Revisiting EM-based Estimation for Locally Differentially Private Protocols

Yutong Ye*‡, Tianhao Wang†, Min Zhang*§, Dengguo Feng*
*Institute of Software, Chinese Academy of Sciences
†University of Virginia
‡Zhongguancun Laboratory, Beijing, PR.China
yutong2017@iscas.ac.cn, *tianhao@virginia.edu*, *zhangmin@iscas.ac.cn*, *fengdg@263.net*

***Abstract*—This paper investigates the fundamental estimation problem in local differential privacy (LDP). We categorize existing estimation methods into two approaches, the unbiased estimation approach, which, under LDP, often gives unreasonable results (negative results or the sum of estimation does not equal to the total number of participating users), due to the excessive amount of noise added in LDP, and the maximal likelihood estimation (MLE)-based approach, which, can give reasonable results, but often suffers from the overfitting issue. To address this challenge, we propose a reduction framework inspired by Gaussian mixture models (GMM). We adapt the reduction framework to LDP estimation by transferring the estimation problem to the density estimation problem of the mixture model. Through the merging operation of the smallest weight component in this mixture model, the EM algorithm converges faster and produces a more robust distribution estimation. We show this framework offers a general and efficient way of modeling various LDP protocols. Through extensive evaluations, we demonstrate the superiority of our approach in terms of mean estimation, categorical distribution estimation, and numerical distribution estimation.**

I. Introduction

Local Differential Privacy (LDP) [18], [32] has been used as one of the standards for collecting large amounts of private user data. In an LDP protocol, an individual perturbs a sensitive record locally, and reports a noisy version to the aggregator; the aggregator collects all sanitized records and computes the statistical results. Keeping private data within the user's device, LDP demonstrates its advantages in enabling privacy-preserving data analysis without involving trustworthy data collectors. LDP techniques have been deployed by companies like Google [22], Apple [2], Microsoft [17].

A series of LDP protocols with different data types and analysis purposes have been proposed [22], [30], [47], [6], [14]. And their LDP aggregation methods can be categorized into two approaches: unbiased estimation and maximal likelihood estimation (MLE). (1) The unbiased estimation approach aims to derive an unbiased estimation of the true distribution. The knowledge of the random perturbation process, represented by a matrix G, is utilized. $G_{i,j}$ is the probability that the i-th input is perturbed to the j-th output. Thus given a hypothetical input distribution $\mathbf{w}$, $G\mathbf{w}$ gives the expected perturbation results $\mathbb{E}[\tilde{\boldsymbol{\omega}}]$. As a result, applying the inverse matrix to $\tilde{\boldsymbol{\omega}}$ gives us the unbiased estimation of $\mathbf{w}$ as $\mathbb{E}\left[G^{-1}\tilde{\boldsymbol{\omega}}\right] = G^{-1}\mathbb{E}[\tilde{\boldsymbol{\omega}}] = G^{-1}G\mathbf{w} = \mathbf{w}$. However, due to the presence of independent noise in each value of $\tilde{\boldsymbol{\omega}}$, this approach may yield unreasonable estimates and distributions. For example, no guarantee of non-negativity or summation to n (total number of participating users), and empirical corrections are generally required [48], [23]. (2) And the MLE approach seeks to find the distribution that maximizes the likelihood of the observed LDP results. This approach employs the construction of a likelihood function for $\mathbf{w}$ and uses an expectation maximization (EM) algorithm to maximize it. The characteristic of this approach is imposing constraints on the estimated values, resulting in smaller overall errors especially when there exists substantial noise [35].

Unfortunately, when attempting to extend the principles of MLE to a broader range of LDP tasks, it encounters the challenge of overfitting to noisy data. The constraints associated with forming a distribution, alongside the impact of noisy data, appear to introduce extra errors (e.g., bias) to the result, particularly when involving many values to be estimated. To address this overfitting issue, Li et al. [34] incorporate a smoothing strategy which imposes smoothness assumptions on the original values. Specifically, in each EM iteration, a smoothing step is applied to the continuous distribution results obtained in the M-step, resulting in improved performance in numerical distribution estimation. However, selecting the smoothing parameter and determining the termination condition for EM remain challenging tasks, and the smoothing technique cannot be applied to categorical value estimation scenarios.

Our research reveals that one of the primary causes of overfitting in EM for LDP data is the excessive number of parameters. In machine learning, regularization terms are commonly employed to penalize excessive parameterization. However, incorporating regularization terms into EM, such as adding in the maximum likelihood function, seems hard to realize. Therefore, we propose leveraging the reduction idea, initially developed for the Gaussian mixture model (GMM) [11], [40], to solve this problem. Reduction is a principled framework

§corresponding author.

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.230521
www.ndss-symposium.org

used in EM algorithms to reduce the number of parameters by iteratively merging or eliminating components that share similar characteristics or contribute minimally to the overall likelihood of the data. We find that this reduction framework is particularly suitable for LDP estimation, as the large size of $\mathbf{w}$ leads to small true counts for many values. Their estimations, having a high probability of being covered by noise, become more effectively addressed through elimination or merging. To use the reduction framework, we adapt the GMM framework to tailor it to the LDP perturbation process (since the random perturbation also creates distributions, although not necessarily following Gaussian).

Our new framework is general and allows the modeling of different LDP protocols. We apply our method to state-of-the-art LDP protocols for different tasks, including one-step basic LDP tasks such as frequency estimation of categorical values, mean estimation of numerical values, and multi-step LDP tasks such as the conditional estimation on key-value data. These tasks serve as building blocks for complicated tasks (e.g., trajectory synthetics [15], graph [55]), and recent LDP poisoning attacks [10], [33] also focus on them). By incorporating our method, we improve all these existing state-of-the-art LDP protocols. Moreover, reduction also improves efficiency: compared to the standard EM algorithm whose running time is proportional to the squared input space, reduction, since it eliminates or merges the estimation of some input variables through the EM process, can greatly reduce the running time in practice.

We conducted evaluations on synthetic and real-world datasets. In terms of mean estimation, our reduction framework applied to the existing method PM [45] consistently achieves the lowest mean absolute error (MAE) compared to its original estimation method. This improvement is particularly significant, with a 70% decrease in MAE, in scenarios with limited data and privacy budget. Regarding frequency estimation, the standard FOs such as GRR [30] and OLH [47], combined with our framework perform comparably to its original unbiased estimation method with consistency-based post-processing method [48], when data and privacy budget are sufficient. But in the opposite case, say $\varepsilon = 0.5$, combining with our framework also provides a relatively $10\% \sim 30\%$ improvement. For numerical distribution estimation, we further apply our reduction framework to Laplace mechanism, PM and SW [34]. After evaluating several metrics (e.g., wasserstein distance, variance), the distribution obtained from our MR reduces the error by 30% compared to EM. As for conditional mean estimation tasks which involves multi-types data, we apply our framework to PCKV [24] and observed a improvement of 40% in mean squared error when $\varepsilon = 1$. In summary, we conclude that MLE with our reduction framework is the preferred choice when the input domain is large (e.g., many values need to estimate) or there exists substantial noise (low ε or insufficient number of users).

To summarize, the main contributions are:

- We summarize the two current LDP aggregation estimation methods and highlight the bottleneck in the application of EM-based MLE is overfitting. To address this challenge, we leverage the reduction idea and propose a mixture reduction framework, incorporating an LDP mixture model to enhance the effectiveness of EM-based MLE.
- We demonstrate the application of our model as a post-processing step to mean estimation, frequency estimation, and numerical distribution estimation tasks, which are the most fundamental and serve as the building blocks of many complicated tasks in LDP field. And we provide access to our code[1].
- We theoretically analyze the mean square error of our method, demonstrating that our reduction framework is the preferred choice when the input domain is large or substantial noise exists. Evaluating our method with both synthetic and real-world datasets, we show that in scenarios with insufficient privacy budget or number of users, our method can reduce the mean absolute error by up to 70%.

II. Background

A. Local Differential Privacy

Throughout the paper, we assume there are many users and an untrusted aggregator. The aggregator's goal is to learn information about users' values. But as the aggregator is untrusted, users apply Local Differential Privacy (LDP) before sending their data to the aggregator.

Definition 1 (LDP). *A randomized function $\Psi(\cdot)$ satisfies ε-local differential privacy if and only if for any possible pairs of x and x' in the domain $\mathcal{X}$, and for any possible output $\tilde{x}$, we have:*

$$\Pr\left[\Psi(x) = \tilde{x}\right] \leq e^{\varepsilon} \cdot \Pr\left[\Psi(x') = \tilde{x}\right] \tag{1}$$

B. LDP Mechanism for Categorical Values

The most basic tools in LDP are mechanisms that can estimate the distribution of users' values. In these mechanisms, users perturb their values locally, and send them to the server. The server, on query of some target value, can output the frequency of that value. These mechanisms are thus also called frequency oracles (shortened as FO's).

Generalized Randomized Response (GRR). This FO protocol generalizes the *randomized response* technique [49]. Here each user with private value $x \in \mathcal{X}$ sends the true value x with probability p, and with probability $1 - p$ sends a randomly chosen $\tilde{x} \in \mathcal{X}$ s.t. $\tilde{x} \neq x$. More formally, the perturbation function is defined as

$$\forall_{x \in X}\ \Pr\left[\Psi_{\mathsf{GRR}(\varepsilon)}(x) = \tilde{x}\right] = \begin{cases} p = \frac{e^{\varepsilon}}{e^{\varepsilon}+K-1} & \text{if } \tilde{x} = x, \\ q = \frac{1}{e^{\varepsilon}+K-1} & \text{if } \tilde{x} \neq x. \end{cases} \tag{2}$$

This satisfies ε-LDP since $\frac{p}{q} = e^{\varepsilon}$.

To estimate the frequency of a target value α (i.e., the ratio of the users who take α as private value to the total number

[1] https://github.com/yyt20080808/LDP-EM-MR

of users), one counts how many times α is reported, denoted as $\sum_{i=1}^{n} 1_{\tilde{x}_i=\alpha}$, and then computes

$$\hat{f}_\alpha = \frac{\sum_{i=1}^{n} 1_{\tilde{x}_i=\alpha}/n - q}{p-q}. \tag{3}$$

In [47], it is shown that this is an unbiased estimation of the true count, and the variance for this estimation is

$$\mathrm{Var}[\hat{f}_\alpha] = \frac{K-2+e^\varepsilon}{(e^\varepsilon-1)^2 \cdot n}. \tag{4}$$

The accuracy of this protocol deteriorates fast when the domain size K increases.

Optimized Local Hashing (OLH) **[47].** This protocol deals with a large domain size K by first using a hash function to map an input value into a smaller domain of size K^* (typically $K^* \ll K$), and then applying randomized response to the hashed value in the smaller domain. The reporting protocol is

$$\Psi_{\mathsf{OLH}(\varepsilon)}(l) \coloneqq \langle H, \Psi_{\mathsf{GRR}(\varepsilon)}(H(l)) \rangle,$$

where H is randomly chosen from a family of hash functions that hash each value in $\mathcal{X}$ to $\{1 \dots K^*\}$. $\Psi_{\mathsf{GRR}(\varepsilon)}$ is given in Equation 2, while operating on the domain $\{1 \dots K^*\}$ and K^* is $\lceil e^\epsilon + 1 \rceil$. The variance is

$$\mathrm{Var}[\hat{f}_\alpha] = \frac{4e^\varepsilon}{(e^\varepsilon-1)^2 \cdot n}. \tag{5}$$

Compared with Equation (4), the factor $K - 2 + e^\varepsilon$ is replaced by $4e^\varepsilon$. This suggests that for smaller K (such that $K - 2 < 3e^\varepsilon$), one is better with GRR; but for large K, OLH is better and has a variance independent of K.

C. LDP Mechanism for Numeric Values

We assume the numerical values are all in the range of $[-1, 1]$. For the case where the value range is different, we can first map the domain into $[-1, 1]$. After the result is obtained, we can map it back to the original domain.

Stochastic Rounding (SR)**.** In detial, this method uses stochastic rounding to estimate the mean of a numerical domain [19]. We call it Stochastic Rounding (SR). The main idea is to round x to $\overline{x}$

$$\overline{x} = \begin{cases} 1 & w/p \quad (x+1)/2, \\ -1 & w/p \quad (-x+1)/2, \end{cases} \tag{6}$$

and then perturb $\overline{x}$ to $\tilde{x}$ with binary randomized response. Formally, the perturbation function is defined as

$$\Pr\left[\Psi_{\mathsf{SR}(\varepsilon)}(x) = \tilde{x}\right] = \begin{cases} \frac{e^\varepsilon-1}{2e^\varepsilon+2}x + \frac{1}{2} & \text{if } \tilde{x} = \frac{e^\varepsilon+1}{e^\varepsilon-1}, \\ \frac{1-e^\varepsilon}{2e^\varepsilon+2}x + \frac{1}{2} & \text{if } \tilde{x} = \frac{e^\varepsilon+1}{1-e^\varepsilon}. \end{cases}$$

The SR method is unbiased in that $\mathbb{E}[\tilde{x}] = x$. The variance is $\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2 - x^2$. Recently, Zhao et al. [54] extends the SR from two outputs to three outputs, and it shows that the three outputs have a smaller worst-case variance than the two when $\varepsilon > 0.69$.

Piecewise Mechanism (PM) **[45].** In this method, the output domain is continuous and in the range of $[-C, C]$ where $C = \frac{\exp(\varepsilon/2)+1}{\exp(\varepsilon/2)-1}$. For each $x \in [-1, 1]$, there is an associated range $[\ell(x), r(x)]$ close to x, such that, with a higher probability p, the output value $\tilde{x}$ is in $[\ell(x), r(x)]$, and with a lower probability q, $\tilde{x}$ is outside the range:

$$\Pr\left[\Psi_{\mathsf{PM}(\varepsilon)}(x) = \tilde{x}\right] = \begin{cases} p & \text{if} \quad \tilde{x} \in [\ell(x), r(x)], \\ q & \text{if} \quad \tilde{x} \in [-C, \ell(x)) \cup (r(x), C]. \end{cases} \tag{7}$$

PM sets $\ell(x) = \frac{C+1}{2}x - \frac{C-1}{2}$ and $r(x) = \ell(x) + C - 1$, and satisfies ε-LDP by setting $q = p/e^\varepsilon$. It is also shown to be unbiased and its variance is $\frac{x^2}{e^{\varepsilon/2}-1} + \frac{e^{\varepsilon/2}+3}{3(e^{\varepsilon/2}-1)^2}$.

Square Wave mechanism (SW) **[34].** The perturbation part of SW is similar to PM. And the output field of SW is fixed to $[-b, 1+b]$, where b is correlated to the privacy budget. Given a record x, the pdf of SW's perturbation step is

$$\Pr\left[\Psi_{\mathsf{SW}(\varepsilon)}(x) = \tilde{x}\right] = \begin{cases} p \cdot e^\varepsilon & \text{if } \tilde{x} \in [x-b, x+b], \\ p & \text{otherwise.} \end{cases}$$

where $p = 1/(2be^\varepsilon + 1)$. At last, they proved that when the parameter is set as:

$$b = \frac{\varepsilon e^\varepsilon - e^\varepsilon + 1}{2e^\varepsilon(e^\varepsilon - 1 - \varepsilon)},$$

the upper bound of mutual information between the input and output is maximized. Moreover, this method combines a estimation step (Expectation Maximization), and can be used to estimate numerical distributions. With the distribution, the mean is indirectly obtained.

D. Notations

Throughout the paper, we use "tilde" to denote perturbed values by LDP, "hat" or "widehat" to denote the estimated results. For example, x is a user's private numerical value, and $\tilde{x}$ is the perturbed value; μ is the ground-truth mean of all users' values, and $\widehat{\mu}$ is the estimated mean. We also use bold letters to denote vectors, such as $\mathbf{w}$. And use w_i to denote the i-th value in $\mathbf{w}$.

III. Statistical Inference Methods under LDP

We focus on addressing the fundamental estimation problems on the aggregator side in the context of LDP. LDP aggregation methods can be categorized into two approaches. The first one tries to derive an unbiased estimation, and the second one aims at finding the distribution that most likely leads to the observed LDP reports. We now discuss them in more detail.

1. Unbiased Estimation. Although we do not know the true distribution $\mathbf{w}$, we know the random perturbation process, which can be represented as a matrix G. Thus, given any hypothetical input distribution $\mathbf{w}$, we can compute the expected perturbation results $\mathbb{E}[\tilde{\boldsymbol{\omega}}] = G\mathbf{w}$, where $\tilde{\boldsymbol{\omega}}$ denotes the perturbation results. Therefore, fundamentally, deriving an unbiased estimation is equivalent to solving matrix inversion problem $\hat{\mathbf{w}} = G^{-1}\tilde{\boldsymbol{\omega}}$, as we can show that

$$\mathbb{E}[\hat{\mathbf{w}}] = \mathbb{E}\left[G^{-1}\tilde{\boldsymbol{\omega}}\right] = G^{-1}\mathbb{E}[\tilde{\boldsymbol{\omega}}] = G^{-1}G\mathbf{w} = \mathbf{w}.$$

The matrix often becomes too large when the output alphabet has many possibilities. Fortunately, in LDP, there are specific patterns in the random perturbation process that allows us to derive simpler estimation processes, as in Equation 3.

Post-processing Method. Observing the unbiased estimation may provide unreasonable results (e.g., Equation 3 may give negative values due to noise), one can apply post-processing calibration algorithms to revise and improve accuracy, most of which are based on consistency and some assumptions. Specifically, consistency means that each estimated value is non-negative and their sum is 1. To attain non-negativity, a common approach involves the implementation of significance threshold [22], [47] to discard negative and tiny-value estimates. To ensure that the estimated frequencies sum to one, one can use a normalization algorithm, such as the addition of a small value, denoted by δ, to all elements within the vector $\tilde{\boldsymbol{\omega}}$. Normsub and Basecut [48], [14] are the typical consistency-based methods currently. Recently, Fang et al., [23] assume the existence of continuity or smoothness in adjacent values of the original frequency distribution, and employ convolution techniques (Improved Iterative Wiener (IIW) filter algorithm) to smooth the frequency estimation results. The above post-processing methods are empirically useful and are mostly applied to frequency estimation tasks on categorical data.

2. Maximal Likelihood Estimation (MLE). This approach finds a distribution that most likely leads to the observed LDP results. Specifically, given any hypothetical input $\hat{\mathbf{w}}$, the random perturbation process G and the observed perturbation results $\tilde{\boldsymbol{\omega}}$ (here $\tilde{\boldsymbol{\omega}}$ is the histogram of observed reports $\{\tilde{x}_1, \ldots, \tilde{x}_n\}$), one can derive the likelihood (probability) of G turning $\hat{\mathbf{w}}$ into $\tilde{\boldsymbol{\omega}}$. That is,

$$\mathcal{L}(\hat{\mathbf{w}}) = Q_n(\tilde{\boldsymbol{\omega}}; \hat{\mathbf{w}}) \tag{8}$$

where the term $Q_n(\tilde{\boldsymbol{\omega}}; \hat{\mathbf{w}})$ represents the joint probability mass (or density) function for n random variables which construct a histogram $\tilde{\boldsymbol{\omega}}$. The MLE is to find the values of the model parameters $\hat{\mathbf{w}}$ that maximize the above function, while the $\hat{\mathbf{w}}$ satisfies consistency constraints.

$$\arg\max_{\hat{\mathbf{w}}} \mathcal{L}(\hat{\mathbf{w}}) \quad \text{s.t.} \ \sum \hat{w}_i = 1, \ \hat{w}_i \geq 0$$

Expectation-maximization (EM) algorithm [7], [16] is the general approach to optimize the above. Briefly speaking, EM iteratively invokes (1) an E-step that estimates the likelihood given $\hat{\mathbf{w}}$ and (2) an M-step to update $\hat{\mathbf{w}}$ that maximize the likelihood function (by taking the derivative of the likelihood function).

Discussion. The estimation problem is particularly important in the LDP settings (in the central DP setting, we typically do not need estimation because the observation is usually an unbiased estimation; for example, Laplace and Gaussian distributions are symmetric). Essentially, unbiased estimation approach is equivalent to applying the MLE independently to each value in $\mathbf{w}$, and thus the whole may produce an unreasonable distribution (e.g., sum not to n and negative values). And the accumulated error across all values may be large, especially when the number of users or ε is small. And EM-based MLE naturally deliver a distribution, while each value is not guaranteed to be unbiased. Therefore, we can consider the former as an unconstrained MLE and the latter as a constrained MLE. Typically, EM-based MLE is considered when the simple form of unbiased estimation (i.e., Equation 3) cannot be directly derived and when it is desirable to keep the overall error smaller.

Nevertheless, it is essential to acknowledge the drawbacks associated with MLE. As the number of values in $\mathbf{w}$ increases, the consistency constraint inherent in EM-based MLE also extends its impact to a greater set of values. In such scenarios, the iterative process of EM also involves a larger number of parameters, making it susceptible to overfitting to noisy data. For instance, when employing Gaussian Mixture Models (GMMs) as a part of the EM algorithm, an overemphasis on maximizing the likelihood can lead to overfitting, resulting in an excessively large number of components, each having small covariance [29], [11]. Additionally, in the LDP protocol SW [34] for numerical distribution estimation, the over-fitting issue cause the output continuous distribution to be overly spiky. In an attempt to address this, the authors introduced a smoothing step after each M-step in the EM algorithm. However, this will also lead to the phenomenon of over-smoothing, and it is not suitable to other types of LDP tasks such as frequency estimation. Moreover, there is currently no generalized framework for EM-based MLE in most of existing LDP protocols.

In this paper, we aim to explore the broad applicability of EM-based MLE on basic LDP protocols, while addressing the challenge of overfitting issues.

IV. EM with Reduction

In this section, we propose to leverage the reduction idea [11], [40], originally from the Gaussian mixture model (GMM) [38], to solve the overfitting problem (we call our solution mixture reduction, abbreviated as MR). Reduction is a principled framework used in EM algorithms to reduce the number of parameters by iteratively merging or eliminating components that share similar characteristics or contribute minimally to the overall likelihood of the data. In particular, we find it well-suited for EM-based LDP estimation, since many values' true counts are small when the size of $\mathbf{w}$ increases, and their estimations are more likely to be covered by noise, and are better eliminated or merged. To use the reduction framework, we adapt the GMM framework to tailor it to the LDP perturbation process (since the random perturbation also creates distributions, although not necessarily following Gaussian). Then, we apply the reduction operation during the EM procedure.

A. Reduction

Reduction in EM arises in the context of Gaussian Mixture Models (GMMs) [38], which assumes observed data is sampled from a mixture of different Gaussian distributions, and

tries to estimate the unknown weights of each distribution. When there are too many possible Gaussian components, naively running EM will lead to overfitting problems, and the intuition behind reduction is to focus on significant components and ignore components whose estimated weights are small. Generally, pruning and merging are the two primary operations for reducing the number of components [11], [40] in Gaussian. Pruning entails removing a specific component from the mixture, while adjusting the weights of the remaining components to ensure the integrated mixture sums up to unity. Merging involves consolidating two or more similar components into a single component, such as $(\pi', \mu', \sigma'^2) \leftarrow \{(\pi_1, \mu_1, \sigma_1^2), (\pi_2, \mu_2, \sigma_2^2)\}$ for Gaussian.

We give the reduction algorithm with EM in the Algorithm 1. It starts by initializing the mixture model with K components. It then iteratively uses a predefined strategy to decrease the number of components, and re-estimates the mixture model using the EM algorithm. While the EM algorithm in LDP demonstrates convergence, it is crucial to note that the reduction operation does not inherently possess the convergence property. Consequently, the algorithm's convergence is tracked through the computation of the change in Bayesian Information Criterion (BIC) [41], [36], denoted as $\Delta\ell$, which is a pervasively used tool in statistical model selection. The BIC quantifies the trade-off between model fit and complexity using the formula $\mathrm{BIC} = -2\log(\mathcal{L}) + K'\log(n)$, where $\mathcal{L}$ represents the likelihood function, n is the number of samples and K' is the remained number of components. A model with a lower BIC value is considered to be a better-fitting model, striking an optimal balance between fit and complexity. The termination of Algorithm 1 also occurs when the change in BIC surpasses a specified threshold, signifying negligible improvement. In such cases, the last operation of reduction should be cancelled.

Remark. Overfitting is a well-studied issue in machine learning, and there, a popular solution is to use regularization. For example, we can suggest a penalized log-likelihood as

$$\log \mathcal{L}(\hat{\mathbf{w}}) = \log Q_n(\tilde{\boldsymbol{\omega}}; \hat{\mathbf{w}}) - \lambda\Omega(\hat{\mathbf{w}}),$$

where $\Omega(\hat{\mathbf{w}})$ represents ℓ_1 or ℓ_2 regularization term. Unfortunately, in the LDP setting, the ℓ_1 regularization term has no effect because there is a constraint that the sum of elements in $\hat{\mathbf{w}}$ is fixed to one. This constraint also diminishes the efficacy of ℓ_2 regularization. Thus, we choose the reduction strategy.

B. LDP Mixture Model

In order to execute the reduction operation, it is necessary to first adapt the Gaussian mixture model to the LDP mixture model:

Definition 2 (LDP Mixture model)**.** *A LDP mixture model is represented by a convex combination of component densities or distributions. It consists of proportions or weights (denoted as π) for each component, satisfying $\pi_k \geq 0$ and $\sum \pi_k = 1$. The mixed probability mass function, denoted as $\phi(\tilde{x}; \boldsymbol{\pi}, \boldsymbol{\alpha})$, is obtained by summing the weighted contributions of each component density $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k) = \tilde{x}\right]$ for $k = 1$ to K.*

$$\phi(\tilde{x}; \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{k=1}^{K} \pi_k \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k) = \tilde{x}\right] \tag{9}$$

where the $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k) = \tilde{x}\right]$ represents the probability mass function of the perturbation that generates report $\tilde{x}$ from the private data $x = \alpha_k$ in LDP protocols.[2]

Mixture models have a flexible and probabilistic nature, which can preserve important characteristics for the data after reduction, and we can use the metrics derived from GMM to judge the reduction effect. Specifically, assuming that the input space for users' private data consists of K distinct values $\alpha_1, \alpha_2, \ldots, \alpha_K$, and the data of n users in total form a normalized frequency histogram $\mathbf{w}$. The perturbation process for each private data can be viewed as a sample $\tilde{x}$ drawn from a perturbation function $\Psi_\varepsilon(x)$ with the private data x ($x \in \{\alpha, \ldots\}$) and the privacy budget ε as input.

And the objective is to fit the observed data. Thus, the log-likelihood function is the objective function to optimize:

$$\log \mathcal{L}(\hat{\mathbf{w}}) = \sum_{i=1}^{n} \log\left(\sum_{j=1}^{K} \hat{w}_j \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j) = \tilde{x}_i\right]\right) \tag{10}$$

For the mixture model with cleared likelihood function, the expectation-maximization algorithm only needs to find the weights of the mixture model, since the shapes of components are clearly fixed in perturbation functions.

Reduction Operations. Pruning means removing an LDP component from the target log-likelihood function. In general, in the absence of any prior knowledge, a greedy strategy is used to continuously select and remove the component with the smallest weight. In the final results, those estimates that are eliminated will be replaced with zeroes. So, the pruning operation is more suitable for distributions with many zeros.

In addition, merging is a more general operation that combines multiple components into a single component.

Algorithm 1 Mixture Reduction Algorithm

Input: Noisy data $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n$, number of components K, min number of components $K_{\min}$, threshold τ
Output: A simplified mixture model
Initialize mixture model with K components, $K' \leftarrow K$
while $K' > K_{\min}$ **do**
  Compute $\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{K'}$ with EM (Algorithm 2)
  Select components $\{k^*, ...\}$ based on a strategy
  Apply reduction operation
  Compute change in BIC: $\Delta\ell \leftarrow \mathrm{BIC}_{\text{new}} - \mathrm{BIC}_{\text{old}}$
  if $\Delta\ell > \tau$ **then**
    Break
  Update K'
return Mixture model with K' components

[2] If the LDP protocol incorporates continuous perturbation methods (such as Laplace), the component can be substituted with a probability density function.

Algorithm 2 Expectation-Maximization Algorithm

Input: Noisy data $\tilde{x}_1, \ldots, \tilde{x}_n$, number of components K, initialized weights $\hat{\mathbf{w}}$, perturbation functions of LDP.
Output: Estimates for weights $\hat{w}_k$.

Initialization:
Calculate the initial log-likelihood $\mathcal{L}(\hat{\mathbf{w}})$ by Equation 10.
repeat
  (E-step)
  for $i \leftarrow 1$ to n **do**
    for $k \leftarrow 1$ to K **do**
      Calculate the posterior probability

$$\gamma_{ik} \leftarrow \frac{\hat{w}_k \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k) = \tilde{x}_i\right]}{\sum_{j=1}^{K} \hat{w}_j \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j) = \tilde{x}_i\right]}$$

  (M-step)
  for $k \leftarrow 1$ to K **do**
    Update the weights

$$\hat{w}_k \leftarrow \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}$$

  Calculate the log-likelihood $\mathcal{L}(\hat{\mathbf{w}})$.
until convergence criterion is met

The following is a merging step, $(w_{12}, \Psi_\varepsilon(\alpha_{12})) \leftarrow \{(w_1, \Psi_\varepsilon(\alpha_1)), (w_2, \Psi_\varepsilon(\alpha_2))\}$:

$$w_{12} = w_1 + w_2$$

$$\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_{12}) = \tilde{x}\right] = \sum_{i=1}^{2} \frac{w_i}{w_{12}} \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_i) = \tilde{x}\right].$$

Since the components' weights are all influenced by the noise, we only merge small weights components and assume $\frac{w_i}{w_{12}} = 1/2$. Then, the strategy for selecting merged components is to choose those that are as close or similar in properties as possible, such as selecting the components with the lowest weights, or based on adjacency information. In the remaining sections, we assume no prior knowledge about the data distribution (e.g., many zeros). Therefore, we default to using only the merge operation and not using pruning.

Proposition 1. *The mean squared error of our reduction algorithm (Algorithm 1), which involves a total of t times merging operations where each merging operation merges $h_1, \ldots, h_t$ mixture components, is given by:*

$$\mathsf{MSE}_{\mathsf{Ours}} = \frac{K'}{K} \mathsf{MSE}_{\mathrm{EM}} + \frac{1}{K} \sum_{i=1}^{t} h_i \sigma_i^2. \tag{11}$$

where σ_i is the variance of the true weights of the components merged at the i-th operation.

We defer the proof to Appendix B. The first term in Equation (11) is the estimation errors when the EM algorithm executes for remaining (or non-merged) components and the newly generated components, it is of order $O(n^{-1})$. The second term is the error introduced by the merging operation on the estimates of the merged components.

Roughly speaking, when the variance of the merged weights is much smaller than the EM estimation error (or $\frac{1}{K}\sum_{i=1}^{t} h_i \sigma_i^2 < \frac{K-K'}{K} \mathsf{MSE}_{\mathrm{EM}}$), our algorithm performs better. When many values need to estimate (large K), it is more likely that some values will be close to each other, resulting in a small variance among the true weights of the merged components. Moreover, the MSE of EM can also be quite large when data size n is small or when the ε is insufficient. To summarize, when these conditions are met,

$$\mathsf{MSE}_{\mathsf{Ours}} \approx \frac{K'}{K} \mathsf{MSE}_{\mathrm{EM}}.$$

V. Applying EM Reduction

In this section, we investigate the employment of EM-based MLE with MR, on different state-of-the-art LDP protocols.

A. Categorical Data

Specifically, in the case of GRR (described in Section II-B), we refer to Algorithm 2 where the target output $\hat{w}$ represents the estimated frequency for each category value. The term $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j) = \tilde{x}_i\right]$ takes the value p as defined in Equation 2 when $\tilde{x}_i$ is the true answer. Since the output domain and input domain are the same, we can represent the transformation (also perturbation) probabilities from an input α_j to any output $\tilde{x}_i$ equal to α_i using a matrix $G \in [0,1]^{K \times K}$. Thus, the space for storing is $O(K^2)$.

To our knowledge, there is no existing research that applies the EM algorithm to handle the noisy data produced by the OLH mechanism, particularly due to its infinite output domain. However, we find it also applicable within our framework. For any noisy data $\langle H_i, \tilde{x}_i \rangle$, one need to get its likelihoods to all components (all possible α). For example, the term $\mathsf{Pr}\left[\Psi_\varepsilon(\langle H_i, \alpha_j \rangle) = \langle H_i, \tilde{x}_i \rangle\right]$ takes the value $\frac{e^\varepsilon}{e^\varepsilon + K^* - 1}$ when hash matches $\tilde{x}_i = H_i[\alpha_j]$, or otherwise takes value $\frac{1}{e^\varepsilon + K^* - 1}$. In this situation, there are totally nk values should be pre-calculated. And these values need to be used multiple times in the EM algorithm, storing them would require approximately $O(nK)$ space.

Reduction. The above-discussed protocols can be directly equipped with mixture reduction algorithm. When no prior knowledge is available, it is common to employ a greedy strategy to select those LDP components with the smallest weights, and merge them. This corresponds to lines $4 \sim 5$ of Algorithm 1. Note that a threshold should be applied to prevent the merging of actually frequent items (or components) with relatively low estimated results. In the Base-cut method [48], they employ such a threshold. Additionally, for any value whose original count is 0, the probability that it will have an estimated frequency less than 2σ (σ is the standard deviation of LDP mechanism, as in the Equation 4) is at most 95%. Consequently, when observing an estimated frequency above 2σ, the probability that the true frequency of the value is 0, is at most 5%. In our approach, we use the merge operation for values that are likely to be covered by the noise. Given that the noise level is approximately up to 2σ (related to n and ε), we empirically set this threshold (denoted by τ) to 2σ. When the data size (n) is small, the noise level of the EM algorithm tends

to be high. Our strategy merges more components, as many are likely noise-dominant, resulting in less error compared to the EM algorithm. As n increases, fewer components will be selected for merging, and the overall estimation aligns more closely with EM. If there is prior knowledge about the distribution (like sparsity), one can manually choose those that are more likely to have lower original values.

Additionally, merging two components at one time, becomes computationally inefficient when the number of initial components K is large. Thus, we suggest two approaches to accelerate the process. First, employing the binary search concept to progressively select and retain components with the highest weights, or alternatively, merging at most half of the components with weights below τ in each step. This will reduce the number of merging step invocations from $O(K)$ to $O(\log K)$. Second, when initializing the number of components, pre-selecting and merging the components whose corresponding unbiased estimation results are negative or tiny small, will reduce the size of K.

Selection of Perturbation Methods. According to the Cramer-Rao lower bound theorem, the MLE accuracy is higher when the fisher information is larger. To investigate the factors influencing the estimated results in Algorithm 2, we first differentiate the log-likelihood function (Equation 10) twice with respect to each $\hat{w}_j$, and get

$$\sum_{i=1}^{n} \frac{\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j)=\tilde{x}_i\right]}{(\sum_{k=1}^{K}\hat{w}_k \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k)=\tilde{x}_i\right])^2}. \tag{12}$$

To satisfy LDP, it is clear that for any output $\tilde{x}$, there exist constraints that $\frac{\mathsf{Pr}[\Psi_\varepsilon(\alpha_i)]=\tilde{x}}{\mathsf{Pr}[\Psi_\varepsilon(\alpha_j)]=\tilde{x}} \leq e^\varepsilon$. Assuming no prior knowledge and considering a uniform distribution for w, we have:

$$-\frac{\partial^2 L}{\partial \hat{w}_j^2} = \sum_{i=1}^{n} \frac{K^2 \mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j)=\tilde{x}_i\right]}{(\sum_{k=1}^{K}\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k)=\tilde{x}_i\right])^2} \tag{13}$$

Then, the accuracy of the estimator is influenced by the ratio between a sample's contribution to its true component and its contributions to other components. For protocols like GRR, where $\sum_{k=1}^{K}\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k)=\tilde{x}_i\right]=1$, and $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k)=\tilde{x}_i\right]$ is bounded by $O(\frac{\varepsilon}{\varepsilon+K})$, and $\mathsf{Var}[\hat{w}_j]$ is $O(\frac{K}{ne^{2\varepsilon}})$. Similarly, the variance of OLH by EM is bounded by $O(\frac{1}{ne^{2\varepsilon}})$. This indicates that the selection of perturbation methods also depends on ε and K.

B. Numerical Data

We observe that using MLE for SR is almost equivalent to its unbiased estimation because SR constrains the input and output domain to $\{-1,1\}$ (just two components in EM), and the only two components do not need reduction. Consequently, we focus on other mechanisms introduced in Section II-C. Interestingly, we also observe that PM and SW actually belong to the same category of perturbation mechanisms. The key distinction lies in the inference method employed by each mechanism: PM utilizes a formula for mean estimation because of its special parameter setting towards unbiasedness, while SW employs MLE for density. To facilitate a more comprehensive comparison between the two inference methods within the context of the same perturbation algorithm, we first introduce the steps of combining PM with a mixture model.

Algorithm 3 Mixture Reduction for PM

Input: Noisy data $\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n\}$, number of components K, min number of components $K_{\min}$, weight threshold τ
Output: A simplified Mixture model

1: Initialize the mixture model $\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K$
2: Set $K^{(1)} \leftarrow K$, $t \leftarrow 1$
3: **while** $K^{(t)} > K_{\min}$ **do**
4: Compute weights of $\hat{w}_{1^{(t)}}, \hat{w}_{2^{(t)}}, \ldots, \hat{w}_{K^{(t)}}$
5: **for** each component $\hat{w}_{k^{(t)}}$ in remaining set **do**
6: **if** $\{\hat{w}_{k^{(t)}}, \hat{w}_{k+1^{(t)}}, \ldots, \hat{w}^{(t)}_{k+\lceil K/2^t \rceil}\}$ all exist **then**
7: Compute their sum $S_k \leftarrow \sum \hat{w}_{k^{(t)}}$
8: $S_{k^*} \leftarrow \min_{k=1}^{K^{(t)}} S_k$
9: **if** $S_{k^*} < \tau$ **then**
10: Merge the k^*-th component and its neighbors
11: $K^{(t+1)} \leftarrow K^{(t)} - \lceil K/2^t \rceil + 1$
12: Re-estimate the mixture model
13: $\Delta\ell \leftarrow \text{BIC}_{\text{new}} - \text{BIC}_{\text{old}}$
14: **if** $\Delta\ell < 0$ **then**
15: Break
16: $t = t + 1$
17: **return** Mixture model with $K^{(t)}$ components

PM Mixture Model. To model the PM's noised reports for numerical estimation, a crucial preliminary step is discretization, where the input and output domains are divided into equal-width ranges or bins denoted as B_j^{in} and B_ℓ^{out}. Each input bin can be seen as a distinct value α, and its PM perturbation process corresponds to a specific PM component, characterized by a matrix $M \in [0,1]^{K\times K}$ that represents the transformation (perturbation) probabilities from an input bin j to any output bin ℓ. The element $M_{\ell,j}$ indicates the probability $\mathsf{Pr}\left[\tilde{x} \in B_\ell^{\text{out}} \mid x \in B_j^{\text{in}}\right]$, which is obtained by integrating the PM's probability density function over the interval of B_ℓ^{out},

$$M_{\ell,j} = \int_{\tilde{x}\in B_\ell^{\text{out}}} \mathsf{pdf}(\tilde{x}|x \in B_j^{\text{in}})\,\mathrm{d}\tilde{x}. \tag{14}$$

Referring to the outlined Algorithm 2, for each noisy value generated by PM, the term $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j)=\tilde{x}_i\right]$ takes the value $M_{\ell,j}$ if $\tilde{x}_i$ falls within the output bin B_ℓ^{out}. And the n noised reports can form a histogram $\tilde{\omega}$, and we can derive the likelihood of M turning $\hat{\mathbf{w}}$ into $\tilde{\omega}$ (see Equation 8). Subsequently, utilizing the computed weights $\hat{\mathbf{w}}$, the center of the estimated mixture model can serve as a mean estimator for the population mean, in the form of a weighted average of the means of each of the individual PM components.

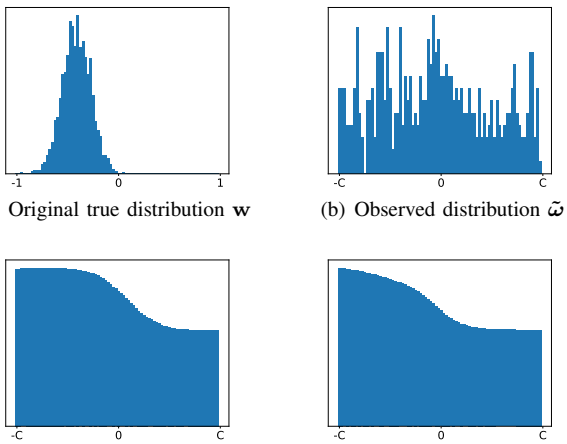
Reduction. Here the parameter K in PM-EM is set to 1024, following a common practice in SW [34]. Given that the large value of K, we use a binary search selection strategy similar to that for discrete data but with a sliding window to merge components with similar weights (Lines $5 \sim 8$ of Algorithm 3), leveraging the continuous nature of the data to merge components associated with small-value ranges. At each step, we retain half of the range with the larger values,

TABLE I
SUMMARY OF METHODS IN EM-BASED MLE

Methods	Description	Pre-process	Probability mass or density function	Time complexity
GRR	FO in small K scenario	-	Equation (2)	$O(K^2 \log(K) I)$
OLH	FO in large K scenario	hash matching	$\frac{e^\varepsilon}{e^\varepsilon + K^* - 1}$ if hash matches	$O(nK \log(K) I)$
PM & SW	numerical FO and mean estimator	binning	Equation (7) and (14)	$O(K^2 \log(K) I)$
Laplace	numerical perturbation	binning	the pdf of Laplace distribution	$O(nK \log(K) I)$
Gaussian	(ε, δ)-LDP for high-dimensional data	binning	the pdf of Gaussian distribution	$O(nK \log(K) I)$
PCKV-PM	key-value data analysis	binning	joint pmf from the combination of PM and FOs	$O(Kd^2 \log(d) I)$

discarding the other half whose sum of weights is less than a threshold. This threshold τ (Line 9 of Algorithm 3) is also related to the twice standard deviation, which is the same as that in category protocols. Since there is no unbiased estimation of $\hat{w}_i$ in PM (or SW), we use the Cramer-Rao lower bound [43] (e.g., the inverse of Equation (13)) to obtain a reference value for its standard deviation.

Naturally, for PM, it is important to determine when to use the sample mean estimator (unbiased estimation, $\frac{1}{n}\sum \tilde{x}$) and when to use the EM-based estimator ($\sum \hat{w}_k \hat{u}_k$). Then, we use an example in Figure 1 to show that the EM-based estimator is suitable in small sample situations. The intuition is that the aggregated LDP data given in Fig 1(b) is like an under-sampling of a mixed distribution given in Fig 1(c). EM-based MLE facilitates noise smoothing by implicitly padding the data, and as a result, the revised distribution given in Fig 1(d) is more likely to provide a better mean result. In our experiments, we observed that when the dataset size exceeds $300,000$ and the privacy budget ε is greater than 2, the use of unbiased estimation can have a superior result.



(a) Original true distribution $\mathbf{w}$ (b) Observed distribution $\tilde{\boldsymbol{\omega}}$

(c) Ideal noisy distribution $M\mathbf{w}$ (d) Estimated distribution of the mixture model

Fig. 1. Examples to show the model parameter estimation can handle small sample problems: (a) shows the original distribution; (c) is the ideal or expected noised output using PM when $\varepsilon = 0.5$ and $n \to \infty$, and it equals $M\mathbf{w}$; (b) is the noised distribution from $n = 2000$ observed reports; and (d) is the mixture model distribution estimated by Algorithm 3.

Laplace and Gaussian Mechanisms. The Laplace mechanism is commonly employed in central DP to protect numerical values by adding a sample $\sigma \sim \text{lap}(\Delta f/\varepsilon)$ from the Laplace distribution. In the case of pure LDP for mean estimation, the sensitivity $\Delta f = 2$ if we scale the overall input domain to $[-1, 1]$. Furthermore, we can also apply the EM algorithm to Laplace noised reports through discretization, similar to PM with EM. In addition, the Laplace mechanism is well-suited for graph analysis under ε-edge LDP [26], [25], where the sensitivity for degree estimation of a single user for edge LDP is 1. Specifically, if a user's degree is $x \in \mathbb{R}$, the reported value would be $x + \text{lap}(1/\varepsilon)$, which satisfies edge LDP. By applying EM to the aforementioned Laplace noised reports, we can not only obtain the mean of the degrees in the graph, as in PM, but also valuable distribution information for each degree. Assuming the degrees are in the range $[0, k-1]$, for each noised report, the term $\Pr[\Psi_\varepsilon(j)] = \tilde{x}_i$ takes the value $\frac{\varepsilon}{2} e^{-|\tilde{x}_i - j|\varepsilon}$, where $j \in [0, k-1]$.

Then for the Gaussian mechanism, it can be directly employed in high-dimensional data. Here the sensitivity is ℓ_2 loss. Specifically, if a user's data is $x \in \mathbb{R}^d$, the reported value would be $x + \text{N}(\mathbf{0}, \sigma^2)$. The term $\Pr[\Psi_\varepsilon(j)] = \tilde{x}$ is replaced by a Gaussian's pdf ($\mathcal{N}(\mathbf{j}, \sigma^2)$).

C. Multi-step LDP Protocols

In fact, various LDP protocols that feature a finite input domain and explicit perturbation probabilities can be augmented with the EM algorithm and reduction technique. In this subsection, we further introduce the employment of the mixture model for multi-step LDP protocols, particularly in scenarios where diverse data types are involved, and the computation entails the aggregation of multi-step LDP estimation outcomes.

We illustrate this through a case study involving key-value data collection. In this context, each user has a private pair $(z_i, x_i), z_i \in \{\alpha_1, \ldots, \alpha_K\}, x_i \in [-1, 1]$, and one of the estimation target is the mean of each value from users whose categorical value $z = \alpha$:

$$\mu^{(\alpha)} = \frac{\sum_{i=1}^{n} 1_{z_i = \alpha} x_i}{\sum_{i=1}^{n} 1_{z_i = \alpha}}. \tag{15}$$

We follow the state-of-the-art method PCKV [24] (details in Appendix D), and replace its constituent module SR with PM. The rationale behind this substitution stems from the superior consistency of PM in the EM algorithm, as compared to SR, which incorporates a rounding step (Section II-C). The noised reports are of two kinds, one is the noised category value set $\{\tilde{z}_1, \ldots\}$ which is used for estimating the denominator of Equation 15. The other is the noised numerical value sets $\{\{\tilde{x}_1^{(\alpha_1)}, \ldots\}, \ldots, \{\tilde{x}_1^{(\alpha_K)}, \ldots\}\}$, which are used for estimat-

ing the numerator. To satisfy LDP, numerical value sets exist dummy values sampled from a uniform distribution, and the percentage of dummy values corresponds to the revised result from noised category value set. Thus, utilizing this connection between two types of noised data, the likelihood is:

$$\mathcal{L}(\hat{\boldsymbol{\theta}}) = Q_n(\tilde{\boldsymbol{\omega}}; \hat{\boldsymbol{\theta}}).$$

- $\hat{\boldsymbol{\theta}} = \{\hat{\mathbf{w}}^{(\alpha_1)}, \ldots, \hat{\mathbf{w}}^{(\alpha_K)}, \hat{\mathbf{f}}\}$.
- $\hat{\mathbf{w}}^{(\alpha_1)} = \{\hat{w}_1, \ldots, \hat{w}_d\}$ represents the hypothetical numerical distribution for categorical value α_1. d is used here to denote binning number, to distinguish K.
- $\hat{\mathbf{f}}$ represents the hypothetical frequency distribution for α.
- $\tilde{\boldsymbol{\omega}} = \{\tilde{\omega}^{(\alpha_1)}, \ldots\}$ represents the histograms for each kind of observed set.

By introducing Lagrange multipliers to enforce the sum-to-one constraints on these hypothetical inputs, we can apply our MR to merge PM components or FO components in estimation. It is not limited to the aforementioned category, numerical data types, and their combined key-value data types. Directional data [50], location data [44], and other similar data types, can also be subjected to analysis using a mixture model if a suitable approach to computing the posterior probability γ is available.

Time Complexity of Reduction procedure. The while loop (line 2 of Algorithm 1) runs as long as $K' > K_{\min}$. Our approaches discussed above will make the loop iterate approximately $\log K$ times. The EM algorithm typically has a time complexity of $O(nKI)$, where n is the number of observed data and I is the number of iterations until convergence. And the selection of components involves a sorting operation, and is typically $O(K' \log(K'))$. Thus, the overall time complexity is $O(K' \log(K') \log(K) + nK \log(K) I) = O(nK \log(K) I)$. In addition, protocols with a finite of perturbation values (e.g., GRR, SW, and PM) can reduce the complexity by aggregating the same observed report as $O(K^2 \log(K) I)$. For key-value protocols PCKV-PM, the number of bins d corresponds to the number of PM components. With K FO components, the total complexity is K times that of a single PM. Note that the original EM algorithm has a time complexity of $O(nKI)$, which seems to be smaller than ours. However, the reduction in the number of components also reduces the number of iterations required for convergence (I), ours is faster in practice. In the evaluation, we compared the efficiency (iteration times) and runtime between ours and the original EM.

Summary. Table I gives a summary of the methods discussed in this section. For category data, there exists a pre-processing step to transfer the noised reports to the probability of generated probability by each component. For numerical value, the pre-processing step is binning, and it can directly use the pdf for iteration.

VI. Evaluation

In this section, we present the demonstration of our reduction technique when applied to existing LDP protocols. We have selected mean estimation, frequency estimation, and numerical distribution estimation as three representative types of estimation tasks for evaluation purposes.

Firstly, regarding mean estimation, PM-MR, which applies our MR method to the noised reports of PM, consistently achieves the lowest Mean Absolute Error (MAE) in most scenarios (different datasets and ε). Notably, our approach demonstrates a 70% decrease in MAE, particularly in situations where the amount of noise is large (available data $n < 2000$ or privacy budget $\varepsilon < 2$). Through statistical analysis of the error in multiple mean results, we observe that EM-based MLE (e.g., PM-EM) reduces the estimation error by introducing bias compared to the unbiased estimation approach (e.g., PM and SR), whereas our MR remedies the issue of bias.

Secondly, in terms of frequency estimation, we combine the GRR and OLH methods with our framework, respectively. For both full-domain frequency estimation and frequent-value estimation, our MR achieves the lowest MAE when the amount of noise is substantial. When the amount of noise is small, MR is comparable to the existing post-processing method Normsub, and the MAE of the MR is lower than that of the EM by 20% to 50%.

Thirdly, in the context of numerical distribution estimation, we evaluate the performance of the Laplace, SW, and PM with both MR and EM. Our evaluation considers performance metrics, including the wasserstein distance, variance, range query, and quantiles. The metrics show the distribution obtained from MR reduces the error by 10% to 20% compared to EM. Additionally, key-value task is placed in Appendix D.

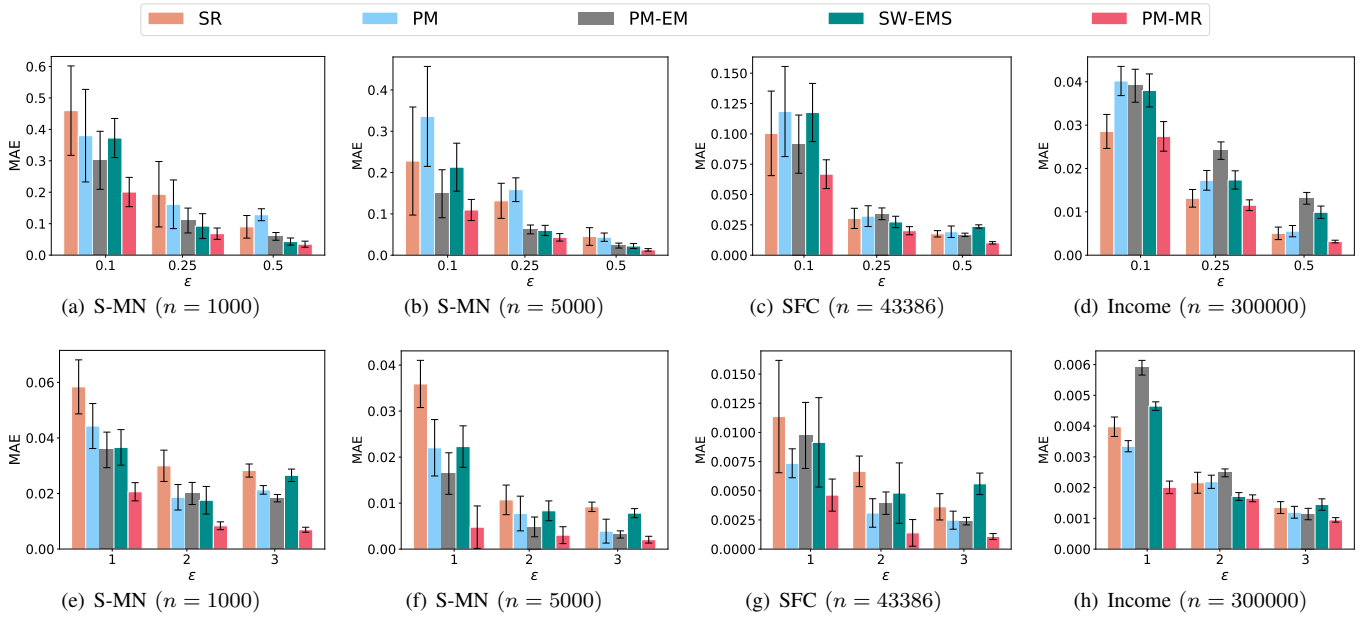
A. Experimental Setup

Datasets and Parameters. In the experiments, we use one synthetic and two real-world datasets, namely, S-MN, SFC, and Income. We normalize the domain of each numerical attribute into $[-1, 1]$.

- *Synthetic mixture of normal distributions (S-MN).* We synthesize two Normal distributions $N(0.7, 0.2^2)$ and $N(0.2, 0.1^2)$, and we form a mixture of these two different distributions with weights $(0.6, 0.4)$. For simulating scenarios of insufficient users, we generate two datasets of different sizes.
- *San Francisco Employee Compensation (SFC)* [42]. There are $43,386$ records in fiscal years 2019 and 2020. We use the "Total Compensation" as the private numeric data, and combine attributes of age group and region as category data.
- *Tax Stats (Income)* [27]. This dataset is publicly available on the website of the Internal Revenue Service. It contains income and tax data for about $300,000$ individuals. Here we use "total income" as the numerical value, and "ZIP code" as the categories for frequency estimation.

Perturbation Protocols. We consider these protocols introduced in Section II-B and Section II-C:

- SR and PM: Existing numerical protocols for mean estimation, SR performs better than PM when ε is small.



(a) S-MN ($n = 1000$) (b) S-MN ($n = 5000$) (c) SFC ($n = 43386$) (d) Income ($n = 300000$)

(e) S-MN ($n = 1000$) (f) S-MN ($n = 5000$) (g) SFC ($n = 43386$) (h) Income ($n = 300000$)

Fig. 2. MAE of mean estimation, varying privacy budget. (First row: low privacy budgets. Second row: large privacy budgets.)

- GRR and OLH: Existing FOs for frequency estimation, GRR performs better than OLH when $K < 3e^{\varepsilon} + 2$.
- SW and Laplace: Used for numerical distribution estimation.

Estimation Methods. We compare our estimation method with EM-based MLE as well as unbiased estimation and its post-processing methods. Here we use suffixes below to denote the corresponding estimation method:

- -BaseCut and -NormSub: Consistency post-processing methods introduced in [48], which revise negative results in frequency estimation result.
- -IIW: Recently proposed method with convolution framework to suppressing the added noise [23], we get their implementation from open source (https://github.com/SEUNICK/LDP).
- -EM and -EMS: EM-based MLE and its combining smoothing step version. The smoothing used in [34] averages the estimates between adjacent weights, represented by the formula $\hat{w}_i^{\text{new}} \leftarrow \frac{1}{4}\hat{w}_{i-1} + \frac{1}{2}\hat{w}_i + \frac{1}{4}\hat{w}_{i+1}$.
- -MR: Our proposed method. The implementation is available at https://github.com/yyt20080808/LDP-EM-MR.

In the setting of the EM algorithm for numerical protocols, we set the number of bins (or an initial number of components) to 1024 as in [34] and set the minimum component $K_{\min}$ in the mixture reduction step to 256. In the setting of frequency estimation, the $K_{\min}$ is also set to $K/4$, to prevent losing details of the distribution. In terms of reduction strategy, we assume that there exists no prior knowledge about the distribution of original data, and use the greedy operations that aim at merging the smallest weights. And τ is set to twice the standard deviation, analyzed and explained in Section V.

Evaluation Environment. The algorithms are implemented using Python 3.8 and Numpy 1.15, executed on a desktop computer equipped with an Intel Core i9-10910 CPU and 64GB of memory. For each dataset and each method, we report the average results over 100 runs.

B. Mean Estimation Accuracy

We compare our method (MR) with the below baseline methods in terms of MAE in this task. The unbiased estimation baselines are SR and PM, the sample mean of their LDP reports represent unbiased estimation. The EM-based MLE methods chosen are SW-EMS and our modified method PM-EM. SR-EM is not selected in this context because SR's LDP noise reports have only two output values, and the results obtained with the EM method are identical to unbiased estimation. Specifically, we conduct an evaluation of the mean on four different datasets by setting ε within the range of 0.1 to 3.0, as depicted in Figure 2. The first row displays the results of the low privacy budget scenario, while the second row illustrates the large privacy budget scenario.

Influence of ε and Size of Dataset. In an overall view of Figure 2, our PM-MR consistently achieves the smallest MAE across all values of ε. When the privacy budget is small, the MAE of MR is reduced by more than 70% compared to other methods. For example in Fig 2(f), the MAE of PM-MR at $\varepsilon = 1$ is 0.004, and its corresponding unbiased estimation method, PM, has an MAE of 0.022. When the privacy budget is larger, the accuracy gap between PM-MR and the unbiased estimation methods, PM, and SR, gradually close to the same level, but MR still outperforms PM-EM. Take the number of users into consideration, when the number of users is small and the privacy budget is small (Fig 2(a) and Fig 2(b)), the EM-based MLE methods (PM-EM, SW-EMS, PM-MR) have overall advantages over unbiased approaches. As the number of users increases and the privacy budget increases, the error of unbiased estimation results decreases more significantly,

becoming gradually better than EM-based MLE. In Fig 2(h), when $\varepsilon = 3$, the MAE of MR, PM-MR, and PM are basically at the same level because the noise is minimal.

Comparing Methods in Terms of Bias. Figure 3 illustrates the distribution of ERRORs for each estimation method. We plot the error (denoted as ERROR, $\mu - \hat{\mu}$) distribution for the 100 runs estimated by each methods, as the box plot. The horizontal green line at ERROR = 0 represents the baseline. If a method's blue square (mean of the errors) is close to the baseline, it indicates that the estimation of this method tends to be unbiased. A large distance between the bottom and top of the box indicates a large variance, and vice versa. In short, a small gray range and a blue square close to 0 suggest a small overall error.

Based on Figure 3, it is evident that the mean estimates derived through the mixture model (SW-EMS and PM-EM) tend to be biased, as indicated by their blue squares being significantly distant from zero. And the blue squares for the unbiased estimates (PM and SR) are very close to the 0-baseline. Comparing Fig 3(c) with Fig 3(d), an increase in privacy budget reduces the variance of all methods and reduces the bias of EM-based MLEs. Then, as demonstrated in Fig 3(a), in the setting of smaller n and ε, and the bias is not as noticeable when compared to the variance for EM-based MLE. Thus, the whole error of EM-based MLE method is comparatively smaller than unbiased approach (SR or PM), which agrees with the observations in Figure 2.

At last, in all sub-figures, we observe that MR has less bias than the methods using -EM and -EMS and less variance than the unbiased estimation, suggesting that the estimate are more accurate.

C. Frequency estimation Accuracy

The unbiased estimation baselines for this task are GRR and OLH. Additionally, we employ post-processing methods Normsub, Basecut as well as IIW to enhance the estimations, respectively. The EM-based MLE baselines include GRR-EM and our modified method OLH-EM. Our methods are OLH-MR and GRR-MR. We calculate the MAEs for both the full-domain frequency estimation tasks and frequent-value frequency estimation tasks and set ε to $\{0.5, 1, 2\}$.

Full Domain Accuracy. Fig 4(a)(b)(e)(f) compare the full-domain accuracy of our MR with other methods. Our proposed GRR-MR and OLH-MR consistently outperform the baselines in most scenarios. In comparison to the EM method, MAE is reduced by $10\% \sim 30\%$, and in comparison to unbiased estimation methods, MAE is reduced by approximately 50%. Unbiased estimation shows the highest MAE, as full-domain estimation considers the error across the entire distribution, and the unbiased approach is likely to get an unreasonable distribution. Additionally, because adjacent frequency values do not exhibit enough smoothness in these two real datasets, the -IIW method does not demonstrate its advantages, which is consistent with its paper's conclusion. In Fig 4(b) and Fig 4(f) where there is a larger number of individuals and candidate items (Income dataset with $K = 300$ and $n = 300,000$), our MR exhibits a reduction in MAE ranging from 20% to 30% compared to EM. The reason is that a larger number of parameters ($K = 300$) increases the risk of overfitting to noise, while our reduction can decrease the number of parameters in EM. This is consistent with our Proposition 1.

Frequent-value Accuracy. Fig 4(c)(d)(g)(h) show the comparison on only frequent values. Here, we take the estimates of the top 10% of candidates with the largest actual frequency values for MAE comparison. In all four subfigures, when $\varepsilon \leq 1$, our MR performs best, consistent with the full domain conclusion. However, when the ε is large ($= 2$), the unbiased estimation methods of OLH and GRR are the ones with the minimum MAE, followed by our MR. The reason stated in [23] is that the post-processing method would actually amplify noise to the high frequent values. And this is the common limitation of post-processing methods on unbiased estimation and ours.

D. Numerical Distribution Accuracy

Metric. We measure the distance between two density distributions, denoted as $\mathbf{w}$ and $\hat{\mathbf{w}}$. Here we use Wasserstein distance (WD) because it is the cost of moving the probability mass (or density) from one distribution to another distribution. In this paper, our estimated numerical distribution is discrete. So for a cumulative function $\mathbf{P}: [0,1]^d \times \mathcal{B} \rightarrow [0,1]^d$ that takes a distribution $\mathbf{w}$ and a value β, and output $\mathbf{P}(\mathbf{w}, \beta) = \sum_{i=1}^{\beta} \mathbf{w}_\beta$, the ℓ_1 loss of Wasserstein distance is

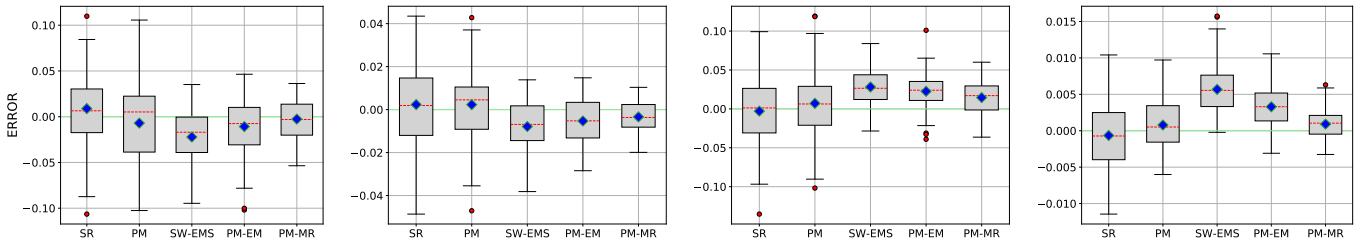
$$\mathrm{W}_1(\mathbf{w}, \hat{\mathbf{w}}) = \sum_{\beta \in \mathcal{B}} |\mathbf{P}(\mathbf{w}, \beta) - \mathbf{P}(\hat{\mathbf{w}}, \beta)|.$$

Then we also measure the variance (Var), quantiles (Quan), and range query (RQ) between the estimated distribution and real distribution.

Numerical Distribution. The Figure 5 shows the comparison of errors between using EM and MR for distribution estimation, under the perturbed data of three protocols (the Laplace method, PM, SW method). Overall, employing MR for distribution estimation results in a noticeable reduction of errors, exceeding 10% across multiple metrics compared to EM. The results are consistent and stable across two datasets with varying numbers of users and distributions. Among the four metrics, the most significant reduction in error is observed in the variance, indicating that MR's estimation is more accurate in identifying the concentration of the distribution. Regarding the influence of different protocols, SW and PM belong to the same type of perturbation protocol, leading to similar estimation results with MR. Since Laplace method is not designed for distribution estimation task, its errors remain higher, whether using EM or MR.

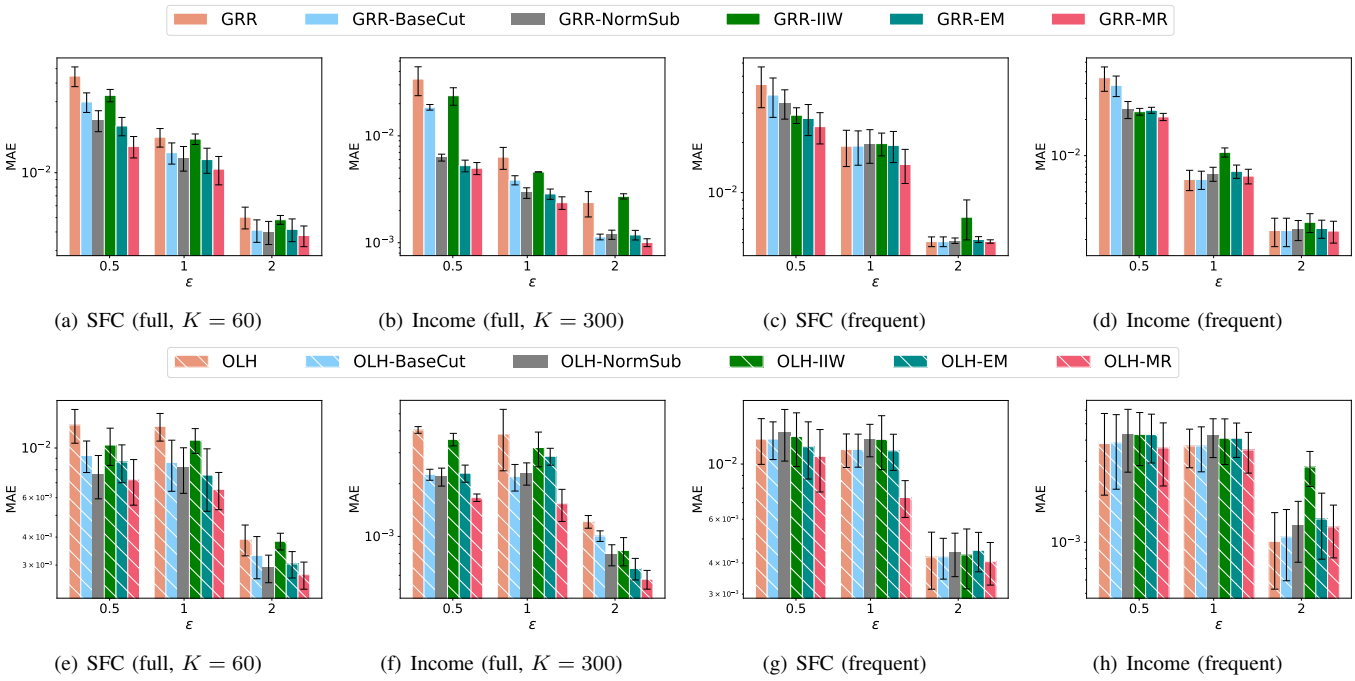
E. Efficiency comparison between MR and EM

We empirically conduct a comparative analysis of the convergence behavior and execution time between the original EM algorithm and the MR framework using two real-world



(a) S-MN ($n = 1000, \varepsilon = 1.5$) (b) S-MN ($n = 5000, \varepsilon = 1.5$) (c) SFC ($n = 43386, \varepsilon = 0.25$) (d) SFC ($n = 43386, \varepsilon = 2.5$)

Fig. 3. Box plots of mean estimation errors for all methods. (a) and (b) are S-MN datasets with different user populations, and their privacy budgets are the same ($\varepsilon = 1.5$). (c) and (d) are conducted on SFC with different ε. The bottom and top of the box represent the first and third quartiles, respectively. The red line inside the box represents the median, while the blue square represents the mean. And red circles are outliers.



(a) SFC (full, $K = 60$) (b) Income (full, $K = 300$) (c) SFC (frequent) (d) Income (frequent)

(e) SFC (full, $K = 60$) (f) Income (full, $K = 300$) (g) SFC (frequent) (h) Income (frequent)

Fig. 4. MAE of frequency estimation results, for full domain and frequent-value estimation on S-MN and Income datasets, varying ε.

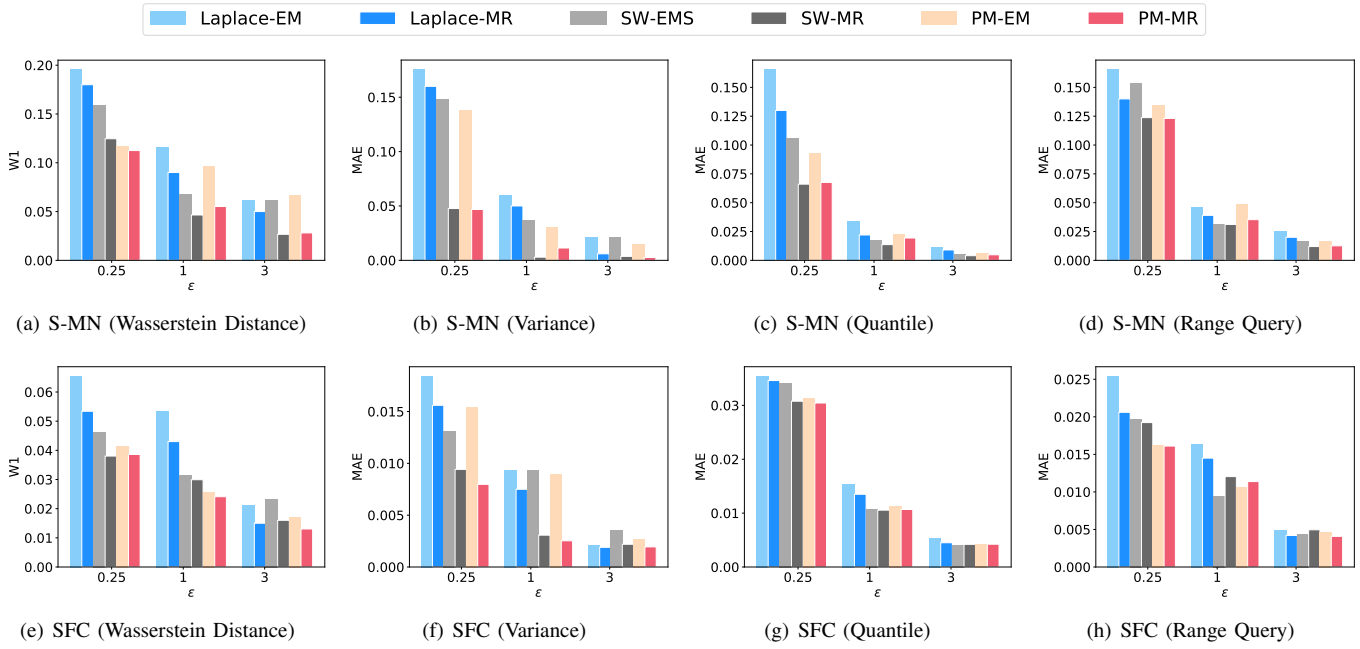
datasets, as shown in Figure 6 and Table II. We focus on the full-domain frequency estimation and track their MAEs during successive EM iterations. Convergence is deemed attained when the error curve stabilizes and assumes a horizontal trajectory. Due to space issues, we only present the convergence analysis for the OLH-EM and OLH-MR methods because they have significantly longer run times.

In Fig 6(a) and Fig 6(b), we notice that a larger value of ε leads to fewer iterations for convergence. For example, with $\varepsilon = 0.75$, the line of OLH-EM requires over 8000 iterations to stabilize, whereas in Fig 6(b) ($\varepsilon = 3$), only 1000 iterations suffice. Furthermore, in cases where the reduction step is effective, as seen in Fig 6(c) (4000 iterations), OLH-MR achieves quicker convergence compared to OLH-EM (e.g., 8000 iterations versus 20000). This acceleration is attributed to the merging of components in EM, which reduces the model complexity. Then, in the Income dataset, the OLH-EM requires a matrix of size 300000×300 ($n \times K$) to store the transformation (perturbation) probabilities, while GRR-EM only needs 300×300 (K^2). Each iteration of EM will use this matrix once, explaining the significantly slower runtime of OLH-EM compared to GRR-EM, as illustrated in Table II.

In addition, for the numerical protocols including PM, SW, and Laplace, we find PM and SW are quite similar to GRR, because their outputs are all bounded and belong to the class of random response. So the running time difference of PM and SW with EM and MR is similar to that of GRR with EM and MR. And for the Laplace mechanism, since Laplace's output is unbounded, we should calculate every sample's posterior probability for each component. And Laplace-MR only takes 30% time compared to Laplace-EM.

VII. Discussions

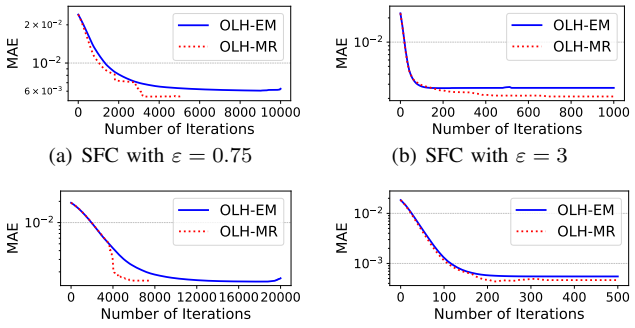
When to Use Our Method. At the high level, when the input domain is large (e.g., many values need to be estimated) or there exists substantial noise (low ε or small n), our method often yields more accurate results than EM. For more precise guidance for various settings (different ε, user counts n, and



(a) S-MN (Wasserstein Distance) (b) S-MN (Variance) (c) S-MN (Quantile) (d) S-MN (Range Query)

(e) SFC (Wasserstein Distance) (f) SFC (Variance) (g) SFC (Quantile) (h) SFC (Range Query)

Fig. 5. Comparing EM, MR method for numerical distribution, varying ε.



(a) SFC with $\varepsilon = 0.75$ (b) SFC with $\varepsilon = 3$

(c) Income with $\varepsilon = 0.75$ (d) Income with $\varepsilon = 3$

Fig. 6. The convergence of -EM and -MR approach on the SFC and Income dataset, varying ε.

TABLE II
RUNTIME TABLE (SECONDS) OF -EM AND -MR ON DIFFERENT DATASETS, VARYING ε.

	method	ε			
		0.75	1	2	3
SFC	GRR-EM	19	12	9	6
	GRR-MR	10	5	4	4
	OLH-EM	765	502	115	58
	OLH-MR	311	204	83	37
	Laplace-EM	2317	931	416	125
	Laplace-MR	1156	665	306	90
Income	GRR-EM	23	17	12	7
	GRR-MR	11	8	5	4
	OLH-EM	15684	6482	1126	154
	OLH-MR	2837	1697	279	67
	Laplace-EM	12317	8152	2516	823
	Laplace-MR	5457	3003	1026	412

data distributions). We suggest a bootstrap-based heuristic approach: (1) The collector can generate a synthetic dataset based on estimations using our method. (2) Simulate the whole process to compare different methods. And finally (3) select the best inference method for statistical tasks.

Employing Our Method in the Shuffle Model. Shuffle DP [9], [13] provides a level of privacy and data utility that lies between DP and LDP. Specifically, each user first applies LDP perturbation to their own data, encrypts it, and sends it to a shuffler. The shuffler then shuffles the data to break the correspondence between user IDs and data, and sends the shuffled result to the server. The server decrypts the data and performs the analysis. The inclusion of the shuffler provides a privacy amplification effect for the task of LDP collection [12]. Our MR estimation framework is applied directly on the server side, enabling the direct enjoyment of privacy amplification benefits, particularly in scenarios with small sample sizes.

Limitations. Our framework has only been validated through simulations on real-world datasets, limiting practical evaluation. The use of the EM algorithm with LDP protocols that produce unbounded outputs (e.g., OLH, Laplace) leads to higher time complexity for large datasets (n) or high-dimensional data (K) compared to matrix-based methods, reducing efficiency in real-time applications. Additionally, we do not account for prior knowledge of the data distribution to enhance performance. For instance, when handling datasets with smooth distributions, methods such as IIW [23] or EMS [34] may yield better results than ours.

VIII. Related Work

Differential privacy [20] is a strong privacy standard that provides semantic, information-theoretic guarantees on individuals' privacy. So far, most existing works focus on the centralized setting, i.e., they assume there exists a trusted data curator who collects and possesses the private genuine information of individuals. As for the local setting, i.e., there

is no such data curator. Kasiviswanathan et al. [31] systematically investigate the framework of local differential privacy and connect it to the classical randomized response technique [49], which is now the foundation.

Categorical Frequency Oracle. The building block of categorical data collection is the frequency oracle, where each user possesses a categorical value, and the aggregator aims to estimate the frequency of all values within the domain. There have been several methods [22], [30], [47], [6], [52], [46], [5], [1] handling this tasks. To further improve the utility, one can apply post-processing calibration algorithms to revise the frequency results, most of which are based on consistency [22], [48], some smoothness assumptions and prior knowledge [23], [28]. In addition, the EM-based MLE [35], [21], [3] serves as an alternative method capable of identifying a reasonable distribution that is most likely to generate the observed FO results. We equipped our framework on the SOTA FO mechanisms (GRR [30] and OLH [47]) for analysis.

Numerical Data Collection. Recent work on numerical data collection under LDP mainly focuses on two tasks, mean estimation [4] and distribution estimation. In [19], the authors propose the stochastic rounding (SR) technique to estimate the mean. Wang et al. [45] propose the piecewise mechanism (PM) for mean estimation. In [34], the authors propose the square wave mechanism and explore the use of EM for distribution estimation. These three methods have been discussed in Section II-C and compared in the experiments.

Over-fitting Issues in EM. Over-fitting occurs when the model learns to fit the training data too well, capturing noise or random fluctuations in the data rather than the underlying patterns. Reducing model complexity [11], [40] and regularization [53], [37] are two major ways to prevent overfitting in the EM algorithm for GMM. In the context of LDP, previous work [35] has explored the use of the EM algorithm in GRR, and states that the consistency property makes this method suitable for small sample situations. They also claimed that the EM results need a correction based on Rilstone et al. [51], which also mitigates the EM overfitting problem. In addition, the correction step requires an inverse matrix G^{-1}, which requires the input alphabet size to be equal to the output alphabet size. Li et al. [34] proposed to use smoothing to solve the overfitting problem, and can be applied to numerical settings. In our paper, we also discuss the use of the EM algorithm in hash-based protocols whose output alphabet size is not constrained, and adopt the reduction method for alleviating the overfitting issue.

IX. Conclusions

Our work introduces a reduction framework to address overfitting in maximum likelihood estimation (MLE) for Local Differential Privacy (LDP) data. Evaluations using synthetic and real-world datasets reveal that MLE with reduction can outperform unbiased estimation methods in fundamental LDP tasks, especially in scenarios with limited data and privacy budgets.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by National Key R&D Program of China (2022YFB4501500, 2022YFB4501503). This work was supported by Ant Group. Wang participated in discussion and writing in his personal capacity and did not receive any support from the above-mentioned funding.

References

[1] J. Acharya, Z. Sun, and H. Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. *AISTATS*, 2018.

[2] Apple. Apple differential privacy team, learning with privacy at scale, 2017. Available at https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf.

[3] H. H. Arcolezi, S. Cerna, and C. Palamidessi. On the utility gain of iterative bayesian update for locally differentially private mechanisms. In *Data and Applications Security and Privacy XXXVII*, pages 165–183. Springer Nature Switzerland, 2023.

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

[5] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta. Practical locally private heavy hitters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2285–2293, 2017.

[6] R. Bassily and A. D. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 127–135. ACM, 2015.

[7] J. A. Bilmes. Gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture. *International Computer Science Institute*, 1998.

[8] J. A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International computer science institute*, 4(510):126, 1998.

[9] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, pages 441–459. ACM, 2017.

[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)*, pages 947–964. USENIX Association, Aug. 2021.

[11] H. Chen, K. Chang, and C. Smith. Constraint optimized weight adaptation for gaussian mixture reduction. In *Signal Processing, Sensor Fusion, and Target Recognition XIX*, volume 7697, pages 281–290. SPIE, 2010.

[12] A. Cheu. Differential privacy in the shuffle model: A survey of separations. *arXiv preprint arXiv:2107.11839*, 2021.

[13] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *38th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2019)*, pages 375–403. Springer, 2019.

[14] G. Cormode, S. Maddock, and C. Maple. Frequency estimation under local differential privacy. *Proc. VLDB Endow.*, 14(11):2046–2058, jul 2021.

[15] T. Cunningham, G. Cormode, H. Ferhatosmanoglu, and D. Srivastava. Real-world trajectory sharing with local differential privacy. *Proc. VLDB Endow.*, 14(11):2283–2295, Jul 2021.

[16] A. P. Dempster. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39, 1977.

[17] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *NIPS 30*, December 2017.

[18] 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*, pages 429–438. IEEE, 2013.

[19] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.
[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284. Springer Berlin Heidelberg, 2006.
[21] E. ElSalamouny and C. Palamidessi. Generalized iterative bayesian update and applications to mechanisms for privacy protection. In *2020 IEEE European Symposium on Security and Privacy (EuroSP)*, pages 490–507. IEEE, 2020.
[22] 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*, pages 1054–1067, 2014.
[23] H. Fang, L. Chen, Y. Liu, and Y. Gao. Locally differentially private frequency estimation based on convolution framework. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2208–2222. IEEE, 2023.
[24] X. Gu, M. Li, Y. Cheng, L. Xiong, and Y. Cao. PCKV: Locally differentially private correlated key-value data collection with optimized utility. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 967–984. USENIX Association, Aug. 2020.
[25] J. Imola, A. R. Chowdhury, and K. Chaudhuri. Robustness of locally differentially private graph analysis against poisoning. *arXiv preprint arXiv:2210.14376*, 2022.
[26] J. Imola, T. Murakami, and K. Chaudhuri. Locally differentially private analysis of graph statistics. In *30th USENIX Security Symposium*, pages 983–1000, 2021.
[27] IRS. Statistics of income. [EB/OL]. https://www.irs.gov/statistics.
[28] J. Jia and N. Z. Gong. Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2008–2016, 2019.
[29] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. *Advances in neural information processing systems*, 29, 2016.
[30] P. Kairouz, S. Oh, and P. Viswanath. Extremal mechanisms for local differential privacy. *J. Mach. Learn. Res.*, 17(1):492–542, jan 2016.
[31] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008.
[32] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.
[33] X. Li, Z. Li, N. Li, and W. Sun. On the robustness of ldp protocols for numerical attributes under data poisoning attacks. *https://arxiv.org/abs/2403.19510*, 2024.
[34] Z. Li, T. Wang, M. Lopuhaä-Zwakenberg, N. Li, and B. Škoric. Estimating numerical distributions under local differential privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 621–635. ACM, 2020.
[35] T. Murakami, H. Hino, and J. Sakuma. Toward distribution estimation under local differential privacy with small samples. *Proc. Priv. Enhancing Technol.*, 2018(3):84–104, 2018.
[36] A. A. Neath and J. E. Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.
[37] D. T. Phan and T. Idé. *L0-Regularized Sparsity for Probabilistic Mixture Models*, pages 172–180. SIAM, 2019.
[38] D. A. Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.
[39] P. Rilstone, V. Srivastava, and A. Ullah. The second-order bias and mean squared error of nonlinear estimators. *Journal of Econometrics*, 75(2):369–395, 1996.
[40] A. R. Runnalls. Kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, 2007.
[41] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.
[42] S.F.C.Office. Sf employee compensation. [EB/OL]. https://www.kaggle.com/san-francisco/sf-employee-compensation/.
[43] P. Stoica and A. Nehorai. Music, maximum likelihood, and cramer-rao bound: further results and comparisons. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(12):2140–2150, 1990.
[44] H. Wang, H. Hong, L. Xiong, Z. Qin, and Y. Hong. L-srr: Local differential privacy for location-based services with staircase randomized response. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 2809–2823. ACM, 2022.
[45] 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)*, pages 638–649, April 2019.
[46] 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, 2016.
[47] T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *25th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.
[48] T. Wang, M. Lopuha-Zwakenberg, Z. Li, B. Skoric, and N. Li. Locally differentially private frequency estimation with consistency. In *Network and Distributed System Security Symposium(NDSS)*, 2020.
[49] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
[50] B. Weggenmann and F. Kerschbaum. Differential privacy for directional data. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 1205–1222. ACM, 2021.
[51] L. Xu and M. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.
[52] M. Ye and A. Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. *IEEE Transactions on Information Theory*, 64(8):5662–5676, 2018.
[53] Y. Zhao, A. K. Shrivastava, and K. L. Tsui. Regularized gaussian mixture model for high-dimensional clustering. *IEEE Transactions on Cybernetics*, 49(10):3677–3688, 2019.
[54] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal*, 8(11):8836–8853, 2021.
[55] X. Zhu, V. Y. F. Tan, and X. Xiao. Blink: Link local differential privacy in graph neural networks via bayesian estimation. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 2651–2664. ACM, 2023.

APPENDIX A
DERIVATION OF THE EM ALGORITHM

First of all, the LDP mixture model is defined as Equation 9, where

- $\tilde{x}_i$ is the i-th noised report.
- $\theta = \{\hat{w}_k, \alpha_k\}_{k=1}^{K}$ are the parameters for the model.
- K is the number of components.
- $g(\tilde{x}; \alpha_k)$ is the pdf or pmf ($\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_k) = \tilde{x}_i\right]$) of perturbation that generates $\tilde{x}$.

The E-step is responsible for calculating the posterior probabilities of each data point belonging to each LDP component. Let γ_{ik} denote the posterior probability that data point $\tilde{x}_i$ belongs to the k-th LDP component. We use these probabilities to obtain a lower bound on the log-likelihood function. By introducing Lagrange multipliers to enforce the sum-to-one constraint on γ_{ik}, we can express the lower bound as follows:

$$J(\theta, \theta^{(t)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \left[\log \hat{w}_k + \log g(\tilde{x}_i; \alpha_k)\right]$$

where $\theta^{(t)}$ represents the current parameter estimates at iteration t. To find γ_{ik}, we use the responsibility formula, which represents the probability of the k-th LDP component generating data point $\tilde{x}_i$:

$$\gamma_{ik} = \frac{\hat{w}_k \cdot g(\tilde{x}_i; \alpha_k)}{\sum_{j=1}^{K} \hat{w}_j \cdot g(\tilde{x}_i; \alpha_j)}$$

The M-step is responsible for updating the parameters based on the posterior probabilities calculated in the E-step. Here we only consider $\hat{w}_k$, and set the derivative of J with respect to $\hat{w}_k$ to zero and solve for $\hat{w}_k$:

$$\hat{w}_k^{new} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}$$

The EM algorithm alternates between the E-step and the M-step until convergence. And the following proof shows that algorithm converges to MLE.

Proof. To prove the EM algorithm converges to the maximum likelihood estimator, it is enough to show the loglikelihood function, Equation 10, is concave [8]. As the second partial derivative of this function is Equation (12), where the term $\mathsf{Pr}\left[\Psi_\varepsilon(\alpha_j) = \tilde{x}_i\right]$ is positive and fixed, and the value of derivatives is always negative. Thus, $\mathcal{L}(\hat{\mathbf{w}})$ is concave function. □

APPENDIX B
PROOF OF THE PROPOSITION 1

To derive the overall error, we first need the lemma below to show the MSE of the original EM algorithm on LDP estimations.

Lemma 1 (from [35])**.** *According to the theory of Rilstone et al. [39], the mean squared error of EM-based MLE is*

$$\mathsf{MSE}_{\mathrm{EM}} = \mathbb{E}[||(\frac{1}{n}\sum_{i=1}^{n}(-\mathbf{Q}^{-1}\frac{1}{\hat{\mathbf{w}}^T\mathbf{g}_i}\mathbf{g}_i)||_2^2] + O(n^{-3/2}).$$

where $\mathbf{g}_i = (\Pr[\Psi(\alpha_1) = \tilde{x}_i], \ldots, \Pr[\Psi(\alpha_K) = \tilde{x}_i])^T \in \mathbb{R}^{K\times 1}$ *is the likelihood vector of noise sample* $\tilde{x}_i$ *for each model in the mixture, and it is constant,* $\mathbf{Q} = \mathbb{E}[-\frac{1}{n}\sum_{i=1}\frac{1}{(\hat{\mathbf{w}}^T\mathbf{g}_i)^2}\mathbf{g}_i\mathbf{g}_i^T] \in \mathbb{R}^{K\times K}$ *is the expectation of the second derivative of the log-likelihood function.*

Since the log-likelihood $\log\mathcal{L}(\hat{\mathbf{w}})$ (see Equation (10)) can be written by matrix like $\sum_{i=1}^{n}\log\hat{\mathbf{w}}\mathbf{g}_i$, the term $\frac{1}{\hat{\mathbf{w}}^T\mathbf{g}_i}\mathbf{g}_i$ is the first derivative of log-likelihood function ($= \nabla\mathcal{L}(\hat{\mathbf{w}}; \tilde{x}_i)$). $-\mathbf{Q}^{-1}$ is the inverse of Fisher information matrix, which provides the lower bound of the covariance matrix (i.e., Crámer-Rao inequality [43]). Therefore, the $\mathsf{MSE}_{\mathrm{EM}}$ is influenced by ε (the value in $\mathbf{g}$) and n.

Then, for our proposed Algorithm 1, the mean squared errors incurred in estimation are combined with two parts: (1) the errors of the EM algorithm executed, which provide the estimations for the weights of remaining components (denoted by $\mathbf{w}_\mathrm{r}$), and (2) the errors incurs for the merged weights (denoted by $\mathbf{w} \setminus \mathbf{w}_\mathrm{r}$). Thus,

$$\begin{aligned}\mathsf{MSE}_{\mathsf{Ours}} &= \frac{1}{K}(\mathbb{E}[||\hat{\mathbf{w}} - \mathbf{w}||_2^2]) = \frac{1}{K}\mathbb{E}[\sum_{i=1}^{K}(\hat{w}_i - w_i)^2] \\ &= \frac{1}{K}\sum_{w_i\in\mathbf{w}_\mathrm{r}}\mathbb{E}[(\hat{w}_i - w_i)^2] + \\ &+ \frac{1}{K}\sum_{w_j\in\mathbf{w}\setminus\mathbf{w}_\mathrm{r}}\mathbb{E}[(\hat{w}_j - w_j)^2]\end{aligned} \tag{16}$$

Step 1. MSE **of Remaining Weights.**

Let the number of remaining components be K_r. Based on Lemma 1, the first term in Equation (16) equals to

$$\mathbb{E}[\sum_{w_k\in\mathbf{w}_\mathrm{r}}(\frac{1}{n}\sum_{i=1}^{n} -Q_k^{-1}\frac{1}{(\hat{\mathbf{w}}^T\mathbf{g}_i)}\mathbf{g}_{ik})^2] + O(n^{-\frac{3}{2}}).$$

When K is large or ε is small (values in $\mathbf{g}_i$ is almost uniformly distributed), the original EM algorithm has almost the same MSE for each $\hat{w}$, so the first term approximates $\frac{K_r}{K}\mathsf{MSE}_{\mathrm{EM}}$.

Step 2. MSE **of Merged Weights.**

Consider that there exists a single merging step that combines a set of mixture components, denoted by S, with size h_m, into a single component. The corresponding merged weight, denoted by $\hat{w}_{\mathsf{new}}$, is given by $\hat{w}_{\mathsf{new}} = \sum_{w_j\in S}\hat{w}_j$. The $\hat{w}_j$ is asymptotically unbiased, $\mathbb{E}[\hat{w}_i] = w_i$ if $n \to \infty$. And we have $\mathbb{E}[\hat{w}_{\mathsf{new}}] = \sum w_j$, $\mathsf{Var}\left[\hat{w}_{\mathsf{new}}\right] = \mathsf{Var}\left[\sum_{w_i\in S}\hat{w}_i\right] = \sum_{w_i\in S}\mathsf{Var}\left[\hat{w}_i\right] + 2\sum_{w_i,w_j\in S}\mathbf{Cov}(\hat{w}_i, \hat{w}_j)$. For most $\hat{w}_i$ and $\hat{w}_j$, they are negatively correlated because the sum of all $\hat{w}_i$ is constrained to be 1, resulting in a negative covariance term. So $\mathsf{Var}\left[\hat{w}_{\mathsf{new}}\right] \le \sum_{w_i\in S}\mathsf{Var}\left[\hat{w}_i\right]$.

Denote the variance of $\hat{w}_{\mathsf{new}}$ as σ_m^2. In the final estimation results, our algorithm assigns the value $\bar{w}_{\mathsf{new}} = \frac{\hat{w}_{\mathsf{new}}}{h_\mathrm{m}}$ to those merged weights. Thus, the MSE for the merged components depends on the error of the assigned $\bar{w}$. And its variance is $\mathrm{Var}[\bar{w}_{\mathsf{new}}] = \mathrm{Var}\left[\frac{1}{h_\mathrm{m}}\hat{w}_{\mathsf{new}}\right] = \frac{\sigma_\mathrm{m}^2}{h_\mathrm{m}^2}$.

Based on MSE, which is the combination of variance and bias: $\mathbb{E}[(\hat{w}_j - w_j)^2] = \mathrm{Var}(\hat{w}_j) + (\mathbb{E}[\hat{w}_j] - w_j)^2$. We can write

$$\mathsf{MSE}(\hat{w}_j) = \frac{\sigma_\mathrm{m}^2}{h_\mathrm{m}^2} + (\frac{\sum_{w_i\in S} w_i}{h_\mathrm{m}} - w_j)^2$$

Therefore, the overall error of merged components

$$\begin{aligned}\sum_{w_j\in S}\mathbb{E}[(\hat{w}_j - w_j)^2] &= \sum_{w_j\in S}(\frac{\sigma_\mathrm{m}^2}{K_\mathrm{m}^2} + (\frac{1}{h_\mathrm{m}}\sum_{w_k\in S} w_k - w_j)^2) \\ &= \frac{\sigma_\mathrm{m}^2}{h_\mathrm{m}} + \frac{1}{h_\mathrm{m}}\sum(\overline{w} - w_i)^2 \\ &= \frac{\sigma_\mathrm{m}^2}{h_\mathrm{m}} + \mathrm{Var}[w_j]\end{aligned}$$

The first term approximates the variance of $\hat{w}_i, w_i \in S$, and we can also approximate it to the MSE of $\hat{w}_i$ because MSE is larger than the variance. The final number of components is K', which includes K_r remaining components and the newly generated components. After one merging operation, $K' = K_\mathrm{r} + 1$. The second term of the above equation represents the variance of the true distribution of the values of the merged components, denoted by $\mathrm{Var}(w) = \sigma^2$. In the end, we get:

$$\mathsf{MSE}_{\mathsf{Ours}} = \frac{K'}{K}\mathsf{MSE}_{\mathrm{EM}} + \frac{h_\mathrm{m}}{K}\sigma^2$$

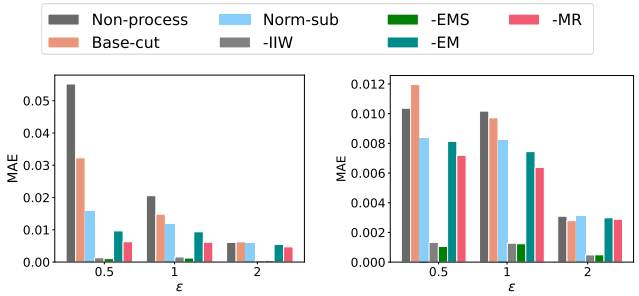
To generalize the given formula to a scenario with a total of t merging times, where each merging time involves merging

$h_1, \ldots, h_t$ mixture components, we can revise the equation as follows:

$$\mathsf{MSE}_{\mathsf{Ours}} = \frac{K'}{K}\mathsf{MSE}_{\mathsf{EM}} + \frac{1}{K}\sum_{i=1}^{t} h_i \sigma_i^2$$

Note that the $\mathsf{MSE}_{\mathsf{EM}}$ in the above equation is the value corresponding to the remaining components after updating the model parameters as described in Lemma 1. Actually, the contribution of each observed data to the remaining components remains unchanged during the merging operation. When the true values of the selected components' weights are small, the covariance (non-diagonal values in $\mathbf{Q}$) between remaining component and merged component is negligible, we assume that the MSE of the updated model is approximately equal to the MSE of the original EM model.

Appendix C
Additional Experiments



(a) Uniform (GRR) (b) Uniform (OLH)

Fig. 7. MAE of frequency estimates, for domain size of $K = 100$ in uniform dataset, varying ε.

Worst-case Scenario. The worst-case scenario refers to the situation where the original dataset follows a relatively uniform distribution, which is considered in the minimax analysis of existing solutions [52]. Figure 7 empirical compare the accuracy of different post-processing methods with EM-based MLEs on a uniform dataset ($n = 50000$, $K = 100$, and for all $f_\alpha \approx 0.01$). We can see that our estimation method, MR, achieves lower MAE than EM on both GRR and OLH protocols. And EM-based MLE is better than unbiased estimations (Non-process in figure). In addition, because the uniform dataset is definitely smoothing, -IIW performs as well as the -EMS (EM-based MLE with smoothing technique in M-step).

Bias Comparison of the Frequency Estimation Task. Figure 8 demonstrates the comparison between the true distribution and the estimated distributions on two datasets. In order to decrease random errors, we use the average of the three estimates of each estimation to judge its unbiasedness property. It can be found that the distribution of EM has a bias compared to the true distribution. There is a negative bias for high-frequent values and a positive bias for low-frequent values, whereas MR reduces the amount of bias introduction to some extent compared to EM (the red line is closer to the black line). The reason for the bias introduction is the normalization and non-negative nature attached to EM-based MLE methods. As can be seen from the subplot comparisons (e.g., Fig 8(a) and Fig 8(c)), the amount of bias is related to the noise introduced by the protocol (OLH-MR has less bias than GRR-MR when $K > 3e^{\varepsilon} + 2$).

Impact of Data Sizes in Numerical Distribution. Figure 9 illustrates the comparison between estimated distributions of -MR and -EM across different data sizes. As the data size increases from $1,000$ to $300,000$, the advantage of -MR over -EM in the four metrics decreases. Notably, at $n = 300,000$, the estimation results for both PM and SW perturbed data are similar between EM and MR. This is because, as n grows, the amount of noise becomes smaller, and the number of noise-dominant components selected by our MR algorithm also decreases, aligning the overall estimation results more closely with EM.

Appendix D
EM-based MLE for Key-Value Data

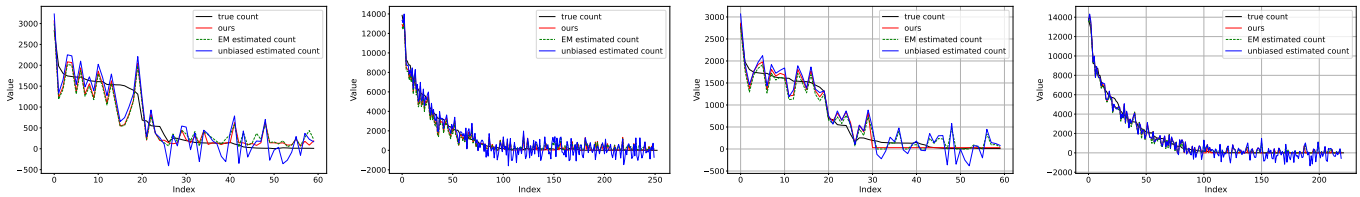
PCKV. This is the state-of-the-art method on key-value data collection for conditional estimation. The unary encoding version of PCKV encodes a key-value pair to a length d vector where the k-th position is $(1, v)$. Then it perturbs the key and value in a correlated manner, each with ϵ_1 and ϵ_2. For example, if the value of the key is unchanged after OUE perturbation, then use SR to perturb the real value v; otherwise, randomly select a value in the output domain. In this evaluation, we replace its constituent module SR with PM, and apply our LDP mixture model to get the conditional density and mean. In this task, for each "key", only a portion of values are real and valid. The accuracy of conditional estimation is bounded by $\mathcal{O}\left(\frac{1}{f^2}\right)$, which means there exists significant noise.

Conditional Estimation Accuracy. We use the dataset SFC for evaluation, and treat the "Total Compensation" attribute as the private numeric data, and "Age groups" as the keys. Here we select the 4 different frequent candidates in SFC, and evaluate the mean squared error (MSE) of their conditional mean estimates. The Figure 10 shows the results for existing methods PCKV-OUE and PCKV-GRR [24], and our method PCKV-EM and PCKV-MR (described in Section V-C).

When ε is set to 0.25 or 0.5, the MSE of conditional mean on low frequent key (0.03, 0.06) is too much. Considering the overall domain size is just 2, the value of MSE that is up to 0.2 (almost 0.25 in Fig 10(a) and Fig 10(b)) shows that the estimated values are significantly different from the true value. But it can be reduced to half using a mixture model (the MSE of PCKV-MR is only a half compared to others).

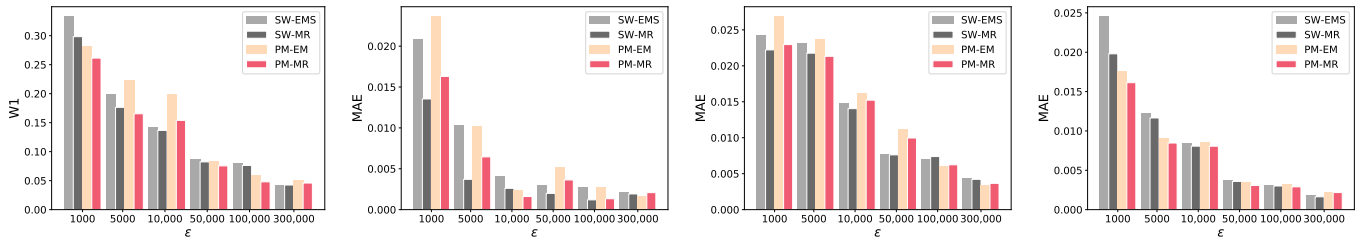
When ε is set to 1 or 2, the MSE of conditional mean on low frequent keys (0.03, 0.06) is $\left(\frac{f_{\text{high}}}{f_{\text{low}}}\right)^2$ times that of high frequent keys, which aligns with the analysis of PCKV. Additionally, our PCKV-MR method consistently outperforms others.

We believe that the reason for this is that the key-value data collection is often insufficient for the data. Each user only provides one key-value pair, and if there are ten keys, on average there are only $n/10$ valid data points for each key.



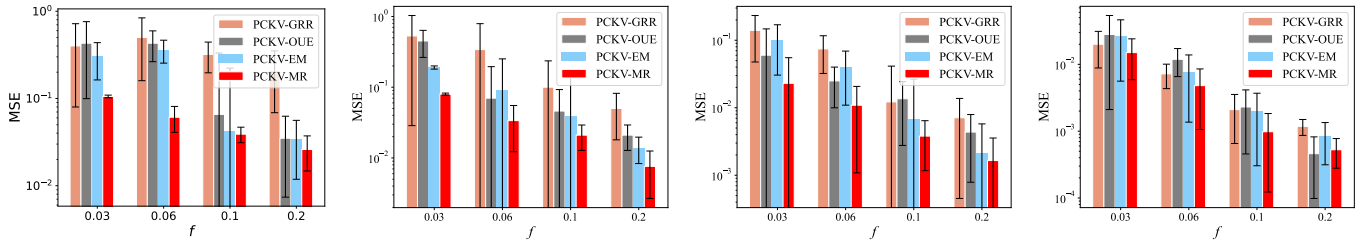
(a) SFC (GRR, $\varepsilon = 1$) (b) Income (GRR,$\varepsilon = 3$) (c) SFC (OLH, $\varepsilon = 1$) (d) Income (OLH, $\varepsilon = 3$)

Fig. 8. Comparison of average estimated counts for unbiased approach and EM-based MLE approach.



(a) Wasserstein Distance (b) Variance (c) Quantile (d) Range Query

Fig. 9. Comparing EM, MR method for numerical distribution on S-MN, with fixed $\epsilon = 1$, varying the size of dataset.



(a) SFC ($\varepsilon = 0.25$) (b) SFC ($\varepsilon = 0.5$) (c) SFC ($\varepsilon = 1$) (d) SFC ($\varepsilon = 2$)

Fig. 10. Mean squared error of conditional mean estimates in SFC, Varying ε and f.

Therefore, using EM can provide more accurate estimation results, and our MR can further improve the results.

APPENDIX E
DETAILS OF POST-PROCESSING METHODS IN EVALUATION

Post-processing is useful since the output of frequency oracles can be quite noisy: negative frequencies or outputs that sum to more the number of inputs. We adopt these techniques:

(1) Basecut. When estimating the whole domain, we sort our frequency estimates in decreasing order and keep them until we get a total frequency, which is n. At this point, we round every remaining estimate down to 0.

(2) Normsub. We round negative estimates to 0. For the rest of the values, we add/subtract some constant δ to ensure that $\sum_{v \in \mathcal{D}_{>0}} (\tilde{f}_\alpha + \delta = n$ where $\mathcal{D}_{>0} = \alpha : \tilde{f}_\alpha > 0$.

(3) IIW. Employ convolution techniques to modify the frequency estimation results and smooth the adjacent estimated frequencies.

(4) EMS. This method adds a smoothing step on the values of $\mathbf{w}$ at the M-step, like

$$\hat{w}_i^{\text{new}} \leftarrow \frac{1}{4}\hat{w}_{i-1} + \frac{1}{2}\hat{w}_i + \frac{1}{4}\hat{w}_{i+1}.$$