N-output Mechanism: Estimating Statistical Information from Numerical Data under Local Differential Privacy

Incheol Baek
Korea University
Seoul, Korea
inch307@korea.ac.kr

Yon Dohn Chung
Korea University
Seoul, Korea
ydchung@korea.ac.kr

ABSTRACT

Local Differential Privacy (LDP) has addressed significant privacy concerns arising from the collection of data containing sensitive information. LDP mechanisms enable users to perturb their data before sharing it with an aggregator. In this work, we focus on numerical data collection and statistical estimation under LDP.

Most LDP mechanisms for numerical data focus on the accuracy of mean estimation but lack optimization in both accuracy and communication efficiency. To address these limitations, we propose the N-output mechanism, which converts continuous numerical data into one of N discrete outputs, enhancing accuracy while reducing communication overhead. We further develop a Hybrid Mechanism (HM-NP) that combines the N-output mechanism with a sub-optimal Piece-wise mechanism, improving theoretical accuracy.

Additionally, we introduce Two-Phase Expectation Maximization (2PEM), a post-processing method that extends HM-NP for distribution estimation and allows the derivation of other statistical measures, such as variance. Finally, we present NP-AAA, which integrates HM-NP with 2PEM and the existing Advanced Adaptive Additive (AAA) mechanism for precise mean estimation.

Our experiments validate the effectiveness of these mechanisms in statistical measure estimation using synthetic and real-world datasets. We also demonstrate their applicability in practical settings through a federated learning case study.

PVLDB Reference Format:
Incheol Baek and Yon Dohn Chung. N-output Mechanism: Estimating Statistical Information from Numerical Data under Local Differential Privacy. PVLDB, 14(1): XXX-XXX, 2025.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/inch307/N-output_mechanism.

1 INTRODUCTION

In today's interconnected world, collecting and analyzing data from various devices, such as computers, phones, and IoT devices, has become increasingly important. However, these data often contain sensitive information, raising significant privacy concerns.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

Local Differential Privacy (LDP) [14] addresses these concerns. When users report their data to an aggregator, they perturb their data using an LDP mechanism in their local setting before sending them to the aggregator. This process ensures that the true data remains hidden from the aggregator, which only access to the perturbed data. Despite this, the aggregator can still estimate statistical information and support data-driven applications while strongly protecting individual privacy. In practice, big tech companies have adopted LDP techniques: Google in its Chrome browser [8], Apple for emoji usage [19], and Microsoft for telemetry data [5].

LDP mechanisms require a privacy parameter, denoted by ϵ, which is referred to as the privacy budget. The value of ϵ rigorously limits the amount of individual information disclosure. For example, lower ϵ values reduce the amount of information that can be utilized, making the analysis less useful but providing stronger privacy protection. In this context, there is a trade-off between privacy protection and estimation accuracy, controlled by ϵ.

Numerous studies have focused on improving the estimation accuracy of LDP mechanisms to mitigate this trade-off. Various LDP mechanisms have been developed for different types of data, such as categorical data [1, 8, 13, 21, 24], numerical data [6, 15, 20, 23, 25], and spatial data [3, 12], each designed for specific tasks like frequency estimation[1, 8, 13, 15, 21, 24], heavy hitter estimation [18, 22], and mean estimation [6, 20, 23, 25].

In this work, we focus on numerical data and various statistical estimations, with a particular emphasis on mean estimation. Most research on numerical data has concentrated on improving mean estimation [6, 20, 23, 25]. Most of these mechanisms utilize the minimax principle, originating from game theory, which aims to minimize the maximum possible loss. In the context of mean estimation, this translates to minimizing the worst-case error, commonly referred to as the worst-case noise variance.

An LDP mechanism, denoted as $\mathcal{M} : \mathcal{X} \rightarrow \Omega$, maps an input value to an perturbed value. Generally, $\mathcal{X}$ is assumed to be $[-1, 1]$, but the number of possible outputs, $|\Omega|$, varies across different mechanisms and is a critical factor influencing their design and effectiveness.

In Duchi's mechanism [6], the output space is defined as $\Omega = \{-\omega, \omega\}$, where $|\Omega| = 2$. An input value is randomly encoded into one of the two values, $\{-\omega, \omega\}$. To preserve the statistical information, the original data point is more likely to be encoded as the nearest value. The perturbed value is reported to the aggregator, who can estimate the mean by averaging the received values. For $|\Omega| = 3$, Zhao et al. studied a three-output mechanism [25].

Additionally, there is a family of Piecewise mechanism (PMs) [20, 25] that utilize continuous output space, where $|\Omega| = \infty$. Each of these mechanisms offers different advantages depending on ϵ.

As shown in Table 1, mechanisms with larger $|\Omega|$ tend to have lower worst-case noise variance in settings with larger ϵ. However,

Table 1: Best mechanisms according to ϵ

ϵ	Best mechanism	
$0 < \epsilon < 0.69$	Duchi's	$\|\Omega\| = 2$
$0.69 < \epsilon < 2.56$	Three-output	$\|\Omega\| = 3$
$\epsilon > 2.56$	PM-sub	$\|\Omega\| = \infty$

it is notable that there are currently no mechanisms available between $|\Omega| = 3$ and $|\Omega| = \infty$, which remains an open problem in this field.

Contribution. Motivated by this gap, we develop a novel LDP mechanism called the N-output mechanism. The N-output mechanism encodes a value into one of N possible outputs, where $|\Omega| = N$. This mechanism is designed to adapt to the value of ϵ by appropriately choosing N and Ω to minimize the worst-case noise variance. Typically, the optimal value of N increases as ϵ increases. Consequently, the N-output mechanism generalizes Duchi's mechanism and the three-output mechanism, achieving the best worst-case noise variance for $0 < \epsilon < 3.5$ and $3.7 < \epsilon < 4.15$. In the other ranges, the N-output mechanism achieves the worst-case noise variance that is close to that of PM-sub, with only a slight difference. Additionally, the N-output mechanism is communication-efficient due to its discrete outputs.

We also develop a hybrid mechanism that combines the N-output mechanism and PM-sub to further reduce the worst-case noise variance. We refer to this hybrid mechanism as HM-NP. By optimizing HM-NP, it outperforms other LDP mechanisms in terms of minimizing the worst-case noise variance.

In addition to mean estimation, we extend our approach to distribution estimation. Previous work on distribution estimation for numerical data introduced Square Wave (SW) [15] report mechanism, which focuses on optimizing the Wasserstein distance, an error measure for distribution estimation. They utilzes Expectation-Maximization (EM) [4] algorithm on results of SW to estimate distributions. However, applying EM directly to HM-NP results in poor accuracy due to the heterogeneity of the two mechanism involved in HM-NP. To address this, we develop a Two-Phase Expectation-Maximization (2PEM) to extend HM-NP for distribution estimation with high accuracy. This advancement in distribution estimation also opens the door to estimating other statistical measures, such as variance, that can be derived from the estimated distribution.

The mechanisms we presented above all assume that the original data distribution is unknown. The Advanced Adaptive Additive (AAA) mechanism [23], however, is a distribution-aware mechanism designed for mean estimation. It operates in two phases: the first phase focuses on distribution estimation, while the second phase is dedicated to mean estimation. Consequently, the effectiveness of the AAA mechanism is dependent on the accuracy of the distribution estimation. Additionally, the data used for distribution estimation cannot be reused for mean estimation.

To address these limitations, we propose a method called NP-AAA, which leverages HM-NP and 2PEM in the first phase. This approach enhances distribution estimation while allowing the data used in the first phase to be reused in the second phase for mean estimation. This approach improves the accuracy of the overall estimation process.

Our contribution can be summarized as follows:

- We propose a novel LDP mechanism, the N-output mechanism, which generalizes existing mechanisms like Duchi's and the three-output mechanism, and achieves the best worst-case noise variance for a wide range of ϵ values while being communication-efficient.
- We introduce a hybrid mechanism, HM-NP, that combines the N-output mechanism and PM-sub, optimized to further minimize the worst-case noise variance, outperforming existing LDP mechanisms.
- We extend our approach to distribution estimation by developing 2PEM, which is applied to the outputs of HM-NP. This method can accurately estimate the distribution and derive other statistical measures from the estimated distribution.
- We address the limitations of the AAA mechanism by proposing NP-AAA that enhances distribution and mean estimation using HM-NP and 2PEM.
- We empirically demonstrate that the N-output mechanism and HM-NP exhibit competitive performance compared to existing LDP mechanisms in mean estimation. Specifically, NP-AAA outperforms others in mean estimation. Furthermore, HM-NP combined with 2PEM excels in distribution estimation and the estimation of other statistical measures. Additionally, we conduct a case study on federated learning, showing that our approach performs well in practical application.

2 PRELIMINARIES

2.1 Local Differential Privacy

LDP stems from Differential Privacy (DP) [7], a widely recognized standard for preserving privacy in data publishing and analysis. In DP models, a data curator releases a statistical information and analysis, perturbed by randomized algorithm, preventing adversaries from inferring individuals' information. However, this model has a vulnerability: the data curator initially has access to raw data, posing a risk if the data publisher is untrustworthy, or adversaries have access to raw data. LDP addresses this risk by having each client perturb their data before sending them to the aggregator. The formal definition of LDP is as follows:

Definition 1. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \Omega$ satisfies ϵ-local differential privacy if and only if, for any two input values $x, x' \in \mathcal{X}$ and an output $y \in \Omega$, the following inequality holds:

$$Pr[\mathcal{M}(x) = y] \leq e^{\epsilon} Pr[\mathcal{M}(x') = y]$$

2.2 LDP mechanisms for mean estimation

Notation. Let the input domain be $\mathcal{X} = [-1, 1]$ and the output space be Ω. Let $x \in \mathcal{X}$ be the true value of a client, $y \in \Omega$ be the be the perturbed value by an LDP mechanism. The users transmit y to the aggregator. Let X and Y be the random variables representing the true value and perturbed value, respectively. The probability of reporting y given that the true value is x is denoted as $\Pr[Y =$

$y|X = x]$, which we simplify as $\Pr[y|x]$. The expected value and variance of the perturbed value Y given the true value x are denoted as $\mathbb{E}[Y|x]$ and $\mathrm{Var}[Y|x]$, respectively.

LDP mean estimation. The aggregator collects a single numerical attribute and estimates unknown $\mathbb{E}(X)$ from the perturbed values. However, as the distribution is unknown, LDP mechanisms are designed to minimize the worst-case noise variance under ϵ-LDP. In this context, the mechanism's performance varies according to ϵ. Generally, mechanisms with a larger $|\Omega|$ tend to have better worst-case noise variance as ϵ increases.

The Duchi's mechanism randomly produces one of the two possible outputs ($|\Omega| = 2$). The output values are constants, which are determined by the privacy budget ϵ. The probabilities of reporting these output values depend on the true value x and are defined as follows:

$$Pr[y|x] = \begin{cases} \frac{e^\epsilon - 1}{2e^\epsilon + 2}x + \frac{1}{2}, & \text{if } y = \frac{e^\epsilon + 1}{e^\epsilon - 1}, \\ -\frac{e^\epsilon - 1}{2e^\epsilon + 2}x + \frac{1}{2}, & \text{if } y = -\frac{e^\epsilon + 1}{e^\epsilon - 1}. \end{cases}$$

Similarly, the three-output mechanism [25] randomly produces one of three possible outputs ($|\Omega| = 3$).

On the other hand, PMs randomly produce a real number with $\Omega = [L(-1), R(1)]$ which is defined as (1). The initial version of PM is proposed by [20], and later, Zhao et al. [25] proposed both optimal and sub-optimal variants. The probability density function is given by:

$$f[y|x] = \begin{cases} e^\epsilon q & \text{if } y \in [L(x), R(x)], \\ q & \text{if } y \in [L(-1), L(x)) \cup (R(x), R(1)] \end{cases} \tag{1}$$

where $q, L(x)$, and $R(x)$ are defined as follows:

$$\begin{aligned} q &= \frac{\zeta(e^\epsilon - 1)}{2(e^\epsilon + \zeta)^2} \\ L(x) &= \frac{(e^\epsilon + \zeta)(x\zeta - 1)}{\zeta(e^\epsilon - 1)} \\ R(x) &= \frac{(e^\epsilon + \zeta)(x\zeta + 1)}{\zeta(e^\epsilon - 1)} \end{aligned} \tag{2}$$

The parameter ζ shapes the characteristics of PMs. Specifically, when $\zeta = e^{\epsilon/2}$, it becomes the original PM. For $\zeta = e^{\epsilon/3}$, it becomes the sub-optimal variant, PM-sub. For PM-opt, since it involves complex formulation for ζ, refer to [25].

In contrast, the AAA mechanism [23] focuses on minimizing the average noise variance. Furthermore, while other mechanisms assume that data distribution is unknown, this mechanism is distribution-aware. The AAA mechanism first estimates the underlying distribution of the data to be collected using randomized response techniques [10]. Based on the estimated distribution, it then constructs an LDP mechanism that is optimized for this distribution. The number and values of the possible outputs are determined by specific hyperparameters.

Multidimensional data. While we assume one-dimensional data in this paper to clearly explain the mechanisms, LDP mechanisms can also be applied to multidimensional data. When collecting multidimensional data, these mechanisms typically employ a sampling strategy. In this approach, a subset of k dimensions is randomly sampled for perturbation. k is defined as [20, 25]:

$$k = \max\{1, \min\{dim, \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\}\} \tag{3}$$

Each selected dimension is perturbed using a privacy budget of ϵ/k, and the perturbed data is then transmitted to the aggregator.

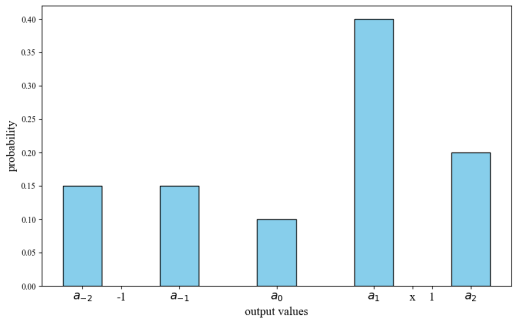


Figure 1: Probability distribution $\Pr[Y|x]$ for $N = 5$.

3 N-OUTPUT MECHANISM

In this section, we introduce the N-output mechanism for privately reporting users' numerical data to an aggregator. The underlying intuition behind the N-output mechanism stems from the observation that mechanisms with larger $|\Omega|$ tend to perform better in larger ϵ. While there are existing mechanisms for $|\Omega| = 2, 3, \infty$, there is a notable gap in the literature regarding mechanisms for arbitrary finite output spaces.

We begin by presenting the abstract design of the N-output mechanism in Section 3.1. Next, we formulate the objective function to optimize the N-output mechanism and outline the challenges to solve the optimization problem in Section 3.2. We then simplify the problem to make it more tractable in Section 3.3. Following this, we provide the optimization of the N-output mechanism in Section 3.4. Finally, we offer a comparative and theoretical analyses of the N-output mechanism in Section 3.5

3.1 Mechanism design

The N-output mechanism involves three steps: setup, randomization, and aggregation of data. The summarized process is presented in Algorithm 1.

Setup. The domain of input data is defined as $\mathcal{X} = [-1, 1]$ and the output space is defined as Ω with $|\Omega| = N$.

(Line 1) The output space for the N-output mechanism is defined as:

$$\Omega = \begin{cases} \{a_{-n}, \ldots, a_{-1}, a_1, \ldots, a_n\} & \text{if } N = 2n \\ \{a_{-n}, \ldots, a_{-1}, a_0, a_1, \ldots, a_n\} & \text{if } N = 2n + 1 \end{cases}$$

where $a_{-i} = -a_i$, $a_0 = 0$, and the sequence satisfies $a_i < a_{i+1}$. We discuss the optimal setting of N and Ω in Section 3.4.

Randomization. (Lines 2-4) The N-output mechanism takes an input $x \in \mathcal{X}$ and yields a perturbed value $y \in \Omega$ by randomly selecting an element from Ω. The probability of selecting each a_i depends on the value of x. The mechanism is designed such that x

is more likely to be perturbed to a value close to it, as illustrated in Figure 1. The probability of perturbing x as a_i is denoted as $\Pr[a_i|x]$. Then, the perturbed output y is determined as:

$$y = a_i \text{ with } \Pr[a_i|x], \quad i \in \{-n, \ldots, -1, 0, 1, \ldots, n\}$$

Aggregation. The aggregator receives the perturbed values from all users. Since the expected values of perturbed values are equal to the expected values of true values, the aggregator can estimate the population mean by simply averaging the perturbed data. This estimation leverages the unbiased nature of the mechanism.

Algorithm 1 N-output Mechanism

Input: privacy budget ϵ, true value $x \in [-1, 1]$
Output: perturbed value $y \in \Omega$.
Set the output space
$\Omega \leftarrow \{a_{-n}, \ldots, a_n\}$
Randomization
for $i \in \{-n, \ldots, n\}$ **do**
  $Pr[a_i|x] \leftarrow Calculate\_probability(\Omega, x, i)$
$y \leftarrow$ sample a value from Ω according to $Pr[a_i|x]$
return y

3.2 Objective formulation

The concept behind the N-output mechanism is straightforward, but its configuration—specifically, the determination of Ω and $\Pr[Y|x]$—is crucial for ensuring both the correctness and accuracy of mean estimation.

Correctness requires that the expected value of the perturbed outputs is an unbiased estimator of the original input, i.e., $\mathbb{E}[Y|x] = x$. Additionally, the probability distribution of Y must be valid, meaning $\Sigma_{a_i \in \Omega} \Pr[a_i|x] = 1$ and $\Pr[a_i|x] \geq 0$. Furthermore, the N-output mechanism must guarantee ϵ-LDP, i.e., $\Pr[y|x] \leq e^{\epsilon} \Pr[y|x']$ for different two any values $x, x' \in \mathcal{X}$.

Accuracy is measured by the worst-case noise variance in the mean estimation. Given Ω and Y, we denote the noise variance as $\mathrm{Var}[Y|x; \Omega]$, which is dependent on x. The noise variance is defined as:

$$\mathrm{Var}[Y \mid x; \Omega] = \sum_{i=-n}^{n} a_i^2 \Pr[a_i|x] - x^2$$

Our objective is to find Ω and Y that minimize the worst-case noise variance over all possible values of x. Thus, we formulate the objective function as follows:

$$\begin{aligned} \text{Minimize} \quad & \max_{x \in [-1,1]} \mathrm{Var}[Y \mid x; \Omega] \\ \text{subject to} \quad & \Pr[y|x] \leq e^{\epsilon} \Pr[y|x'] \\ & \mathbb{E}[Y|x] = x \\ & \Sigma_{a_i \in \Omega} \Pr[a_i|x] = 1 \\ & \Pr[a_i|x] \geq 0 \end{aligned} \tag{4}$$

The noise variance can be interpreted as the expected squared distance between x and Y. Therefore, when the probability distribution of the outputs is concentrated around x, the variance will be lower. Since the probabilities $\Pr[a_i|x]$ are constrained by the LDP condition, we define x_j such that:

$$\Pr[a_j|x_j] = \begin{cases} e^{\epsilon} \Pr[a_j| - 1] & \text{if } a_j \geq 0 \\ e^{\epsilon} \Pr[a_j|1] & \text{if } a_j < 0 \end{cases}$$

As done in previous work [6, 25], $\Pr[a_i|x]$ can be expressed as a piecewise linear function over each interval $x_{j-1} \leq x \leq x_j$. Let $p_{i,j}(x)$ denote $\Pr[a_i|x]$ for $x_{j-1} \leq x \leq x_j$. The function $p_{i,j}(x)$ can then be defined as:

$$\begin{aligned} p_{i,j}(x) &= \frac{p_{i,j}(x_j) - p_{i,j}(x_{j-1})}{x_j - x_{j-1}}(x - x_j) + p_{i,j}(x_j) \qquad (5) \\ &= \frac{p_{i,j}(x_j) - p_{i,j}(x_{j-1})}{\sum_k \left(p_{k,j}(x_j) - p_{k,j}(x_{j-1})\right) a_k} \left(x - \sum_i p_{i,j}(x_j) a_i\right) + p_{i,j}(x_j) \end{aligned}$$

where $p_{i,j}(x_j)$ for $i = -n, \ldots, n$ and $j = -n, \ldots, n$ and a_i for $i = -n, \ldots, n$ are variables that need to be determined to minimize the worst-case noise variance. As $p_{i,j}(x)$ is defined as piecewise linear function, the noise variance $\mathrm{Var}[Y|x]$ is defined as piecewise quadratic function according to the value of x:

$$\mathrm{Var}_j[Y \mid x; \Omega] = \sum_{i=-n}^{n} a_i^2 p_{i,j}(x) - x^2 \quad \text{if } x_{j-1} \leq x \leq x_j \qquad (6)$$

Challenges. When optimizing the worst-case noise variance, we have $n(N-2)$ free variables for $N = 2n$ and $n(N-1) - 1$ free variables for $N = 2n + 1$ to determine $p_{i,j}(x_j)$. Additionally, there are $n - 1$ free variables associated with the output values a_i. For example, Duchi's mechanism ($N = 2$) [6] has no free variables, and the three-output mechanism ($N = 3$) [25] has only one free variable. In contrast, the N-output mechanism has $O(N^2)$ free variables that need to be optimized, making the problem more complex.

Moreover, there is the inherent interdependence between $p_{i,j}(x)$ and a_i, which affects the overall distribution of Y and the noise variance. This interdependence implies that optimizing one set of variables without considering its impact on the others is infeasible, leading to nontrivial.

Additionally, the noise variance comprises $N - 1$ quadratic functions with respect to x. The objective is not merely to minimize the noise variance at a single point but to ensure that the maximum possible variance across the entire interval is minimized. This requirement introduces an additional layer of difficulty to the optimization process.

3.3 Our probability setting

As discussed in the previous section, finding a theoretical solution to the problem is challenging. While numerical optimization methods can be applied, they may not guarantee a global solution, as the problem appears to be non-convex. However, obtaining a theoretical solution would offer significant advantages. A theoretical solution provides stable and predictable performance, enables rigorous analysis, and supports the development of hybrid mechanisms that integrate existing approaches. Therefore, we simplify the problem to make it more tractable, aiming to find a theoretical solution that can offer these benefits.

To obtain a theoretical solution, we simplify the problem by reducing the number of free variables to $O(N)$ and the interdependence between $p_{i,j}(x)$ and a_i. To achieve this, we establish the following probability setting:

First, for $i = 0$, we define:

$$p_{i,j}(x_j) = \begin{cases} e^\epsilon p_0 & \text{if } i = j \\ p_0 & \text{otherwise} \end{cases} \tag{7}$$

where p_0 is a free variable to be determined.

Next, for $|i| > 1$, we set:

$$p_{i,j}(x_j) = \begin{cases} e^\epsilon p & \text{if } i = j \\ p & \text{otherwise} \end{cases} \tag{8}$$

where p is a dependent variable determined by p_0.

Finally, for $|i| = 1$, we define:

$$p_{i,j}(x_j) = \begin{cases} e^\epsilon p & \text{if } i = j \\ p^* & \text{if } j = 0 \\ p & \text{otherwise} \end{cases} \tag{9}$$

where p^* is a dependent variable determined by p_0.

Substituting these settings into (5), for $j = 0, 1$, the probability function $p_{i,j}(x)$ is defined over the interval $x_{j-1} \le x \le x_j$ as follows:

$$p_{i,j}(x) = \begin{cases} \frac{j(p-e^\epsilon p)+e^\epsilon p-p^*}{x_1}|x| + p^* & \text{if } i = -1 \\ \frac{(1-e^\epsilon)p_0}{x_1}|x| + e^\epsilon p_0 & \text{if } i = 0 \\ \frac{j(e^\epsilon p-p)+p-p^*}{x_1}|x| + p^* & \text{if } i = 1 \\ p & \text{otherwise} \end{cases} \tag{10}$$

Otherwise, $p_{i,j}(x)$ is defined on $x_{j-1} \le x \le x_j$ as follows:

$$p_{i,j}(x) = \begin{cases} \frac{x-ta_{j-1}}{a_j-a_{j-1}} + p & \text{if } i = j \\ \frac{ta_j-x}{a_j-a_{j-1}} + p & \text{if } i = j-1 \\ p & \text{otherwise} \end{cases} \tag{11}$$

where $t = (e^\epsilon - 1)p$ and $x_j = ta_j$. Here, $\mathbb{E}[Y|x] = \sum_i p_{i,j}(x)a_i = x$ for every j.

Given the sum of probability condition, $\sum_i p_{i,n}(1) = 1$, the variable p can be defined as follows:

$$p = \frac{1 - p_0}{e^\epsilon + 2n - 1}$$

Similarly, using the sum of probabilities condition $\sum_i p_{i,1}(0) = 1$, the variable p^* is defined as:

$$p^* = \frac{1 - 2(n-1)p - e^\epsilon p_0}{2}$$

When $N = 2n$, p_0 naturally becomes 0. For cases where N is odd, optimizing the value of p_0 becomes crucial to achieve the best accuracy.

Furthermore, by constraining $0 \le p_0 \le p$, the N-output mechanism guarantees ϵ-LDP.

Theorem 1. *The N-output mechanism is ϵ-LDP.*

Proof. For $|i| > 1$, every probability $Pr[a_i|x]$ falls within the range $[p, e^\epsilon p]$. For any two input values x, x', we have:

$$max_{x,x'} \frac{Pr[a_i|x]}{Pr[a_i|x']} = \frac{e^\epsilon p}{p} = e^\epsilon$$

Similarly, for $i = 0$, we have:

$$max_{x,x'} \frac{\Pr[0|x]}{\Pr[0|x']} = \frac{e^\epsilon p_0}{p_0} = e^\epsilon$$

For $|i| = 1$, we demonstrate that $p \le p^* \le e^\epsilon p$ to prove the theorem. Since p^* decreases as p_0 increases, and given $0 \le p_0 \le p$, we need to show two cases:

(1) For $p_0 = 0$, we have $p = \frac{1}{e^\epsilon+2n-1}$. Thus, the inequality $p^* \le e^\epsilon p$ is equivalent to:

$$\begin{aligned} & \frac{1-2(n-1)p}{2} \le e^\epsilon p \\ \iff & \frac{1}{2} - \frac{n-1}{e^\epsilon+2n-1} \le \frac{e^\epsilon}{e^\epsilon+2n-1} \\ \iff & 1 \le e^\epsilon \end{aligned}$$

Since $\epsilon > 0$, the inequality $p^* \le e^\epsilon p$ holds.

(2) For $p_0 = p$, we have $p^* = p$. Thus, the inequality $p \le p^*$ holds trivially.

□

3.4 Optimizing N-output mechanism

By simplifying the optimization problem, our goal is to optimize the N-output mechanism by finding the optimal p_0 and Ω that minimize the worst-case noise variance. Specifically, we aim to solve the following optimization problem:

$$\min_{\Omega, p_0} \max_{x \in [-1,1]} \mathrm{Var}[Y \mid x; \Omega]$$

For $N = 2$, the parameters p_0 and Ω are already determined, meaning they are fixed and not subject to further optimization. The case for $N = 3$ requires the optimization of only p_0, and this has been previously studied in [25]. Thus, we focus on solving the problem for $N \ge 4$.

To achieve this, we first analyze the structure of the noise variance function $\mathrm{Var}[Y|x; \Omega]$ and identify the key factors that influence the optimization. For simplicity, we will omit parameters like Ω when the context is clear. Additionally, since the N-output mechanism is symmetric at $x = 0$ (i.e., $\mathrm{Var}[Y|x; \Omega] = \mathrm{Var}[Y| - x; \Omega]$), it is sufficient to consider the case where $0 \le x \le 1$.

Let $\mathrm{Var}_i[Y|x]$ be the noise variance over the interval $x_{i-1} \le x \le x_i$ for $i \in \{1, \ldots, n\}$. The expression for $\mathrm{Var}_i[Y|x]$ is given by:

$$\mathrm{Var}_i[Y \mid x] = \begin{cases} -x^2 + \frac{a_1(e^\epsilon p+p-2p^*)}{(e^\epsilon-1)p}x + 2a_1^2 p^* + 2\sum_{j=2}^n a_j^2 p & \text{if } i = 1 \\ -x^2 + (a_{i-1}+a_i)x - (e^\epsilon-1)pa_{i-1}a_i + 2\Sigma_{j=1}^n a_j^2 p & \text{if } i \ge 2 \end{cases}$$

The following lemmas provide conditions necessary for minimizing the worst-case noise variance.

Lemma 2. *For $i \ge 2$, let $x_i^* = \frac{a_{i-1}+a_i}{2}$. Given ϵ and N, $Var_i[Y|x]$ must attain its maximum value at $x = x_i^*$ to minimize the worst-case noise variance.*

Proof. The proof is in the full version of this paper [cite] □

Lemma 3. *To minimize the worst-case noise variance, $Var_n[Y|x_n^*]$ must be the worst-case noise variance.*

Proof. The proof is in the full version of this paper [cite] □

With Lemma 2, Lemma 3, the following theorem identifies Ω that minimizes the worst-case noise variance.

Theorem 4. *Given ϵ, N, and p_0, the sequence a_i that minimizes the worst-case noise variance is defined as follows. Define the sequences P_i and Q_i as follows:*

$$P_n = 0, \quad P_{n-1} = 1$$
$$P_i = (4t-2)P_{i+1} - P_{i+2} \quad \text{for } i \in \{1, 2, \ldots, n-2\}$$

$$Q_n = 1, \quad Q_{n-1} = 0$$
$$Q_i = (4t-2)Q_{i+1} - Q_{i+2} \quad \text{for } i \in \{1, 2, \ldots, n-2\}$$

Then, the sequence a_i is given by:

$$a_n = \frac{1}{t}$$
$$a_{n-1} = \frac{(2t-1) - 8p\sum_{i=1}^{n} P_i Q_i}{1 + 8p\sum_{i=1}^{n} P_i^2} a_n$$
$$a_i = (4t-2)a_{i+1} - a_{i+2} \quad \text{for } i \in \{1, 2, \ldots, n-2\}$$

This holds only if the following conditions are satisfied:

$$a_i < a_{i+1} \quad \text{for } i \in \{0, 1, \ldots, n-1\}$$
$$Var_n[Y \mid x_n^*; p_0] \geq Var_1[Y \mid x_1^*; p_0]$$

where $x_1^* = \frac{a_1(e^\epsilon p + p - 2p^*)}{2(e^\epsilon - 1)p}$

Proof. The proof is in the full version of this paper [cite] □

For even values of N, it follows that $p_0 = 0$. Under these circumstances, Ω derived by Theorem 4 ensures that the worst-case noise variance is minimized, provided the conditions specified in the theorem are satisfied. These conditions are related to the value of N and ϵ. If N is too large relative to ϵ, the condition $a_i < a_{i+1}$ fails to hold, requiring a reduction in N. Conversely, when N is too small, we encounter a different issue: $\mathrm{Var}_1[Y|x_1^*]$ exceeds $\mathrm{Var}_n[Y|x_n^*]$, thereby violating Lemma 3. In this case, the configuration described in Theorem 5 should be employed to achieve the desired minimization.

For odd values of N, an additional optimization of p_0 is required. Decreasing p_0 has the effect of lowering $\mathrm{Var}_n[Y|x_n^*]$ while simultaneously increasing $\mathrm{Var}_1[Yx_1^*]$. Since p_0 must satisfy $0 \leq p_0 \leq p$, three scenarios can be considered:

- If $\mathrm{Var}_n[Y|x_n^*; p_0 = p] \geq \mathrm{Var}_1[Y|x_1^*; p_0 = p]$, then the value of p_0 that minimizes the worst-case noise variance is given by:

$$p_0 = \underset{p_0 \in [0,p]}{\operatorname{argmin}} \left| \mathrm{Var}_n[Y \mid x_n^*; p_0] - \mathrm{Var}_1[Y \mid x_1^*; p_0] \right| \tag{12}$$

- If $\mathrm{Var}_n[Y|x_n^*; p_0 = 0] > \mathrm{Var}_1[Y|x_1^*; p_0 = 0]$, this indicates that N is too large relative to the current ϵ.
- If $\mathrm{Var}_n[Y|x_n^*; p_0 = p] < \mathrm{Var}_1[Y|x_1^*; p_0 = p]$, then Theorem 5 offers the appropriate setting to minimize the worst-case noise variance.

Here, $p_0 = 0$ when $N = 2n$ and $p_0 = p$ when $N = 2n+1$, where p is given by $p = \frac{1}{e^\epsilon + N - 1}$.

Theorem 5. *Given $p = \frac{1}{e^\epsilon + N - 1}$, if Ω derived by Theorem 4 satisfies $a_i < a_{i+1}$ and $Var_1[Y|x_1^*; p] > Var_n[Y|x_n^*; p]$, then the worst-case noise variance is minimized by the following sequences. Define the sequence C_i as:*

$$C_1 = \begin{cases} \frac{1}{4t-1} & \text{if } N = 2n \\ \frac{1}{4t-2} & \text{if } N = 2n+1 \end{cases}$$

$$C_{i+1} = \frac{1 - 2t + \sqrt{C_i^2 + 2C_i - 4tC_i + (2t-1)^2}}{C_i^2 + 2C_i - 4tC_i}$$

Then, the sequence a_i is defined as:

$$a_n = \frac{1}{t}$$
$$a_i = C_i a_{i+1} \quad \text{for } i \in \{1, 2, \ldots, n-1\}$$

Proof. The proof is in the full version of this paper [cite] □

Optimization process In Algorithm 2, we provide an optimization process designed to find an optimal N-output mechanism. The algorithm specifically addresses cases where $N \geq 4$. We omit the cases for $N = 2$ and $N = 3$ from the algorithm because they can be handled separately.

- For $N = 2$, the optimal solution is easily derived with $p = \frac{1}{e^\epsilon + 1}$.
- For $N = 3$, the process involves optimizing p_0 by solving the following equation:

$$p_0 = \underset{p_0 \in [0,p]}{\operatorname{argmin}} \mathrm{Var}[Y|x_1^*]$$

These cases are simple and do not require iterative optimization. The main focus of the algorithm is on cases where $N \geq 4$, which are more complex and require a systematic approach to finding the optimal parameters.

(Lines 1-21) The algorithm iteratively adjusts N, starting from $N = 4$, to find the optimal set of parameters Ω and p that minimize the worst-case noise variance. For convenience, the algorithm is designed to determine p instead of p_0. (Line 2) An empty list $\mathcal{S}$ is initialized to store tuples of the form $(\Omega, p, max\mathrm{Var})$. (Lines 4-5) For each iteration, use Theorem 4 with $p = \frac{1}{e^\epsilon + N - 1}$ to obtain initial configuration Ω_0. (Lines 6-7) If the condition $a_i < a_{i+1}$ does not hold for every i, it indicates that N is too large, so the iteration stops. (Line 20) The optimal configuration is then found by selecting the tuple with the smallest $max\mathrm{Var}$ from $\mathcal{S}$. (Lines 8-14) If the condition of Line 8 holds, Theorem 4 provides the optimal configuration for the given N. (Lines 15-18) If the condition does not hold, Theorem 5 is used to provide the optimal configuration for the given N.

3.5 Comparative and theoretical Analyses

Comparison to existing methods. As the N-output mechanism extends and generalizes both the Duchi's mechanism [6] and the three-output mechanism [25], we compare our mechanism with PM-sub. The blue line of Figure 2 illustrates the percentage deviation in worst-case noise variance between PM-sub and the N-output mechanism, calculated as $(\frac{PM-sub}{N-output} - 1) \times 100(\%)$, while the red horizontal dashed line at 0% serves as a reference line. This metric quantifies the improvement in worst-case noise variance achieved

Algorithm 2 Optimization algorithm for minimizing the worst-case noise variance

Input: ϵ
Output: Ω, p

$N \leftarrow 4$
$\mathcal{S} \leftarrow []$
while True **do**
  $p = \frac{1}{e^{\epsilon}+N-1}$
  Let Ω_0 be the result from Theorem 4 with p
  if there exists an i such that $a_i >= a_{i+1}$ **then**
    break
  else if $Var_n[Y|x_n^*; \Omega_0, p] < Var_1[Y|x_1^*; \Omega_0, p]$ **then**
    if $N = 2n$ **then**
      $\mathcal{S}$.append($(\Omega_0, p, Var_n[Y|x_n^*; \Omega_0, p])$)
    else
      Let Ω_1, p_0 be the result from (12) and Theorem 4.
      $p \leftarrow \frac{1-p_0}{e^{\epsilon}+2n-1}$
      $\mathcal{S}$.append($(\Omega_1, p, Var_n[Y|x_n^*; \Omega_1, p])$)
  else
    $p \leftarrow \frac{1}{e^{\epsilon}+N-1}$
    Let Ω_2 be the result from Theorem 5 with p
    $\mathcal{S}$.append($(\Omega_2, p, Var_n[Y|x_n^*; \Omega_1, p])$)
  $N \leftarrow N + 1$
Find Ω, p that minimize maxVar from $\mathcal{S}$
return Ω, p

by the N-output mechanism over PM-sub. As shown in the Figure 2, the N-output mechanism demonstrates superior performance in $0 < \epsilon < 3.5$ and $3.7 < \epsilon < 4.15$. Outside of these ranges, PM-sub exhibits slightly better performance, but the difference is marginal, with a maximum deviation of only -4% for values of ϵ up to 8.

Additionally, the N-output mechanism is also communication-efficient. It requires only $\log_2 N$ bits to encode each report. Table 2 details the required number of bits according to ϵ. In comparison, PM-sub typically requires 32 or 64 bits to represent floating-point numbers.

Table 2: Required bits according to ϵ

	$\epsilon < 0.69$	$\epsilon < 2.54$	$\epsilon < 5.41$	$\epsilon < 7.8$
bits	1	2	3	4
	$\epsilon < 10.0$	$\epsilon < 12.1$	$\epsilon < 14.26$	$\epsilon < 16.35$
bits	5	6	7	8

Theoretical analysis. The accuracy guarantee is the sames as that of Duchi's [6] or three-output mechanism[25], since the N-output mechanism generalizes them. Therefore, we omit the detailed accuracy guarantee.

The following lemma addresses the lower bound of worst-case noise variance.

Lemma 6. *The worst-case noise variance of the N-output mechanism:*

$$\inf \max_{x \in [-1,1]} Var[Y|x; N, \Omega] = \frac{1}{(N-1)^2}$$

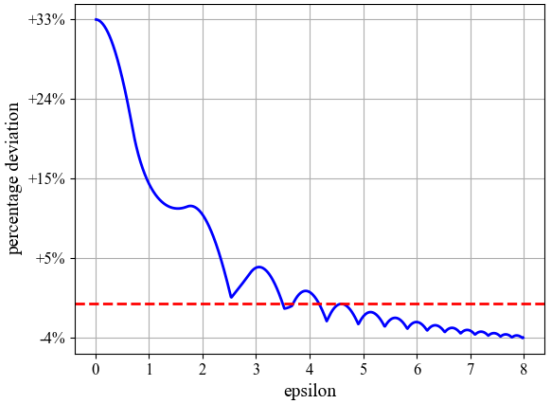


Figure 2: Percentage deviation of worst-case noise variance between PM-sub and the N-output mechanism.

Proof. The proof is in the full version of this paper [cite] □

Even as $\epsilon \to \infty$, the worst-case noise variance of Duchi's mechanism and three-output mechanism are bounded at 1 and 1/4, respectively. In other words, even if the value of ϵ increases, the accuracy of these mechanisms does not significantly improve due to the limitations of output possibilities. In contrast, the worst-case noise variance of the N-output mechanism approaches 0 as $\epsilon \to \infty$ because N also approaches infinity. Therefore, the N-output mechanism maintains effectiveness even at high ϵ.

4 EXTENDING N-OUTPUT MECHANISM

Building on the N-output mechanism, we propose HM-NP to further reduce the worst-case noise variance. Next, we propose 2PEM to enable distribution estimation from the result of HM-NP. Finally, we integrate HM-NP and 2PEM into the AAA mechanism [23] to improve the accuracy of mean estimation.

4.1 HM-NP

In HM-NP, a user randomly selects either the N-output mechanism or PM-sub to report their data, with the selection probability denoted by $\alpha \in [0, 1]$. Here α represents the probability of selecting the N-output mechanism.

The value of α plays a crucial role in determining the worst-case noise variance. Therefore, it is essential to find the optimal value of α that minimizes this variance.

We denote the N-output mechanism, PM-sub, and HM-NP as $\mathcal{N}, \mathcal{P}$, and $\mathcal{H}$, respectively. Let the noise variance of PM-sub be given by:

$$\text{Var}_{\mathcal{P}}[Y|x] = Ax^2 + B$$

$$where \quad A = \frac{e^{\epsilon/3}+1}{e^{\epsilon}-1}, \quad B = \frac{(e^{\epsilon/3}+e^{\epsilon})((e^{\epsilon/3}+1)^3+e^{\epsilon}-1)}{3e^{2\epsilon/3}(e^{\epsilon}-1)^2}$$

Then, the noise variance of HM-NP is given by:

$$\text{Var}_{\mathcal{H}}[Y|x; \alpha] = \alpha \text{Var}_{\mathcal{N}}[Y|x] + (1-\alpha)\text{Var}_{\mathcal{P}}[Y|x]$$

Our goal is to find the value of α that minimizes the worst-case noise variance of $\mathrm{Var}_{\mathcal{H}}[Y|x;\alpha]$.

The N-output mechanism has its worst-case noise variance at $x = x_n^*$ and $\mathrm{Var}_{\mathcal{P}}[Y|x]$ increases monotonically with x. Therefore, the worst-case noise variance for HM-NP must occur in the range $x_n^* \le x \le 1$. Thus, we can formulate the problem as finding α that minimizes the maximum value of $V(x,\alpha)$ over $x_n^* \le x \le 1$, defined as:

$$\begin{aligned} V(x,\alpha) &= \alpha \mathrm{Var}_n[Y|x] + (1-\alpha)\mathrm{Var}_{\mathcal{P}}[Y|x] \\ &= (A - \alpha A - \alpha)x^2 + 2\alpha x_n^* x + \alpha b + (1-\alpha)B \end{aligned}$$

where $b = 2\Sigma_{i=1}^{n} a_i^2 p - a_{n-1}$.

Lemma 7 specifies the location of the worst-case noise variance, and Theorem 8 finds an optimal α that minimizes the worst-case noise variance.

Lemma 7. *Given ϵ, N, and Ω, the point of worst-case noise variance according to α is:*

$$\underset{x}{\mathrm{argmax}}\ Var_{\mathcal{H}}[Y|x;\alpha] = \begin{cases} 1 & if\ 0 \le \alpha \le \alpha_1 \\ x_{\mathcal{H}}^* & if\ \alpha_1 < \alpha \le 1 \end{cases}$$

$$where \quad \alpha_1 = \frac{e^{\epsilon/3}+1}{e^{\epsilon/3} + e^{\epsilon} - x_n^*(e^{\epsilon}-1)}, \quad x_{\mathcal{H}}^* = \frac{\alpha x_n^*}{\alpha + A\alpha - A}$$

Proof. The proof is in the full version of this paper [cite] □

Theorem 8. *Given ϵ, N, and Ω, define:*

$$D = -A - B + 2\Sigma_{i=1}^{n} a_i^2 p + a_n - 1$$

$$\gamma_1 = \frac{(x_n^*)^2 + (1+A)(b-B)}{(1+A)^2}, \quad \gamma_2 = \left(\frac{Ax_n^*}{1+A}\right)^2$$

$$\alpha_2 = \frac{(e^{\epsilon}-1)\sqrt{\frac{\gamma_2}{\gamma_1}} + e^{\epsilon/3} + 1}{e^{\epsilon} + e^{\epsilon/3}}$$

Then, the value of α that minimizes the worst-case noise variance is given as follows:

1. *If $D > 0$, then $\alpha = 0$.*
2. *If $D \le 0$, then:*
   (a) *If $\gamma_1 \le 0$, then $\alpha = 1$.*
   (b) *If $\gamma_1 > 0$:*

$$\alpha = \begin{cases} \alpha_2 & if\ \alpha_2 \le 1 \\ 1 & if\ \alpha_2 > 1 \end{cases}$$

Proof. The proof is in the full version of this paper [cite] □

Fixing $N = 3$, HM-NP becomes HM-TP [25], which is the hybrid mechanism combining the three-output mechanism with PM-sub. Setting $\alpha = 0$, HM-NP becomes equivalent to PM-sub. Therefore, HM-NP generalizes these mechanisms, and by selecting the optimal N and α as determined Theorem 8. HM-NP achieves the best worst-case noise variance among these mechanisms.

HM-NP is also more communication-efficient. Let F represent the number of bits required for floating-point representation. The average number of bits required by HM-NP is $\lceil \log_2 \alpha N \rceil + (1-\alpha)F$. As α decreases, the number of bits required increases due to the heavier reliance on floating-point representation.

However, for HM-TP, the value of α decreases consistently as ϵ increases, leading to higher bit requirements. Conversely, in HM-NP, the value of α increases with larger N, which reduces the communication overhead. For example, when $\epsilon = 4$ and $F = 32$, HM-TP requires 28.12 bits on average, whereas HM-NP requires 23 bits on average.

4.2 2PEM

Building on HM-NP, we propose 2PEM which can be applied to the results of HM-NP for distribution estimation. Consider an aggregator receiving m perturbed values from users, represented as $\mathbf{y} = \{y_1, \ldots, y_m\}$. Dividing the domain $\mathcal{X} = [-1, 1]$ into d bins, we denote the distribution of true values as $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_d\}$, which represents the relative frequency of true values in each bin. The goal of distribution estimation is accurately estimate the vector $\boldsymbol{\pi}$. Since 2PEM is a post-processing method, it does not require any additional privacy budget or data collection.

EM is a method that estimates unknown parameters based on observed data. Specifically, EM finds the distribution that maximizes the likelihood of observing the perturbed data, functioning as a Maximum Likelihood Estimation (MLE).

However, directly applying the EM approach proposed by [15] to HM-NP results in poor estimation accuracy due to the heterogeneous probability distribution arising from the two distinct mechanisms: one being continuous and the other discrete.

To overcome this challenge, 2PEM employs a two-phase approach that separately handles the data from PM-sub and the N-output mechanism. In the first phase, 2PEM uses EM on the PM-sub data to estimate the distribution via MLE. In the second phase, the estimated distribution from the first phase serves as prior knowledge for the EM process. This prior knowledge is then used in the EM algorithm applied to the N-output mechanism data, employing a Maximum A Posteriori (MAP) approach. Through this two-phase process, 2PEM effectively utilizes all available information, leading to more accurate distribution estimation.

First phase. The algorithm for 2PEM is detailed in Algorithm 3. In this process, the input domain and output space of PM-sub are divided into d bins and $\tilde{d}$ bins, respectively. Let $R_i = [r_{i-1}, r_i]$ denote the i-th bin of the divided input domain and $\tilde{R}_i = [\tilde{r}_{i-1}, \tilde{r}_i]$ denote the i-th bin of the divided output space. Let c_i' represent the number of outputs falling into $\tilde{R}_i$. The estimated distribution in the first phase is denoted by $\mu = \{\mu_1, \ldots, \mu_d\}$.

The objective is to find μ that maximizes the log-likelihood $\mathcal{L}(\mu) = \ln Pr[\mathbf{y}|\mu]$. We apply EM to the result of PM-sub to estimate μ. Following the EM process outlined [15], we define Q_i as:

$$Q_i = \mu_i \sum_{j=1}^{\tilde{d}} c_j' \frac{\Pr[y \in \tilde{R}_j | x \in R_i, \mu]}{\Pr[y \in \tilde{R}_j | \mu]}$$

where $\Theta_{j,i}^{\mathcal{P}} = \Pr[y \in \tilde{R}_j | x \in R_i, \mu]$. The distribution μ is then updated iteratively as:

$$\mu_i = \frac{Q_i}{\Sigma_{k=1}^{d} Q_k}$$

which maximizes the log-likelihood. The iteration stops when the relative improvement in the log-likelihood becomes sufficiently small, as suggested by [9, 15].

Second phase. In this phase, EM is applied to the results from the N-output mechanism, using the distribution μ estimated in the first phase as prior knowledge. Since this phase follows a MAP estimation, the log-likelihood we aim to maximize is given by:

$$\mathcal{L}(\hat{\pi}) = \ln \Pr[\hat{\pi}|\mathbf{y}]$$

Maximizing this function is equivalent to maximizing:

$$\ln \Pr[\mathbf{y}|\hat{\pi}] + \ln \Pr[\hat{\pi}]$$

where $\ln \Pr[\hat{\pi}] = -\frac{1}{2\sigma^2} \sum_{i=1}^{d} (\hat{\pi}_i - \mu_i)^2$, assuming Gaussian prior distribution. We define Q_i for the second phase as:

$$Q_i = \hat{\pi}_i \sum_{j=-n}^{n} c_j \frac{\Pr[y = a_j | x \in R_i, \mu]}{\Pr[y = a_j | \mu]}$$

where $\Theta^{\mathcal{N}}_{j,i} = \Pr[ya_j | x \in R_i, \mu]$ and c_j is the number of outputs a_j. The distribution $\hat{\pi}$ is then updated iteratively as:

$$\hat{\pi}_i = \frac{Q_i + \lambda \mu_i}{\Sigma_{k=1}^{d} Q_k + \lambda}$$

which maximizes the log-likelihood. Here, $\lambda = \frac{1}{\sigma^2}$ controls the influence of the prior knowledge on the second phase. If $\lambda = 0$, the second phase becomes to a simple MLE approach that does not incorporate the prior from the first phase.

Algorithm 3 2PEM

Input: $y, \Theta^{\mathcal{P}}, \Theta^{\mathcal{N}}, \tau, \lambda$
Output: $\hat{\pi}$

while $|\mathcal{L}_{t+1}(\mu) - \mathcal{L}_t(\mu)| > \tau$ **do**
  for $i \in \{1, \ldots, d\}$ **do**
    $Q_i = \mu_i \Sigma_{j=1}^{\tilde{d}} c'_j \frac{\Theta^{\mathcal{P}}_{j,i}}{\Sigma_{k=1}^{d} \Theta^{\mathcal{P}}_{j,k} \mu_k}$
  for $i \in \{1, \ldots, d\}$ **do**
    $\mu_i = \frac{Q_i}{\Sigma_{k=1}^{d} Q_k}$
while $|\mathcal{L}_{t+1}(\hat{\pi}) - \mathcal{L}_t(\hat{\pi})| > \tau$ **do**
  for $i \in \{1, \ldots, d\}$ **do**
    $Q_i = \hat{\pi}_i \Sigma_{j=-n}^{n} c_j \frac{\Theta^{\mathcal{N}}_{j,i}}{\Sigma_{k=1}^{d} \Theta^{\mathcal{N}}_{j,k} \hat{\pi}_k}$
  for $i \in \{1, \ldots, d\}$ **do**
    $\hat{\pi}_i = \frac{Q_i + \lambda \mu_i}{\Sigma_{k=1}^{d} Q_k + \lambda}$
return $\hat{\pi}$

Due to the page limitation, the derivation of $\Theta^{\mathcal{P}}_{i,j}$ and $\Theta^{\mathcal{N}}_{i,j}$ is provided in the full version of this paper [cite].

Hyperparameters. The selection of d, τ, and λ is discussed in Section 5. For $\tilde{d}$, we propose the following choice:

$$\tilde{l} = \frac{2(e^{\epsilon} + e^{\epsilon/3})}{d(e^{\epsilon} - 1)}$$
$$\tilde{d} = \left\lceil \frac{R(1) - L(1)}{\tilde{l}} \right\rceil \tag{13}$$

The intuition behind this choice is that the distribution of true values is well reflected in the distribution of noisy values.

The length of each bin R_i is $l = \frac{2}{d}$, and the length of each bin $\tilde{R}_j$ is $\tilde{l} = \frac{R(1) - L(1)}{\tilde{d}}$. To match the intervals of noisy values with the intervals of true values, we define the $\tilde{R}_j$ corresponding to R_i as $[L(r_{i-1}), L(r_i)]$. We then set $\tilde{l} = L(r_i) - L(r_{i-1})$. This approach helps to choose a value that consistently works well across different values of d.

4.3 NP-AAA

In this section, we present the method for combining HM-NP with the AAA mechanism. Unlike other mean estimation mechanisms that assume underlying data distribution is unknown, the AAA mechanism is distribution-aware and tailors its approach based on the estimated distribution. As a result, this mechanism requires an accurate distribution estimation in the first phase. In the subsequent phase, the AAA mechanism constructs an optimized report protocol tailored to the estimated distribution. Therefore, the performance of mean estimation heavily depends on the quality of the distribution estimation.

In the original approach, the AAA mechanism uses the generalized randomized response [10] to estimate the distribution. However, this approach is not focused on the numerical distribution estimation, and the data used for distribution estimation cannot be reused for mean estimation.

By contrast, our approach —HM-NP and 2PEM —is focused on numerical distribution estimation. Moreover, the outputs of HM-NP used for distribution estimation can also be reused for mean estimation, expressed as:

$$\mathbb{E}[X] = \mathbb{E}[Y_{\mathcal{N}} + Y_{\mathcal{A}}]$$

where $Y_{\mathcal{N}}$ and $Y_{\mathcal{A}}$ represent the outputs from HM-NP and the AAA mechanism, respectively.

5 PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed methods: the N-output mechanism, HM-NP, 2PEM, and NP-AAA. Our evaluation is conducted in three parts.

First, we evaluate the accuracy of mean estimation using our mechanisms comparing them against several benchmarks: Duchi's mechanism [6], the three-output mechanism [25], the AAA mechanism [23], HM-TP [25], and PM-sub [25].

Next, we evaluate the effectiveness of our mechanisms in distribution estimation. For this we benchmark HM-NP with 2PEM against: Direct encoding (DE) [21], Optimized unary encoding (OUE) [21], and Square Wave (SW) mechanism [15]. We also examine the statistical information, such as variance and quantile, derived from the estimated distribution.

Finally, we conduct a case study on federated learning using logistic regression. This case study demonstrates the practical application of our mechanisms.

5.1 Mean estimation

Dataset. We evaluate the performance of our proposed mechanisms using one synthetic dataset and two real-world datasets. *Beta* dataset is a synthetic dataset generated from a Beta distribution

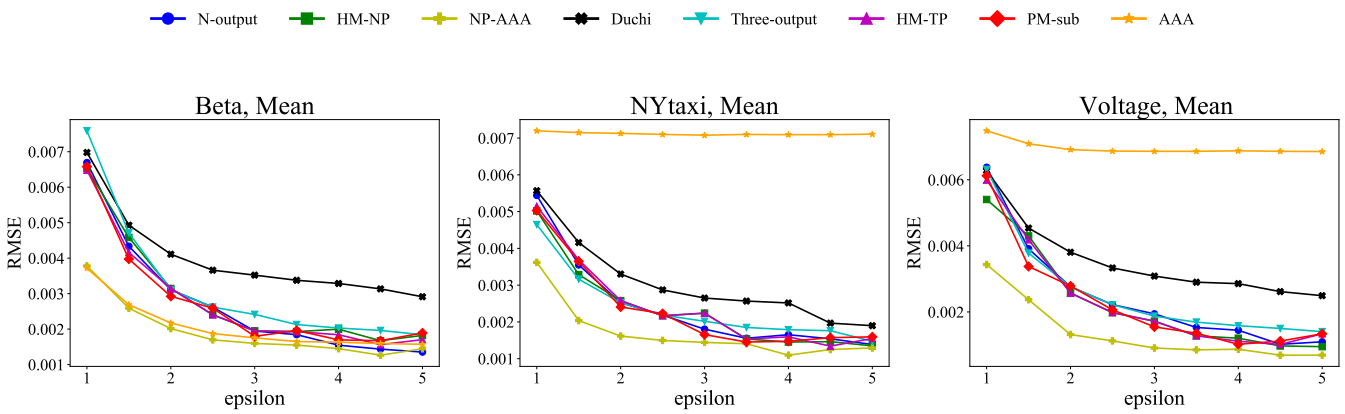


Figure 3: Mean estimation performance (Lower RMSE indicates better performance)

characterized by shape parameters $\alpha = 3$ and $\beta = 2$. This dataset consists of 10 dimensions and contains 1,000,000 samples. *NYtaxi* dataset is a real-world dataset derived from New York City yellow cab taxi data from January 2023 [16]. It has 19 dimensions and 3,066,766 samples. For our experiments, we focus on the pickup time attribute, which is expressed in seconds within a 24-hour period. *Voltage* dataset is also a real-world dataset sourced from Individual Household Electric Power Consumption dataset [11]. This dataset includes 9 dimensions and 1,044,506 samples, and we specifically analyze the voltage attribute.

Experimental settings. Given the multi-dimensional nature of the data, we sample $\frac{k}{dim}$ data points, where k is defined differently depending on the mechanism. For the three-output mechanism, HM-TP, and PM-sub, we use k as defined in equation (3). For the other mechanisms, k is set to 1. We set $\mathcal{X} = [-1, 1]$ for all datasets.

In the AAA mechanism and NP-AAA, we use 10% of the data for distribution estimation, set the bin size to 0.125, and fix the ratio $\frac{|\Omega|}{|\mathcal{X}|} = 4$ for the second phase. Specifically, for the distribution estimation in NP-AAA, we apply 2PEM with a hyperparameter $\tau = 10^{-5}$. The experiments are conducted with varying ϵ within the range $[1, 5]$.

Metric. We measure the performance using the Root Mean Squared Error (RMSE) of the mean estimation, which is calculated as the difference between the true mean and the estimated mean. Each experiments is repeated 100 times and the results are averaged. A lower RMSE indicates better performance.

Experimental results. The performance of mean estimation is illustrated in Figure 3. Overall, our NP-AAA mechanism outperforms the other mechanisms, with exception of $\epsilon = 5$ in the Beta dataset, where the N-output mechanism slightly surpasses NP-AAA. Although the AAA mechanism is effective mechanism, it heavily relies on the accuracy of the distribution estimation in its first phase. In contrast to the mechanism, which performs well on the Beta dataset but struggles significantly with the other datasets, NP-AAA exhibits notable performance improvements across all datasets. The experiments demonstrate that HM-NP and 2PEM effectively address the issue of the AAA mechanism.

The N-output mechanism, HM-NP, HM-TP, and PM-sub show similar performance, with no single mechanism consistently outperforming the others. This is because the mechanisms have different variances depending on the data and ϵ. However, it is important note that HM-NP theoretically guarantees minimized worst-case noise variance. Additionally, both the N-output mechanism and HM-NP are highly communication-efficient.

5.2 Distribution estimation

Experimental settings. For the distribution estimation, we use the same dataset as mean estimation experiments. For hyperparameters, we use $d = 1024$ for all dataset. For the SW mechanism, we set $\tilde{d} = 1024$. For HM-NP with 2PEM, we use $\tau = 10^{-5}$ and $\tilde{d}$ is determined using equation (13).

Metric. To evaluate the distribution estimation, we use the Wasserstein distance. The Wasserstein distance, also known as the Earth Mover's Distance (EMD), is a measure of the distance between two probability distribution. Intuitively, it represents the minimum effort required to transform one distribution into another.

In addition to Wasserstein distance, we also estimate the variance from the estimated distribution. The accuracy of variance is evaluated by calculating the RMSE between the true variance and the estimated variance. Furthermore, we assess the estimation of 10-quantiles, which are data points dividing the dataset into intervals containing 10 % of data each. This is evaluated by calculating the RMSE between the true quantile and the estimated quantile.

Experimental results. The performance of distribution, variance, and quantile estimation is illustrated in Figure 4. The RMSE values for variance and quantile estimation are presented in logarithmic scale due to the significant differences in error magnitudes. As shown in the figure, HM-NP with 2PEM consistently outperforms other distribution estimation mechanisms across all datasets and for each type of estimation. Our mechanisms not only accurately estimates numerical distributions but also provides reliable estimations of other statistical information, such as variance and quantile.

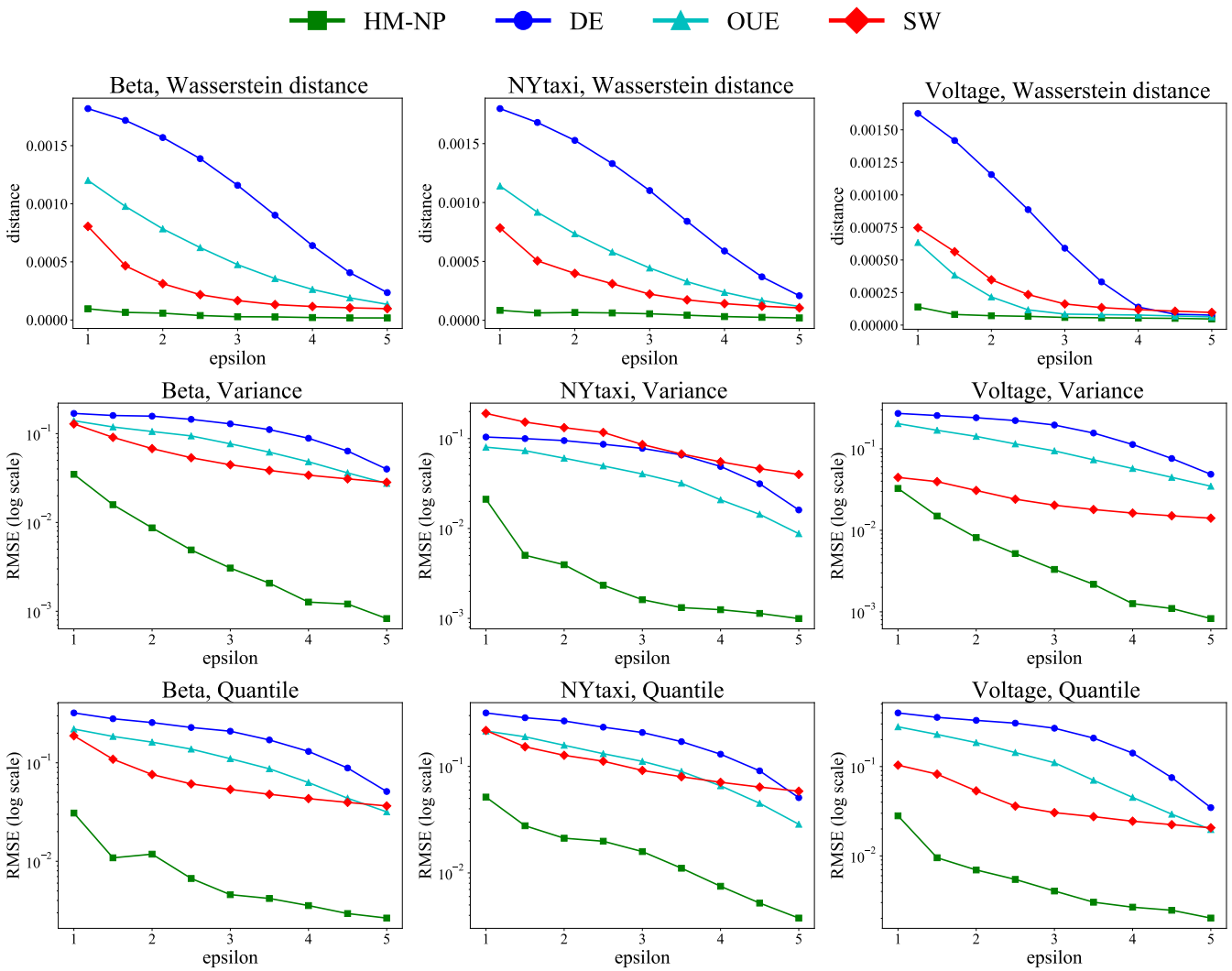


Figure 4: Distribution, Variance, and Quantile estimation (Lower Wasserstein distance and RMSE indicate better performance)

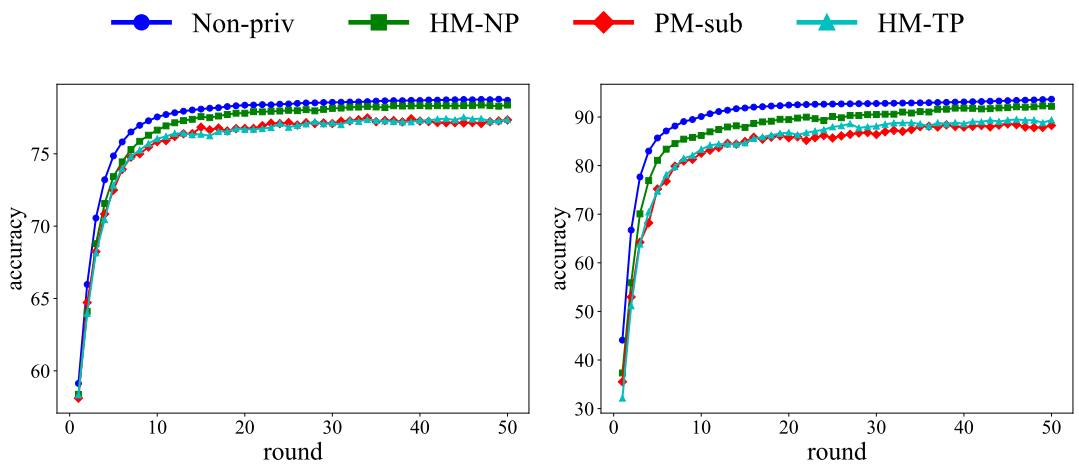
5.3 Case study: federated learning

Dataset. In this case study on federated learning, we evaluate the performance of our proposed mechanisms using two real-world datasets. The first dataset is the *Gamma* dataset [2], which is used for classifying gamma rays and hadrons, and involves binary classification task. This dataset contains 19,020 samples with 10 dimensions. The second dataset is the *Shuttle* dataset [17], which is a part of Statlog collection, a benchmark suite of datasets for evaluating machine learning algorithm. The shuttle dataset specifically deals with space shuttle and involves multi class classification with 7 classes. It consists of 14,500 samples with 9 dimensions.

Experimental settings. In our federated learning setup, data are distributed among 500 clients without overlap. The experiments spans 50 rounds. At each round, the global model is sent to randomly selected 10 % of clients. Each client trains the model using a batch size of 50, 5 epochs, with a learning rate 0.1. The clients report their gradients back to the server. HM-NP, HM-TP, and PM-sub is applied to the gradients before sending it to the sever. However, the AAA mechanism and NP-AAA are not suitable for federated learning in typical settings. These mechanism rely on accurate distribution estimation, but the typical federated learning setup involves high-dimensional data and does not involve enough clients to enable precise estimation.

For the non-private setting, we apply a weight decay of 0.001 to regularize the model. However, for the LDP mechanisms, we set the weight decay to 0, as these models are already regularized by perturbation. For the LDP mechanisms used in federated learning, the input domain is set to $\mathcal{X} = [-0.3, 0.3]$.

We evaluate the performance at $\epsilon = 5$ and present the results as the average over 50 runs. The accuracy metric, where higher values indicate better performance, is used to asses the model's performance.



(a) Gamma **(b) Shuttle**

Figure 5: Federated learning (Higher accuracy indicates better performance)

Experimental results. The accuracy of federated learning is illustrated in Figure 5. In our experiments, we observe that HM-NP consistently outperforms both PM-sub and HM-TP, with HM-TP showing slightly better performance than PM-sub. Notably, in the Gamma dataset, HM-NP achieves an accuracy close to that of non-private learning, highlighting the effectiveness of our mechanism in practical applications.

In our experiments, the domain of gradients is fixed to $[-0.3, 0.3]$ because the distribution of gradients is unknown. Although this aspect was not the focus of our study, tuning the gradient domain based on the distribution estimation using 2PEM could potentially improve the aggregation of gradients in federated learning. We leave this as future work.

6 CONCLUSION

In this work, we proposed the N-output mechanism, which enables private data reporting with high accuracy and communication efficiency. Building on this mechanism, we introduced HM-NP, a hybrid approach that combines the N-output mechanism with PM-sub to effectively minimize the worst-case noise variance—a key measure for mean estimation—while also enhancing communication efficiency. Additionally, we developed 2PEM, a post-processing method that leverages the results of HM-NP to estimate distributions accurately. Finally, we adopt out mechanisms to the AAA mechanism, a state-of-the-art method, and demonstrated significant performance improvements.

Our experiments show that the proposed mechanisms excel in estimating means, distributions, and other statistical information. Moreover, we show that our study is highly effective in practical applications, such as federated learning. We believe that our study contributes valuable insights and tools for implementations in privacy-preserving data analysis.

ACKNOWLEDGMENTS

This work was supported in part by (1) the National Research Foundation of Korea (NRF) funded by Korean Government through Ministry of Science and ICT (MSIT), South Korea, under Grant NRF-2020R1A2C2013286; (2) MSIT under the ICT Creative Consilience Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2024-2020-0-01819; (3) the Basic Science Research Program through NRF funded by the Ministry of Education under Grant NRF-2021R1A6A1A13044830.

REFERENCES

[1] Raef Bassily and Adam Smith. 2015. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 127–135.
[2] R. Bock. 2007. MAGIC Gamma Telescope. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52C8B.
[3] Rui Chen, Haoran Li, A Kai Qin, Shiva Prasad Kasiviswanathan, and Hongxia Jin. 2016. Private spatial data aggregation in the local setting. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 289–300.
[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)* 39, 1 (1977), 1–22.
[5] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. *Advances in Neural Information Processing Systems* 30 (2017).
[6] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2018. Minimax optimal procedures for locally private estimation. *J. Amer. Statist. Assoc.* 113, 521 (2018), 182–201.
[7] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.
[8] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.
[9] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. 2016. Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries. *Proceedings on Privacy Enhancing Technologies* (2016).
[10] Bernard G Greenberg, Abdel-Latif A Abul-Ela, Walt R Simmons, and Daniel G Horvitz. 1969. The unrelated question randomized response model: Theoretical framework. *J. Amer. Statist. Assoc.* 64, 326 (1969), 520–539.
[11] Georges Hebrail and Alice Berard. 2012. Individual Household Electric Power Consumption. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58K54.

[12] Daeyoung Hong, Woohwan Jung, and Kyuseok Shim. 2021. Collecting geospatial data with local differential privacy for personalized services. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2237–2242.

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

[14] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[15] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Škoric. 2020. Estimating numerical distributions under local differential privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 621–635.

[16] NYC Taxi and Limousine Commission. 2024. TLC Trip Record Data. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page Accessed: 2024-08-31.

[17] Statlog project. [n.d.]. Statlog (Shuttle). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5WS31.

[18] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 192–203.

[19] Apple Differential Privacy Team. 2017. *Learning with Privacy at Scale*. https://machinelearning.apple.com/research/learning-with-privacy-at-scale

[20] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 638–649.

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

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

[23] Fei Wei, Ergute Bao, Xiaokui Xiao, Yin Yang, and Bolin Ding. 2024. AAA: An Adaptive Mechanism for Locally Differentially Private Mean Estimation. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1843–1855.

[24] Min Ye and Alexander Barg. 2018. Optimal schemes for discrete distribution estimation under locally differential privacy. *IEEE Transactions on Information Theory* 64, 8 (2018), 5662–5676.

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