, which outperform existing mechanisms in utility across all privacy levels. Wang et al. (2019) [37] proposed the PEM protocol, optimizing utility and computational complexity by grouping users and reporting value prefixes. Gutiérrez et al. Murakami and Kawamoto (2019) [38] introduced Utility-optimized LDP (ULDP), specifically designed to improve utility in contexts containing large amounts of non-sensitive data.

D. Direction

LDP demonstrates remarkable diversity and adaptability across various application scenarios, including federated learning and data mining, with extensive exploration into the application and optimization of LDP under privacy protection. Although significant strides have been made, ongoing challenges remain in refining privacy-utility trade-offs and enhancing the adaptability of protocols for complex data environments, pointing to future areas for research and improvement. Based on GRR, this article proposes a new mechanism to improve the utility of data while protecting privacy.

III. FOUNDATIONAL KNOWLEDGE

In this subsection, we delve into the concept of ϵ-differential privacy, local differential privacy and associated noise-adding mechanisms, including Randomized Response (RR), Generalized Randomized Response (GRR), the Exponential Mechanism, and the Laplace Mechanism. These methods constitute pivotal techniques for implementing differential privacy, and each will be elucidated in detail below.

A. ϵ-Differential Privacy and Local Differential Privacy

DP and LDP are both privacy-preserving mechanisms used to protect sensitive data. While both methods aim to ensure privacy, they differ in their implementation and application scenarios.

Differential Privacy is defined as follows. Given a randomized algorithm $\mathcal{M}$ that takes a dataset D as input and produces an output, $\mathcal{M}$ satisfies ϵ-differential privacy if for all neighboring datasets D and D' (datasets that differ in only one data point) and for any subset S of the output space, the following condition holds:

$$\Pr[\mathcal{M}(D) \in S] \leq e^\epsilon \cdot \Pr[\mathcal{M}(D') \in S]. \quad (1)$$

Here, D and D' are neighboring datasets, differing by only one data point. S is a subset of the algorithm's output space. ε is the privacy parameter that controls the level of privacy protection.

Differential Privacy ensures that the presence or absence of a single data point does not significantly affect the output, thus safeguarding individual data privacy.

Local Differential Privacy differs from traditional Differential Privacy in that it does not require a trusted third party to process the data. Instead, data is privatized at the user level before it is sent to a central authority. An algorithm $\mathcal{M}$ satisfies ϵ-Local Differential Privacy if, for any two possible inputs x and x' and for any output y, the following condition holds:

$$\Pr[\mathcal{M}(x) = y] \leq e^\varepsilon \cdot \Pr[\mathcal{M}(x') = y]. \quad (2)$$

$\mathcal{M}$ is the mechanism that adds noise to the data. ε is the privacy budget, controlling the level of privacy protection. x and x' are any two distinct input values. y is the output of the algorithm.

In LDP, each user adds noise locally before sending the perturbed data to the central data collector. This ensures that the data collector only receives the noisy version of the data and cannot infer the original data, effectively preserving individual privacy.

B. Exponential Mechanism

The Exponential Mechanism, introduced by McSherry and Talwar in 2007 [39], is a versatile differential privacy mechanism suitable for arbitrary query ranges. The fundamental principle involves selecting outputs from the output space based on a score function. The score function $q(D, r)$ measures the quality of output r for the database D. The Exponential Mechanism is defined as:

$$\Pr[\mathcal{M}(D) = r] \propto \exp\left(\frac{\epsilon q(D, r)}{2\Delta q}\right), \quad (3)$$

where Δq is the sensitivity of the score function, representing the maximum change in the score function due to a single-element change in the database.

C. Laplace Mechanism

The Laplace Mechanism([40], [41]) is one of the most commonly employed methods for achieving differential privacy. It accomplishes privacy protection by adding noise drawn from a Laplace distribution to the query results. For a given query function f, the noisy output is defined as:

$$\mathcal{M}(D) = f(D) + \text{Laplace}\left(\frac{\Delta f}{\epsilon}\right). \qquad (4)$$

Here, Laplace(λ) denotes a Laplace distribution with a mean of 0 and scale parameter λ, and Δf is the sensitivity of the query function f.

D. Randomized Response (RR)

Randomized Response, originally proposed by Warner in 1965 [3], is a technique for anonymized surveys. Its core idea is that respondents answer sensitive questions according to a probabilistic mechanism. Respondents use a probabilistic method (typically involving a coin flip or another random process) to determine their answers to sensitive questions. This method ensures that even if researchers know the respondents' answers, they cannot ascertain whether the respondents are providing truthful responses, thereby protecting the respondents' privacy.

E. Generalized Randomized Response (GRR)

Unlike the classical method, which typically deals with binary responses, GRR is designed to handle multi-category problems. It introduces various probabilistic noise mechanisms into the response process to protect privacy. For a question with N possible answers, a respondent might randomly select an answer based on a probability distribution. The GRR method preserves individual privacy while enabling unbiased estimation of the overall data.

To satisfy ϵ-differential privacy in the GRR context, each possible answer can be chosen with the following probabilities:

$$\Pr[y|x] = \begin{cases} \frac{e^\epsilon}{e^\epsilon+N-1} & \text{if } y = x, \\ \frac{1}{e^\epsilon+N-1} & \text{if } y \neq x. \end{cases} \qquad (5)$$

Where x is true answer, which is the actual choice of the respondent. And y represents the reported answer, which may or may not be the same as the true answer x. This setup ensures that the mechanism adheres to the ϵ-differential privacy requirements. By adjusting the probabilities, the GRR mechanism effectively balances the trade-off between privacy protection and data accuracy across multiple categories.

IV. DETAILED DESIGN AND IMPLEMENTATION OF BRR

In this section, we first introduce the mathematical expression of BRR. Then we primarily focus on determining the number of terms m in the BRR mechanism, dividing the discussion into the optimal cases for local real terms and globally prior real occurring terms.

A. Bipartite Randomized Response (BRR)

In the BRR mechanism, a pivotal aspect lies in elevating the publishing probability of m other values similar to the true value to match that of the true value itself. When $m = 1$, indicating the absence of other values closely resembling the true value, $p = p^*$, thereby equating BRR to GRR. Let Y_m denote the set containing the true value and its similar values. Clearly, $Y_m \subseteq Y$.

Under the premise of satisfying differential privacy, BRR can be expressed as:

$$\Pr[y|x] = \begin{cases} \frac{e^\epsilon}{me^\epsilon+N-m} & \text{if } y \in Y_m, \\ \frac{1}{me^\epsilon+N-m} & \text{if } y \notin Y_m. \end{cases} \qquad (6)$$

Through the careful allocation of publishing probabilities p^* and q^*, BRR reduces the likelihood of disclosing erroneous values that deviate significantly from x_i. This balance between data utility and privacy enhancement makes BRR a promising mechanism in the realm of differential privacy.

B. m in Local Bipartite Randomized Response Mechanism

For local publishing, starting from a determined real term, let λ_i denote the similarity of each term to it. The utility value can be simply defined and directly uses the similarity (or a functional relationship of the similarity). After sorting, we have:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0.$$

The corresponding probability distribution weights s_i satisfy:

$$e^\epsilon = s_1 \geq s_2 \geq \cdots \geq s_N = 1.$$

The sum of s_i is normalized to w_i such that $\Sigma_i w_i = 1$. The utility is defined as $Q = \Sigma_i \lambda_i w_i$, where $w_i = \frac{s_i}{\Sigma_j s_j}$. Therefore, $Q = \frac{\Sigma_i \lambda_i s_i}{\Sigma_j s_j}$. To further explore the effect of s_i on Q, we take the partial derivative of Q with respect to s_i :

$$\frac{\partial Q}{\partial s_i} = \frac{\lambda_i \Sigma_j s_j - \Sigma_j \lambda_j s_j}{(\Sigma_j s_j)^2} = \frac{\Sigma_j (\lambda_i - \lambda_j) s_j}{(\Sigma_j s_j)^2}. \qquad (7)$$

It can be observed that the monotonicity of Q with respect to s_i is independent of the values of s_i. Based on this, to maximize the utility Q, we start by initializing it as the GRR mechanism, i.e., $s_1 = e^\epsilon$, $s_2, \ldots, s_N$ are uniformly assigned the minimum value 1. Next, we incrementally increase s_2 to the maximum value $s_1 = e^\epsilon$, then successively increase $s_3, s_4, \ldots$, until some s_i satisfies $\frac{\partial Q}{\partial s_i} \leq 0$. At this point, $s_i, \ldots, s_N$ remain unchanged and are uniformly assigned the minimum value 1. The detailed algorithmic process is presented comprehensively in Algorithm 1. However, for different given occurrences, the optimal discrete probability distribution and the corresponding high probability count m can vary, which violates differential privacy protection.

Algorithm 1 Local highest-utility response search algorithm

Input: utility sequence $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$, privacy budget ϵ

1: Initialize: $s_1 = e^\epsilon$, $s_2 = \cdots = s_N = 1$, $w_1 = \frac{e^\epsilon}{e^\epsilon + N - 1}$, $w_2 = \ldots = w_N = \frac{1}{e^\epsilon + N - 1}$, $Q = \Sigma_i \lambda_i w_i$, $m = 1$
2: **for** $i = 2$ to N **do**
3:     Compute $\frac{\partial Q}{\partial s_i} = \frac{\Sigma_j(\lambda_i - \lambda_j)s_j}{(\Sigma_j s_j)^2}$
4:     **if** $\frac{\partial Q}{\partial s_i} > 0$ **then**
5:       $s_i = e^\epsilon$, $m = i$
6:     **else**
7:       Break
8:     **end if**
9: **end for**
10: **for** $i = 1$ to N **do**
11:     $w_i = \frac{s_i}{\Sigma_j s_j}$
12: **end for**

Output: $(w_1, \ldots, w_N)$ and m

C. m in Bipartite Randomized Response Mechanism

For global processing similar to local, considering the real occurrence k, we first sort the similarity $\lambda_i^{(k)}$ and weights $s_i^{(k)}$ as follows:

$$\lambda_1^{(k)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k)}, \quad s_1^{(k)} \geq s_2^{(k)} \geq \cdots \geq s_N^{(k)}.$$

The weights $s_i^{(k)}$ are summed and normalized to generate $w_i^{(k)} = \frac{s_i^{(k)}}{\Sigma_j s_j^{(k)}}$ with satisfying $\Sigma_i w_i^{(k)} = 1$.

For the occurrence item k, the initial weight vector $s^{(k)}$ is distributed according to the GRR mechanism, where the first weight is e^ϵ and the remaining weights are 1. Based on this, the utility function is defined as $Q^{(k)} = \sum_i \lambda_i^{(k)} w_i$.

Starting from the second weight $s_2^{(k)}$, each weight is adjusted sequentially. During the adjustment process, the partial derivative of the utility function $Q^{(k)}$ with respect to $s_i^{(k)}$ is calculated to determine whether increasing the current weight can improve the utility function.

Once the weights of all occurrence items k have been optimized, the minimum value $m = \min_k m^{(k)}$ is selected as the global final threshold.

To put it simply, we compare the optimal $m^{(k)}$ derived from each local BRR mechanism and select the minimum value, This allows us to construct a unified global BRR mechanism. In this global BRR mechanism:

- For each local occurrence k, $m^{(k)}$ high-weighted items are assigned a probability of e^ϵ, while the remaining items are assigned a probability of 1.
- m is chosen as the smallest $m^{(k)}$ across all occurrences, ensuring uniformity and efficiency across different local scenarios.

The core idea of BRR mechanism is to dynamically adjust the weight distribution under the constraints of privacy protection, improving utility and achieving a balance between utility and privacy. Algorithm 2 outlines the process of creating a global BRR mechanism by combining several local BRR

Algorithm 2 Bipartite Randomized Response Mechanism (BRR)

Input: utility matrix $\{\lambda_j^{(k)}\}_{N \times N}$ with $\lambda_1^{(k)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k)} > 0$, privacy budget ϵ

1: Initialize: $s_1^{(k)} = e^\epsilon$, $s_2^{(k)} = \cdots = s_N^{(k)} = 1$, $m^{(k)} = 1$
2: **for** $k = 1$ to N **do**
3:     $w_1 = \frac{e^\epsilon}{e^\epsilon + N - 1}$, $w_2 = \ldots = w_N = \frac{1}{e^\epsilon + N - 1}$, $Q^{(k)} = \Sigma_i \lambda_i^{(k)} w_i$
4:     **for** $i = 2$ to N **do**
5:       Compute $\frac{\partial Q^{(k)}}{\partial s_i^{(k)}} = \frac{\Sigma_j(\lambda_i^{(k)} - \lambda_j^{(k)})s_j^{(k)}}{(\Sigma_j s_j^{(k)})^2}$
6:       **if** $\frac{\partial Q^{(k)}}{\partial s_i^{(k)}} > 0$ **then**
7:         $s_i^{(k)} = e^\epsilon$, $m^{(k)} = i$
8:       **else**
9:         Break
10:       **end if**
11:     **end for**
12: **end for**
13: Compute $m = \min_k m^{(k)}$
14: **for** $i = 1$ to N **do**
15:     **if** $i \leq m$ **then**
16:       $w_i = \frac{e^\epsilon}{me^\epsilon + N - m}$
17:     **else**
18:       $w_i = \frac{1}{me^\epsilon + N - m}$
19:     **end if**
20: **end for**

Output: $(w_1, \ldots, w_N)$ and m

---

mechanisms. Ensuring that all local occurrences are treated consistently by using the same number of high-weighted items across the entire system. By selecting the smallest m, the global mechanism avoids overestimating the number of high-weighted items. This unified approach balances the trade-offs between local variations and global consistency in a differentially private setting.

D. BRR mechanism for natural number sequence

Let λ represent the distance between two numbers, using the Euclidean distance as the measure of utility loss, which is represented by the absolute value of the difference. Starting from any integer point, when the distance between two numbers is smaller, λ is smaller. After sorting λ, we have $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. For the numbers $1 \sim N$, the distance array $\lambda_{[i]}$ for item 1 is $[0, 1, 2, 3, \cdots, N - 1]$; the distance array for item 2 is $[0, 1, 1, 2, 3, \cdots, N - 2]$. Initially, the corresponding probability distribution is $s_1 = e^\epsilon$, with the rest s_i being 1. To minimize utility loss, we need to increment $s_2, \cdots$ successively until some s_i satisfies $\frac{\partial Q}{\partial s_i} = \frac{\Sigma_j(\lambda_i - \lambda_j)s_j}{(\Sigma_j s_j)^2} \geq 0$, which depends on the sign of the numerator $\Sigma_j(\lambda_i - \lambda_j)s_j$ in the partial derivative result.

Since the distances are sorted in ascending order, for adjacent items k' and k each starting from their respective prior positions, there are j_0 distances that are equal, as shown in

the elliptical region in Figure 3, i.e., $\lambda'_1 = \lambda_1$, $\lambda'_2 = \lambda_2, \cdots,$ $\lambda'_{j_0} = \lambda_{j_0}$. For the remaining $N - j_0$ distances, we have $\lambda'_{j_0+1} = \lambda_{j_0+1} + 1$, $\lambda'_{j_0+2} = \lambda_{j_0+2} + 1, \cdots, \lambda'_N = \lambda_N + 1$.

When considering increasing the probability weight s_i, if $i \leq j_0$, then $\lambda'_i = \lambda_i$. when $j \leq j_0$, $\lambda'_j = \lambda_j$; and when $j > j_0$, $\lambda'_j = \lambda_j + 1$. Let

$$\sum_j (\lambda'_i - \lambda'_j)s_j$$

$$= \sum_{j=1}^{j_0} (\lambda'_i - \lambda'_j)s_j + \sum_{j=j_0+1}^{N} (\lambda'_i - \lambda'_j)s_j$$

$$= \sum_{j=1}^{j_0} (\lambda_i - \lambda_j)s_j + \sum_{j=j_0+1}^{N} (\lambda_i - (\lambda_j + 1))s_j$$

$$= \sum_{j=1}^{j_0} (\lambda_i - \lambda_j)s_j + \sum_{j=j_0+1}^{N} (\lambda_i - \lambda_j)s_j - \sum_{j=j_0+1}^{N} s_j$$

$$= \sum_j (\lambda_i - \lambda_j)s_j - \sum_{j_0+1}^{N} s_j$$

$$\leq \sum_j (\lambda_i - \lambda_j)s_j.$$

In Figure 3, this shows the monotonicity relationship within the same column in the lower triangular red region, specifically indicating that the numerator factor of the partial derivative of utility loss for the item below is greater than the corresponding factor for the item above when considering increasing the probability weight.

If $i > j_0$, then $\lambda'_i = \lambda_i + 1$. When $j \leq j_0$, $\lambda'_j = \lambda_j$; and when $j > j_0$, $\lambda'_j = \lambda_j + 1$. Therefore,

$$\sum_j (\lambda'_i - \lambda'_j)s_j$$

$$= \sum_{j=1}^{j_0} (\lambda'_i - \lambda'_j)s_j + \sum_{j=j_0+1}^{N} (\lambda'_i - \lambda'_j)s_j$$

$$= \sum_{j=1}^{j_0} (\lambda_i + 1 - \lambda_j)s_j + \sum_{j=j_0+1}^{N} (\lambda_i + 1 - (\lambda_j + 1))s_j$$

$$= \sum_{j=1}^{j_0} (\lambda_i - \lambda_j)s_j + \sum_{j=j_0+1}^{N} (\lambda_i - \lambda_j)s_j + \sum_{j=1}^{j_0} s_j$$

$$= \sum_j (\lambda_i - \lambda_j)s_j + \sum_{j=1}^{j_0} s_j$$

$$> \sum_j (\lambda_i - \lambda_j)s_j.$$

In Figure 3, this demonstrates the monotonicity relationship within the same column in the upper triangular green region, specifically indicating that the numerator factor of the partial derivative of utility loss for the item below is less than the corresponding factor for the item above.

Based on Figure 3, if the threshold value i for changing the sign of the partial derivative appears before j_0 (in the

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	......	λ_n
1	0	1	2	3	4	5	6	7	8	9	10	11	......	N-1
2	0	1	1	2	3	4	5	6	7	8	9	10	......	N-2
3	0	1	1	2	2	3	4	5	6	7	8	9	......	N-3
4	0	1	1	2	2	3	3	4	5	6	7	8	......	N-4
5	0	1	1	2	2	3	3	4	4	5	6	7	......	N-5
6	0	1	1	2	2	3	3	4	4	5	5	6	......	N-6
7	0	1	1	2	2	3	3	4	4	5	5	6	......	N-7
⋮														⋮
N/2	0	1	1	2	2	3	3	4	4	5	5	6	......	N/2

Fig. 3. Monotonicity comparison of the BRR mechanism with Euclidean distance utility loss for array 1-N

red region), the lower rows will always reach the threshold before the upper rows. In this case, starting from the midpoint $N/2$ as the prior, the bottom-most row achieves the smallest m. Conversely, if the threshold appears after j_0 (in the green region), the row k will reach the threshold before k'. In this scenario, starting from the edge points 1 or N as the prior, the top-most row achieves the smallest m. Figure 3 depicts the situation where N is even; the case for odd N is similar and will not be elaborated further.

Hence, it can be seen that the minimum m is obtained when the true term is at the extreme point or the middle point. If it is at the extreme point, we have

$$\sum_{j=1}^{i-1} (\lambda_i - \lambda_j)e^\epsilon + \sum_{j=i+1}^{N} (\lambda_i - \lambda_j)1$$

$$= \frac{(1+i-1)(i-1)}{2}e^\epsilon - \frac{(1+N-i)(N-i)}{2}$$

$$= \frac{i(i-1)}{2}e^\epsilon - \frac{(N-i)^2}{2} - \frac{N-i}{2} \quad (8)$$

$$= \frac{1}{2}(e^\epsilon - 1)i^2 + \left(N - \frac{1}{2}e^\epsilon + \frac{1}{2}\right)i - \frac{1}{2}N^2 - \frac{1}{2}N$$

which is a quadratic function in i, where $\Delta = N^2 e^\epsilon + \frac{1}{4}(1 - e^\epsilon)^2 > 0$. i is a positive integer, and we get $m_1 = i = \left\lfloor \frac{\sqrt{N^2 e^\epsilon + \frac{1}{4}(1-e^\epsilon)^2} - (N - \frac{e^\epsilon}{2} + \frac{1}{2})}{e^\epsilon - 1} \right\rfloor$.

Since the arrangement of the median depends on N, if the minimum m is obtained at the middle point, it is necessary to consider the two different cases when N is odd or even. In the median, there are many paired values, making the paired i always odd. Here, only the case where i is odd needs to be considered.

For even N:

$$\sum_{j=1}^{i-1} (\lambda_i - \lambda_j)e^\epsilon + \sum_{j=i+1}^{N} (\lambda_i - \lambda_j) \times 1$$

$$= \frac{(e^\epsilon - 1)}{4}i^2 + \frac{N - (e^\epsilon - 1)}{2}i + \frac{(e^\epsilon - 1) - N^2 - 2N}{4} \quad (9)$$

where $\Delta = \frac{e^\epsilon}{4}N^2$. We have $i = \left\lfloor \frac{N}{e^{\frac{\epsilon}{2}} + 1} + 1 \right\rfloor$.

For odd N:

$$\sum_{j=1}^{i-1}(\lambda_i - \lambda_j)e^\epsilon + \sum_{j=i+1}^{N}(\lambda_i - \lambda_j) \times 1$$

$$= \frac{i-1}{2}e^\epsilon + (1 + 2 + \cdots + \frac{i-1}{2} - 1)e^\epsilon \times 2$$

$$- (1 + 2 + \cdots + \frac{N-1}{2} - \frac{i-1}{2}) \times 2$$

$$= \frac{(e^\epsilon - 1)}{4}i^2 + \frac{N - (e^\epsilon - 1)}{2}i + \frac{(e^\epsilon - 1) - 2N - N^2 + 1}{4}$$

$$(10)$$

where $\Delta = \frac{e^\epsilon}{4}(N^2 - 1) + \frac{1}{4}$. We have $i = \left\lfloor \frac{\sqrt{e^\epsilon(N^2-1)+1}-N}{e^\epsilon-1} + 1 \right\rfloor$. If i is odd, then $m_2 = i$; if i is even, then $m_2 = i + 1$.

In conclusion, we just need to compare the m values obtained at the extreme point and the middle point, and take the smaller one as the final result $m = \min\{m_1, m_2\}$.

For ease of discussion, let $y = \sum_{j=1}^{i-1}(\lambda_i - \lambda_j)e^\epsilon + \sum_{j=i+1}^{N}(\lambda_i - \lambda_j)1$. If the minimum m is obtained at the extreme point, we have $y_1 = \frac{(e^\epsilon-1)}{2}i^2 + \left(N - \frac{e^\epsilon}{2} + \frac{1}{2}\right)i - \frac{N^2}{2} - \frac{N}{2}$. If the minimum m is obtained at the middle point, then for even N, we have $y_2 = \frac{(e^\epsilon-1)}{4}i^2 + \frac{N-(e^\epsilon-1)}{2}i + \frac{(e^\epsilon-1)}{4} - \frac{N}{2} - \frac{N^2}{4}$, and for odd N, we have $y_3 = \frac{(e^\epsilon-1)}{4}i^2 + \frac{N-(e^\epsilon-1)}{2}i + \frac{(e^\epsilon-1)}{4} - \frac{N}{2} - \frac{N^2}{4} + \frac{1}{4}$. It can be observed that y_3 can be obtained by shifting y_2 upwards by $\frac{1}{4}$ units. Therefore, the left root of y_3 is greater than the left root of y_2, and the right root of y_3 is smaller than the right root of y_2. The left root of y_3 is $i_{3l} = 1 - \frac{N+\sqrt{e^\epsilon(N^2-1)+1}}{e^\epsilon-1} < 1$, thus the left root of y_2 is also less than 1.

To analyze which of m_1 or m_2 is larger, it is necessary to compare the right root of y_1, i_{1r}, with the right root of y_3 when N is odd, i_{3r}.

$$i_{1r} = \frac{\sqrt{N^2e^\epsilon + \frac{1}{4}(1-e^\epsilon)^2} - (N - \frac{e^\epsilon}{2} + \frac{1}{2})}{e^\epsilon - 1}$$

$$< \frac{Ne^{\frac{\epsilon}{2}} + \frac{1-e^\epsilon}{2} - (N - \frac{e^\epsilon}{2} + \frac{1}{2})}{e^\epsilon - 1} = \frac{Ne^{\frac{\epsilon}{2}} - N}{e^\epsilon - 1},$$

$$i_{3r} = \frac{\sqrt{e^\epsilon(N^2-1)+1} - (N - e^\epsilon + 1)}{e^\epsilon - 1}$$

$$> \frac{e^{\frac{\epsilon}{2}}(N-1) - (N - e^\epsilon + 1)}{e^\epsilon - 1}.$$

$$(11)$$

Let $a = Ne^{\frac{\epsilon}{2}} - N$, $b = e^{\frac{\epsilon}{2}}(N-1) - (N - e^\epsilon + 1)$, then $b - a = e^\epsilon - e^{\frac{\epsilon}{2}} - 1 = (e^{\frac{\epsilon}{2}} - \frac{1}{2})^2 - \frac{5}{4}$. Setting $b - a > 0$, we solve for $\epsilon > 2\ln\left(\frac{\sqrt{5}+1}{2}\right) \approx 0.9624$. Therefore, when $\epsilon \geq 1$, we always have $i_{3r} > i_{1r}$, and $m_1 = \lfloor i_{1r} \rfloor$, $m_2 = \lfloor i_{3r} \rfloor$, hence $m_1 \leq m_2$. This shows that when $\epsilon \geq 1$, the minimum m is obtained at the extreme point.

V. EXPERIMENTS

In this section, we conducted four distinct sets of comparative experiments. In the first set of experiments, we utilized the generalized Jaccard similarity coefficient (Tinimoto

coefficient) as the similarity function to compare the utility performance of BRR, GRR, and the exponential mechanism under this similarity definition. The second set of experiments employed the Euclidean distance as the similarity function, using it as a measure of utility loss to compare the utility loss under different noise addition schemes. In the third set of experiments, we applied the BRR mechanism to privacy-preserving decision tree pruning. Lastly, in the fourth set of experiments, we incorporated the BRR mechanism into the stochastic gradient descent (SGD) process.

A. Experiments with Jaccard Similarity

For a sequence of N numbers, $1 \sim N$, the similarity is defined as the pairwise generalized Jaccard similarity coefficient (Tinimoto coefficient):

$$\lambda(x, y) = \frac{xy}{x^2 + y^2 - xy}.$$

In the experimental setup, we aim to compare and analyze the performance of three different differential privacy protection mechanisms—BRR, GRR, and the Exponential mechanism, under varying privacy budgets, with sample sizes N set at 20, 40, 60, 80, and 100 respectively. The objective is to understand the applicability and utility of these mechanisms across different data scales and levels of privacy protection. Through experimental data, we hope to uncover the differences in how these mechanisms maintain data quality while ensuring data privacy, as N varies.

Fig. 4. Experimental results comparing the Q_value of BRR, GRR, and Exponential mechanisms using Jaccard similarity with varying privacy budgets and sample sizes

From Figure 4, it can be observed that, for the same N, the utility increases with higher privacy budgets across all mechanisms. This trend is inherent to the properties of differential privacy: with higher privacy budgets, less noise is added, thereby enhancing data utility. When examined vertically, it is evident that the results obtained using the BRR mechanism surpass those of the GRR and exponential mechanisms. These findings indicate that the BRR mechanism enhances data utility. The experimental results thus substantiate our proposed theory.

B. Experiments with Euclidean Distance

In Section 3, we have detailed the specific location of the m-value in the BRR mechanism when the similarity function is considered as the Euclidean distance. In this context, we need to identify the specific value that makes the partial derivative $\frac{\partial Q}{\partial s_i} < 0$. This step is crucial for optimizing the BRR mechanism, as it ensures that the selected m-value can maximize the reduction of utility loss within the Euclidean distance framework of the similarity function.

Fig. 5. Utility loss comparison of BRR, GRR, and Exponential mechanisms using Euclidean distance similarity with varying privacy budgets and sample sizes

From figure 5 we observed that, under the same privacy budget, the utility loss with the BRR mechanism is consistently lower than that of the GRR and exponential mechanisms. Since our goal is to minimize utility loss, this indicates that the BRR mechanism provides more effective data. The superior performance of the BRR mechanism in maintaining data utility underscores its efficacy and potential for practical applications in privacy-preserving data analysis.

By comparing the utility performance of the generalized Jaccard correlation coefficient (Tinimoto coefficient) and Euclidean distance as similarity functions. We verify the superiority of the BRR mechanism under different similarity function definitions, which further supports the views and conclusions we put forward in the theoretical analysis. Experimental results show that under the same privacy budget, the BRR mechanism can not only provide higher data utility, but also have lower utility loss than other data perturbation mechanisms, such as GRR and exponential mechanism. These findings not only prove the effectiveness of the BRR mechanism, but also provide strong support for its practical application.

C. Application of BRR in Privacy-Preserving Decision Tree Training

Decision tree is a supervised learning algorithm employed for both classification and regression tasks, extensively utilized in various data mining and machine learning applications.

In the realm of Gradient Boosting Decision Trees (GBDT), pruning entails curtailing the expansion of decision trees to mitigate overfitting and bolster the model's generalization capability. While GBDT primarily hinges on the aggregation of multiple weak learners (usually shallow decision trees), pruning remains instrumental in the construction of these foundational learners. Within the paradigm of differential privacy, appending noise to each leaf node subsequent to pruning can synergistically harness the advantages of both pruning and noise addition. This experiment is mainly compared with DPBoost by Li et al. (2022) [42]. When training DP-GBDT, we need to perturb the gradient value of the leaf node. According to the literature [42], the sensitivity of the noise added during each clipping is $\Delta f = \min(\frac{g_l^*}{1+\lambda}, 2g_l^*(1-\eta)^{t-1})$. During the DP-GBDT training process, the gradient clipping range c needs to be consistent with the sensitivity Δf of the leaf node to control the noise scale. To control the range of perturbations for leaf node values, an interval length L is set to evenly divide the leaf node value range $[-c, c]$ into $n = \frac{2c}{L} + 1$ discrete values. Then, the leaf node values are adjusted to the nearest discrete points to ensure alignment. Finally, the perturbed leaf node values after applying the BRR mechanism are used.

For the regression tasks on the abalone, Bias and superconduct(sup) datasets, RMSE (Root Mean Squared Error) was used as the evaluation metric. The number of instances and features for these datasets are (4177, 8), (7588, 23), and (21263,82) respectively. The experiment compared the performance of applying BRR and Laplace mechanisms for noise addition under different privacy budgets. As shown in figure 6, the results indicate that the BRR mechanism consistently outperforms the Laplace mechanism in terms of regression performance. Moreover, the BRR mechanism demonstrates greater stability across various privacy budgets, highlighting its advantage in balancing privacy preservation with model accuracy.

D. Application of BRR in SGD

The application of differential privacy techniques to safeguard the privacy of training data has gained considerable traction in deep learning research (e.g., [21], [33]). Stochastic Gradient Descent (SGD), a widely adopted optimization algorithm, has been refined through critical enhancements, particularly in the treatment of gradient clipping and noise injection. These modifications ensure that the final model parameters adhere to differential privacy standards. By incorporating noise and applying gradient clipping at each iteration of the descent process, the influence of individual data points on the overall model is rigorously constrained.

The experiment conducted is based on the differential private SGD algorithm [21]. In the experiment , the BRR mechanism is compared with the Laplace mechanism, with a focus on analyzing how these two noise mechanisms affect model accuracy under varying privacy budgets. NP-SGD is the base scheme where no noise is added. The MNIST dataset, a widely-used benchmark for handwritten digit recognition, was

Fig. 6. Comparison of BRR and Laplace mechanisms in privacy-preserving decision tree training based on RMSE with varying privacy budgets

Fig. 7. Accuracy dynamics of differentially private SGD with BRR and Laplace mechanisms compared to Non-Private SGD (NP-SGD) across training iterations under varying privacy budgets

chosen for the classification task. Throughout the experiment, model performance is evaluated using accuracy as the primary metric, the prediction results of the model are compared with the real labels of the test set and the accuracy is output, with the aim of assessing the effectiveness of these noise mechanisms across different privacy budget conditions.

As shown in Figure 7, with smaller privacy budgets, the injected noise increases, resulting in a decline in the accuracy of model updates. Under these conditions, the Laplace mechanism introduces larger noise magnitudes, leading to significant fluctuations in accuracy and difficulty in maintaining stability. In contrast, the BRR mechanism demonstrates better stability, although its convergence is slower. As the privacy budget

increases, the noise intensity decreases, reducing its impact on model updates. The Laplace mechanism can achieve higher accuracy more quickly; however, its fluctuations persist. On the other hand, the BRR mechanism exhibits a more stable convergence process and eventually reaches, or even surpasses, the accuracy of the Laplace mechanism. At high privacy budget levels, both mechanisms inject smaller amounts of noise, and the model's performance approximates a noise-free scenario. At this stage, the gap in accuracy between the Laplace and BRR mechanisms narrows, and both ultimately achieve similarly high accuracy levels. Overall, the BRR mechanism proves to be more advantageous in terms of stability and final accuracy, particularly when balancing privacy protection with

(a) Gowalla   (b) FourSquare

Fig. 8. Dataset partition distribution map

(a) Gowalla   (b) FourSquare

Fig. 9. Data utility comparison of L-SRR, GRR, and BRR under the varying privacy budgets.

(a) US Census   (b) bank

Fig. 10. Comparison of model performance between the BRR and the Laplace-based DNN-DP approach on the US Census and bank datasets under varying privacy budgets.

maintaining strong model performance.

E. Application of BRR in LBS

In location-based services (LBS), ensuring user privacy while maintaining data utility is a significant challenge. L-SRR, proposed by Wang et al. (2022) [43], employs a staircase mechanism to enhance privacy protection while effectively improving data utility. Therefore, in our evaluation of BRR mechanism in the LBS context, we compare its performance with that of L-SRR.

We experimented using Gowalla and FourSquare datasets. Extract 1,000 locations from the Gowalla dataset and 3,000 locations from the FourSquare dataset to preprocess the dataset. To ensure consistency and facilitate comparison, following the hierarchical coding approach of the L-SRR scheme, we partition the dataset accordingly based on a common prefix, as shown in Figure 8. This zoning ensures that when BRR and GRR are applied to location perturbations, the perturbations are confined to each zone and do not affect locations in the other zones. Make sure that the average size of the domains is equal to each other for all schemes. The size of each domain is represented by the number of locations inside.

The experimental results are shown in Figure 9, where we compare the performance of three schemes (L-SRR, GRR, and BRR) under different privacy budgets (ϵ ranging from 0.25 to 2). From the results, it is observed that the BRR scheme performs the best across all privacy budgets. On the Gowalla dataset, the average Qloss of the BRR is reduced

by 8.7% compared to the L-SRR, with the most significant improvement of 23.4% when $\epsilon = 2.0$. Similarly, on the FourSquare dataset, the average Qloss of BRR decreased by 6.7%, with the highest improvement observed at $\epsilon = 0.25$ by 14.3%. Overall, when the privacy budget is low, the data utility of all schemes is significantly affected, but BRR still outperforms the other schemes, demonstrating better noise resistance. As the privacy budget increases, the overall utility loss gradually decreases, with BRR maintaining relatively optimal accuracy across the entire range.

F. Application of BRR in Vector Perturbation

To evaluate the effectiveness of the BRR mechanism in vector perturbation, we adopted the DNN-DP framework proposed by Wang et al. [44] as a comparative baseline. This framework is a deep neural network training architecture that supports differential privacy by incorporating the Laplace mechanism for data perturbation. To ensure a fair comparison, we replaced the original Laplace noise by BRR mechanism while keeping the rest of the DNN-DP architecture unchanged. This allows us to improve the two mechanisms' abilities by uniform hyper-parameter optimization to preserve model performance under privacy constraints.

The experiments were conducted on two publicly available datasets: the US Census dataset and the bank-additional-full.csv dataset, which contains records from a direct marketing campaign conducted by a Portuguese bank. The former contains 14 features that were used by Wang et al. in the initial DNN-DP experiments, while the latter has been widely adopted in taxonomic studies and contains 20 features. In order to ensure the fairness and comparability of experimental results, we adopted a uniform hyperparameter optimization strategy in the experiments. During the experiments, we varied the privacy budget within the range of 0.25 to 3.5 to systematically evaluate the performance differences between the two mechanisms under different levels of privacy protection.

As shown in Figure 10, the experimental results on the US Census dataset demonstrate that the BRR-based perturbation approach consistently outperforms the traditional DNN-DP

framework and also the results shown in [44] as the privacy budget increases from 0.25 to 3.5.

Similar results were observed when the same experiments were conducted on the bank dataset, where the BRR mechanism again exhibited superior performance compared to the Laplace-based DNN-DP method. These findings suggest that BRR, as a novel noise injection strategy, can more effectively preserve—or even enhance—model utility while ensuring data privacy.

VI. DISCUSSION

A. Bipartite Randomized Response using Weighted Average m

The BRR mechanism proposed in this paper has been described in detail above, focusing on how to determine how many items have a publishing weight from 1 to e^ϵ, i.e., to determine the value of m. However, if we know the a prior probability distribution on the set of real items N, we can find a more suitable m. The detailed process will be described next. In the same way, considering the real occurrence k, we first sort the similarity $\lambda_i^{(k)}$ and weights $s_i^{(k)}$ as follows:

$$\lambda_1^{(k)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k)}, \quad s_1^{(k)} \geq s_2^{(k)} \geq \cdots \geq s_N^{(k)}.$$

The weights $s_i^{(k)}$ are summed and normalized to generate $w_i^{(k)} = \frac{s_i^{(k)}}{\Sigma_j s_j^{(k)}}$ with satisfying $\Sigma_i w_i^{(k)} = 1$. The global utility Q_g is defined as:

$$Q_g = \Sigma_k \pi(k) \Sigma_i \lambda_i^{(k)} w_i^{(k)} = \Sigma_k \frac{\pi(k) \Sigma_i \lambda_i^{(k)} s_i^{(k)}}{\Sigma_j s_j^{(k)}}, \quad (12)$$

where π represents the prior occurrence probability distribution over the set of real terms N, which can be determined based on historical frequency records or recommended frequency collection. Here, π can be assumed to be uniform, with $\pi(k) = 1/N$. Similarly, for the derivative with respect to weights $s_i^{(k)}$:

$$\frac{\partial Q_g}{\partial s_i^{(k)}} = \frac{\pi(k) \sum_j (\lambda_i^{(k)} - \lambda_j^{(k)}) s_j^{(k)}}{(\sum_j s_j^{(k)})^2}. \quad (13)$$

To ensure a unified count m of high-weighted terms, we can treat $s_i^{(k)}$ uniformly as s_i. Thus, Q_g becomes:

$$Q_g = \sum_k \frac{\sum_i \lambda_i^{(k)} s_i}{\sum_j s_j}. \quad (14)$$

Taking the derivative:

$$\frac{\partial Q_g}{\partial s_i} = \frac{\pi(k)}{(\sum_j s_j)^2} \sum_j \left(\sum_k (\lambda_i^{(k)} - \lambda_j^{(k)}) \right) s_j, \quad (15)$$

with $\pi(k) = \frac{1}{N}$. Similar to the case of local single-point release, in the expression above, the factor before s_i (numerator part), $\sum_k (\lambda_i^{(k)} - \lambda_j^{(k)}) = 0$, where $i = j$. Similarly, the monotonicity of Q_g with respect to s_i is independent of the values of s_i. Based on this, to maximize the utility Q_g, we initialize it as the GRR mechanism, where $s_1 = e^\epsilon$

Algorithm 3 BRR*: Modified BRR with average m

Input: utility matrix $\{\lambda_j^{(k)}\}_{N \times N}$ with $\lambda_1^{(k)} \geq \lambda_2^{(k)} \geq \cdots \geq \lambda_N^{(k)} > 0$, privacy budget ϵ

1: Initialize: $s_1 = e^\epsilon$, $s_2 = \cdots = s_N = 1$, $w_1 = \frac{e^\epsilon}{e^\epsilon + N - 1}$, $w_2 = \ldots = w_N = \frac{1}{e^\epsilon + N - 1}$, $Q_g = \frac{1}{N} \Sigma_k \Sigma_i \lambda_i^{(k)} w_i$, $m = 1$

2: **for** $i = 2$ **to** N **do**
3:     Compute $\frac{\partial Q_g}{\partial s_i}$ by (15)
4:     **if** $\frac{\partial Q_g}{\partial s_i} > 0$ **then**
5:         $s_i = e^\epsilon$, $m = i$
6:     **else**
7:         Break
8:     **end if**
9: **end for**
10: **for** $i = 1$ **to** N **do**
11:     $w_i = \frac{s_i}{\Sigma_j s_j}$
12: **end for**
Output: $(w_1, \ldots, w_N)$ and m

---

and $s_2, \ldots, s_N$ are uniformly assigned the minimum value 1. We then incrementally increase s_2 to the maximum value s_1, and successively increase $s_3, s_4, \ldots$ until some s_i satisfies $\frac{\partial Q_g}{\partial s_i} \leq 0$. At this point, $s_i, \ldots, s_N$ remain unchanged and are uniformly assigned the minimum value 1. Algorithm 3 outlines the detailed procedure for determining the modified BRR with average m.

Thus, for the global scenario, considering all possible real occurrences (prior terms), under the condition of a unified count m of high-weighted terms, we can find an optimized discrete probability distribution for m. For each prior real term, the distribution ensures that m items each have the weight e^ϵ, and the remaining items have the weight 1. This approach towards global BRR considers all prior real occurrences and optimizes the distribution of m high-weighted items, ensuring differential privacy. We refer to this mechanism derived from the extension of BRR as BRR*.

B. Comparison of m-Values in BRR and BRR*

The distribution trend of m-values was studied under the condition where similarity is measured by Euclidean distance and the privacy budgets $\epsilon = 1, 3, 5$. A linear model $y = kx$ was used to fit the BRR scheme and the BRR* scheme separately. The $y = kx$ model effectively captured the trend under varying privacy budgets, with the fitted curve closely matching the data points.

Figure 11 clearly demonstrates how an increase in the privacy budget leads to a decrease in the slope, meaning that the growth rate of the output value m with respect to the input n slows down as the privacy budget increases. This is because as the privacy budget ϵ increases, the strength of privacy protection weakens, reducing the dependency of the output value m on the input n. Consequently, the BRR and BRR* mechanisms become closer to the GRR mechanism.

Fig. 11. Fitted trends of value m in BRR and BRR* mechanisms using linear models

The slope of the BRR* is always greater than that of the BRR under different privacy budgets. This is reasonable, because the m chosen in the BRR mechanism is the minimum m value obtained from different terms, while the BRR* mechanism considers the global utility, which can better maintain data dependencies while protecting privacy.

C. Utility Loss Comparison of BRR, BRR*, and GRR

This experiment aims to evaluate the utility performance of the BRR and BRR* mechanisms under different privacy budgets (ϵ), comparing them to the GRR mechanism. The primary objective is to explore the trade-off between privacy protection and utility loss among these three mechanisms. The GRR mechanism adopts a fixed value of $m = 1$, representing a classical randomized response method. The BRR mechanism determines the value of m using Algorithm 2, which calculates the optimal m for each point based on utility maximization conditions and selects the smallest m as the global value. In contrast, the BRR* mechanism leverages prior probabilities and determines m using Algorithm 3, aiming to minimize global utility loss. The experiment employs Euclidean distance as the similarity function and utility loss as the evaluation metric, setting privacy budgets at $\epsilon = 3, 5, 8$, and examines the utility loss variations of each mechanism across the region.

The results, presented in Figure 12, demonstrate that under a low privacy budget ($\epsilon = 3$), the GRR mechanism exhibits the highest utility loss and performs the worst. This is primarily due to severe information loss caused by the strong randomization with fixed $m = 1$. In contrast, the BRR and BRR* mechanisms show significantly reduced utility loss, with comparable overall performance. Notably, BRR* achieves lower utility loss at certain points, highlighting its advantage of global optimization. When the privacy budget increases to $\epsilon = 5$, the utility loss of the GRR mechanism decreases but remains inferior to the other two mechanisms. The BRR mechanism, which adopts a local optimization strategy by selecting the smallest m, further reduces utility loss. The BRR* mechanism, benefiting from the global utility

optimization strategy, demonstrates superior performance in most regions. As the privacy budget increases further to $\epsilon = 8$, the m value of the BRR mechanism degrades to 1, resulting in utility loss identical to that of the GRR mechanism, making their performances equivalent. However, the BRR* mechanism continues to maintain lower utility loss in most regions due to its global optimization approach. While it exhibits slightly higher utility loss than the BRR mechanism at a few points, its overall utility remains advantageous.

These findings indicate that under low privacy budget conditions, the BRR* mechanism outperforms both the GRR and BRR mechanisms, demonstrating superior utility. This highlights that incorporating prior probabilities and designing mechanisms based on global utility optimization can effectively reduce utility loss or enhance utility while maintaining privacy protection.

VII. CONCLUSION

In practical applications, the core challenge for LDP mechanisms lies in how to maximize data utility while ensuring privacy protection. This paper addresses this issue by proposing a novel adaptive LDP mechanism, the Bipartite Randomized Response (BRR), which aims to balance the trade-off between privacy and data utility. The paper provides a theoretical analysis and experimental validation of this approach. Initially, it systematically reviews traditional differential privacy mechanisms, including RR, GRR, the Exponential Mechanism, and the Laplace Mechanism. While these mechanisms have made significant contributions to privacy protection, they exhibit limitations in terms of data utility. In particular, the GRR mechanism treats all non-true data values equally, failing to account for data similarity, which reduces the utility of data queries.

To address this limitation, the BRR mechanism introduces data similarity by partitioning the data domain into values close to the true value and those farther away, assigning different release probabilities to each group. Specifically, BRR optimizes the allocation of release probabilities, ensuring that values close to the true data receive higher release probabilities, while distant values receive lower probabilities. This optimization not only improves data utility but also enhances privacy protection. The paper theoretically proves that the problem of maximizing privacy and utility can be resolved by determining the optimal allocation of release probabilities, which can be efficiently solved as a linear programming problem. The computational complexity of this solution is theoretically low, making it feasible for practical applications.

In the experimental section, four sets of experiments were conducted to validate the superiority of the BRR mechanism across various application scenarios. First, the utility performance of BRR, GRR, and the Exponential Mechanism was compared using the generalized Jaccard similarity coefficient and Euclidean distance as similarity functions, with results showing that BRR outperformed the other mechanisms in terms of utility loss. Subsequently, BRR was applied to

Fig. 12. Comparison of normalized_Q_Loss among the BRR, BRR*, and GRR mechanisms under varying privacy budgets. The analysis evaluates the impact of mechanism-specific m-value selection strategies on utility performance across different regions

privacy-preserving decision tree pruning and stochastic gradient descent (SGD), further demonstrating its ability to enhance model accuracy while maintaining strong privacy protection.

Finally, we explored the utilization of the prior information regarding the publication probability of occurrence term within publication domain N, in order to identify a more precise and appropriate value of m that is grounded in a global context. We provided a detailed description of the specific derivation process.

REFERENCES

[1] Lea Demelius, Roman Kern, and Andreas Trügler. Recent Advances of Differential Privacy in Centralized Deep Learning: A Systematic Survey. *ACM Computing Surveys*, 57(6), 2025.

[2] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[3] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[4] Naoise Holohan, Douglas J. Leith, and Oliver Mason. Optimal Differentially Private Mechanisms for Randomised Response. *IEEE Transactions on Information Forensics and Security*, 12(11):2726–2735, 2017.

[5] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal Mechanisms for Local Differential Privacy. *Journal of Machine Learning Research*, 17(17):1–51, 2016.

[6] Héber H. Arcolezi, Jean François Couchot, Bechara Al Bouna, and Xiaokui Xiao. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. *Digital Communications and Networks*, 10(2):369–379, 2024.

[7] Makhlouf Karima, Arcolezi Heber H., Zhioua Sami, Ben Brahim Ghassen, and Palamidessi Catuscia. On the impact of multi-dimensional local differential privacy on fairness. *Data Mining And Knowledge Discovery*, 38(4):2252–2275, 2024.

[8] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local Privacy and Statistical Minimax Rates. *CoRR*, abs/1302.3203, 2013.

[9] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. A Primer on ε-Differential Privacy. In *Differential Privacy: From Theory to Practice*, pages 7–31. Springer, 2017.

[10] Rui Chen, Haoran Li, A Kai Qin, Shiva Prasad Kasiviswanathan, and Hongxia Jin. Private spatial data aggregation in the local setting. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 289–300. IEEE, 2016.

[11] Haina Song, Tao Luo, Xun Wang, and Jianfeng Li. Multiple Sensitive Values-Oriented Personalized Privacy Preservation Based on Randomized Response. *IEEE Transactions on Information Forensics and Security*, 15:2209–2224, 2020.

[12] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.

[13] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 212–229, 2018.

[14] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

[15] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal*, 8(11):8836–8853, 2020.

[16] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Hang Su, Bo Zhang, and H Vincent Poor. User-level privacy-preserving federated learning: Analysis and performance optimization. *IEEE Transactions on Mobile Computing*, 21(9):3388–3401, 2021.

[17] Xuanyu Bai, Jianguo Yao, Mingxuan Yuan, Ke Deng, Xike Xie, and Haibing Guan. Embedding differential privacy in decision tree algorithm with different depths. *Science China Information Sciences*, 60:1–15, 2017.

[18] Sam Fletcher and Md Zahidul Islam. Decision Tree Classification with Differential Privacy. *ACM Computing Surveys*, 52(4):1–33, 2019.

[19] Huiyu Fang, Liquan Chen, Yali Liu, and Yuan Gao. Locally Differentially Private Frequency Estimation Based on Convolution Framework. In *2023 IEEE Symposium on Security and Privacy (S&P)*, pages 2208–2222. IEEE, 2023.

[20] Mengmeng Yang, Taolin Guo, Tianqing Zhu, Ivan Tjuawinata, Jun Zhao, and Kwok Yan Lam. Local differential privacy and its applications: A comprehensive survey. *Computer Standards & Interfaces*, 89:103827–, 2024.

[21] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[22] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 127–135, 2015.

[23] Adam Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 58–77. IEEE, 2017.

[24] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. *Advances in Neural Information Processing Systems*, 31, 2018.

[25] Yan Yan, Jianzhuang Chen, Adnan Mahmood, Xingying Qian, and Pengbin Yan. LDPORR: A localized location privacy protection method based on optimized random response. *Journal of King Saud University-Computer and Information Sciences*, 35(8):101713, 2023.

[26] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (S&P)*, pages 304–317. IEEE, 2020.

[27] Norma Gutiérrez, Beatriz Otero, Eva Rodríguez, Gladys Utrera, Sergi Mus, and Ramon Canal. A Differential Privacy protection-based federated deep learning framework to fog-embedded architectures. *Engineering Applications of Artificial Intelligence*, 130:107689, 2024.

[28] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Data poisoning attacks to local differential privacy protocols. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 947–964, 2021.

[29] Teng Wang, Jun Zhao, Zhi Hu, Xinyu Yang, Xuebin Ren, and Kwok-Yan Lam. Local differential privacy for data collection and analysis. *Neurocomputing*, 426:114–133, 2021.

[30] Yanling Wang, Qian Wang, Lingchen Zhao, and Cong Wang. Differential privacy in deep learning: Privacy and beyond. *Future Generation Computer Systems*, 148:408–424, 2023.

[31] Mehmet Emre Gursoy, Acar Tamersoy, Stacey Truex, Wenqi Wei, and Ling Liu. Secure and utility-aware data collection with condensed local differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2365–2378, 2019.

[32] Ben Niu, Yahong Chen, Boyang Wang, Zhibo Wang, Fenghua Li, and Jin Cao. AdaPDP: Adaptive Personalized Differential Privacy. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[33] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. Local Differential Privacy for Deep Learning. *IEEE Internet of Things Journal*, 7(7):5827–5842, 2020.

[34] Héber H Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. Random sampling plus fake data: Multidimensional frequency estimates with local differential privacy. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 47–57, 2021.

[35] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. "I need a better description": An Investigation Into User Expectations For Differential Privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3037–3052, 2021.

[36] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR, 2016.

[37] Tianhao Wang, Ninghui Li, and Somesh Jha. Locally differentially private heavy hitter identification. *IEEE Transactions on Dependable and Secure Computing*, 18(2):982–993, 2019.

[38] Takao Murakami and Yusuke Kawamoto. {Utility-Optimized} local differential privacy mechanisms for distribution estimation. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1877–1894, 2019.

[39] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[40] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[41] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[42] Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 784–791, 2020.

[43] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. L-SRR: Local differential privacy for location-based services with staircase randomized response. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2809–2823, 2022.

[44] Yufeng Wang, Min Gu, Jianhua Ma, and Qun Jin. Dnn-dp: Differential privacy enabled deep neural network learning framework for sensitive crowdsourcing data. *IEEE Transactions on Computational Social Systems*, 7(1):215–224, 2020.