An Attack-Agnostic Defense Framework Against Manipulation Attacks under Local Differential Privacy

Puning Zhao[1], Zhikun Zhang[2], Jiawei Dong[2], Jiafei Wu[3], Zhe Liu[3#] Shaowei Wang[4#], Yunjun Gao[2]
[1] *Shenzhen Campus of Sun Yat-sen University* [2] *Zhejiang University* [3] *Zhejiang Lab* [4] *Guangzhou University*

***Abstract*—Protection of *local differential privacy* (LDP) protocols against *manipulation attacks* is an important and challenging problem. We hope to design an *attack-agnostic* framework, which does not rely on any knowledge of attackers. An early work [1] restricts the attacker's capability by converting each sample into a binary signal. However, the compression of signal leads to severe loss of information, and thus results in unnecessary sacrifice of utility, especially when $\epsilon > 1$. In this paper, we propose a general estimation framework RobustLDP for robust estimation under LDP. The general idea is to send carefully crafted pre-defined information to all users, and then aggregate the feedback at the server. We strike a better tradeoff between preserving information and restricting the attacker's capability. We instantiate RobustLDP for frequency estimation and mean estimation in ℓ_1 and ℓ_2 support, which serve as building blocks for more advanced tasks. We also establish theoretical guarantees for all possible attacks. The result shows that our method significantly outperforms the existing one for $\epsilon > 1$. Extensive experiments on multiple real-world datasets validate the effectiveness of our method.**

1. Introduction

Local differential privacy (LDP) has become the de facto standard for sensitive user data collection [2]. The general idea of LDP is to allow individual users to perturb their raw sensitive data before uploading it to the server. It ensures that attackers with access to the perturbed outputs cannot recover the raw data while guaranteeing the aggregator's ability to extract useful statistical information from the whole population. As such, LDP has been adopted by a number of high-tech giants. For example, Google integrated RAPPOR into the Chrome browser [3], Apple deployed LDP protocols in iOS [4] to collect user preferences, Microsoft also developed LDP protocols to collect application usage data [5].

Most existing studies on LDP assume that all the data providers are trustworthy and honestly report data following the LDP protocols [6], [7]. However, some data providers can be malicious in the real world. For instance, in Google, Twitter, and Hotmail, attackers can either compromise a number of existing accounts or purchase many new fake accounts from the underground market at very low prices [8].

Corresponding authors

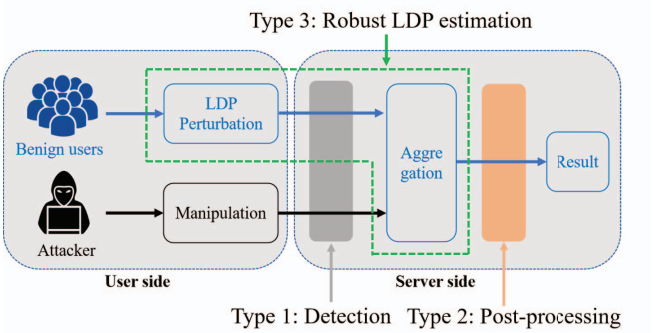


Figure 1: Illustration of three types of defense mechanisms against manipulation attacks under LDP.

These corrupted local data providers can report carefully designed data to the server, aiming to skew the final estimates, referred to as *manipulation attack*. Therefore, it is important to design effective defense strategies that can withstand these manipulation attacks.

Existing Defenses. Recent years have witnessed several defense mechanisms to mitigate the manipulation attacks [1], [9], [10], [11]. These strategies can be roughly categorized into three types: detection, post-processing, and robust estimation, as illustrated in Figure 1.

The *detection-based* methods aim to identify possibly corrupted reported data and remove them before aggregation, such as malicious user detection (MUD), conditional probability-based detection (CPAD) [10], and LDPGuard [9]. However, these methods rely on some information that is unlikely to be known in advance, such as the attack strategy and the number of corrupted samples. Moreover, according to recently established theories on robust statistics [12], [13], even under non-private setting, an attacker can corrupt a sample by roughly $\Theta(\sqrt{d})$ without being detected. Under LDP, the undetectable manipulation can become more serious. The *post-processing-based* methods aim to correct the aggregation results if they appear anomalous, such as LDPRecover [11]. However, these defense mechanisms can be easily bypassed. For instance, for the frequency estimation problem, the attacker can refine its attack to ensure that the final estimated values are nonnegative in all components and sum up to 1. Such an attack is unlikely to be corrected by post-processing based methods.

Both the detection-based and post-processing-based

2025 IEEE Symposium on Security and Privacy (SP) | 979-8-3315-2236-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/SP61157.2025.00255

© 2025, Puning Zhao. Under license to IEEE.
DOI 10.1109/SP61157.2025.00255

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

TABLE 1: Comparison of error bounds of RobustLDP and HST/EST [1] with attack, and that of without attack.

Tasks	Range of ϵ	HST/EST	RobustLDP	Without attack
Frequency estimation	$(0, 1)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d}}{n\epsilon}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d}}{n\epsilon}\right)$	$O\left(\frac{d}{\sqrt{n\epsilon^2}}\right)$ (OUE [15], RAPPOR [3])
	$[1, \ln d]$	$\tilde{O}\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{ne^\epsilon}} + \frac{q\sqrt{d}}{n\sqrt{e^\epsilon}}\right)$	$O\left(\frac{d}{\sqrt{ne^\epsilon}}\right)$ (SS [16], [17], Hadamard [18])
	$(\ln d, +\infty)$	$\tilde{O}\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{ne^\epsilon}} + \frac{q}{n}\right)$	$O\left(\frac{d}{\sqrt{ne^\epsilon}}\right)$ (kRR [14])
ℓ_1 mean estimation	$(0, 1)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d}}{n\epsilon}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d}}{n\epsilon}\right)$	$O\left(\frac{d}{\sqrt{ne^\epsilon}}\right)$
	$[1, d]$	$\tilde{O}\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon}} + \frac{q\sqrt{d}}{n\epsilon}\right)$	$O\left(\frac{d}{\sqrt{n\epsilon}}\right)$
	$(d, d^2]$	$\tilde{O}\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon}} + \frac{qd}{n\sqrt{\epsilon}}\right)$	$O\left(\frac{d}{\sqrt{n\epsilon}}\right)$
	$(d^2, +\infty)$	$\tilde{O}\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$	$\tilde{O}\left(\frac{d}{\sqrt{n\epsilon}} + \frac{q}{n}\right)$	$O\left(\frac{d}{\sqrt{n\epsilon}}\right)$
ℓ_2 mean estimation	$(0, 1)$	$O\left(\sqrt{\frac{d}{n\epsilon^2}} + \frac{q}{n}\right)$	$O\left(\sqrt{\frac{d}{n\epsilon^2}} + \frac{q}{n}\right)$	$O\left(\sqrt{\frac{d}{n\epsilon^2}}\right)$ ([2], [19], [20], [21], [22])
	$[1, +\infty)$	$O\left(\sqrt{\frac{d}{n}} + \frac{q}{n}\right)$	$O\left(\sqrt{\frac{d}{n\epsilon}} + \frac{q}{n}\right)$	$O\left(\sqrt{\frac{d}{n\epsilon}}\right)$ ([20], [21], [22])

methods are temporary fixes, which cannot address the root causes of manipulation attacks. To this end, a few researchers attempted to redesign both the perturbation and aggregation mechanisms to make them inherently robust to manipulation attacks, which we referred to as *robust estimation-based* methods. Concretely, Cheu et al. [1] proposed HST for frequency estimation and ℓ_1 mean estimation, and EST for ℓ_2 mean estimation. They design different parameters for the LDP protocols of different users. Under the guidance of such parameters, each user compresses their private data into a binary signal and then uploads it to the server passing a randomized response (RR) mechanism [14]. Intuitively, the compression operation leads to a smaller output space, thus mitigating the impact of adversarial manipulation. Nevertheless, compression also results in loss of information, which eventually leads to the degradation of performance.

Our Proposal. In this paper, we propose a new robust estimation-based method, RobustLDP, by carefully adjusting the complexity of the output space in LDP protocols. This approach mitigates manipulation while still restoring useful information. Specifically, the server sends predefined information to users, who then return a perturbed signal based on this information and their local sample. The output space complexity is determined by the privacy budget. For a small ϵ, the attacks on LDP protocols can be highly destructive, so we use a simple feedback signal with limited information to reduce the impact of manipulation. Conversely, with a larger ϵ, users can send more information to enhance the estimation accuracy. This design compresses each sample appropriately, balancing manipulation mitigation and information preservation. Additionally, the predefined information depends solely on ϵ, ensuring no additional privacy loss from parameter selection. We further instantiate RobustLDP for three basic data analysis tasks, including frequency estimation and mean estimation in ℓ_1 and ℓ_2 unit balls, which serve as building blocks for more advanced data analysis tasks [23]. Furthermore, we theoretically analyze the worst-case error bound when the robust estimation-based defenses face the optimal attack, as shown in Table 1. With $\epsilon \leq 1$, we achieve the same performance with HST/EST. With $\epsilon > 1$, RobustLDP significantly improves the error bound over HST/EST.

The improvement made by our method with $\epsilon > 1$ is important in practice. In real industrial applications, from an accuracy-first perspective, companies usually use $\epsilon > 1$ instead of $\epsilon \leq 1$. For example, Apple uses $\epsilon = 4$ to discover popular emojis and identify high energy usage in Safari [4]. In addition, to mitigate server-side computational and communication bottlenecks in large-scale LDP systems, subsampling strategies are commonly adopted. Due to privacy amplification by subsampling, it suffices to let the local privacy budget to be higher than 1 [24]. Furthermore, the shuffle model has become increasingly popular in recent years. A shuffle model usually adopts ϵ-LDP randomizers with privacy budgets exceeding 1, and the final budget under central DP after shuffling can still be smaller than 1 [25]. Therefore, cases with $\epsilon > 1$ are widely used in practice. Large ϵ is both necessary since it avoids too much performance loss, and useful since in many applications, after subsampling or shuffling, the final value of ϵ is smaller than 1.

Intuitively, such improvement can be explained by our adaptive design of the complexity of feedback signals. In particular, as shown in Table 1, in the absence of attacks (i.e. $q = 0$), the performance of our method matches the best performance among LDP protocols without considering attacks, such as OUE and kRR. Extensive experiments on multiple real-world datasets validate the effectiveness of RobustLDP. For instance, when $\epsilon = 3$, for the distribution of city from the US population in the IPUMS dataset with 2% users being attacked, our method achieves ℓ_1 error of 0.148, a 62% reduction compared with HST.

We summarize the contributions as follows.

- We propose a general framework RobustLDP for defending against manipulation attacks using robust estimation.
- We instantiate RobustLDP for frequency estimation and mean estimation in ℓ_1 and ℓ_2 support.
- We provide theoretical analysis with results summarized in Table 1. The result shows that compared with existing

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

methods, the performances are significantly improved under $\epsilon \geq 1$. Moreover, if there is actually no attack, then our results match the state-of-the-art LDP protocols, without introducing additional cost of performance.
- For each task, we design corresponding optimal attack strategies that is adaptive to sample values and the privacy budget ϵ.
- We conduct extensive experiments on multiple real-world datasets to validate the effectiveness of RobustLDP.

Roadmap. In Section 2, we present background knowledge about LDP. We introduce the problem definition and the existing solutions in Section 3. In Section 4, we provide the general idea for RobustLDP. Section 5, Section 6, and Section 7 instantiate RobustLDP for frequency estimation, ℓ_1, and ℓ_2 mean estimation. We discuss related work in Section 9 and conclude the paper in Section 10.

2. Preliminaries

2.1. Local Differential Privacy

Suppose there are n users, in which the i-th user has data $\mathbf{x}_i \in \mathcal{X}$. $\mathcal{X}$ is the support of the data. For frequency estimation problem, $\mathcal{X} = [d]$. For ℓ_1 and ℓ_2 mean estimation problems, $\mathcal{X}$ is $\mathbb{B}_1^d := \{\mathbf{u} | \|\mathbf{u}\|_1 \leq 1\}$ and $\mathbb{B}_2^d := \{\mathbf{u} | \|\mathbf{u}\|_2 \leq 1\}$, respectively. In this work, for simplicity, we only consider non-interactive LDP, which means that the output of each user does not depend on that of other users. The i-th user has a privacy mechanism Q_i. The definition of LDP is stated as follows.

Definition 1. *(LDP [26], [27]) A local randomizer Q_i satisfies ϵ-LDP if for every $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and $T \subseteq Range(Q_i)$,*

$$P(Q_i(\mathbf{x}) \in T) \leq e^{\epsilon} P(Q_i(\mathbf{x}') \in T), \tag{1}$$

in which $Range(Q_i)$ denotes the set of all possible outputs of Q_i. The randomness comes only from the local randomizer Q_i.

The user does not report the local sample $\mathbf{x}_i$ to the server. It only sends $Q_i(\mathbf{x}_i)$, thus the privacy of users are guaranteed even if the output is controlled by the attacker.

2.2. LDP Protocols

Frequency estimation protocols are used to estimate the frequency distribution of the *categorical attributes*, such as gender and race, which serves as the building block for advanced tasks, such as marginal release, range query, etc [28], [29]. Similarly, *mean estimation* protocols aim to estimate the mean value of the continuous attributes, such as weight and height, which serves as the building block for advanced tasks, such as trust evaluation, location estimation, stochastic optimization, etc [30], [31], [32]. Therefore, in this paper, we design the defense mechanism for frequency estimation and mean estimation. In the following, we introduce the commonly used protocols for both tasks.

2.2.1. Frequency Estimation. Here we introduce two common frequency estimation protocols.

k-Randomized Response (kRR). Given the input $x \in [d]$, the output follows the following distribution:

$$\mathrm{P}(Y = j|x) = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+d-1} & \text{if } x = j \\ \frac{1}{e^{\epsilon}+d-1} & \text{otherwise.} \end{cases} \tag{2}$$

The data collector receives the reported values $Y_1, \ldots, Y_n$ generated from $x_1, \ldots, x_n$. Then the frequency estimate is

$$\hat{\mu}_j = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(Y_i = j) - q}{p - q}, \tag{3}$$

in which $p = e^{\epsilon}/(e^{\epsilon} + d - 1)$, $q = 1/(e^{\epsilon} + d - 1)$.

Optimized Unary Encoding (OUE). Given $x \in [d]$, the local randomizer generates $\mathbf{Y} \in \{0, 1\}^d$ according to the following distribution:

$$\mathrm{P}(Y(j) = 1|x) = \begin{cases} \frac{1}{2} & \text{if } x = j \\ \frac{1}{e^{\epsilon}+1} & \text{otherwise.} \end{cases} \tag{4}$$

Then the frequency estimate is

$$\hat{\mu}_j = \frac{\frac{1}{n}\sum_{i=1}^{n} Y(j) - q}{p - q}, \tag{5}$$

in which $p = 1/2$, $q = 1/(e^{\epsilon} + 1)$.

If ϵ is large or d is small, then kRR is more suitable. Otherwise, OUE performs better.

2.2.2. Mean Estimation. For one dimensional mean estimation, [33] has proposed a piecewise mechanism:

$$p(y|x) = \begin{cases} p_0 & \text{if} \quad y \in [l(x), r(x)], \\ p_0/e^{\epsilon} & \text{if} \quad y \in [-C, C] \setminus [l(x), r(x)], \end{cases} \tag{6}$$

in which

$$p_0 = \frac{e^{\epsilon/2}(e^{\epsilon/2} - 1)}{2(e^{\epsilon/2} + 1)}, C = \frac{e^{\epsilon/2} + 1}{e^{\epsilon/2} - 1}, \tag{7}$$

and

$$l(x) = \frac{e^{\epsilon/2}x - 1}{e^{\epsilon/2} - 1}, r(x) = \frac{e^{\epsilon/2}x + 1}{e^{\epsilon/2} - 1}. \tag{8}$$

One dimensional mean estimation serves as a building block for mean estimation in multi-dimensional spaces [2], [19].

3. Problem Formulation and Existing Solutions

3.1. Threat Model

Attacker's goals. This paper considers the untargeted attack, in which the attacker's goal is to increase the error of the estimation results. For frequency estimation and ℓ_1 mean estimation problems, the attacker maximizes the ℓ_1 error $\|\hat{\mu} - \mu\|_1$. In ℓ_2 mean estimation problem, the attacker aims at maximizing the ℓ_2 error $\|\hat{\mu} - \mu\|_2$.

Attacker's capabilities. We assume that the attacker can control up to q users out of n users. Denote $\mathcal{C}$ as the set of corrupted users. If a user is corrupted, then it reports an

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

arbitrary message determined by the attacker, otherwise it receives signal $\mathbf{s}_i$ and sends $\mathbf{Y}_i = Q(\mathbf{x}_i, \mathbf{s}_i)$ honestly to the server:

$$\mathbf{Z}_i = \begin{cases} \mathbf{Y}_i & \text{if} \quad i \notin \mathcal{C} \\ \star & \text{if} \quad i \in \mathcal{C}. \end{cases} \tag{9}$$

Throughout the process, we assume that the attacker can manipulate the communication between the server and users arbitrarily, which means that the value of $\mathbf{Z}_i$ is entirely determined by the attacker.

Attacker's knowledge. We assume that the attacker knows the input domain $\mathcal{X}$, encoded domain $\mathcal{Y}$ and the privacy budget ϵ. Moreover, it also knows the aggregator $\mathcal{A}$, the local randomizer Q and signals $\mathbf{s}_1, \ldots, \mathbf{s}_n$. We also assume that the attacker can observe the feedback signal $\mathbf{Y}_i$ sent from users to the server. Furthermore, the attacker knows the ground truth μ. Based on this information, the attacker can design a strategy carefully to skew the estimation.

3.2. Problem Formulation

This paper proposes and analyzes robust estimation methods for the following tasks under LDP.

Frequency estimation. Suppose that samples are supported on $\mathcal{X} = [d]$. Let

$$\mu_l = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(x_i = l), \tag{10}$$

and $\mu = (\mu_1, \ldots, \mu_d)$. μ is the true frequency of the dataset. The goal is to estimate μ. Frequency estimators are evaluated by the expected ℓ_1 error $\mathbb{E}[\|\hat{\mu} - \mu\|_1]$.

ℓ_1 mean estimation. Secondly, we extend our work to ℓ_1 mean estimation. It is assumed that all samples are supported in a region constrained by ℓ_1 norm, i.e. $\mathcal{X} = \mathbb{B}_1 := \{\mathbf{u} | \|\mathbf{u}\|_1 \leq 1\}$. The goal is to estimate the sample mean $\mu = (1/n) \sum_{i=1}^{n} \mathbf{x}_i$. An important application of ℓ_1 mean estimation is the user-level frequency estimation problem, in which we have n users and each user has m samples. The user conducts one-hot encoding to all local samples and calculates their average $\mathbf{x}_i$ first, and then we need to estimate the frequency with each $\mathbf{x}_i$ protected under ϵ-LDP. For simple frequency estimation with n samples, each user contains no more than $\log_2 d$ bits. However, for ℓ_1 mean estimation, each sample can take any value in $\mathbb{B}_1$, thus the ℓ_1 mean estimation problem is more complex.

ℓ_2 mean estimation. Finally, we explore ℓ_2 mean estimation. It is assumed that all samples are supported in a unit ball with Euclidean distance, i.e. $\mathcal{X} = \mathbb{B}_2 := \{\mathbf{u} | \|\mathbf{u}\|_2 \leq 1\}$. The goal is to estimate the sample mean $\mu = (1/n) \sum_{i=1}^{n} \mathbf{x}_i$. Now we use ℓ_2 error $\|\hat{\mu} - \mu\|_2$ to evaluate the quality of mean estimation.

3.3. Existing Solutions

Here we explain the methods in [1], including the HST for frequency estimation and ℓ_1 mean estimation, and EST for ℓ_2 mean estimation.

HST. The server first sends pre-defined information $\mathbf{s}_i$, which is a vector randomly sampled from $\{-1, 1\}^d$, to each user. Given $x_i \in [d]$, let

$$Q(\mathbf{x}_i, \mathbf{s}_i) = \begin{cases} \frac{e^\epsilon + 1}{e^\epsilon - 1} s_{i,x_i} & \text{with probability} \quad \frac{e^\epsilon}{e^\epsilon + 1} \\ -\frac{e^\epsilon + 1}{e^\epsilon - 1} s_{i,x_i} & \text{with probability} \quad \frac{1}{e^\epsilon + 1}. \end{cases} \tag{11}$$

The user i reports $Y_i = Q(\mathbf{x}_i, \mathbf{s}_i)$ to the server. The final result is $\hat{\mu} = (1/n) \sum_{i=1}^{n} Y_i \mathbf{s}_i$. [1] uses the same method for frequency estimation and ℓ_1 mean estimation.

EST. The pre-defined information $\mathbf{s}_i$ is a vector drawn uniformly from the surface of d-dimensional unit ball. Given $\mathbf{x}_i \in \mathbb{B}_2$, let $w_i = \text{sign}(\langle \mathbf{s}_i, \mathbf{x}_i \rangle)$, and let

$$Q(\mathbf{x}_i, \mathbf{s}_i) = \begin{cases} \frac{e^\epsilon + 1}{e^\epsilon - 1} w_i & \text{with probability} \quad \frac{e^\epsilon}{e^\epsilon + 1} \\ -\frac{e^\epsilon + 1}{e^\epsilon - 1} w_i & \text{with probability} \quad \frac{1}{e^\epsilon + 1}. \end{cases} \tag{12}$$

The user i reports $Y_i = Q(\mathbf{x}_i, \mathbf{s}_i)$ to the server. The final result is $\hat{\mu} = (c_d/n) \sum_{i=1}^{n} Y_i \mathbf{s}_i$, in which

$$c_d = \frac{d\sqrt{\pi}\Gamma\left(\frac{d+1}{2}\right)}{2\Gamma\left(\frac{d}{2} + 1\right)}. \tag{13}$$

Drawbacks. The limitation of HST/EST is that Y_i can only take binary values, and can thus convey only one bit of information. However, the sample $\mathbf{x}_i$ contains more information, thus sending Y_i is not enough to transmit the necessary information to the server. Such information loss eventually results in some utility drop. With larger ϵ, we expect better accuracy, but such a utility drop significantly hinders the performance improvement with ϵ. This work significantly improves the performance with $\epsilon \geq 1$ by redesigning the pre-defined information as well as the feedback signals.

4. Our Proposal

4.1. Intuitions

The server sends predefined parameters to each user, and the feedback signal depends on these parameters as well as the local data. The crucial step is the design of the feedback signal. A simple feedback signal can effectively reduce the impact of manipulation attacks because attackers cannot significantly alter the result without being detected. However, a simple signal also limits the information the server receives. Conversely, a complex feedback signal allows the server to gather more information but gives attackers more opportunities for manipulation. We need to design a LDP protocol with an appropriately complex output space to balance information preservation and manipulation mitigation.

This work designs an adaptive output space. The general guideline is to let the complexity increase with the privacy budget ϵ. For small ϵ, the attack on LDP protocols can be highly destructive. Therefore, we let the server receive only limited information from each user, in order to mitigate the effect of manipulation attack. If ϵ is large, then preserving the information of users becomes more important. Therefore, we let the output space be more complex, so that the server can receive sufficient information conduct an accurate final aggregation.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

4.2. General Framework

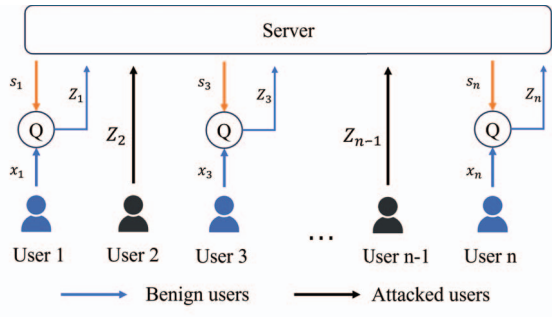


Figure 2: The general framework of robust LDP estimation.

The server, users, and the attacker need to engage in the following steps:

(1) The server generates pre-defined information $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathcal{S}$ and send $\mathbf{s}_i$ to user i;

(2) Each user i generates $\mathbf{Y}_i = Q_i(\mathbf{x}_i)$. The LDP protocols of different users have different pre-defined parameters $\mathbf{s}_i$, i.e. $Q_i(\mathbf{x}_i) = Q(\mathbf{x}_i, \mathbf{s}_i)$.

(3) The attacker modifies the signal of up to q users. The results are denoted as $\mathbf{Z}_i$, in which $\mathbf{Z}_i = \mathbf{Y}_i$ for at least $n - q$ users;

(4) The aggregator receives $\mathbf{Z}_i$ from user i and calculates the output $\hat{\mu} = \mathcal{A}(\mathbf{Z}_1, \ldots, \mathbf{Z}_n, \mathbf{s}_1, \ldots, \mathbf{s}_n)$.

The above procedures are summarized in Figure 2.

5. Frequency Estimation

5.1. Method Description

The original kRR method performs poorly for large alphabet size d. To reduce the effect of LDP perturbation, we can divide the alphabet into k groups, in which $k \leq d$, and then let each user only report the group it belongs to. Such a report is still randomized by the kRR randomizer. Since $k < d$, now the perturbation leads to less performance loss. However, the drawback of such a grouping approach is that we can only estimate the sum of frequency within each group, instead of the frequency of individual components. To address this issue, our solution is to let each user divide the alphabet in different ways so that the aggregator can estimate the frequency of each component. The detailed design is shown as follows.

Pre-defined information $\mathbf{s}_i$. We let $\mathbf{s}_i \in [k]^d$ be a vector whose elements take value from $[k] := \{1, \ldots, k\}$. We assume that d is an integer multiple of k, otherwise we can just pad samples with zeros. For each value $j = 1, \ldots, k$, it is ensured that $\mathbf{s}_i$ has d/k elements with value j, i.e.

$$\sum_{l=1}^{d} \mathbf{1}(s_{il} = j) = \frac{d}{k}, \forall j = 1, \ldots, k. \tag{14}$$

$\mathbf{s}_i$ is selected randomly under this constraint.

The signal $\mathbf{s}_i$ divides the alphabet into k groups equally. For user i possessing local sample x_i, s_{i,x_i} is the index of the group containing user i.

Local randomizer. In this step, the user reports the group index s_{i,x_i} to the server. To make the result satisfy ϵ-LDP, we randomize the result using kRR. The whole mechanism for encoding and perturbation is shown as follows:

$$\mathrm{P}(Y_i = j | \mathbf{s}_i, x_i) = \begin{cases} \frac{e^\epsilon}{e^\epsilon + k - 1} & \text{if} \quad s_{i,x_i} = j \\ \frac{1}{e^\epsilon + k - 1} & \text{if} \quad s_{i,x_i} \neq j, \end{cases} \tag{15}$$

for $j = 1, \ldots, k$. According to (15), if x_i belongs to the j-th group, then the user reports j with higher probability. An example of the randomization procedure with $d = 6$ and $k = 3$ is shown in Figure 3.

Due to the manipulation attack, the server may not receive Y_i. Denote Z_i as the value received from user i, $i = 1, \ldots, n$. Denote $\mathcal{C}$ as the set of attacked users. For all $i \in \mathcal{C}$, Z_i is determined by the attacker, otherwise $Z_i = Y_i$.

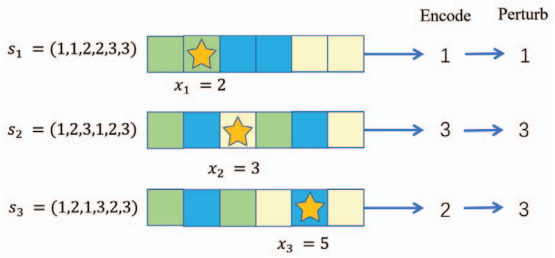


Figure 3: An example of local randomization with $d = 6$ and $k = 3$. For each user, the alphabet is divided into $k = 3$ groups. Light green, blue, and yellow color corresponds to $s_{ij} = 1, 2, 3$ respectively. At the encoding step, user i reports the group it belongs to. The perturbation step just uses kRR.

The aggregator. The server then estimates the frequency after receiving Z_i, $i = 1, \ldots, n$. To estimate the frequency of the l-th component, we count the number of users whose feedback is exactly the group index of l, i.e. $\sum_{i=1}^{n} \mathbf{1}(Z_i = s_{il})$. It can be shown that without attack, the expectation of this value is linearly dependent on the real frequency μ_l. Therefore, it remains to calibrate this result with appropriate slope and intercept. To be more precise, let

$$\hat{\mu}_l = \frac{1}{c}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(Z_i = s_{il}) - a\right), \tag{16}$$

in which a and c are selected to ensure the unbiasedness of $\hat{\mu}_l$ in the absence of attack:

$$a = \frac{(d-k)e^\epsilon + d(k-1)}{k(d-1)(e^\epsilon + k - 1)}, \tag{17}$$

$$c = \frac{d(k-1)(e^\epsilon - 1)}{k(d-1)(e^\epsilon + k - 1)}. \tag{18}$$

The final result is $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_d)$. The whole procedure is summarized in Algorithm 1. The derivation of (17) and (18) is shown in Appendix B.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Frequency estimation

Input: samples $x_i \in [d]$, $i = 1, \ldots, n$; privacy budget ϵ
Output: Frequency estimate $\hat{\mu}$
Parameter: k

if d is not an integer multiple of k **then**
  $d \leftarrow k\lceil d/k \rceil$
end if
for $i = 1, \ldots, n$ (in parallel) **do**
  Server: Randomly divide $[d]$ into k groups with equal size
  for $j = 1, \ldots, k$ **do**
    For all elements l in j-th group, let $s_{il} = j$
  end for
  User i: Generate Y_i according to (15)
end for
Attacker: Determine the set of corrupted users $\mathcal{C} \subseteq [n]$
for $i = 1, \ldots, n$ **do**
  if $i \in \mathcal{C}$ **then**
    The attacker determines Z_i
  else
    $Z_i = Y_i$
  end if
end for
Server: Receive Z_i, $i = 1, \ldots, n$
Calculate $\hat{\mu}_l$ according to (16)
return $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_d)$

Finally, we intuitively explain why the proposed method is robust to manipulation attacks. Roughly speaking, the manipulation can be more effective with a larger output space of LDP protocols. For OUE, the output space is $\{0,1\}^d$, whose size is 2^d. For kRR, the output space is $[d]$, with size d. In our framework, the output space of Z_i is $[k]$, whose size is only k, thus the attacker's capability is more restricted. For small ϵ, we use small k to ensure robustness. For large ϵ, we make k larger to reduce the information loss caused by the randomizer (15).

Optimal attack. Finally, we design strategies for both targeted and untargeted attacks to our method. The goal of the targeted attack is to enhance the frequency estimate at some components. For the untargeted attack, the goal is to maximize the overall error $\|\hat{\mu} - \mu\|_1$. The detailed design is shown in Appendix A.

5.2. Theoretical Analysis

In Theorem 1, we prove that the proposed method is ϵ-LDP and bound the ℓ_1 estimation error.

Theorem 1. *(Privacy guarantee) Algorithm 1 is ϵ-LDP.*

(Performance guarantee) If $\epsilon < 1$, then let $k = 2$. If $1 \le \epsilon \le \ln d$, then pick $k \in [e^\epsilon, 2e^\epsilon]$. If $\epsilon > \ln d$, then let $k = d$. With this selection of k, the ℓ_1 error of frequency estimation is bounded by

$$\mathbb{E}\left[\|\hat{\mu} - \mu\|_1\right] \lesssim \begin{cases} \frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d \ln n}}{n\epsilon} & \text{if} \quad \epsilon < 1 \\ \frac{d}{\sqrt{ne^\epsilon}} + \frac{q\sqrt{d \ln n}}{n\sqrt{e^\epsilon}} & \text{if} \quad 1 \le \epsilon \le \ln d \\ \frac{d}{\sqrt{ne^\epsilon}} + \frac{q\sqrt{\ln n}}{n} & \text{if} \quad \epsilon > \ln d, \end{cases} \tag{19}$$

in which μ is the ground truth defined in (10).

In (19), the first term is caused by honest execution, which means the estimation error without attack. The second term is caused by manipulation by attackers.

For the proof of Theorem 1, the privacy guarantee is relatively easy to prove. For all $\mathbf{x}_i$ and $\mathbf{x}_i'$, from (15),

$$\frac{\mathrm{P}(Y_i = j | \mathbf{s}_i, x_i)}{\mathrm{P}(Y_i = j | \mathbf{s}_i, x_i')} \le \frac{e^\epsilon/(e^\epsilon + k - 1)}{1/(e^\epsilon + k - 1)} = e^\epsilon. \tag{20}$$

Therefore, the privatization mechanism from x_i to Y_i is ϵ-LDP.

The proof of the performance guarantee, i.e. (19), is relatively involved. The general idea is to decompose the estimation error into a term caused by honest execution, which means the error without attack, and another term caused by manipulation. These two terms are analyzed separately. The details of the proof are shown in Appendix C.

Our results significantly improve over HST in [1]. According to Theorem V.7 in [1], the expected ℓ_1 error of HST is $O\left(\frac{d\sqrt{\ln n}}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d \ln n}}{\epsilon n}\right)$ for $\epsilon < 1$, and $O\left(\frac{d\sqrt{\ln n}}{\sqrt{n}} + \frac{q\sqrt{d \ln n}}{n}\right)$ for $\epsilon \ge 1$. For $\epsilon < 1$, our result (19) is nearly the same as HST. For $\epsilon > 1$, we make significant improvements in both the honest execution term ($\frac{d}{\sqrt{ne^\epsilon}}$ versus $\frac{d\sqrt{\ln n}}{\sqrt{n}}$) and the manipulation term ($\frac{q\sqrt{\ln n}}{n}\sqrt{\frac{d}{\min\{e^\epsilon, d\}}}$ versus $\frac{q\sqrt{d \ln n}}{n}$). These results indicate that our method has significantly better performance with medium or large ϵ. Intuitively, for HST, a user only sends a binary signal to the server, which leads to severe information loss. On the contrary, our method sends the group index that has k possible values, thus the server can collect more information from users. Therefore we achieve significantly improved performance with $\epsilon > 1$.

Moreover, although this paper focuses on robust estimation under manipulation attacks, it is worth mentioning that our method is also an optimal frequency estimator in its own right. This method combines the advantages of OUE and kRR. OUE is suitable for large d or small ϵ. On the contrary, with small d or large ϵ, kRR is more preferred. Compared with the error of OUE and kRR in [15], our error bound in Theorem 1 is comparable or better for all privacy levels ϵ and dimensionality d.

6. ℓ_1 Mean Estimation

6.1. Method Description

As discussed earlier, since ℓ_1 mean estimation is more complex than the frequency estimation, we design protocols for these two tasks separately.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

For one-dimensional mean estimation under LDP, currently, the most effective method is the piecewise mechanism [33]. Now we extend to multi-dimensional mean estimation. There are two tempting methods. The first one is to estimate each component separately. For d-dimensional samples, the estimation of each component needs to satisfy ϵ/d-LDP, which may lead to strong noise. Therefore, this method is only suitable for large ϵ. Another one is to divide users into d groups, and each group is used to estimate one component under ϵ-LDP. However, this method compresses each sample into only one dimension, resulting in a loss of information. With large ϵ, such information loss is obvious, therefore this method is only suitable for small ϵ. Motivated by the pros and cons of these two methods, we propose a new strategy that lies between these two extremes. We estimate each sample $\mathbf{x}_i$ along k pre-defined directions. The details are shown as follows.

Pre-defined information $\mathbf{s}_i$. Let $\mathbf{s}_i$ contain k vectors, i.e. $\mathbf{s}_i = (\mathbf{s}_{i1}, \ldots, \mathbf{s}_{ik})$, with $\mathbf{s}_{ij}$ randomly drawn from $\{-1, 1\}^d$ for all i and j. Intuitively, $\mathbf{s}_{ij}$, $j = 1, \ldots, k$ specify k directions. Each sample $\mathbf{x}_i$ is only evaluated along these directions.

Local randomizer. We use a local randomizer that is inspired by the piecewise mechanism in [33]. After receiving $\mathbf{s}_{ij}$, $j = 1, \ldots, k$ from the server, it generates Y_{ij}, $j = 1, \ldots, k$. Let $\epsilon_0 = \epsilon/k$. Let the probability density function (pdf) of Y_{ij} be

$$p(y_{ij}|\mathbf{s}_{ij}, \mathbf{x}_i) = \begin{cases} p_0 & \text{if} \quad y_i \in [l(\mathbf{s}_{ij}, \mathbf{x}_i), r(\mathbf{s}_{ij}, \mathbf{x}_i)] \\ \frac{p_0}{e^{\epsilon_0}} & \text{if} \quad y_i \in [-C, C] \setminus [l(\mathbf{s}_{ij}, \mathbf{x}_i), r(\mathbf{s}_{ij}, \mathbf{x}_i)] \end{cases}, \tag{21}$$

in which

$$p_0 = \frac{e^{\frac{\epsilon_0}{2}}(e^{\frac{\epsilon_0}{2}} - 1)}{2(e^{\frac{\epsilon_0}{2}} + 1)}, C = \frac{e^{\frac{\epsilon_0}{2}} + 1}{e^{\frac{\epsilon_0}{2}} - 1}, \tag{22}$$

and

$$l(\mathbf{s}_{ij}, \mathbf{x}_i) = \frac{e^{\frac{\epsilon_0}{2}} \langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle - 1}{e^{\frac{\epsilon_0}{2}} - 1}, \tag{23}$$

$$r(\mathbf{s}_{ij}, \mathbf{x}_i) = \frac{e^{\frac{\epsilon_0}{2}} \langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle + 1}{e^{\frac{\epsilon_0}{2}} - 1}. \tag{24}$$

(21) has the same format as the piecewise mechanism [33], except that the output Y_{ij} now depends on the internal product of $\mathbf{s}_{ij}$ and $\mathbf{x}_i$. The difference is that the piecewise mechanism is designed for one-dimensional samples. Now we estimate mean values for high dimensional samples, therefore we project each sample in k directions and estimate these k components separately. Since the estimation of each component satisfies ϵ_0-LDP, in which $\epsilon_0 = \epsilon/k$, by the basic composition theorem [34], the estimation of k components satisfies ϵ-LDP.

With the mechanisms above, n users generate Y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k$. The attacker can modify the response of up to q users. After manipulation, the feedback signals become Z_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k$. It is ensured that $Z_{ij} = Y_{ij}$ for all $i \notin \mathcal{C}$, and $|\mathcal{C}| \leq q$.

The aggregator. Now the aggregator receives Z_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k$. Based on these feedback messages, we calculate

$$\hat{\mathbf{x}}_{i0} = \frac{1}{k} \sum_{j=1}^{k} \mathbf{s}_{ij} Z_{ij}. \tag{25}$$

Now we show that without attack, (25) is an unbiased estimate of $\mathbf{x}_i$. According to the pdf in (21), it can be shown that $\mathbb{E}[Y_{ij}|\mathbf{s}_{ij}, \mathbf{x}_i] = \langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle$. Therefore

$$\mathbb{E}[\mathbf{s}_{ij} Y_{ij} | \mathbf{x}_i] = \mathbb{E}[\mathbf{s}_{ij} \mathbf{s}_{ij}^T \mathbf{x}_i | \mathbf{x}_i] = \mathbf{x}_i, \tag{26}$$

in which the last step holds because $\mathbf{s}_{ij}$ is randomly selected from $\{-1, 1\}^d$, thus $\mathbb{E}[\mathbf{s}_{ij} \mathbf{s}_{ij}^T] = \mathbf{I}_d$, in which $\mathbf{I}_d$ denotes the $d \times d$ identity matrix. (26) indicates that if the user i is not attacked, i.e. $Z_{ij} = Y_{ij}$, then (25) is an unbiased estimator of $\mathbf{x}_i$. Since (25) is an average over k independent vectors, if k is large and the user i is not attacked, $\hat{\mathbf{x}}_{i0}$ is expected to be close to $\mathbf{x}_i$. Recall that in ℓ_1 mean estimation problem, $\|\mathbf{x}_i\|_1 \leq 1$, thus as long as user i is benign, $\|\mathbf{x}\|_{i0}$ should also not be too large. In other words, a large $\|\mathbf{x}\|_{i0}$ indicates that user i is likely to be corrupted. Therefore, a simple defense strategy is to clip the ℓ_1 norm to some fixed parameter T:

$$\hat{\mathbf{x}}_i = \min\left\{1, \frac{T}{\|\hat{\mathbf{x}}_{i0}\|_1}\right\} \hat{\mathbf{x}}_{i0}. \tag{27}$$

The clipping operation in (27) ensures the robustness of our method. The final estimate of μ is just the average of the estimation of each sample $\hat{\mathbf{x}}_i$:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \hat{\mathbf{x}}_i. \tag{28}$$

The whole procedure is summarized in Algorithm 2.

Optimal attack. Similar to the frequency estimation problem, we also define the optimal targeted and untargeted attack strategy for ℓ_1 mean estimation. Given a target direction $\mathbf{u}$, the optimal targeted attack maximizes the error along $\mathbf{u}$. The optimal untargeted attack maximizes $\|\hat{\mu} - \mu\|_1$. We refer to Appendix A for the detailed design.

6.2. Theoretical Analysis

The theoretical results are shown as follows.

Theorem 2. *(Privacy guarantee) Algorithm 2 is ϵ-LDP. (Performance guarantee) Let*

$$T \sim \frac{e^{\frac{\epsilon}{2k}} + 1}{e^{\frac{\epsilon}{2k}} - 1} \frac{d}{\sqrt{k}} \ln(nd), \tag{29}$$

then the ℓ_1 estimation error of Algorithm 2 is bounded by

$$\mathbb{E}\left[\|\hat{\mu} - \mu\|_1\right] \lesssim \frac{e^{\frac{\epsilon}{2k}} + 1}{e^{\frac{\epsilon}{2k}} - 1} \left(\frac{d}{\sqrt{nk}} + \frac{q}{n} \min\left\{\sqrt{d}, 1 + \frac{d}{\sqrt{k}} \ln(nd)\right\}\right). \tag{30}$$

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: ℓ_1 Mean Estimation

Input: samples $\mathbf{x}_i \in \mathbb{B}_1$, $i = 1, \dots, n$; privacy budget ϵ
Output: Mean estimate $\hat{\mu}$
Parameter: k, T

for $i = 1, \dots, n$ (in parallel) **do**
  Server:
  for $j = 1, \dots, k$ **do**
    Generate $\mathbf{s}_{ij}$ by randomly draw from $\{-1, 1\}^d$
    User i: Generate Y_{ij} according to (21)
  end for
end for
Attacker: Determine the set of corrupted users $\mathcal{C} \subseteq [n]$
for $i = 1, \dots, n$ **do**
  if $i \in \mathcal{C}$ **then**
    The attacker determines Z_{ij}, $j = 1, \dots, k$
  else
    $Z_{ij} = Y_{ij}$, $j = 1, \dots, k$
  end if
end for
Server: Receive Z_{ij}, $i = 1, \dots, n$, $j = 1, \dots, k$
for $i = 1, \dots, n$ **do**
  Calculate $\hat{\mathbf{x}}_i$ according to (27)
end for
Calculate $\hat{\mu}$ according to (28)
return $\hat{\mu}$

From (21), the generation of Y_{ij} satisfies ϵ_0-LDP. Since there are k feedback signals from each user, by the basic composition rule [34], $Y_{i1}, \dots, Y_{ik}$ satisfies ϵ-LDP. The proof of the performance guarantee is shown in Appendix D.

Now we discuss (30).

- *1)* $\epsilon < 1$. In this case, we can just let $k = 1$. Then the ℓ_1 error is $O\left(\frac{d}{\sqrt{n\epsilon^2}} + \frac{q\sqrt{d}}{\epsilon n}\right)$, which matches the result in [1] (with slight improvement in logarithmic factor).
- *2)* $1 \le \epsilon \le d$. In this case, we can pick k that minimizes $(e^{\frac{\epsilon}{2k}} + 1)/(\sqrt{k}(e^{\frac{\epsilon}{2k}} - 1))$, then the right hand side of (30) becomes $O\left(\frac{d}{\sqrt{n\epsilon}} + \frac{q\sqrt{d}}{n}\right)$, which improves over $O\left(\frac{d}{\sqrt{n}} + \frac{q\sqrt{d}}{n}\right)$ in [1].
- *3)* $\epsilon > d$. The estimation error further decreases with $\epsilon > d$. In particular, with $\epsilon \to \infty$, the right hand side of (30) converges to $O\left(\frac{q}{n}\right)$, which matches the non-private error.

In general, our theoretical results show that the new method has similar performance with [1] for small ϵ. For larger ϵ, our method significantly improves over [1].

7. ℓ_2 Mean Estimation

7.1. Method Description

[2] has proposed a popular protocol for mean estimation in ℓ_2 ball, as is shown in Section 2.2. To design robust LDP protocols for all privacy levels, we need to face two challenges. The first one is to control the effectiveness of the attack, and the second one is to refine the protocol to make it optimal for all privacy levels. To address these issues, we restrict the output space with pre-defined information $\mathbf{s}_i$, which contains k unit vectors, and k can be adjusted based on ϵ. The detailed design is shown as follows.

Pre-defined information $\mathbf{s}_i$. Similar to the ℓ_1 mean estimation problem, let $\mathbf{s}_i$ contain k vectors, i.e. $\mathbf{s}_i = (\mathbf{s}_{i1}, \dots, \mathbf{s}_{id})$. Now we let $\mathbf{s}_{ij}$ to be randomly drawn from the surface of unit sphere, i.e. $\mathbb{S}^d = \{\mathbf{u} | \|\mathbf{u}\|_2 \le 1\}$, for all $i = 1, \dots, n$ and $j = 1, \dots, k$.

Local randomizer. User i receives $\mathbf{s}_{ij}$, $j = 1, \dots, k$ from the server. It then generates Y_{ij} according to the following distribution:

$$\mathrm{P}(Y_{ij} = 1 | \mathbf{s}_{ij}, \mathbf{x}_i) = \frac{1}{2} + \frac{1}{2} \|\mathbf{x}_i\|_2 c_\epsilon \operatorname{sign}(\langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle) \quad (31)$$

$$\mathrm{P}(Y_{ij} = -1 | \mathbf{s}_{ij}, \mathbf{x}_i) = \frac{1}{2} - \frac{1}{2} \|\mathbf{x}_i\|_2 c_\epsilon \operatorname{sign}(\langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle) \quad (32)$$

in which

$$c_\epsilon = \frac{e^{\epsilon/k} - 1}{e^{\epsilon/k} + 1}. \quad (33)$$

This mechanism is inspired by [2]. In particular, if there are no attacked users, with $k = 1$, our method reduces to [2]. The difference is that now $\mathbf{s}_{ij}$ for $j = 1, \dots, k$ are predefined and can not be modified by the attackers. Moreover, we repeat the estimation process k times, and ϵ/k privacy budget is allocated to each output Y_{ij}, in order to make our method suitable for all ϵ.

After manipulation, the server receives Z_{ij}, $i = 1, \dots, n$, $j = 1, \dots, k$. Let $\mathbf{Z}_i = (Z_{i1}, \dots, Z_{ik})$, $\mathbf{Y}_i = (Y_{i1}, \dots, Y_{ik})$. Then $\mathbf{Z}_i \neq \mathbf{Y}_i$ for no more than q users.

The aggregator. The final aggregation is

$$\hat{\mu} = \frac{c_d}{c_\epsilon} \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{s}_{ij} Z_{ij}, \quad (34)$$

in which c_d has been defined in (13). Now we explain (34).

Lemma 1. *Without attack,* (34) *is unbiased, i.e.*

$$\mathbb{E}\left[\frac{c_d}{c_\epsilon} \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{s}_{ij} Y_{ij}\right] = \mu. \quad (35)$$

We prove Lemma 1 as follows. According to (31) and (32), given $\mathbf{s}_{ij}$ and $\mathbf{x}_i$,

$$\mathbb{E}[Y_{ij} | \mathbf{s}_{ij}, \mathbf{x}_i] = \|\mathbf{x}_i\|_2 c_\epsilon \operatorname{sign}(\langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle). \quad (36)$$

Since $\mathbf{s}_{ij}$ is uniformly distributed on the surface of unit sphere $\mathbb{S}^d$, it can be shown that as long as $\|\mathbf{x}_i\|_2 > 0$, the following equation holds:

$$\mathbb{E}[\mathbf{s}_{ij} \operatorname{sign}(\langle \mathbf{s}_{ij}, \mathbf{x}_i \rangle)] = \frac{1}{c_d} \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}. \quad (37)$$

(35) can then be proved using (36) and (37).

Optimal attack. For ℓ_2 mean estimation, the targeted attack maximizes the estimation error along a fixed direction $\mathbf{u}$, while the untargeted attack maximizes $\|\hat{\mu} - \mu\|_2$. The detailed design is shown in Appendix A.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: ℓ_2 Mean Estimation

Input: samples $\mathbf{x}_i \in \mathbb{B}_2$, $i = 1, \ldots, n$; privacy budget ϵ
Output: Mean estimate $\hat{\mu}$
Parameter: k

for $i = 1, \ldots, n$ (in parallel) **do**
  Server:
  for $j = 1, \ldots, k$ **do**
    Generate $\mathbf{s}_{ij}$ by randomly draw from $\{-1, 1\}^d$
    User i: Generate Y_{ij} according to (31) and (32)
  end for
end for
Attacker: Determine the set of corrupted users $\mathcal{C} \subseteq [n]$
for $i = 1, \ldots, n$ **do**
  if $i \in \mathcal{C}$ **then**
    The attacker determines Z_{ij}, $j = 1, \ldots, k$
  else
    $Z_{ij} = Y_{ij}$, $j = 1, \ldots, k$
  end if
end for
Server: Receive Z_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k$
Calculate $\hat{\mu}$ according to (34)
return $\hat{\mu}$

7.2. Theoretical Analysis

The theoretical results are shown as follows.

Theorem 3. *(Privacy guarantee) Algorithm 3 is ϵ-LDP.*

(Performance guarantee) The ℓ_2 error of the estimator above is bounded by

$$\mathbb{E}\left[\|\hat{\mu} - \mu\|_2\right] \lesssim c_d \frac{e^{\epsilon/k}+1}{e^{\epsilon/k}-1}\frac{1}{\sqrt{nk}} + \frac{e^{\epsilon/k}+1}{e^{\epsilon/k}-1}\frac{q}{n}. \tag{38}$$

The privacy guarantee holds because from (31) and (32),

$$\frac{\mathrm{P}(Y_{ij}=1|\mathbf{s}_{ij},\mathbf{x}_i)}{\mathrm{P}(Y_{ij}=-1|\mathbf{s}_{ij},\mathbf{x}_i')} \le \frac{1+c_\epsilon}{1-c_\epsilon} = e^{\epsilon/k}. \tag{39}$$

Similar bound holds for $\mathrm{P}(Y_{ij} = -1|\mathbf{s}_{ij}, \mathbf{x}_i')/\mathrm{P}(Y_{ij} = 1|\mathbf{s}_{ij}, \mathbf{x}_i)$. Therefore the generation of Y_{ij} satisfies ϵ/k-LDP, and thus $(Y_{i1}, \ldots, Y_{ik})$ is ϵ-LDP. For the performance guarantee, (38) is proved in Appendix E.

For $\epsilon < 1$, we can just let $k = 1$. According to (13), $c_d \lesssim \sqrt{d}$. Then the ℓ_2 error is $O\left(\sqrt{\frac{d}{n\epsilon^2}} + \frac{q}{\epsilon n}\right)$, which matches the result in [1]. With $\epsilon \ge 1$, we can pick k that minimizes $(e^{\epsilon/k}+1)/(\sqrt{k}(e^{\epsilon/k}-1))$. As a result, the ℓ_2 error becomes $O\left(\sqrt{\frac{d}{n\epsilon}} + \frac{q}{n}\right)$, which improves over $O\left(\sqrt{\frac{d}{n}} + \frac{q}{n}\right)$ in [1].

8. Evaluation

In this section, we show some results of numerical experiments.

8.1. Experiments of Frequency Estimation

Experimental setup. We run experiments under $\epsilon = 3$ first. The experiment uses the following datasets:

- Synthesized dataset following Zipf distribution: $p_j = (1/j)/\sum_{l=1}^{d}(1/l)$, with $d = 16$. In this experiment, the sample size is $N = 10,000$.
- Synthesized dataset following zipf distribution, $d = 64$.
- Fire dataset [35]. The Fire dataset was from the San Francisco Fire Department, which records information about calls for service. Following [10], we use the unit ID as the value. Now it has $N = 754,061$ samples, and the alphabet size is $d = 321$.
- IPUMS [36]. The IPUMS dataset contains the US census data. We use the data in 2022 and treat the city attribute as values. As a result, there are $N = 388,525$ samples, and the alphabet size is $d = 110$.

For all these datasets, we test the performance of mean estimation under three attacks: random attack, maximal gain attack, and optimal attack. Under the random attack, each attacked local randomizer generates random output from the output domain. This attack is exactly RPA in [10]. The maximal gain attack is just the optimal targeted attack defined in Appendix A. We generate the target randomly, such that each component belongs to the target with $1/2$ probability. The optimal attack is an untargeted attack that maximizes the overall estimation error over all possible directions. For HST, the corresponding optimal attack has been designed in [1]. For our method, the optimal attack is designed in Appendix A.

The quality of estimation is evaluated using ℓ_1 metric. We let the fraction of corrupted users α to take values from $[0, 0.2]$. In our experiments, we compare our new method with several baselines, including OUE, kRR [15], HST [1], and LDPRecover [11]. LDPRecover attempts to recover true frequency from its corrupted estimates. We use the output from OUE as the base estimator. This method needs to set a hyperparameter η, whose best value is the fraction of corrupted users α. Since we assume that q is unknown, considering that we have already set $\alpha \in [0, 0.2]$, we use $\eta = 0.1$ for all experiments. There are also several methods relying on the detailed knowledge of attacks, such as fake user detection [10] and LDPGuard [9]. We do not compare with these methods since we assume that the details of attacks are unknown to the server.

For all methods, we normalize the final results [37], as a common post-processing strategy. Firstly, for an estimated result $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_d)$, we conduct the following post-processing. Firstly, for any $j \in [d]$, if $\hat{\mu}_j < 0$, then we set $\hat{\mu}_j \leftarrow 0$. Secondly, we let $\hat{\mu} \leftarrow \hat{\mu}/\|\hat{\mu}\|_1$, such that the ℓ_1 norm of the estimated frequency is exactly 1. After such post-processing, since $\|\hat{\mu}\|_1 = 1$, $\|\mu\|_1 = 1$, it is ensured that $\|\hat{\mu} - \mu\|_1 \le 2$.

For our new method, we fix $k = 8$ for all datasets and all corruption levels. Here we would like to remark that the performance may be improved further if we let k be adaptive to the ratio of corruption α, the dimensionality d,

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

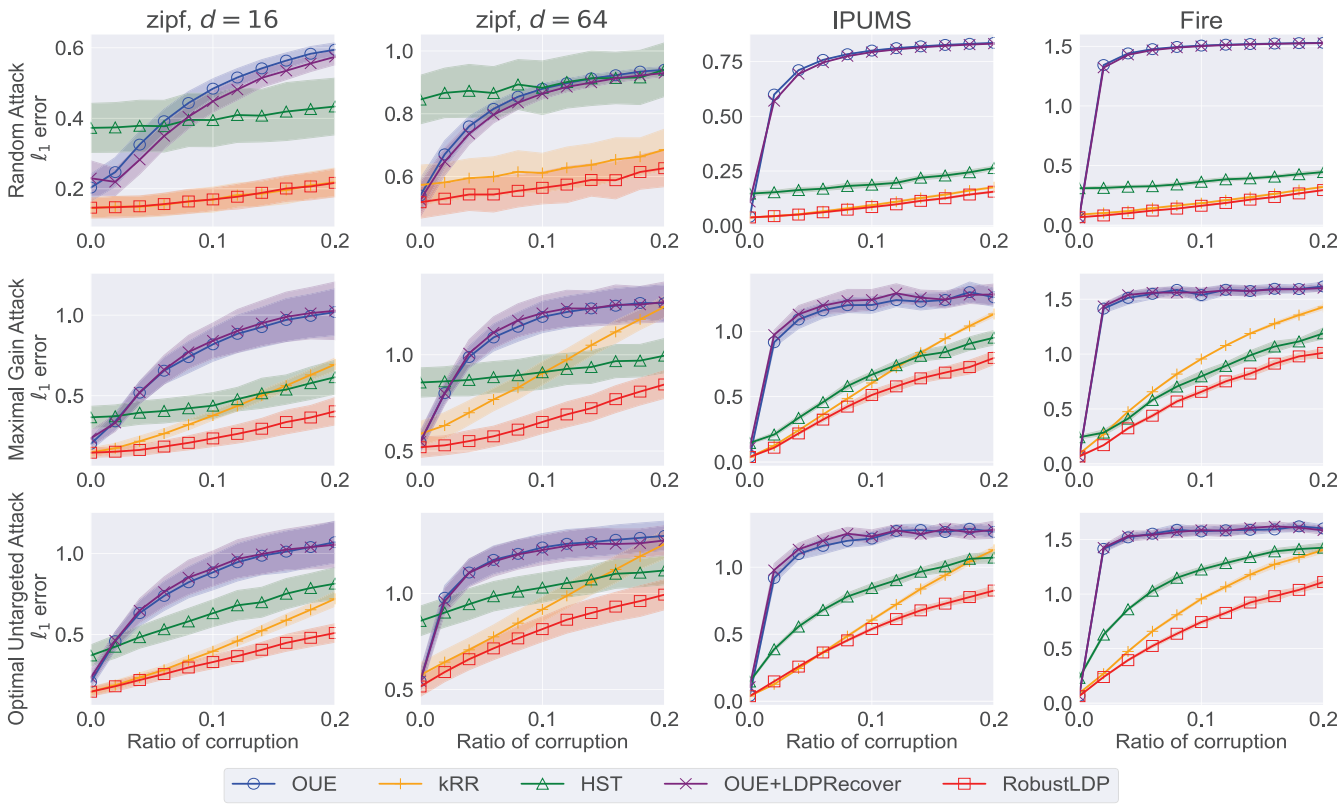


Figure 4: Comparison of frequency estimation methods. We set $\epsilon = 2$ for all experiments.

and sample distributions. However, we do not tune k in such a way. Firstly, we can not assume that such information is known to the server. Secondly, the parameter tuning may introduce additional privacy costs. Therefore, we just pick a fixed k for all experiments.

Overall results. The overall results are shown in Figure 4. In all figures, the horizontal axis represents the ratio of corrupted users $\alpha = q/n$, which varies from 0 to 0.2. The vertical axis corresponds to the ℓ_1 estimation error $\|\hat{\mu} - \mu\|$. The blue, orange, green, and red curves denote the OUE, kRR, HST, and our method, respectively. Each point on the curve denotes the average ℓ_1 error over $M = 500$ random trials.

From these results, it can be observed that in general, the new method (the red curve) significantly outperforms existing methods. Therefore, these experiments validate the effectiveness of RobustLDP and our theoretical analysis.

Impact of ϵ. Here we use random attack and optimal untargeted attack with $d = 64$. The value of k for our new method is set as follows: For $\epsilon \leq 1$, $k = 2$. For $\epsilon > 1$, we set $k = \lfloor e^{\epsilon} \rfloor$. Here k only depends on ϵ. It is ensured that the value of k does not change over datasets, in order to avoid the additional privacy cost caused by parameter tuning. Changing k with respect to ϵ is necessary since the theoretical analysis in Theorem 1 suggests that the optimal k depends on ϵ. Therefore, in our experiments, we let k increase with ϵ. The sample size n is fixed at 1000, while the number of corrupted users q takes values from $0, 100$ and 200. The results are shown in Figure 5.

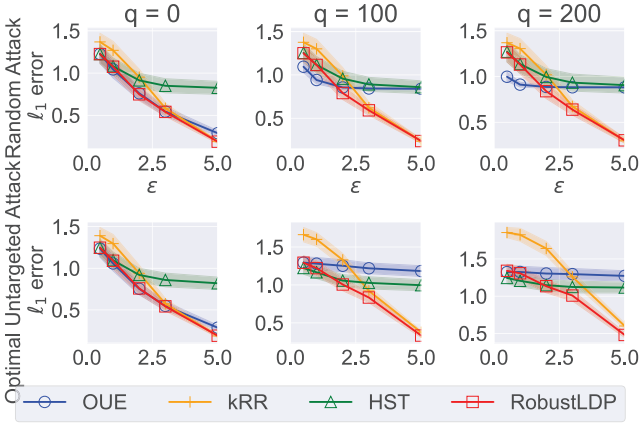


Figure 5: ℓ_1 error of frequency estimation with respect to ϵ.

From Figure 5, it can be observed that with $\epsilon \leq 1$, the new method has nearly the same performance as HST. With ϵ increases, the error of HST and OUE does not converge to zero even if there are no attacks (i.e. $q = 0$). On the contrary, for our method (the red curve), the ℓ_1 estimation error converges to zero as ϵ increases. These results agree with our theoretical analysis. Moreover, it is worth mentioning that in many practical cases, it is not practical to achieve $\epsilon < 1$. As shown in Figure 5, the ℓ_1 error for $\epsilon < 1$ can be larger than 1, which introduces little practical significance. Therefore,

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

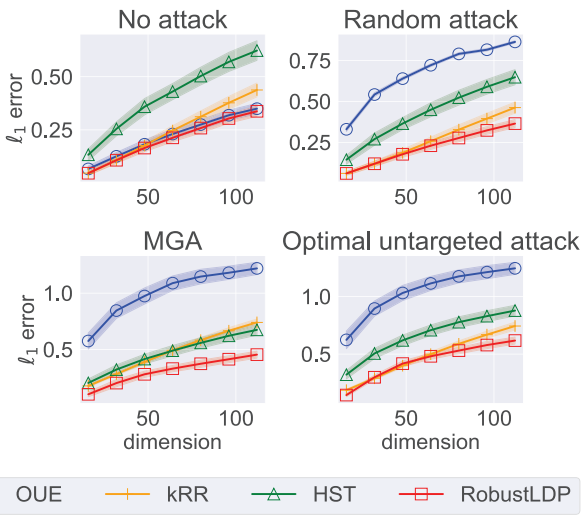


Figure 6: ℓ_1 error of frequency estimation with respect to d, under $\epsilon = 3$.

for a better privacy-accuracy tradeoff, we can only consider larger ϵ, under which our method has clear advantages over existing methods.

Impact of d. To further test the performance of RobustLDP, we run experiments with different dimensionality d. In this experiment, we fix $N = 10,000$, and let the dimension d vary from 0 to 128. We use a synthesized dataset following Zipf distribution. We compare the ℓ_1 estimation of all methods. We run experiments with no attacks, random attack, MGA, and optimal untargeted attack defined in Appendix A, respectively, and let $q = 500$ for all of these attacks. The results are shown in Figure 6, in which each point represents the result by averaging over $M = 1000$ independent trials.

From Figure 6, it can be observed that with the increase of dimensionality d, the performances of all methods degrade significantly. However, compared with existing methods, the red curves in Figure 6 have a relatively smaller slope, indicating that our method is more robust to manipulation compared with existing methods.

8.2. Experiments of ℓ_1 Mean Estimation

Experimental setup. We use the following datasets to evaluate the performance of mean estimation in ℓ_1 support.

- Synthesized dataset. We use the following steps to generate the dataset. Firstly, we draw N samples randomly from $\{\mathbf{u}|u_i > 0, \forall i \in [d] \text{ and } \|\mathbf{u}\|_1 \leq 1\}$, i.e. the subset of ℓ_1 ball with all elements being positive. Then we randomly flip each coordinate of each sample with probability $p_0 = 0.3$. Then samples fill the entire ℓ_1 support, with the mean value being positive for each coordinate. We run experiments with $d = 4$ and $d = 8$, respectively.
- Fire and IPUMS datasets. We consider that a common application of ℓ_1 mean estimation is frequency estimation at user level. Therefore, we divide the samples randomly into m groups and calculate the group-wise averages as the data used in the experiments.

For these datasets, we test the resilience with respect to three attacks: Random Attack, which generates random output from the output domain; Flip Attack, which reverses the sign of output, i.e. $\mathbf{Z}_i = -\mathbf{Y}_i$ for $i \in \mathcal{C}$; and Optimal Attack, which maximize the estimation error with respect to a fixed direction. The detailed design of the optimal attack is shown in Appendix A. In all experiments, we set $k = 2$.

Overall results. The overall results are shown in Figure 7. Following the experiments of frequency estimation, the horizontal axis means the ratio of corruption $\alpha = q/N$. The vertical axis is the ℓ_1 error. The blue curve represents Rappor. The orange curve corresponds to HST in [1]. The result of our method is shown in the green curve. From Figure 7, we have the following observations.

- Under Random Attack and Flip Attack, HST achieves nearly the same performance as Rappor. Compared with HST and Rappor, our method achieves a significantly better performance.
- Under optimal attack, the ℓ_1 error grows fastly with the ratio of corruption. Compared with Rappor, HST significantly reduces the error. With increasing q, the ℓ_1 error only grows in a relatively slow way. Compared with HST, our new method further reduces the estimation error.

In general, these numerical results validate the effectiveness of our proposed approach. For all experiments, we achieve significantly better performance than HST and Rappor. Moreover, with the increase of d, the advantage of our method becomes more obvious. These results agree well with our theoretical analysis.

Impact of ϵ. Finally, we run some experiments to test the impact of ϵ. In this experiment, we let ϵ vary from 0 to 20. Similar to the frequency estimation problem, we let k grow with ϵ. Here we let $k = \lfloor \epsilon/2 \rfloor$. We run the experiments with two attacks, including the random attack and the optimal attack. The dataset contains $n = 1000$ samples, and we test the performances with $q = 0, 100, 200$ separately. The results are shown in Figure 8.

From Figure 8, with $\epsilon = 1$, although our theoretical bound is almost the same as the result of HST in [1], the experimental results show that we still achieve significantly better practical improvements. This is because compared with the randomized response mechanism used in [1], the mechanism (21) leads to an improvement to a constant factor, even if $\epsilon \leq 1$ and $k = 1$. With the increase of ϵ, the advantage of our method becomes more obvious. It can be observed that the Rappor method (with normalization) suffers severely from carefully designed attacks. HST can perform better than Rappor, but the ℓ_1 error of HST still fails to converge to zero with the increase of ϵ, even if $q = 0$. On the contrary, for our method, the error converges to the non-private case.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

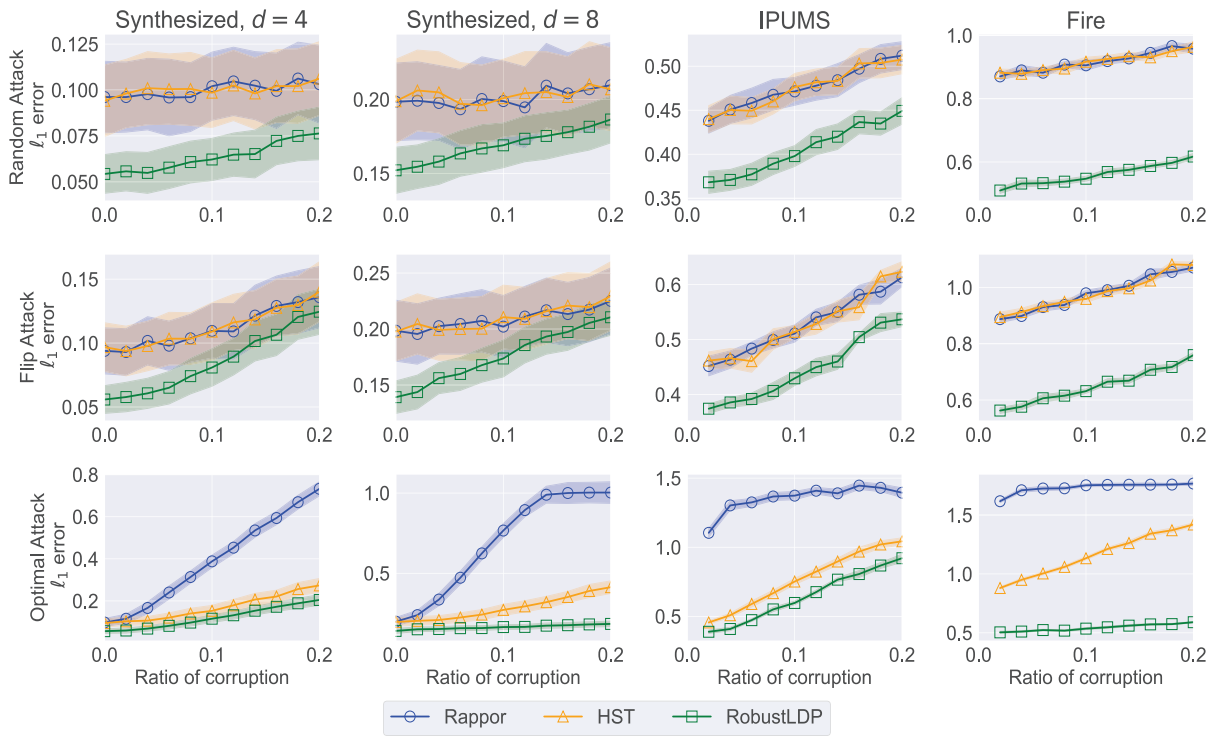


Figure 7: Comparison of ℓ_1 mean estimation methods. We set $\epsilon = 5$ for all experiments.

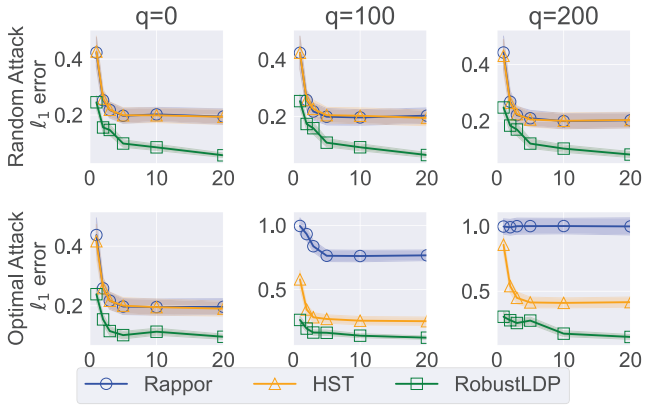


Figure 8: The ℓ_1 error of mean estimation over ℓ_1 support with respect to ϵ.

8.3. Experiments of ℓ_2 Mean Estimation

Experimental setup. For ℓ_2 mean estimation, we use the following datasets.

- Synthesized dataset. We generate samples from Gaussian distribution with mean $0.8\mathbf{e}_1$, and identity covariance matrix. The mean is nonzero only at the first element. For all samples with norms larger than 1, we project them onto the surface of the unit ball, i.e. $\mathbf{x}_i \leftarrow \min(1, 1/\|\mathbf{x}_i\|_2)\mathbf{x}_i$. As a result, all samples fall in $\mathbb{B}_2$.
- IPUMS dataset. Similar to the frequency estimation and ℓ_1 mean estimation task, we still use the IPUMS dataset. For ℓ_2 mean estimation, we need to find continuous variables. Therefore, we use the salary data in 2022 from the dataset and pick three columns: INCTOT, INCWAGE, and INCBUS, which represent the total income, the income from wages, and the income from business, respectively. Samples are then normalized into $\mathbb{B}_2$.
- NBA players dataset [38]. The NBA player's dataset contains performance metrics that allow people to assess players' ability and overall offensive contribution to the team. We use the data in the 2023 season with a subset of attribute values such as FGM(number of field goals made by the player) and 3PA(number of 3-point field goals attempted by the player). Now it has $n = 539$ samples and $d = 9$ attributes. Similar to the IPUMS dataset, samples are normalized into $\mathbb{B}_2$.

Similar to the ℓ_1 mean estimation problem, we test the performance with respect to Random Attack, Flip Attack, and Optimal Attack. The optimal attack is designed in Appendix A. The privacy budget is set to be $\epsilon = 10$. In all experiments, we set $k = 5$.

Overall results. The overall results are shown in Figure 9. The blue curve represents the standard ℓ_2 mean estimation method in [2] without any defense. The orange curve represents EST in [1]. The green curve represents our new method.

From Figure 9, it can be observed that our method performs better than existing methods.

Impact of ϵ. We finally test the performance for different ϵ. In this experiment, we let ϵ vary from 1 to 50. The value of k is selected to to minimize $(e^{\epsilon/k} + 1)/(\sqrt{k}(e^{\epsilon/k} - 1))$.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

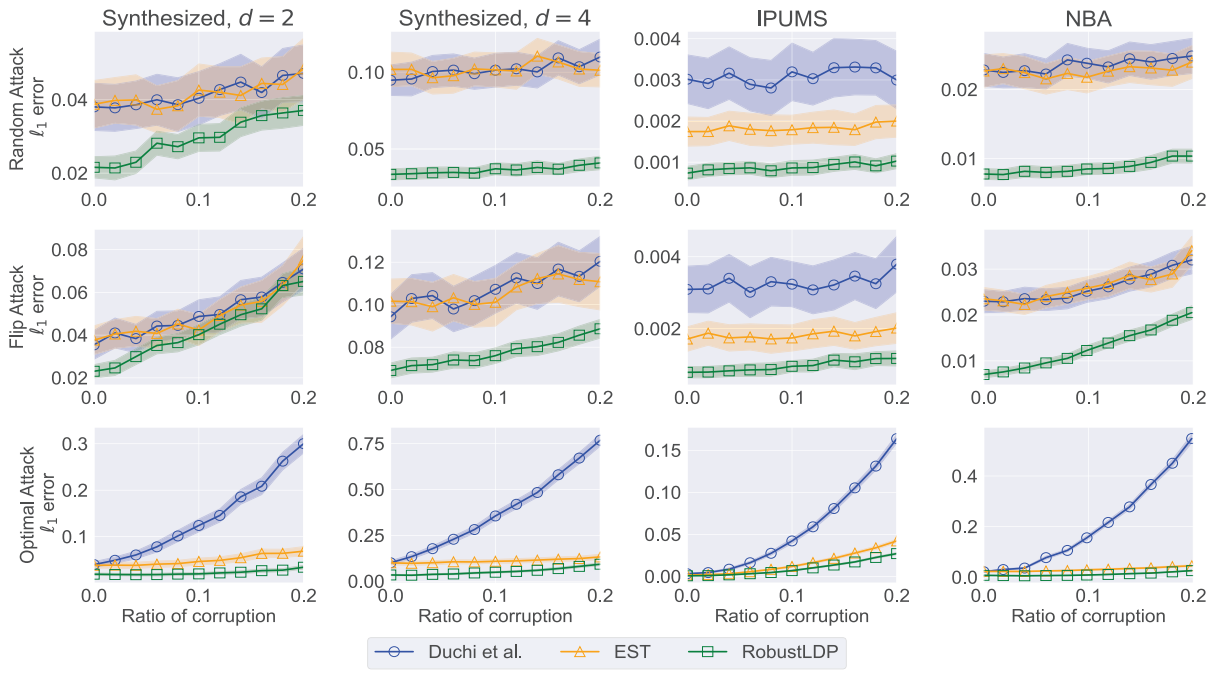


Figure 9: Comparison of ℓ_2 mean estimation methods. We set $\epsilon = 10$ for all experiments.

We use a synthesized dataset with $d = 8$. The results are shown in Figure 10.

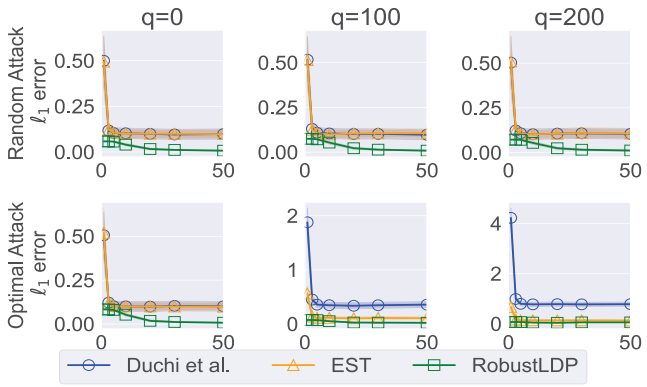


Figure 10: The ℓ_2 error of mean estimation over ℓ_2 support with respect to ϵ.

From Figure 10, it can be observed that the original method in [2] does not perform well, especially with manipulation. Compared with the original method, the EST method in [1] exhibits significantly lower estimation error. However, Figure 10 also shows that with the increase of ϵ, the error of EST does not converge to zero, even if there are no attacks. Compared with EST, our new method achieves nearly the same performance as EST for small ϵ. With larger ϵ, our method has significantly smaller ℓ_2 error than EST, indicating that our method has advantages in large ϵ. The results agree with our theoretical analysis.

9. Related Work

LDP protocols without attacks. For frequency estimation, [3] proposed Rappor. [15] proposed OUE and OLH. [17] and [18] proposed optimal LDP protocols at all privacy regimes and has improved communication complexity. For mean estimation, [2] and [19] proposed estimators that project each sample on the surface of a sphere. Improved methods are then proposed in [20], [22]. [33] proposed a refined mean estimator under LDP. Apart from these basic problems, LDP has been used in some advanced tasks [39], [40], [41], [42], [43].

Manipulation attacks against LDP protocols. LDP protocols are well known to be vulnerable to poisoning attacks [1], [10], [44]. [10] proposed three common attacks: Random Perturbation Attack (RPA), Random Item Attack (RIA) and Maximal Gain Attack (MGA). [44] extended these attacks to key-value data. [1] designed an optimal untargeted attack strategy. [45] proposed an attack that can distort both mean and variance estimation.

Defense against manipulation attacks. Cao et al. [10] proposed several standard methods. [9] proposed LDPGuard. [11] proposed LDPRecover, which attempts to recover the real frequency from its corrupted estimation. [1] proposed general defense strategies, called HST and EST, for frequency estimation and ℓ_2 mean estimation, respectively. [46] generalizes the analysis to general information constraints.

10. Conclusion

In this paper, we have designed a new robust estimation framework under LDP that can withstand output poisoning attacks. We discuss three common tasks: frequency estimation, ℓ_1 and ℓ_2 mean estimation. For frequency estimation, each element of the pre-defined signal can take k possible values. For ℓ_1 and ℓ_2 mean estimation, there are k independent signals for each user. Our theoretical analysis

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

shows that compared with existing approaches, RobustLDP can significantly reduce the estimation error for $\epsilon > 1$. Moreover, we have conducted experiments for all of these tasks. The results have verified the superior performance of our approach.

Acknowledgements

The work of Zhikun Zhang was supported in part by the NSFC under Grants No. 62402431, 62441618, and Zhejiang University Education Foundation Qizhen Scholar Foundation. The work of Shaowei Wang was supported by National Natural Science Foundation of China (No.62372120, 62102108), GuangDong Basic and Applied Basic Research Foundation (No.2022A1515010061), and Science and Technology Projects in Guangzhou (No.2025A03J3182). The work of Zhe Liu was supported by the National Natural Science Foundation of China (62132008, U22B2030), the Natural Science Foundation of Jiangsu Province (BK20220075).

References

[1] A. Cheu, A. Smith, and J. Ullman, "Manipulation attacks in local differential privacy," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 883–900, IEEE, 2021.

[2] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th annual symposium on foundations of computer science*, pp. 429–438, IEEE, 2013.

[3] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

[4] A. D. P. Team, "Learning with privacy at scale," tech. rep., 2017.

[5] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proceedings of the 2018 International Conference on Management of Data*, pp. 1655–1658, 2018.

[7] M. Yang, T. Guo, T. Zhu, I. Tjuawinata, J. Zhao, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *Computer Standards & Interfaces*, p. 103827, 2023.

[8] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *22nd USENIX Security Symposium (USENIX Security 13)*, pp. 195–210, 2013.

[9] K. Huang, G. Ouyang, Q. Ye, H. Hu, B. Zheng, X. Zhao, R. Zhang, and X. Zhou, "Ldpguard: Defenses against data poisoning attacks to local differential privacy protocols," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[10] X. Cao, J. Jia, and N. Z. Gong, "Data poisoning attacks to local differential privacy protocols," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 947–964, 2021.

[11] X. Sun, Q. Ye, H. Hu, J. Duan, T. Wo, J. Xu, and R. Yang, "LDPRecover: Recovering frequencies from poisoning attacks against local differential privacy," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2024.

[12] J. Steinhardt, *Robust learning: Information theory and algorithms*. Stanford University, 2018.

[13] I. Diakonikolas and D. M. Kane, *Algorithmic high-dimensional robust statistics*. Cambridge university press, 2023.

[14] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American statistical association*, vol. 60, no. 309, pp. 63–69, 1965.

[15] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 729–745, 2017.

[16] S. Wang, L. Huang, P. Wang, Y. Nie, H. Xu, W. Yang, X.-Y. Li, and C. Qiao, "Mutual information optimally local private discrete distribution estimation," *arXiv preprint arXiv:1607.08025*, 2016.

[17] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under locally differential privacy," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5662–5676, 2018.

[18] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1120–1129, PMLR, 2019.

[19] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.

[20] H. Asi, V. Feldman, and K. Talwar, "Optimal algorithms for mean estimation under local differential privacy," in *International Conference on Machine Learning*, pp. 1046–1056, PMLR, 2022.

[21] W.-N. Chen, P. Kairouz, and A. Özgür, "Breaking the communication-privacy-accuracy trilemma," *IEEE Transactions on Information Theory*, vol. 69, no. 2, pp. 1261–1281, 2022.

[22] H. Asi, V. Feldman, J. Nelson, H. Nguyen, and K. Talwar, "Fast optimal locally private mean estimation via random projections," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] X. Xiong, S. Liu, D. Li, Z. Cai, and X. Niu, "A comprehensive survey on local differential privacy," *Security and Communication Networks*, vol. 2020, no. 1, p. 8829523, 2020.

[24] K. Talwar, S. Wang, A. McMillan, V. Jina, V. Feldman, P. Bansal, B. Basile, A. Cahill, Y. S. Chan, M. Chatzidakis, *et al.*, "Samplable anonymous aggregation for private federated data analysis," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[25] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479, SIAM, 2019.

[26] C. Dwork, F. McSherry, K. Nissim, and A. 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*, pp. 265–284, Springer, 2006.

[27] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?," *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[28] G. Cormode, T. Kulkarni, and D. Srivastava, "Marginal release under local differential privacy," in *Proceedings of the 2018 International Conference on Management of Data*, pp. 131–146, 2018.

[29] T. Wang, J. Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha, "Continuous release of data streams under both centralized and local differential privacy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1237–1253, 2021.

[30] Y. Zhang, Y. Zhu, S. Wang, and X. Huang, "Mean estimation of numerical data under (, δ)-utility-optimized local differential privacy," *IEEE Transactions on Information Forensics and Security*, 2024.

[31] J. C. Duchi, "Introductory lectures on stochastic optimization," *The mathematics of data*, vol. 25, pp. 99–186, 2018.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

[32] Z. Wang, Y. Sun, D. Liu, J. Hu, X. Pang, Y. Hu, and K. Ren, "Location privacy-aware task offloading in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2269–2283, 2023.

[33] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649, IEEE, 2019.

[34] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[35] "San Francisco fire department calls for service." http://bit.ly/336sddL, 2019.

[36] S. Ruggles, S. Flood, M. Sobek, D. Beckman, A. Chen, G. Cooper, S. Richards, R. Rogers, and M. Schouweiler, "IPUMS USA: Version 15.0 [dataset]," 2024.

[37] T. Wang, M. Lopuhaä-Zwakenberg, Z. Li, B. Skoric, and N. Li, "Locally differentially private frequency estimation with consistency," *arXiv preprint arXiv:1905.08320*, 2019.

[38] "kaggle nba players stats(2023 season)." https://www.kaggle.com/datasets/amirhosseinmirzaie/nba-players-stats2023-season/data, 2023.

[39] Z. Zhang, T. Wang, N. Li, S. He, and J. Chen, "CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy," in *ACM CCS*, 2018.

[40] L. Du, Z. Zhang, S. Bai, C. Liu, S. Ji, P. Cheng, and J. Chen, "AHEAD: Adaptive Hierarchical Decomposition for Range Query under Local Differential Privacy," in *ACM CCS*, 2021.

[41] Y. Du, Y. Hu, Z. Zhang, Z. Fang, L. Chen, B. Zheng, and Y. Gao, "LDPTrace: Locally Differentially Private Trajectory Synthesis," in *VLDB*, 2023.

[42] P. Zhao, L. Shen, R. Fan, Q. Li, H. Wu, J. Wu, and Z. Liu, "Learning with user-level local differential privacy," *arXiv preprint arXiv:2405.17079*, 2024.

[43] P. Zhao, J. Wu, Z. Liu, L. Shen, Z. Zhang, R. Fan, L. Sun, and Q. Li, "Enhancing learning with label differential privacy by vector approximation," in *International Conference on Learning Representations*, 2025.

[44] Y. Wu, X. Cao, J. Jia, and N. Z. Gong, "Poisoning attacks to local differential privacy protocols for key-value data," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 519–536, 2022.

[45] X. Li, N. Li, W. Sun, N. Z. Gong, and H. Li, "Fine-grained poisoning attack to local differential privacy protocols for mean and variance estimation," in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1739–1756, 2023.

[46] J. Acharya, Z. Sun, and H. Zhang, "Robust testing and estimation under manipulation attacks," in *International Conference on Machine Learning*, pp. 43–53, PMLR, 2021.

Appendix A. Optimal Attack

Attack strategy for frequency estimation. For targeted attack, the goal is to maximize the estimated frequency for a subset of alphabets $[d]$. In particular, we fix $\mathbf{u} \in \{0,1\}^d$. The positive elements in $\mathbf{u}$, i.e. $\{l|u_l = 1\}$, is the target set that we would like to maximize the frequency estimation. Denote $\mathcal{C}$ as the set of users the attacker can control. Such problem can be formulated as follows:

$$\max_{Z_i, i\in\mathcal{C}} \sum_{l=1}^{d} u_l(\hat{\mu}_l - \hat{\mu}_{cl}), \tag{40}$$

in which $\hat{\mu}_{cl}$ is the estimation without attack, i.e.

$$\hat{\mu}_{cl} = \frac{1}{c}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(Y_i = s_{il}) - a\right). \tag{41}$$

For all $i \notin \mathcal{C}$, $Z_i = Y_i$. From (16) and (41), the maximization of (40) over Z_i is equivalent to maximizing $\sum_{l=1}^{d} u_l \sum_{i\in\mathcal{C}} \mathbf{1}(Z_i = s_{il})$. Therefore, given $\mathcal{C}$ and $\mathbf{u}$, the attacker can pick Z_i as

$$Z_i = \arg\max_j \sum_{l=1}^{d} u_l \mathbf{1}(s_{il} = j). \tag{42}$$

Intuitively, (42) means that among k groups partitioned by the pre-defined information $\mathbf{s}_i$, the attacker picks the group that has largest overlap with the target $\{l|u_l = 1\}$.

Now it remains to design untargeted attack. Instead of maximizing the skew of estimate along a specific direction, now the attacker aims at maximizing the overall error. Therefore the attacker needs to find the best vector $\mathbf{u} \in \{0,1\}^d$, in order to maximize the overall ℓ_1 error introduced by the targeted attack above using target $\mathbf{u}$. Recall that our threat model allows the attacker to know the ground truth μ and the feedback Y_i, thus the attacker can calculate $\hat{\mu}_c = (\hat{\mu}_{c1}, \ldots, \hat{\mu}_{cd})$. Given such knowledge, a simple strategy is to let $u_l = 1$ if $\hat{\mu}_{cl} > \mu_l$, such that the error caused by manipulation is in the same direction with the error without attack.

Attack strategy for ℓ_1 mean estimation. Similar to the frequency estimation problem, we design optimal strategies for targeted and untargeted attacks separately. Define

$$\hat{\mu}_c = \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{x}}_{ci}, \tag{43}$$

$$\hat{\mathbf{x}}_{ci} = \min\left\{1, \frac{T}{\|\hat{\mathbf{x}}_{ci0}\|_1}\right\}\hat{\mathbf{x}}_{ci0}, \tag{44}$$

$$\hat{\mathbf{x}}_{ci0} = \frac{1}{k}\sum_{j=1}^{k}\mathbf{s}_{ij}Y_{ij}. \tag{45}$$

$\hat{\mu}_c$ is the estimation without attack. For targeted attack, let $\mathbf{u} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ be arbitrary nonzero d dimensional vector. We hope to maximize $\langle \mathbf{u}, \hat{\mu} - \hat{\mu}_c\rangle$. Given $\mathbf{u}$ and the set of corrupted users $\mathcal{C}$, it remains to pick appropriate Z_i to maximize $\langle \mathbf{u}, \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{ci}\rangle$. Recall that in (21), $|Y_{ij}| \le C$, with C defined in (22). To prevent the attack from being directly detected, we need to ensure that $|Z_{ij}| \le C$. By (25) and (27), we let $Z_{ij} = C$ if $\langle \mathbf{u}, \mathbf{s}_{ij}\rangle > 0$, otherwise $Z_{ij} = -C$.

The design of untargeted attack shares similar idea with the frequency estimation problem. The goal is to pick an appropriate $\mathbf{u}$ to maximize $\|\hat{\mu} - \mu\|_1$. A simple solution is to let $\mathbf{u} = \hat{\mu}_c - \mu$, then the error caused by manipulation is in the same direction with the error without attack.

Attack strategy for ℓ_2 mean estimation. Let

$$\hat{\mu}_c = \frac{c_d}{c_\epsilon}\frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{s}_{ij}Y_{ij} \tag{46}$$

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

be the estimation with clean data. Given fixed $\mathbf{u} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$, the targeted attack aims at maximizing $\langle \mathbf{u}, \hat{\mu} - \hat{\mu}_c \rangle$. From (34) and (46), this is equivalent to maximize $\langle \mathbf{u}, \mathbf{s}_{ij} \rangle Z_{ij}$ for each i and j. Therefore, we let $Z_{ij} = 1$ if $\langle \mathbf{u}, \mathbf{s}_{ij} \rangle > 0$, otherwise $Z_{ij} = -1$.

For untargeted attack, the goal is to maximize $\|\hat{\mu} - \mu\|_2$. Therefore, it suffices to just impose targeted attack with $\mathbf{u} = \hat{\mu}_c - \mu$.

Appendix B. Unbiasedness of Frequency Estimate

Note that the probability that $Y_i = s_{il}$ depends on whether $x_i = l$. Given $x_i = l$, According to (15), we have $\mathrm{P}(Y_i = s_{il} | x_i = l) = e^\epsilon/(e^\epsilon + k - 1)$. On the contrary, given $x_i \neq l$, then from (15),

$$\begin{aligned} \mathrm{P}(Y_i = s_{il} | x_i \neq l) &= \frac{e^\epsilon}{e^\epsilon + k - 1} \mathrm{P}(s_{i,x_i} = s_{il} | x_i \neq l) \\ &\quad + \frac{1}{e^\epsilon + k - 1} \mathrm{P}(s_{i,x_i} \neq s_{il} | x_i \neq l). \end{aligned} \quad (47)$$

Recall that the signal $\mathbf{s}_i$ equally and randomly allocate the alphabet into k groups. If $x_i \neq l$, then the probability that x_i and l are allocated into the same group, i.e. $s_{i,x_i} = s_{il}$, is $(d/k - 1)/(d - 1)$. Therefore

$$\begin{aligned} &\mathrm{P}(Y_i = s_{il} | x_i \neq l) \\ &= \frac{e^\epsilon}{e^\epsilon + k - 1} \frac{d - k}{k(d-1)} + \frac{1}{e^\epsilon + k - 1} \frac{d(k-1)}{k(d-1)} = a. \end{aligned} \quad (48)$$

Denote $x_{1:n} := (x_1, \ldots, x_n)$. Then

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \mathbf{1}(Y_i = s_{il}) | x_{1:n} \right] &= \frac{e^\epsilon}{e^\epsilon + k - 1} \mu_l + a(1 - \mu_l) \\ &= c\mu_l + a. \end{aligned} \quad (49)$$

From (49), without attack (i.e. $Z_i = Y_i$ for all i), the estimator (16) is unbiased, i.e. $\mathbb{E}[\hat{\mu}_l] = \mu_l$.

Appendix C. Proof of Theorem 1

Recall the definition of $\hat{\mu}_{cl}$ defined in (41). Let $\hat{\mu}_c = (\hat{\mu}_{c1}, \ldots, \hat{\mu}_{cd})$. $\hat{\mu}_c$ is the estimate without corruption. Now analyze the error caused by honest execution $\|\hat{\mu}_c - \mu\|_1$ and the error caused by manipulation $\|\mu - \mu_c\|$ separately.

The error by honest execution. It can be easily shown that $\mathrm{Var}[\mathbf{1}(Y_i = s_{il}) | x_i = l] = e^\epsilon(k-1)/(e^\epsilon + k - 1)^2$, and $\mathrm{Var}[\mathbf{1}(Y_i = s_{il}) | x_i \neq l] = a(1-a)$. Therefore

$$\begin{aligned} \mathrm{Var}[\hat{\mu}_{cl} | x_{1:n}] &= \frac{1}{n^2 c^2} \sum_{i=1}^n \mathrm{Var}[\mathbf{1}(Y_i = s_{il}) | x_i] \\ &= \frac{1}{nc^2} \left[\mu_l \frac{e^\epsilon(k-1)}{(e^\epsilon + k - 1)^2} + (1 - \mu_l) a(1-a) \right]. \end{aligned} \quad (50)$$

The overall ℓ_2 error is

$$\begin{aligned} \mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2^2 | x_{1:n} \right] &= \sum_{l=1}^d \mathrm{Var}[\hat{\mu}_{cl}] \\ &= \frac{1}{nc^2} \frac{e^\epsilon(k-1)}{(e^\epsilon + k - 1)^2} + \frac{a(1-a)}{nc^2} \sum_{l=1}^d (1 - \mu_l) \\ &\leq \frac{1}{nc^2} \frac{e^\epsilon(k-1)}{(e^\epsilon + k - 1)^2} + \frac{a(1-a)d}{nc^2}. \end{aligned} \quad (51)$$

1) $\epsilon < 1$: Let $k = 2$. From (18), $c \sim \epsilon$. Therefore $\mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2^2 \right] \lesssim d/(n\epsilon^2)$.

2) $1 \leq \epsilon \leq \ln d$: Let $k \sim e^\epsilon$. From (17) and (18), $a \sim 1/e^\epsilon$ and $c \sim 1$. Therefore $\mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2^2 \right] \lesssim d/(ne^\epsilon)$.

3) $\epsilon > \ln d$: Let $k = d$. From (17), now $a = 1/(e^\epsilon + d - 1)$, $c = (e^\epsilon - 1)/(e^\epsilon + d - 1)$. Then $\mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2^2 \right] \lesssim d/(ne^\epsilon)$.

Case 2 and 3 appear to have the same form. Combine these three cases, by Cauchy's inequality,

$$\begin{aligned} \mathbb{E}\left[\|\hat{\mu}_c - \mu\|_1 \right] &\leq \sqrt{d} \mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2 \right] \\ &\lesssim \begin{cases} \frac{d}{\sqrt{n\epsilon^2}} & \text{if} \quad \epsilon < 1 \\ \frac{d}{\sqrt{ne^\epsilon}} & \text{if} \quad \epsilon \geq 1. \end{cases} \end{aligned} \quad (52)$$

The error caused by manipulation. For notational simplicity, in the following steps, $\max_{Z_{1:n}}$ means taking maximum over all $(Z_1, \ldots, Z_n)$ such that $Z_i \neq Y_i$ for at most q elements. From the definition of $\hat{\mu}$ and $\hat{\mu}_c$ in (16) and (41),

$$\begin{aligned} &\max_{Z_{1:n}} \|\hat{\mu} - \hat{\mu}_c\|_1 \\ &= \frac{1}{nc} \max_{Z_{1:n}} \sum_{l=1}^d \left| \sum_{i \in \mathcal{C}} (\mathbf{1}(Z_i = s_{il}) - \mathbf{1}(Y_i = s_{il})) \right| \\ &\leq \frac{2}{nc} \max_{Z_{1:n}} \sum_{l=1}^d \left| \sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) - \frac{q}{k} \right|. \end{aligned} \quad (53)$$

The second step holds because

$$\begin{aligned} &\left| \sum_{i \in \mathcal{C}} (\mathbf{1}(Z_i = s_{il}) - \mathbf{1}(Y_i = s_{il})) \right| \\ &= \left| \left(\sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) - \frac{q}{k} \right) - \left(\sum_{i \in \mathcal{C}} \mathbf{1}(Y_i = s_{il}) - \frac{q}{k} \right) \right| \\ &\leq \left| \sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) - \frac{q}{k} \right| + \left| \sum_{i \in \mathcal{C}} \mathbf{1}(Y_i = s_{il}) - \frac{q}{k} \right|. \end{aligned} \quad (54)$$

Note that

$$\begin{aligned} &\sum_{l=1}^d \left| \sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) \right| \\ &= \max_{\mathbf{u} \in \{-1,1\}^d} \sum_{l=1}^d u_l \left(\sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) - \frac{q}{k} \right). \end{aligned} \quad (55)$$

Since s_{il} is randomly taken from $\{1, \ldots, k\}$, we have $\mathrm{P}(Z_i = s_{il}) = 1/k$. Define

$$V(\mathbf{u}, \mathcal{C}, \mathbf{Z}) = \frac{1}{qd} \left(\sum_{i \in \mathcal{C}} \mathbf{1}(Z_i = s_{il}) - \frac{q}{k} \right), \quad (56)$$

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

in which $\mathbf{Z} = (Z_1, \ldots, Z_n)$. Then $\mathbb{E}[V(\mathbf{u}, \mathcal{C}, \mathbf{Z})] = 0$. From (53),

$$\mathbb{E}\left[\max_{Z_{1:n}} \|\hat{\mu} - \hat{\mu}_c\|_1\right] \leq \frac{2qd}{nc} \mathbb{E}\left[\max_{Z_{1:n}, \mathbf{u}} V(\mathbf{u}, \mathcal{C}, \mathbf{Z})\right]. \quad (57)$$

From Chernoff's inequality, for $t > 0$, $\mathrm{P}(V(\mathbf{u}, \mathcal{C}, \mathbf{Z}) > t) \leq e^{-qdD_{KL}(t+\frac{1}{k}||\frac{1}{k})}$, in which D_{KL} denotes the Kullback-Leibler divergence, and $\mathrm{P}(V(\mathbf{u}, \mathcal{C}, \mathbf{Z}) < -t) \leq e^{-qdD_{KL}(\frac{1}{k}-t||\frac{1}{k})}$. If $t \leq 1/(2k)$, then

$$\max\left\{D_{KL}(\frac{1}{k} - t||\frac{1}{k}), D_{KL}(t + \frac{1}{k}||\frac{1}{k})\right\} \geq \frac{1}{4}kt^2. \quad (58)$$

Therefore $\mathrm{P}(|V(\mathbf{u}, \mathcal{C}, \mathbf{Z})| > t) \leq e^{-\frac{1}{4}qdkt^2}$, and

$$\begin{aligned} &\mathrm{P}\left(\left|\max_{Z_{1:n}, \mathcal{C}, \mathbf{u}} V(\mathbf{u}, \mathcal{C}, \mathbf{z})\right| > t\right) \leq 2^d \binom{n}{q} k^q e^{-\frac{1}{4}qkdt^2} \\ \leq\ & e^{d\ln 2 + 2q\ln n - \frac{1}{4}qkdt^2}. \end{aligned} \quad (59)$$

Hence

$$\mathbb{E}\left[\left|\max_{Z_{1:n}, \mathcal{C}, \mathbf{u}} V(\mathbf{u}, \mathcal{C}, \mathbf{Z})\right|\right] \lesssim \frac{1}{\sqrt{k}}\left(\frac{1}{\sqrt{q}} + \sqrt{\frac{\ln n}{d}}\right). \quad (60)$$

If $\epsilon < 1$, then let $k = 2$. If $1 \leq \epsilon \leq \ln d$, let $k \in [e^\epsilon, 2e^\epsilon]$. Finally, if $\epsilon > \ln d$, let $k = d$. Combine the error by honest execution (52) and by manipulation (57), (60), the overall error bound is

$$\mathbb{E}[\|\hat{\mu} - \mu\|_1] \lesssim \begin{cases} \frac{d}{\sqrt{ne^2}} + \frac{q\sqrt{d\ln n}}{n\epsilon} & \text{if} \quad \epsilon < 1 \\ \frac{d}{\sqrt{ne^\epsilon}} + \frac{q\sqrt{d\ln n}}{n\sqrt{e^\epsilon}} & \text{if} \quad 1 \leq \epsilon \leq \ln d \\ \frac{d}{\sqrt{ne^\epsilon}} + \frac{q\sqrt{\ln n}}{n} & \text{if} \quad \epsilon > \ln d. \end{cases} \quad (61)$$

The proof of Theorem 1 is complete.

Appendix D.
Proof of Theorem 2

Define $\mathbf{x}_{ci} := \mathbb{E}[\hat{\mathbf{x}}_{ci}|\mathbf{x}_i]$. Recall that $\hat{\mathbf{x}}_{ci0}$ is unbiased, i.e. $\mathbb{E}[\hat{\mathbf{x}}_{ci0}|\mathbf{x}_i] = \mathbf{x}_i$. However, the estimator (44) has some bias due to the clipping operation with threshold T. We use $\mathbf{x}_{ci}$ to denote the expectation after clipping.

Then the estimation error can be decomposed as follows.

$$\begin{aligned} \|\hat{\mu} - \mu\| &= \left\|\hat{\mu} - \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|_1 \leq \left\|\hat{\mu}_c - \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{ci}\right\|_1 \\ &+ \left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_{ci} - \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|_1 + \|\hat{\mu} - \hat{\mu}_c\|_1 \\ &:= I_1 + I_2 + I_3. \end{aligned} \quad (62)$$

I_1 is the error caused by randomness of honest execution. I_2 is the clipping bias. I_3 is the error from manipulation.

Bound of I_1. Note that

$$\|\mathbf{s}_{ij}Y_{ij}\|_2 \leq \sqrt{d}\frac{e^{\frac{\epsilon_0}{2}}+1}{e^{\frac{\epsilon_0}{2}}-1} = \sqrt{d}\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}. \quad (63)$$

From (26), $\mathbb{E}[\mathbf{s}_{ij}Y_{ij}|\mathbf{x}_i] = \mathbf{x}_i$. Therefore

$$\mathbb{E}\left[\|\mathbf{s}_{ij}Y_{ij} - \mathbf{x}_i\|_2^2\right] \leq \mathbb{E}\left[\|\mathbf{s}_{ij}Y_{ij}\|_2^2\right] \leq d\left(\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}\right)^2. \quad (64)$$

From the definition of $\hat{\mathbf{x}}_{ci0}$ in (45), since $\mathbf{s}_{ij}Y_{ij}$ are independent for different j, we have

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{ci0} - \mathbf{x}_i\|_2^2\right] \leq \frac{d}{k}\left(\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}\right)^2. \quad (65)$$

The clipping operation always reduces the variance. From (44), $\mathrm{Var}[\hat{\mathbf{x}}_{ci}] \preceq \mathrm{Var}[\hat{\mathbf{x}}_{ci0}]$, in which Var is the covariance matrix of a random vector, i.e. $\mathrm{Var}[\mathbf{U}] = \mathbb{E}[(\mathbf{U} - \mathbb{E}[\mathbf{U}])(\mathbf{U} - \mathbb{E}[\mathbf{U}])^T]$. Note that $\mathbb{E}[\hat{\mu}_c] = \mathbb{E}[\hat{\mathbf{x}}_{ci}] = \mathbf{x}_{ci}$. Therefore

$$\begin{aligned} &\mathbb{E}[\|\hat{\mathbf{x}}_{ci} - \mathbf{x}_{ci}\|_2^2] = \mathbb{E}\left[\mathrm{tr}(\mathrm{Var}[\hat{\mathbf{x}}_{ci}])\right] \\ &\leq \mathbb{E}\left[\mathrm{tr}(\mathrm{Var}[\hat{\mathbf{x}}_{ci0}])\right] = \mathbb{E}\left[\|\hat{\mathbf{x}}_{ci0} - \mathbf{x}_i\|_2^2\right] \\ &= \frac{d}{k}\left(\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}\right)^2, \end{aligned} \quad (66)$$

in which tr denotes the trace of a matrix. Therefore

$$\mathbb{E}\left[\left\|\hat{\mu}_c - \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{ci}\right\|_2^2\right] \leq \frac{d}{nk}\left(\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}\right)^2. \quad (67)$$

The expectation of ℓ_1 distance can thus be bounded by

$$\mathbb{E}[I_1] \leq \mathbb{E}\left[\left\|\hat{\mu}_c - \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{ci}\right\|_1\right] \leq \frac{d}{\sqrt{nk}}\frac{e^{\frac{\epsilon}{2k}}+1}{e^{\frac{\epsilon}{2k}}-1}. \quad (68)$$

Bound of I_2. Now we bound the bias introduced by the clipping operation. From the definition of $\mathbf{x}_{ci}$ at the beginning of Appendix D,

$$\begin{aligned} &\|\mathbf{x}_i - \mathbf{x}_{ci}\|_1 = \|\mathbb{E}[\hat{\mathbf{x}}_{ci0} - \hat{\mathbf{x}}_{ci}]\|_1 \\ \overset{(a)}{=}\ & \left\|\mathbb{E}\left[\hat{\mathbf{x}}_{ci0} - \min\left\{1, \frac{T}{\|\hat{\mathbf{x}}_{ci0}\|_1}\right\}\hat{\mathbf{x}}_{ci0}\right]\right\|_1 \\ \overset{(b)}{\leq}\ & \mathbb{E}\left[\left\|\hat{\mathbf{x}}_{ci0} - \min\left\{1, \frac{T}{\|\hat{\mathbf{x}}_{ci0}\|_1}\right\}\hat{\mathbf{x}}_{ci0}\right\|_1\right] \\ =\ & \mathbb{E}\left[(\|\hat{\mathbf{x}}_{ci0}\|_1 - T)\mathbf{1}(\|\hat{\mathbf{x}}_{ci0}\|_1 > T)\right] \\ =\ & \int_T^\infty \mathrm{P}(\|\hat{\mathbf{x}}_{ci0}\|_1 > t)dt. \end{aligned} \quad (69)$$

(a) comes from the definition of $\hat{\mathbf{x}}_{ci}$ in (44). (b) uses the Jensen's inequality. It remains to get a high probability bound of $\|\mathbf{x}_{ci0}\|_1$. Note that

$$\begin{aligned} \|\hat{\mathbf{x}}_{ci0}\|_1 &= \left\|\frac{1}{k}\sum_{j=1}^k \mathbf{s}_{ij}Y_{ij}\right\|_1 \\ &\leq \left\|\frac{1}{k}\mathbf{s}_{ij}\mathbf{s}_{ij}^T\mathbf{x}_i\right\|_1 + \left\|\frac{1}{k}\sum_{j=1}^k \mathbf{s}_{ij}(Y_{ij} - \mathbf{s}_{ij}^T\mathbf{x}_i)\right\|_1. \end{aligned} \quad (70)$$

For the first term of (70), let $\mathbf{M} = (1/k)\sum_{j=1}^k \mathbf{s}_{ij}\mathbf{s}_{ij}^T$. Define the ℓ_1 operator norm as $\|\mathbf{M}\|_1 := \sup_{\mathbf{u}} \|\mathbf{M}\mathbf{u}\|_1 / \|\mathbf{u}\|_1$. It can be easily shown that $\|\mathbf{M}\|_1 = \max_{l'} \sum_{l=1}^d |M_{ll'}|$, in

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

which $M_{ll'}$ denotes the (l,l') element of matrix $\mathbf{M}$. Note that $M_{ll'} = (1/k)\sum_{j=1}^{k} s_{ijl}s_{ijl'}$. By Hoeffding's inequality, for all $l \neq l'$, $\mathrm{P}(|M_{ll'}| > t) \leq 2e^{-\frac{1}{2}kt^2}$. Thus

$$\mathrm{P}\left(\max_{l\neq l'}|M_{ll'}| > t\right) \leq 2d(d-1)e^{-\frac{1}{2}kt^2}. \tag{71}$$

If $l = l'$, $M_{ll'} = (1/k)\sum_{j=1}^{k} s_{ijl}^2 = 1$. Therefore

$$\mathrm{P}(\|\mathbf{M}\|_1 > 1 + t(d-1)) \leq 2d(d-1)e^{-\frac{1}{2}kt^2}, \tag{72}$$

and thus $\mathrm{P}(\|\mathbf{M}\mathbf{x}_i\|_1 > t) \leq 2d(d-1)e^{-\frac{k(t-1)^2}{2(d-1)^2}}$. For the second term in (70), by Hoeffding's inequality, it can be shown that

$$\mathrm{P}\left(\left\|\frac{1}{k}\sum_{j=1}^{k}\mathbf{s}_{ij}(Y_{ij} - \mathbf{s}_{ij}^T\mathbf{x}_i)\right\|_1 > t\right) \leq de^{-\frac{kt^2}{2c_\epsilon^2 d^2}}. \tag{73}$$

Therefore,

$$\mathrm{P}\left(\|\mathbf{u}_i\|_1 > t\right) \leq 2d(d-1)e^{-\frac{k(\frac{1}{2}t-1)^2}{2(d-1)^2}} + de^{-\frac{kt^2}{8c_\epsilon^2 d^2}}. \tag{74}$$

Let $T \sim \max\{1, c_\epsilon d\ln(nd)/\sqrt{k}\}$, then from (69),

$$\mathbb{E}[I_2] \leq \max_i \|\mathbf{x}_{ci} - \mathbf{x}_i\|_1 \lesssim \frac{1}{n}. \tag{75}$$

Bound of I_3. The error with clipping can then be upper bounded by the corresponding error without clipping. Therefore

$$\begin{aligned}
\|\hat{\mu} - \hat{\mu}_c\|_1 &= \frac{1}{nk}\left\|\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}(Z_{ij} - Y_{ij})\right\|_1 \\
&\leq \frac{2}{nk}\sup_{Z_{ij}}\left\|\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}Z_{ij}\right\|_1 \\
&\leq \frac{2}{nk}\max_{\mathbf{u}\in\{-1,1\}^d}\sup_{Z_{ij}}\mathbf{u}^T\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}Z_{ij}.
\end{aligned} \tag{76}$$

To reach the supremum, Z_{ij} take values in $[-c_\epsilon, c_\epsilon]$. For each fixed value of $\mathbf{u}$ and Z_{ij},

$$\mathrm{P}\left(\mathbf{u}^T\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}Z_{ij} > t\right) \leq e^{-\frac{t^2}{2c_\epsilon^2 qkd}}. \tag{77}$$

$\mathbf{u}$ can take 2^d values. Since i and j take values from $\{1,\ldots,n\}$ and $\{1,\ldots,k\}$, there are 2^{qk} values for $Z_{ij}, i = 1,\ldots,n, j = 1,\ldots,k$. Hence the union bound is

$$\mathrm{P}\left(\max_{\mathbf{u}\in\{-1,1\}^d}\sup_{Z_{ij}}\mathbf{u}^T\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}Z_{ij} > t\right) \leq 2^{d+qk}e^{-\frac{t^2}{2c_\epsilon^2 qkd}}. \tag{78}$$

Therefore

$$\begin{aligned}
&\mathbb{E}\left[\max_{\mathbf{u}\in\{-1,1\}^d}\sup_{Z_{ij}}\mathbf{u}^T\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}Z_{ij}\right] \\
\leq\ & \int_0^\infty \min\{1, 2^{d+qk}e^{-\frac{t^2}{2c_\epsilon^2 qkd}}\}dt \\
=\ & \sqrt{(2\ln 2)qkd(d+qk)}c_\epsilon \\
&+ \int_{\sqrt{(2\ln 2)qkd(d+qk)}c_\epsilon}^{\infty} e^{-\frac{t^2}{2c_\epsilon^2 qkd}}dt \\
\lesssim\ & (d\sqrt{qk} + qk\sqrt{d})c_\epsilon.
\end{aligned} \tag{79}$$

From (76),

$$\mathbb{E}[I_3] \lesssim \frac{c_\epsilon}{nk}(d\sqrt{qk} + qk\sqrt{d}). \tag{80}$$

Now we use the second approach to give another bound of $\mathbb{E}[I_3]$. Recall (28) and (43), we have $\hat{\mu} - \hat{\mu}_c = (1/n)\sum_{i=1}^{n}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{ci})$. Note that $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{ci}\| \leq 2T$. Thus

$$I_3 \leq \frac{1}{n}\sum_{i\in\mathcal{C}}\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{ci}\|_1 \leq \frac{1}{n}\sum_{i\in\mathcal{C}}2T = \frac{2qT}{n}. \tag{81}$$

Combine (80) and (81), recall that $T \sim \max\{1, c_\epsilon d\ln(nd)/\sqrt{k}\}$,

$$\mathbb{E}[I_3] \lesssim c_\epsilon\frac{q}{n}\min\left\{\sqrt{d}, 1 + \frac{d\ln(nd)}{\sqrt{k}}\right\} + c_\epsilon\frac{d\sqrt{q}}{n\sqrt{k}}. \tag{82}$$

Combine (68), (75), (82) and (62),

$$\mathbb{E}\left[\|\hat{\mu} - \mu\|_1\right] \lesssim c_\epsilon\left(\frac{d}{\sqrt{nk}} + \frac{q}{n}\min\left\{\sqrt{d}, \frac{d\ln(nd)}{\sqrt{k}}\right\}\right). \tag{83}$$

The proof of Theorem 2 is complete.

Appendix E.
Proof of Theorem 3

The error caused by honest execution. Define

$$\hat{\mathbf{x}}_{ci} := c_d\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{1}{k}\sum_{j=1}^{k}\mathbf{s}_{ij}Y_{ij}, \tag{84}$$

and

$$\hat{\mu}_c := \frac{1}{n}\sum_{i=1}^{n}\hat{\mathbf{x}}_{ci} = c_d\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{s}_{ij}Y_{ij}. \tag{85}$$

In Lemma 1, we have shown that $\mathbb{E}[\hat{\mathbf{x}}_{ci}] = \mathbf{x}_i$. This result indicates that the estimation is unbiased without manipulation. Therefore, we only need to bound the variance. Since $\|\mathbf{s}_{ij}Y_{ij}\| \leq 1$ always hold, we have $\mathrm{Var}[\mathbf{s}_{ij}Y_{ij}] \preceq \mathbf{I}$, then $\mathrm{Var}[\hat{\mu}_c] \preceq c_d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\frac{1}{nk}$. Hence $\mathbb{E}\left[\|\hat{\mu}_c - \mu\|_2\right] \leq c_d\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{1}{\sqrt{nk}}$.

Error caused by manipulation.

$$\begin{aligned}
\|\hat{\mu} - \hat{\mu}_c\|_2 &= c_d\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{1}{nk}\left\|\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{s}_{ij}(Z_{ij} - Y_{ij})\right\|_2 \\
&= c_d\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{1}{nk}\left\|\sum_{i\in\mathcal{C}}\sum_{j=1}^{k}\mathbf{s}_{ij}(Z_{ij} - Y_{ij})\right\|_2 \\
&\lesssim \frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\frac{q}{n}.
\end{aligned} \tag{86}$$

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.

Appendix F.
Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

F.1. Summary

This paper proposes a robust estimator under local DP called Robust LDP. The estimator is used for frequency and l1/l2 mean estimation. The estimator uses predetermined information sent to users to limit attacker efficacy at skewing the aggregate results. Theoretical guarantees compare against both robust estimators and non-robust estimators and show significant improvement, particularly for $\epsilon > 1$. Empirical results show a similar improvement even under attacks.

F.2. Scientific Contributions

- A valuable step forward in the established literatures of robust estimation and local differential privacy.
- New method for countering manipulation attacks

F.3. Reasons for Acceptance

1) The paper's technical approach is sound and the paper offers a measurable improvement both theoretically and empirically.
2) The paper's technical details were well presented and that the paper was mostly well written.
3) The reviewers found the scientific exploration of the paper to be thorough.

F.4. Noteworthy Concerns

1. One concern was with the presentation for the technical details. There were notation and expressions that were not clearly or fully defined and heavy reliance on references. The paper should be made to be more self-contained and clear in the theoretical details.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 07,2026 at 15:11:34 UTC from IEEE Xplore. Restrictions apply.