Locally Differentially Private Frequency Estimation Based on Convolution Framework

Huiyu Fang*, Liquan Chen*✉, Yali Liu† and Yuan Gao*
*School of Cyber Science and Engineering, Southeast University, Nanjing, China
†School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China
{fanghuiyu, lqchen, yuan_gao}@seu.edu.cn, liuyali@jsnu.edu.cn

2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179389

***Abstract*—Local differential privacy (LDP) collects user data while protecting user privacy and eliminating the need for a trusted data collector. Several LDP protocols have been proposed and deployed in real-world applications. Frequency estimation is a fundamental task in the LDP protocols, which enables more advanced tasks in data analytics. However, the existing LDP protocols amplify the added noise in estimating the frequencies and therefore do not achieve optimal performance in accuracy. This paper introduces a convolution framework to analyze and optimize the estimated frequencies of LDP protocols. The convolution framework can equivalently transform the original frequency estimation problem into a deconvolution problem with noise. We thus add the Wiener filter-based deconvolution algorithms to LDP protocols to estimate the frequency while suppressing the added noise. Experimental results on different real-world datasets demonstrate that our proposed algorithms can lead to significantly better accuracy for state-of-the-art LDP protocols by orders of magnitude for the smooth dataset. And these algorithms also work on non-smooth datasets, but only to a limited extent. Our code is available at https://github.com/SEUNICK/LDP.**

I. Introduction

Differential privacy (DP) [1] has been accepted as the standard of practice for data privacy in the real world. And the U.S. Census Bureau adopts differential privacy to protect 2020 census data [2]. Recently, local differential privacy (LDP) [3] has been proposed, which is a more rigorous differential privacy technique in the local setting for protecting personal privacy. In the local setting, the data collector does not gather the raw data from the users. Instead, each user sends perturbed data with random noise to the data collector, who aims to learn the data distribution from the reports of all users. Since the raw data never leave users, the technique of LDP enables collecting and analyzing data from users while preserving the privacy of every user without relying on a trusted data collector. Nowadays, local differential privacy has been adopted by several major tech companies, including Google [4], Apple [5], Microsoft [6], etc. Examples of usage include collecting web settings and browsing behavior to help identify encountered threats; collecting commonly used emojis and phrases to enhance quick type suggestions; or collecting telemetry data to improve the user experience and make informed business decisions.

A fundamental task in the LDP is frequency estimation. In frequency estimation, there are numerous users and one data collector. Each user owns a value from a specified domain and the data aggregator attempts to estimate the frequency of each value among all users. In recent years, there have been various LDP protocols [4], [5], [7]–[12] proposed for frequency estimation. All these protocols essentially consist of three steps: encoding, perturbation and aggregation [9]. Encoding converts each user's value into a certain format. Perturbation adds noise to the encoded value and sends the perturbed output to the data collector. Aggregation collects all the reports from users to estimate the frequency of each value. Both encoding and perturbation are done on each user's device, and thus the data collector can never know the raw value of each user.

However, since each user has to add noise to his or her private value, the total added noise of LDP is significantly high. Moreover, while the aggregation step can estimate frequencies, we find it further amplifies the added noise. As a consequence, the estimated frequencies may not be accurate enough, and some of them will even be negative. Although there are post-processing algorithms, mostly consistency-based algorithms, that calibrate the estimated frequencies after the aggregation step, they are not effective enough since the added noise has been amplified and they do not fully exploit the properties of the added noise. And these consistency-based algorithms have basically no enhancement for the top k frequent values [13].

To analyze and suppress the noise amplification, we propose a convolution framework. Figure 1 demonstrates existing LDP protocols and our convolution framework. In the convolution framework, the encoding and perturbation steps can be equivalently represented as a one-dimensional circular convolution of true frequency and transfer vector with added noise which will be fully explained in Section 3. Thus, the aggregation step to estimate frequencies can be converted to a deconvolution problem which has been well studied in signal processing [14], [15], image enhancement [16], [17], etc. We can suppress the noise in the deconvolution process and obtain much more accurate estimation results. Most existing protocols fit our proposed framework, especially for all pure LDP protocols.

Since the noise is added by a specified protocol, the distribution of the added noise can be inferred from the known parameters. Our theoretical derivation shows that for pure LDP protocols, the noise follows a zero-mean Gaussian distribution with a known variance. As the Wiener filter is the optimal linear filter for a signal with Gaussian noise and one of the most fundamental noise reduction approaches [18], we introduce Wiener filter-based algorithms to estimate the

DOI 10.1109/SP46215.2023.00148

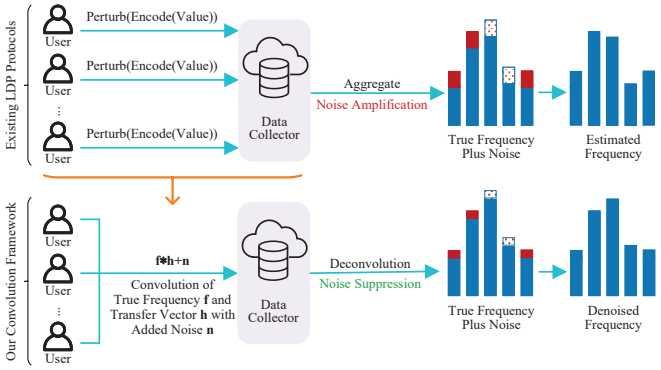


Fig. 1: Our convolution framework.

frequencies in the modified aggregation step. Wiener filter is assumed to know the power spectrum of the original signal and the added noise, as the distribution of the noise is known, the main challenge is to estimate the original signal. To achieve that, we propose the Direct Wiener (DW) filter algorithm and the Improved Iterative Wiener (IIW) filter algorithm. DW algorithm directly uses the output of the existing aggregation step as input to get the generalized spectrum which is easy to implement. IIW algorithm uses the fixed-point iteration to converge to the ideal generalized spectrum of true frequencies, which provides much better accuracy. Moreover, since only the aggregation step is actually modified, existing protocols can easily incorporate our proposed filtering algorithms to improve the accuracy of the estimation results.

The contributions of this paper are as follows:

- We introduce a convolution framework for the existing LDP protocols. This framework enables us to transform the frequency estimation problem into a denoising deconvolution problem, which has been well studied in signal processing, image enhancement, etc. And this allows deconvolution techniques to be applied to locally differentially private frequency estimation.
- Based on the convolution framework, we give the power spectrum of added noise and introduce DW and IIW algorithms to filter the added noise while estimating frequencies to improve the accuracy of existing LDP protocols. Since we only modify the aggregation step, existing LDP protocols can easily append our proposed algorithms to filter noise and boost the accuracy.
- We conduct extensive experiments on real-world datasets based on the two commonly used LDP protocols. Results illustrate that appending our proposed algorithms to the state-of-the-art protocols provides significantly better accuracy for smooth datasets.

Roadmap. In Section 2, we review the LDP definition, existing LDP frequency estimation protocols and post-processing calibration algorithms. In Section 3, We present our convolution framework for LDP protocols and apply it to analyze the noise amplification in existing LDP protocols. We give Wiener filter-based algorithms to suppress the noise in Section 4. We show our experimental results in Section 5 and conclude in Section 6.

II. Background and Related Work

We assume that there are m users and one data collector in frequency estimation. Each user possesses a private value v from a specified domain D which is denoted as $\{1, 2, \ldots, d\}$. The goal of the data collector is to learn the frequencies of each value from all users, and f_i is the frequency of value i. Frequency estimation is a fundamental task in the LDP protocols and is the key building block of other advanced tasks, e.g., heavy hitter identification [12], [19], [20], range queries [21]–[23], etc. Improving the accuracy of frequency estimation will also improve the effectiveness of the protocols for other tasks.

Notational Conventions. In this paper, bold letters are used to denote vectors and bold capital letters denote the vector after Fourier transform. For instance, $\mathbf{f} = [f_1, f_2, \ldots, f_d]$ is the true frequencies of each value in the specified domain D and $\mathbf{F}$ is the Fourier transform of $\mathbf{f}$. If there is a tilde above the notation(e.g., $\widetilde{\mathbf{f}}$), it is an estimate or a theoretical value by the data collector.

Privacy Requirement. A frequency estimation protocol A is used by each user to perturb his or her private value v, and the data collector receives all the perturbed values $\mathrm{A}(v)$ to estimate the frequency of each value. To satisfy the LDP privacy requirement, the output of protocol A must comply with the following property.

Definition 1 (ε-Local Differential Privacy)**.** *A protocol* A *satisfies ε-local differential privacy (ε-LDP), where $\varepsilon \geq 0$, if and only if for any input $v, v' \in D$, we have*

$$\forall y \in \mathrm{A}(D) : \Pr[\mathrm{A}(v) = y] \leq e^{\varepsilon} \Pr[\mathrm{A}(v') = y]$$

where $\mathrm{A}(D)$ denotes the set of all possible outputs of A.

Because the raw value v of each user is never sent to the data collector, even if the collector is malicious, the privacy of every user is still preserved.

A. Existing LDP Frequency Estimation Protocols

The LDP frequency estimation protocols can be traced back to the random response technique proposed in 1965 [24]. Recently, several LDP protocols have been proposed to estimate the frequencies, e.g., RAPPOR [11], KRR [8], OUE [9], OLH [9], Subset Selection [10] and Hadamard Response [7]. All these LDP protocols can be broken down into three steps: encoding, perturbation and aggregation [9]. Each user converts his or her value into a certain format in encoding first and then adds noise to the encoded value in perturbation to achieve local differential privacy. The data collector gathers all the perturbed values to estimate the frequencies in aggregation lastly.

Moreover, most existing LDP protocols for frequency estimation achieve pure local differential privacy [9]. The definition of pure LDP is as follows:

Definition 2 (Pure Local Differential Privacy)**.** *A protocol* A *is pure if and only if there exist two probability values $p^* > q^*$ such that for all v*

$$\Pr[\mathrm{A}(v) \in \{y | v \in \mathrm{Support}(y)\}] = p^*,$$
$$\forall_{v' \neq v} \Pr[\mathrm{A}(v') \in \{y | v \in \mathrm{Support}(y)\}] = q^*$$

where the set $\{y|v \in \mathrm{Support}(y)\}$ denotes all outputs y that "support" v.

For any protocols that satisfy the pure LDP definition, the data collector can estimate the frequency f_i in the aggregation step using the following equation:

$$\widetilde{f_i} = \frac{\sum_{k=1}^{m} \mathbb{1}_{\mathrm{Support}(y_k)}(i) - mq^*}{m(p^* - q^*)} \tag{1}$$

where $\mathbb{1}_{\mathrm{Support}(y_k)}(i)$ is 1 if the output y_k of user k supports the value i, otherwise 0. Thus $\sum_{k=1}^{m} \mathbb{1}_{\mathrm{Support}(y_k)}(i)$ sums all the perturbed outputs that support the value i. Then, the data collector use the probabilities p^*, q^* and the total number m of users to normalize the counts to estimate the frequency f_i.

The variance of the estimated frequency indicates the accuracy of the LDP protocol quantitatively, i.e., the smaller the variance, the more accurate the frequency estimation. And the variance of the estimated frequency f_i for a pure LDP protocol is:

$$\mathrm{Var}[\widetilde{f_i}] = \frac{q^*(1-q^*)}{m(p^*-q^*)^2} + \frac{f_i(1-p^*-q^*)}{m(p^*-q^*)} \tag{2}$$

Formula (1) and (2) are come from [9]. For completeness, we give the relevant derivations in Appendix A. As the $\mathrm{Var}[\widetilde{f_i}]$ in (2) is dominated by the first term, we can use just the first term to represent the variance approximately as $\mathrm{Var}^*[\widetilde{f_i}]$ as in [9]:

$$\mathrm{Var}^*[\widetilde{f_i}] = \frac{q^*(1-q^*)}{m(p^*-q^*)^2} \tag{3}$$

The following are two commonly used state-of-the-art LDP frequency estimation protocols that satisfy pure LDP. We use these two protocols as examples to illustrate the pure LDP and three steps in detail.

K-ary Randomized Response (KRR). KRR [8] generalizes the randomized response technique.

Encoding: In KRR, $\mathrm{Encode}(v) = v$ and $v \in D$.

Perturbation: $\mathrm{Perturb}(v)$ outputs $v' \in D$ as follows:

$$\Pr[\mathrm{Perturb}(v) = v'] = \begin{cases} p = \frac{e^\varepsilon}{e^\varepsilon + d - 1}, & \text{if } v = v' \\ q = \frac{1}{e^\varepsilon + d - 1}, & \text{if } v \neq v' \end{cases}$$

Aggregation: KRR is a pure LDP protocol and $p^* = p, q^* = q$. The data collector estimates the the frequency $\widetilde{f_i}$ by using (1), and the approximate variance of $\widetilde{f_i}$ is:

$$\mathrm{Var}_{\mathrm{KRR}}[\widetilde{f_i}] = \frac{e^\varepsilon + d - 2}{m(e^\varepsilon - 1)^2} \tag{4}$$

Optimized Unary Encoding (OUE). OUE [9] achieves state-of-the-art performance in terms of accuracy.

Encoding: OUE uses unary encoding to encode v. $\mathrm{Encode}(v) = [0, \ldots, 0, 1, 0, \ldots, 0]$, a length-$d$ binary vector $\mathbf{b}$ where only the v-th bit is 1.

Perturbation: OUE perturbs vector $\mathbf{b}$ into $\mathbf{b}'$ bit by bit as follows:

$$\Pr[\mathrm{Perturb}(b'_i) = 1] = \begin{cases} p = \frac{1}{2}, & \text{if } b_i = 1 \\ q = \frac{1}{e^\varepsilon + 1}, & \text{if } b_i = 0 \end{cases}$$

Aggregation: OUE is also a pure LDP protocol and $p^* = p, q^* = q$. The data collector estimates the frequency $\widetilde{f_i}$ of value i also using (1). However, with different encoding and perturbation, the approximate variance is:

$$\mathrm{Var}_{\mathrm{OUE}}[\widetilde{f_i}] = \frac{4e^\varepsilon}{m(e^\varepsilon - 1)^2} \tag{5}$$

Comparing KRR with OUE, KRR is easier and faster to implement on the client side and the communication cost of KRR is obviously lower. But when d is large, OUE has a lower variance and is more accurate. More specifically, OUE outperforms KRR when d is greater than $3e^\varepsilon + 2$.

Other LDP Frequency Estimation Protocols. Here we briefly introduce several other LDP frequency estimation protocols that have been proposed recently.

Optimized Local Hashing (OLH) [9] uses a random hash function to map the input values into a smaller domain of size t, and then perturbs the hash values to achieve lower communication cost than OUE. The p^* and q^* values of OLH are the same as OUE, which means they have the same accuracy performance.

Fast Local Hashing (FLH) [25] is also based on local hashing as OLH. It sacrifices some theoretical guarantees on accuracy to achieve computational gains on the server side compared to OLH. This approach is several times faster than OLH, without much loss of accuracy.

Hadamard Response [7] uses Hadamard transform to compress the communication cost. It is similar to OLH with $t = 2$. However, in OLH t varies with the privacy budget ε rather than being fixed. Therefore, Hadamard Response is suboptimal to OLH and OUE.

Projective Geometry Response (PGR) [26] is based on using projective planes over a finite field to define a small collection of sets that are close to being pairwise independent and a dynamic programming algorithm for approximate histogram reconstruction on the server side. According to [26], it achieves the same variance or accuracy as OUE and OLH.

Piecewise Mechanism [27] consists of two main components, one is used to handle mean estimation over a single numeric attribute and the other is used to handle multidimensional data that can contain both numeric and categorical attributes. And the part of processing multidimensional data with categorical attributes is based on OUE.

Optimized Multiple Encoding (OME) [28] is motivated by OUE and is also designed to handle multidimensional data.

The key idea is to map each real value v_i of the embedding vector into a binary vector with a fixed size l.

B. Existing Calibration Algorithms

Existing LDP protocols introduce significantly high noise in the encoding and perturbation steps, which are further amplified in the aggregation step, thus dramatically affecting the accuracy of the estimation results. To improve the accuracy, several post-processing calibration algorithms have been proposed, most of which are based on consistency.

Specifically, consistency [13] means that each estimated frequency is non-negative and the sum of the estimated frequencies is 1. To make the estimation result non-negative, the most common algorithm is the significance threshold [9], [11].

Statistically, higher estimates are more reliable. And with a fixed distribution of added noise in the specific LDP protocol, the higher the true frequency, the less likely it is to be swamped by noise. Therefore, the significance threshold is proposed to discard unreliable estimates. After applying this algorithm, all estimated frequencies above the threshold are kept, the rest are considered unreliable and need to be discarded.

$$\mathrm{ST} = \Phi^{-1}(1 - \frac{\alpha}{d})\sqrt{\mathrm{Var}^*} \tag{6}$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution and the parameter α controls the magnitude of the threshold. In [9] and [20], α is recommended to be set to 0.05. In (6), $\frac{\alpha}{d}$ can be viewed as the probability that a zero-mean Gaussian random variable with standard deviation $\sqrt{\mathrm{Var}^*}$ is above the threshold. Therefore, the probability that all random variables have a value above the threshold is at most $d \cdot \frac{\alpha}{d} = \alpha$. In other words, α controls the number of values that originally have low frequencies but have high estimated frequencies above the threshold.

To ensure that the estimated frequencies sum to one, the normalization algorithm is firstly proposed for the centralized differential privacy setting [29] and then used in the local differential privacy setting [22], [30]. In normalization algorithm, δ is added to each estimated frequency so that the sum is 1. Since the normalization algorithm does not force all estimated frequencies to be non-negative, Kairouz [8] and Bassily [31] convert negative estimated frequencies to 0 and add δ to each remaining estimated frequency. Instead of adding δ, Wang [13] proposes a different algorithm to convert negative and small estimated frequencies to zero so that the sum is 1.

Besides the consistency-based approaches, Jia [20] proposes an algorithm that assumes the true frequencies follow a certain type of distribution with unknown parameters and estimates the parameters of the distribution to finally update the estimated frequencies to achieve the expected least square. However, it is difficult to have prior knowledge about the distribution of true frequencies which limits the scenarios of this algorithm.

III. Analyzing LDP Protocols with Convolution Framework

A. Our Convolution Framework

LDP protocols essentially consist of three steps: encoding, perturbation and aggregation. Our convolution framework can equivalently convert the encoding and perturbation steps into a one-dimensional circular convolution process with noise. Hence, the aggregation step is equivalently transformed into a deconvolution problem with noise which has been well studied in signal processing, image processing, etc. And several effective deconvolution techniques [17], [32] have been proposed. By applying the deconvolution techniques to the aggregation step, the noise in the estimated frequencies can be greatly suppressed.

Theorem 1. *For protocol* A *that satisfies pure LDP, the encoding and perturbation steps can be equivalently transformed into a one-dimensional circular convolution process with added noise. Specifically, we have:*

$$\mathbf{g} = \mathbf{f} * \mathbf{h} + \mathbf{n} \tag{7}$$

where $\mathbf{g}$ *is a length-d vector consisting of the summation of gathered perturbed outputs,* $g_i = \frac{\sum_{k=1}^{m} \mathbb{1}_{\mathrm{Support}(y_k)}(i)}{m}$, $\mathbf{h}$ *is a length-d transfer vector* $[p^*, q^*, \ldots, q^*]$ *and* $\mathbf{n}$ *is a length-d vector of added noise.*

Proof. From Definition 2, we can build a $d \times d$ circulant matrix M for pure LDP protocols to characterize the randomization process of encoding and perturbation.

$$M = \begin{bmatrix} p^* & q^* & q^* & \cdots & q^* \\ q^* & p^* & q^* & \cdots & q^* \\ \vdots & & \ddots & & \vdots \\ q^* & \cdots & q^* & p^* & q^* \\ q^* & q^* & \cdots & q^* & p^* \end{bmatrix}$$

The matrix M indicates the transformation probabilities, where $M_{i,j}$ represents the probability of input value i supports value j after perturbation. Thus, for an input value i, the probability of output supports value j is $M_{i,j}$. And for m users and the true frequency vector $\mathbf{f}$, we have a total expectation of outputs supporting value j as:

$$\mathrm{E}(\textstyle\sum_{k=1}^{m} \mathbb{1}_{\mathrm{Support}(y_k)}(j)) = \sum_{i=1}^{d} m f_i M_{i,j}$$

Divide both sides by m, we have:

$$\mathrm{E}(g_j) = \textstyle\sum_{i=1}^{d} f_i M_{i,j}$$
$$\mathrm{E}(\mathbf{g}^\top) = M\mathbf{f}^\top$$

Because M is a circulant square matrix, we can rewrite the equation as the circular convolution:

$$\mathrm{E}(\mathbf{g}) = \mathbf{f} * \mathbf{h}$$

where transfer vector $\mathbf{h}$ is the first column of M. Plus the added noise $\mathbf{n}$ in the randomization process, the real gathered outputs can be expressed as:

$$\mathbf{g} = \mathbf{f} * \mathbf{h} + \mathbf{n}$$

□

It is worth noting that Theorem 1 can be further extended to certain LDP protocols, even though they do not satisfy the pure LDP, as long as the transformation probability matrix M of these protocols is circulant.

Theorem 2. *In convolution framework,* $\mathbf{n}$ *can be well approximated as a Gaussian noise with zero-mean and a known variance. More specifically, we have:*

$$n_i \sim \mathcal{N}(0, \sigma_i^2) \tag{8}$$

$$\sigma_i^2 = \frac{q^*(1-q^*)}{m} + \frac{f_i(p^*-q^*)(1-p^*-q^*)}{m} \tag{9}$$

Proof. There are m users and one data collector, and each user with value i sends the perturbed value to the data collector which is equivalent to an independent Bernoulli experiment with parameter $M_{i,j}$ supports value j (resp. $1 - M_{i,j}$). In other words, the random variable mg_j is the summation of m independent variables drawn from the Bernoulli distribution. Therefore,

$$\begin{aligned}
\mathrm{E}(mg_j) &= \textstyle\sum_{i=1}^{d} m f_i M_{i,j} \\
\mathrm{Var}(mg_j) &= \textstyle\sum_{i=1}^{d} m f_i M_{i,j}(1-M_{i,j}) \\
&= m f_j p^*(1-p^*) + m(1-f_j)q^*(1-q^*) \\
&= m q^*(1-q^*) + m f_j (p^*-q^*)(1-p^*-q^*) \\
\mathrm{Var}(g_j) &= \frac{q^*(1-q^*)}{m} + \frac{f_j(p^*-q^*)(1-p^*-q^*)}{m}
\end{aligned}$$

Let $\sigma_j^2 = \mathrm{Var}(g_j)$, according to the central limit theorem, the random variable g_j can be regarded as the expected value plus a Gaussian distributed noise:

$$\begin{aligned}
g_j &= \textstyle\sum_{i=1}^{d} f_i M_{i,j} + \mathcal{N}(0, \sigma_j^2) \\
&= f_j p^* + (1-f_j)q^* + \mathcal{N}(0, \sigma_j^2) \\
&= f_j(p^*-q^*) + q^* + \mathcal{N}(0, \sigma_j^2) \\
\sigma_j^2 &= \frac{q^*(1-q^*)}{m} + \frac{f_j(p^*-q^*)(1-p^*-q^*)}{m}
\end{aligned}$$

That is:

$$n_i \sim \mathcal{N}(0, \sigma_i^2)$$

□

It is worth noting that $\mathbf{n}$ is the noise added to the overall vector in the system of multiple users, which occurs in the perturbation step before the aggregation step. Although previous studies [9], [13], [20] have included analyses of the noise in LDP protocols, these analyses are based on the noise after the aggregation step.

B. Noise amplification in LDP protocols

According to the definitions and derivations in Theorems 1 and 2, we have:

$$g_i = \frac{\sum_{k=1}^{m} \mathbb{1}_{\mathrm{Support}(y_k)}(i)}{m} = f_i p^* + (1-f_i)q^* + n_i \tag{10}$$

Substituting g_i into (1) used in the existing aggregation step, we have:

$$\widetilde{f_i} = f_i + \frac{n_i}{p^*-q^*} \tag{11}$$

$$\begin{aligned}
\mathrm{Var}[\widetilde{f_i}] &= \mathrm{Var}[\frac{n_i}{p^*-q^*}] = \frac{\sigma^2}{(p^*-q^*)^2} \\
&= \frac{q^*(1-q^*)}{m(p^*-q^*)^2} + \frac{f_i(1-p^*-q^*)}{m(p^*-q^*)} \\
&\approx \frac{q^*(1-q^*)}{m(p^*-q^*)^2}
\end{aligned} \tag{12}$$

Equation (11) suggests that the added noise is further amplified by a factor of $1/(p^*-q^*)$ in the existing aggregation step, which leads to further inaccuracies in the estimation results. Equation (12) yields the same variance as in (2) and (3) derived by pure LDP, which verifies the correctness of our convolution framework.

For the commonly used KRR and OUE introduced in Section 2, their σ^2 of added noise $\mathbf{n}$ can be solved by (9) separately as follows:

$$\sigma_{\mathrm{KRR}}^2 = \frac{e^\varepsilon + d - 2}{m(e^\varepsilon + d - 1)^2} \tag{13}$$

$$\sigma_{\mathrm{OUE}}^2 = \frac{e^\varepsilon}{m(e^\varepsilon + 1)^2} \tag{14}$$

It is clear that the added noise of KRR is smaller than that of OUE, especially when d is large, although the final estimated frequencies of OUE are more accurate. This is due precisely to the fact that KRR has a much larger factor of noise amplification in the existing aggregation step.

However, with the convolution framework, the encoding and perturbation steps can be converted equivalently into a one-dimensional convolution process with noise as $\mathbf{g} = \mathbf{f} * \mathbf{h} + \mathbf{n}$, where $\mathbf{f}$ is the input true frequency vector and $\mathbf{g}$ is the output after the perturbation received by the data collector. With known $\mathbf{g}$ and $\mathbf{h}$, estimating the true frequency vector $\mathbf{f}$ is a classical problem of restoring data degraded by a convolution and the addition of Gaussian noise. By using the deconvolution techniques, it is possible to significantly suppress the added noise while estimating the frequencies. And there is a chance that the estimation accuracy of KRR exceeds that of OUE with the deconvolution techniques, as the added noise of KRR is smaller than that of OUE.

IV. Wiener Deconvolution

A. Overview of Wiener deconvolution

Wiener deconvolution is an application of the Wiener filter to the noise problems inherent in deconvolution, especially for such additive Gaussian noise, Wiener filtering is optimal in terms of the mean squared error (MSE) after recovering the data [18]. In other words, it executes an optimal tradeoff between inverse convolution and noise smoothing. Wiener filtering removes the additive noise and inverts the convolution simultaneously to restore the data. To use Wiener deconvolution, Equation (7) needs to be converted to the frequency domain. Since circular convolution in the time domain is

equivalent to the multiplication of the frequency domain, we have:

$$\mathbf{G} = \mathbf{F}\mathbf{H} + \mathbf{N} \tag{15}$$

where $\mathbf{G}$, $\mathbf{F}$, $\mathbf{H}$, and $\mathbf{N}$ are the Fourier transform of $\mathbf{g}$, $\mathbf{f}$, $\mathbf{h}$, and $\mathbf{n}$ respectively.

The Wiener filter in the frequency domain can be demonstrated as:

$$W = \frac{\mathbf{H}^*}{|\mathbf{H}|^2 + P_\mathbf{n}/P_\mathbf{f}} = \frac{1}{\mathbf{H}}[\frac{1}{1 + 1/(|\mathbf{H}|^2 \text{SNR})}] \tag{16}$$

where $\mathbf{H}^*$ is the conjugation of $\mathbf{H}$, $P_\mathbf{f}$ and $P_\mathbf{n}$ are the generalized power spectral density of $\mathbf{f}$ and $\mathbf{n}$ respectively, $\text{SNR} = P_\mathbf{f}/P_\mathbf{n}$ is the signal-to-noise ratio.

The filtering operation is carried out in the frequency domain as:

$$\widetilde{\mathbf{F}} = W\mathbf{G} \tag{17}$$

and then executing an inverse Fourier transform on $\widetilde{F}$ to obtain $\widetilde{\mathbf{f}}$ as:

$$\widetilde{\mathbf{f}} = \text{IFFT}(\widetilde{\mathbf{F}}) \tag{18}$$

As seen in (16), when the signal-to-noise ratio goes to infinite (i.e., the noise goes to zero or the signal goes to infinite), the term inside the square brackets equals 1, and the Wiener filter is just the inverse of the degraded data and has the same effect as the original aggregation step. However, as the signal-to-noise ratio falls at specific frequencies, the term inside the square brackets also falls, which means the Wiener filter automatically adjusts the filtering intensity according to the signal-to-noise ratio at different frequencies.

There are two main assumptions in the Wiener filter. One is the noise is independent of the signal, and the other is the signal is a weak stationary stochastic process. The basic idea of stationarity is the statistical properties of the stochastic process do not change over time. For non-stationary processes, Wiener filter may not achieve optimal filtering performance, but it will still be somewhat effective. For real-life data, they often follow a certain type of distribution. And for numerical data, there will be some continuity or smoothness in the frequency of adjacent numerical values, which can be considered to be approximately stationary on a small scale. In nature, many attributes are ordinal or numerical, e.g., age, height, weight, income, population by age, etc. And all these datasets can be viewed as smooth and have a certain level of stationarity.

To implement the Wiener filter in practice, we need to estimate the power spectral density of the true frequencies $\mathbf{f}$ and the additive noise $\mathbf{n}$.

B. Estimating $P_\mathbf{n}$

Since $\mathbf{n}$ can be modeled as the additive Gaussian noise, it is uncomplicated to obtain the expected power spectral density $\text{E}(P_\mathbf{n})$. In our algorithms, we use the expectation $\text{E}(P_\mathbf{n})$ as the estimate $\widetilde{P_\mathbf{n}}$.

Theorem 3. *For the additive Gaussian noise* $\mathbf{n}$ *which consists* d *independent variables and* $n_i \sim \mathcal{N}(0, \sigma_i^2)$*, its generalized power spectral density expectation is:*

$$\begin{aligned}\text{E}(P_\mathbf{n}) &= \text{E}([P_0, P_1, \ldots, P_{d-1}]) \\ &= [\textstyle\sum_{i=1}^d \sigma_i^2, \sum_{i=1}^d \sigma_i^2, \ldots, \sum_{i=1}^d \sigma_i^2]\end{aligned} \tag{19}$$

$$\textstyle\sum_{i=1}^d \sigma_i^2 = \dfrac{dq^*(1-q^*)}{m} + \dfrac{(p^*-q^*)(1-p^*-q^*)}{m} \tag{20}$$

Proof. According to the definition of Fourier transform, we have:

$$\mathbf{N} = \text{FFT}(\mathbf{n}) = [N_0, N_1, \ldots, N_{d-1}]$$

$$N_k = \textstyle\sum_{l=0}^{d-1} n_{l+1} e^{-j\frac{2\pi}{d}lk} \quad (k = 0, 1, \ldots, d-1)$$

$$n_{l+1} e^{-j\frac{2\pi}{d}lk} = n_{l+1}(\cos\frac{2\pi}{d}lk - j\sin\frac{2\pi}{d}lk)$$

Here we use n_l instead of n_i to prevent confusion with the imaginary number i or j. Since $n_{l+1} \sim \mathcal{N}(0, \sigma_{l+1}^2)$, we have the expectation of $n_{l+1} e^{-j\frac{2\pi}{d}lk}$ as:

$$\text{E}(n_{l+1} e^{-j\frac{2\pi}{d}lk}) = 0$$

$$\begin{aligned}\text{E}(|n_{l+1} e^{-j\frac{2\pi}{d}lk}|^2) &= \text{E}(n_{l+1}^2(\cos^2\frac{2\pi}{d}lk + \sin^2\frac{2\pi}{d}lk)) \\ &= \text{E}(n_{l+1}^2) = \sigma_{l+1}^2\end{aligned}$$

As each variable in n is independent, the expectation of the summation N_k is:

$$\text{E}(N_k) = \text{E}(\textstyle\sum_{l=0}^{d-1} n_{l+1} e^{-j\frac{2\pi}{d}lk}) = 0$$

$$\text{E}(|N_k|^2) = \textstyle\sum_{l=0}^{d-1} \sigma_{l+1}^2 = \sum_{l=1}^{d} \sigma_l^2$$

$P_\mathbf{n}$ can be derived from $\mathbf{N}$ as follows:

$$P_\mathbf{n} = |\mathbf{N}|^2 = [|N_0|^2, |N_1|^2, \ldots, |N_{d-1}|^2]$$

Thus, we have:

$$\begin{aligned}\text{E}(P_\mathbf{n}) &= \text{E}([|N_0|^2, |N_1|^2, \ldots, |N_{d-1}|^2]) \\ &= [\textstyle\sum_{l=1}^d \sigma_l^2, \sum_{l=1}^d \sigma_l^2, \ldots, \sum_{l=1}^d \sigma_l^2]\end{aligned}$$

In addition, as the sum of all frequencies is 1, the $\sum_{l=1}^d \sigma_l^2$ is:

$$\textstyle\sum_{l=1}^d \sigma_l^2 = \dfrac{dq^*(1-q^*)}{m} + \dfrac{(p^*-q^*)(1-p^*-q^*)}{m}$$

□

It is worth noting that for KRR, $P_0 = 0$ is an exception. In KRR, one output supports only one input, and the sum of all estimates is m, which is equal to the sum of the expected counts of each value in domain D. Thus we have $mN_0 = m\sum_{l=0}^{d-1} n_{l+1} e^0 = m(n_1 + n_2 + \cdots + n_d)$ is the subtraction of these two sums equal to 0 and $P_0 = 0$. But for $k \neq 0$, P_k can still be considered as $\sum_{i=1}^d \sigma_i^2$ in KRR. We will demonstrate this in the empirical experiments in Section 5.

Algorithm 1 Improved Iterative Wiener Filter Algorithm

Input: $\mathbf{g}$, $\mathbf{h}$ and $\widetilde{P_{\mathbf{n}}}$
Output: $\widetilde{\mathbf{f}}$
Let $\mathbf{G} \leftarrow \text{FFT}(\mathbf{g}), \mathbf{H} \leftarrow \text{FFT}(\mathbf{h}), \widetilde{P_{\mathbf{f}}}(0) \leftarrow |\frac{\mathbf{G}}{\mathbf{H}}|^2$
$W(0) = \frac{\mathbf{H}^*}{|\mathbf{H}|^2 + \widetilde{P_{\mathbf{n}}}/\widetilde{P_{\mathbf{f}}}(0)}$
$\widetilde{\mathbf{F}}(0) = W(0)\mathbf{G}$
while $\widetilde{\mathbf{F}}(i)$ not converge **do**
$\widetilde{P_{\mathbf{f}}}(i)_{\text{correction}} \leftarrow \frac{\widetilde{P_{\mathbf{n}}}}{|\mathbf{H}|^2 + \widetilde{P_{\mathbf{n}}}/\widetilde{P_{\mathbf{f}}}(i-1)}$
$\widetilde{P_{\mathbf{f}}}(i) \leftarrow |\widetilde{\mathbf{F}}(i-1)|^2 + \widetilde{P_{\mathbf{f}}}(i)_{\text{correction}}$
$W(i) = \frac{\mathbf{H}^*}{|\mathbf{H}|^2 + \widetilde{P_{\mathbf{n}}}/\widetilde{P_{\mathbf{f}}}(i)}$
$\widetilde{\mathbf{F}}(i) = W(i)\mathbf{G}$
end while
$\widetilde{\mathbf{f}} = \text{IFFT}(\widetilde{\mathbf{F}}(i))$
Return $\widetilde{\mathbf{f}}$

C. Estimating $P_{\mathbf{f}}$ and $\mathbf{f}$

Direct Wiener (DW) filter algorithm. Since the variables $P_{\mathbf{f}}$ and $\mathbf{f}$ are related as follows:

$$P_{\mathbf{f}} = |\mathbf{F}|^2 \tag{21}$$

where $\mathbf{F}$ is the Fourier transform of $\mathbf{f}$. Therefore, by using the result $\widetilde{\mathbf{f}}$ estimated by (1), we can estimate $P_{\mathbf{f}}$. After that, given $\widetilde{P_{\mathbf{f}}}$, $P_{\mathbf{n}}$, $\mathbf{G}$ and $\mathbf{H}$, we can re-estimate $\mathbf{f}$ by Wiener filtering. This is the Direct Wiener filter algorithm. The DW algorithm directly uses the original estimate $\widetilde{\mathbf{f}}$ of the aggregation step to obtain the power spectrum $\widetilde{P_{\mathbf{f}}}$, so it can be regarded as a smoothing of the original estimate $\widetilde{\mathbf{f}}$ to some extent. Because the denoising effect of the Wiener filter is directly related to the accuracy of the spectrum estimation $\widetilde{P_{\mathbf{f}}}$, which is obviously still far from the ideal $P_{\mathbf{f}}$, the DW algorithm is not yet optimal. However, the performance of the DW algorithm will be greatly improved if there is prior knowledge of the true power spectrum $P_{\mathbf{f}}$.

Improved Iterative Wiener (IIW) filter algorithm. Although the DW algorithm is not optimal, we can find that the frequency estimation $\widetilde{\mathbf{f}}$ is refined after one DW algorithm processing. The improved frequency estimation can gives a more accurate spectral estimation of $P_{\mathbf{f}}$, which allows the DW algorithm to be used again to improve the frequency estimation. Repeating the process in an iterative fashion is iterative Wiener filtering. However, the simple iterative Wiener filtering converges to more than one fixed point, and the convergence of the iterative Wiener filtering is not the true minimum MSE solution. In order to make the iterative results converge to the true $P_{\mathbf{f}}$, a possible modification is to add a correction factor to the power spectrum and the formal proof of convergence can be found in [32]. The Improved Iterative Wiener filter algorithm is shown in Algorithm 1. In our proposed IIW algorithm, we only need to use $\mathbf{g}$, $\mathbf{h}$ and $\widetilde{P_{\mathbf{n}}}$ as initial values without any other additional preconditioning. The fixed points of the update equation in the IIW algorithm are determined by solving for the roots of $\widetilde{P_{\mathbf{f}}}(i+1) - \widetilde{P_{\mathbf{f}}}(i) = 0$ as follows:

$$\widetilde{P_{\mathbf{f}}}(i+1) - \widetilde{P_{\mathbf{f}}}(i) = \frac{|\mathbf{H}|^2 \widetilde{P_{\mathbf{f}}}^2(i)[|\mathbf{G}|^2 - |\mathbf{H}|^2 \widetilde{P_{\mathbf{f}}}(i) - \widetilde{P_{\mathbf{n}}}]}{[|\mathbf{H}|^2 \widetilde{P_{\mathbf{f}}}(i) + \widetilde{P_{\mathbf{n}}}]^2} \tag{22}$$

There are two fixed points at

$$\widetilde{P_{\mathbf{f}}} = \frac{|\mathbf{G}|^2 - \widetilde{P_{\mathbf{n}}}}{|\mathbf{H}|^2} \quad \text{and} \quad \widetilde{P_{\mathbf{f}}} = 0 \tag{23}$$

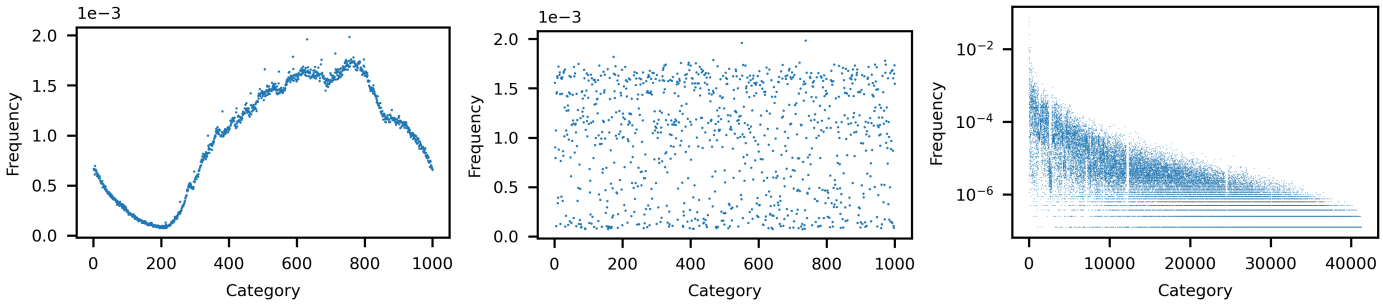
Since the added noise $\mathbf{n}$ can be considered irrelevant to the real frequencies $\mathbf{f}$, from (15) we have:

$$P_{\mathbf{f}} = |\mathbf{F}|^2 = \frac{|\mathbf{G}|^2 - |\mathbf{N}|^2}{|\mathbf{H}|^2} = \frac{|\mathbf{G}|^2 - P_{\mathbf{n}}}{|\mathbf{H}|^2} \tag{24}$$

Comparing (23) and (24), we can find that the proposed IIW algorithm does lead to the ideal power spectrum $P_{\mathbf{f}}$ for any positive starting point. It is also easy to notice from (24) that any error in the estimate of $P_{\mathbf{n}}$ will be dramatically amplified for a small value of $|\mathbf{H}|^2$. Therefore, the result of (24) cannot be used as an estimate of $P_{\mathbf{f}}$ to be applied in the DW algorithm. However, in the IIW algorithm, this problem can be alleviated by controlling the times of iterations. Besides, we find that if the initial value of the power spectrum $\widetilde{P_{\mathbf{f}}}$ in IIW is set to $P_{\mathbf{g}} = |\mathbf{G}|^2$, the performance for KRR on smooth datasets can be substantially improved. But this enhancement is not stable and only works on KRR. For better generality, we set the initial value of $\widetilde{P_{\mathbf{f}}}$ to $|\frac{\mathbf{G}}{\mathbf{H}}|^2$, and $\frac{\mathbf{G}}{\mathbf{H}}$ is equivalent to the original aggregation estimation result, while the result of $\widetilde{\mathbf{F}}(0)$ is equal to the result of DW. Empirically, the IIW algorithm usually has an extremely slow convergent rate after about 30 iterations. Therefore, the stopping criterion is set to 30 iterations to achieve a balance between performance and computing cost.

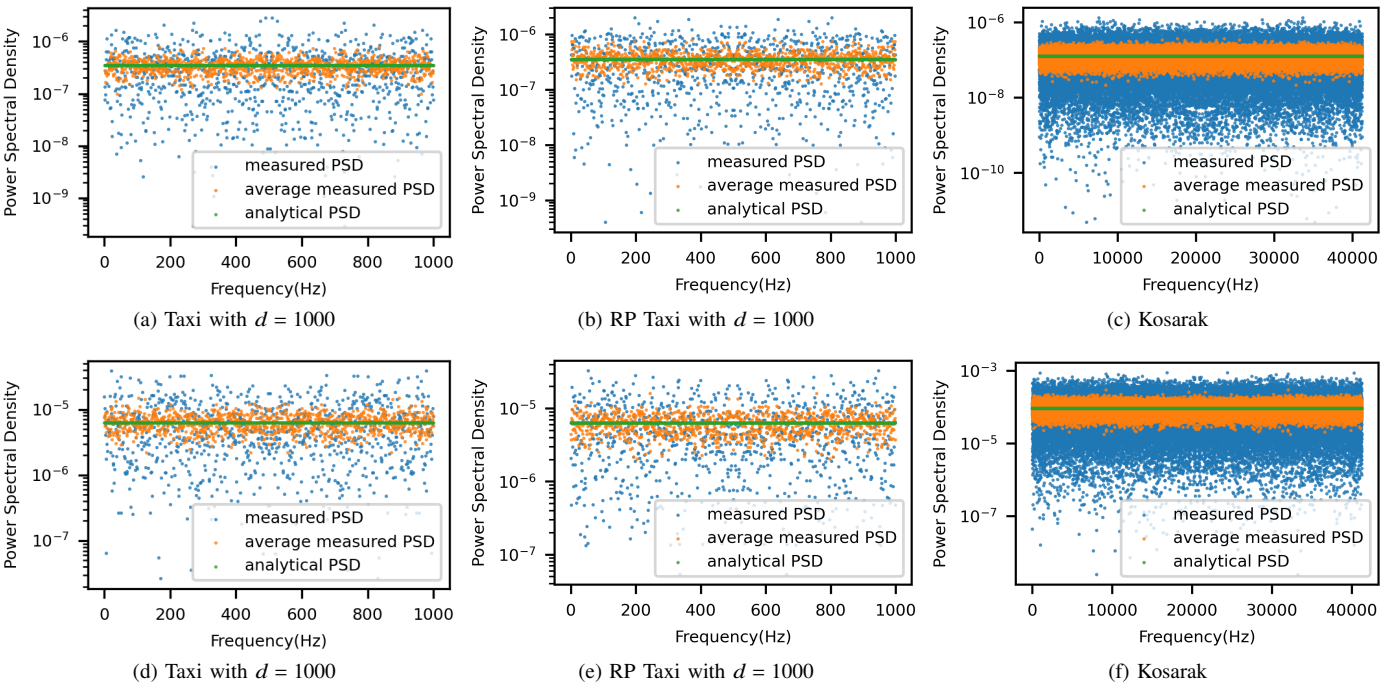
Average Multiple Random Permuted (AMRP). Wiener filtering requires the signal to be weakly stationary to achieve the optimum MSE. Through experiments, our proposed approaches based on Wiener filtering work best on smooth datasets, while less effective on non-smooth datasets, especially on randomly permuted datasets. To overcome this, we propose the AMRP approach for non-smooth datasets. It is inspired by the fact that different re-permutation of the same aggregation results lead to different Wiener filter estimation results. Therefore, we randomly re-permute the observations on the server side and then restore the estimated result to the order before re-permute. We sum the restored estimated result obtained by different randomly re-permute orders and average it to achieve a more accurate result. The number of re-permute estimated results is set to 100.

Intuitively, Wiener deconvolution can be considered a decorrelation process. Usually, the noise signal and the original signal are not correlated, so the noise can be effectively removed after deconvolution. However, after random permutation, the distributions of the input signal and noise signal will have some similarity (but not fully), so Wiener deconvolution will be less effective in denoising (but remain somewhat effective).



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

Fig. 2: Frequencies distribution of all datasets.



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

(d) Taxi with $d = 1000$ (e) RP Taxi with $d = 1000$ (f) Kosarak

Fig. 3: Comparing empirical and analytical power spectral density of added noise (first row: KRR, second row: OUE).

Since the performance of Wiener deconvolution is related to the order of the input signal, the main goal of the AMRP method is to obtain the average performance among multiple random orders. And after averaging, the noise suppression amplitude of each frequency item is also averaged, which helps the noise suppression amplitude to be neither too large nor too small.

V. Experiments

A. Experimental Setup

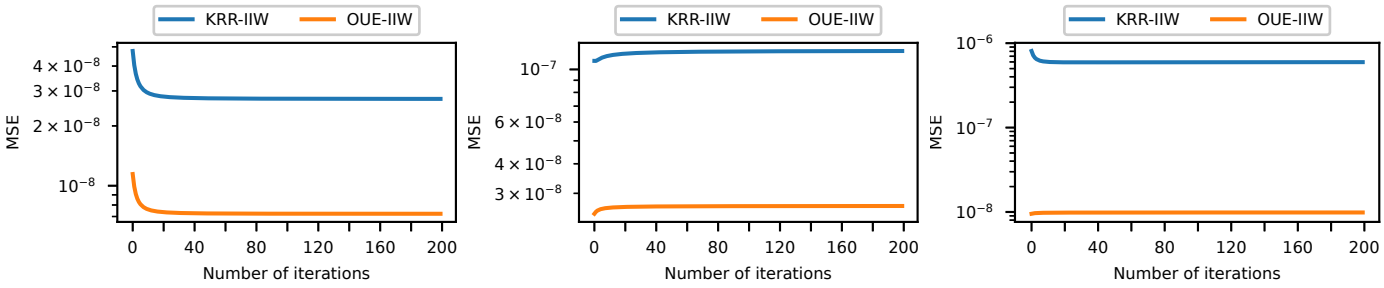
Datasets. This paper uses the following three datasets for the experiments.

Taxi pickup time dataset. Taxi pickup time dataset comes from 2021 June New York Yellow Taxi Trip Records [33]. The recorded taxi pickup time in a day is accurate to a second. There are 2,834,264 records in this dataset. Since it is composed of numerical data, it can be easily normalized to [0,1] and categorized at different granularities.

Randomly Permuted (RP) Taxi pickup time dataset. Since the Taxi pickup time dataset can be viewed as a smooth dataset, we randomly permute its frequency vectors to generate an extremely non-smooth dataset as a comparison.

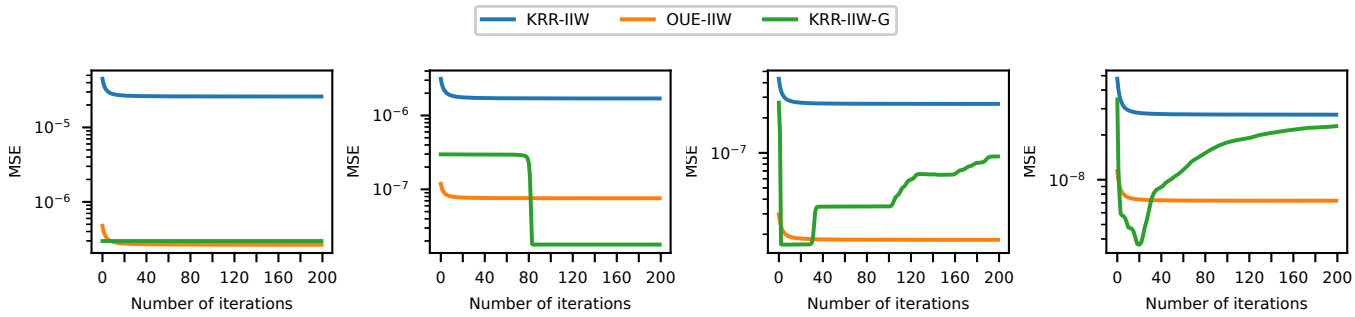
Kosarak dataset. Kosarak dataset [34] contains (anonymized) click-stream data of a hungarian on-line news portal. In total, there are 8,019,015 click events for 41,270 different webpages. This dataset is very non-smooth and the data varies dramatically, where the largest number of clicks on the same page is 601,374, while the 10th largest is only 65,412. As the Kosarak dataset is already categorized, it has a fixed domain size of 41,270.

The Taxi dataset is used as a representative of a continuous or smooth dataset and the RP Taxi dataset is used as a



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

Fig. 4: The convergence of IIW approach with $\varepsilon = 4$.



(a) Taxi with $\varepsilon = 1$ (b) Taxi with $\varepsilon = 2$ (c) Taxi with $\varepsilon = 3$ (d) Taxi with $\varepsilon = 4$

Fig. 5: The convergence of IIW and IIW-G approach with $d = 1000$ on Taxi dataset, varying ε.

worst-case comparison. As in previous studies [9], [20], the Kosarak dataset is used as a representative of the discrete non-smooth dataset. Figure 2 shows all the datasets used for the experiments. The Taxi dataset is relatively smooth, while the RP Taxi dataset and Kosarak dataset are non-smooth and the Kosarak has more drastic data variation.

Metrics. We verify the correctness of the theoretical analysis of the convolution framework and evaluate the effects of frequency estimation in two scenarios, namely, full-domain evaluation and frequent-value evaluation. In full-domain evaluation, we estimate the frequency of each value in the entire domain. And in frequent-value evaluation, we only estimate the frequencies of top k frequent values.

We use the metrics of Mean Squared Error (MSE) to evaluate the performance of the frequency estimation protocols as in the previous study [8], [9], [11]. MSE measures the mean of the squares of the errors—that is, the average squared difference between the estimated values and the true value. And when the estimated frequencies are unbiased, the MSE is equivalent to the variance. For full-domain evaluation, we have

$$\text{MSE} = \frac{1}{d}\sum_{v \in D}(\widetilde{f_v} - f_v)^2$$

As for frequent-value evaluation, we only calculate the MSE of the k most frequent values instead of the full domain.

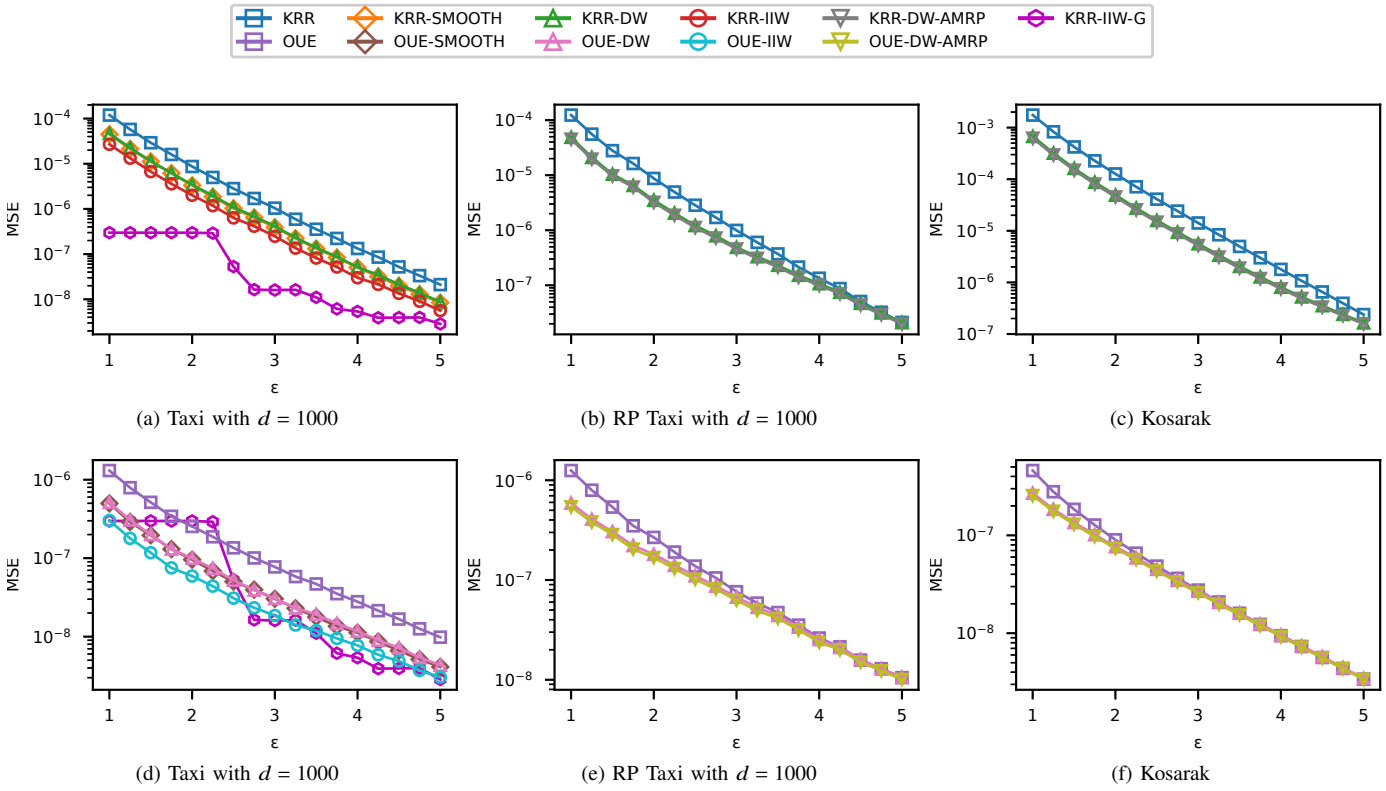
In addition, we use the number of true (false) positives to indicate the number of reliable estimates as in [9]. A true (false) positive is a category value that has a frequency above (below) a certain threshold and is estimated to have a frequency above the threshold. This metric also directly reflects the accuracy of the frequency estimation protocol used for heavy hitter identification, which aims to detect the category values whose frequencies are larger than a given threshold.

Baseline/alternative approaches. The following protocols and algorithms are compared:

KRR and OUE. KRR [8] and OUE [9] are two commonly used LDP frequency estimation protocols. We choose them as the baselines because OUE achieves the best performance in accuracy when d is large, while KRR can achieve better accuracy when d is smaller than $3e^{\varepsilon} + 2$. For more details about these two protocols please refer to Section 2.

KRR-DW, OUE-DW, KRR-IIW, KRR-IIW-G and OUE-IIW. Our proposed DW and IIW algorithms are appended to KRR and OUE separately. Without additional explanation, IIW uses $|\frac{\mathbf{G}}{\mathbf{H}}|^2$ as the initial value of the power spectrum $\widetilde{P}_{\mathbf{f}}$, because this ensures the robustness of the algorithm. And we rename the IIW method with the initial value of $|\mathbf{G}|^2$ as IIW-G. For non-smooth datasets, we only use the DW approach and additionally append the AMRP approach to improve performance.

Smoothing approach. Besides, since the Taxi dataset is relatively smooth, we use the same smoothing approach in [35] as a competitor to DW and IIW on Taxi dataset. This smoothing approach assumes continuity between adjacent data



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

(d) Taxi with $d = 1000$ (e) RP Taxi with $d = 1000$ (f) Kosarak

Fig. 6: MSE results on full-domain (first row: KRR, second row: OUE, and KRR-IIW-G in both (a) and (d)), varying ε.

items, thus averaging the estimates between adjacent data can greatly improve the accuracy. Its specific formula is as follows:

$$\widetilde{f}_i = \frac{1}{2}\widetilde{f}_i + \frac{1}{4}(\widetilde{f}_{i-1} + \widetilde{f}_{i+1}) \tag{25}$$

All the protocols and algorithms are implemented using Python 3.8.6 and Numpy 1.22.3. Because LDP protocols add randomized noise, the frequency estimated by the data collector is not completely determined each time. This means that the frequencies estimated by the data collectors will vary after each execution of the LDP protocols. To reduce randomness, we repeat each experiment 10 times to calculate the average MSE for each protocol.

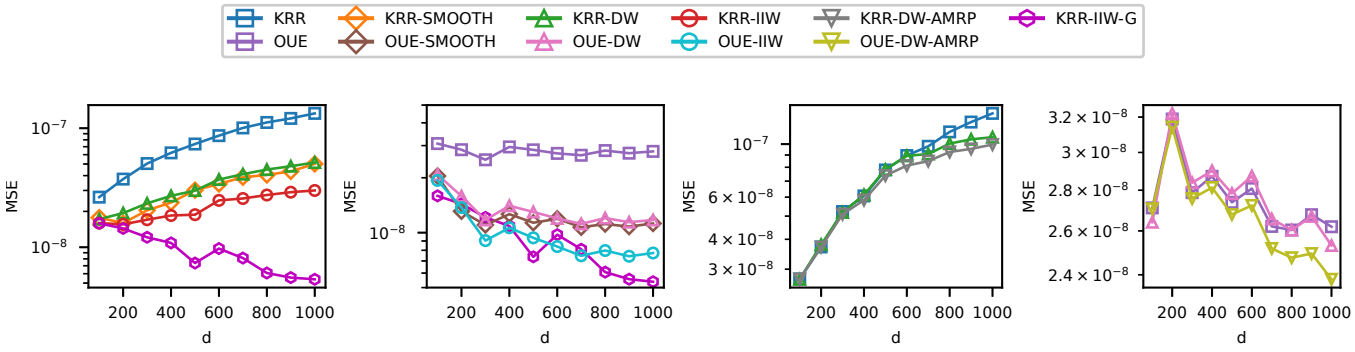
As mentioned in Section 2.2, some post-processing algorithms, mostly consistency-based, have been proposed recently to calibrate the estimation results. However, these consistency-based algorithms have very limited effectiveness in accuracy and no enhancement for the frequent values [9]. Therefore, we do not apply these algorithms to the baselines. It is worth noting that these algorithms can be used together with our proposed algorithms and may help to improve the performance.

B. Verification of Analysis

Estimation of the power spectral density of the additive noise n. In our convolution framework, the encoding and perturbation steps of existing LDP protocols are equivalently considered as a circular convolution process with added Gaussian noise. We now show that our analytical power spectrum of the added noise in collected perturbed outputs matches the actual measured noise. For empirical data, we run the LDP protocols and generate the collected perturbation outputs. We measure the added noise by subtracting the theoretical expectation from the obtained perturbation outputs, namely, $\mathbf{n} = \mathbf{g} - \mathbf{f} * \mathbf{h}$. We then calculate the power spectrum of the measured noise and compare it with the analytical noise power spectrum.

Figure 3 shows the empirical and analytical noise power spectrum for all LDP protocols with $\varepsilon = 4$. The measured power spectrum data of the noise are the values obtained from one experiment and the average of 10 experiments, respectively. The practical power spectrum is found to fluctuate up and down from our analytical value through experiments. This is consistent with our theoretical derivation since our analytical spectrum is the expected value, and it can be seen experimentally that the average measured spectrum of 10 experiments is much closer to our analytical value. The noise power spectrums of the Taxi and RP Taxi datasets are very similar because the random permutation to the same dataset has essentially no effect on the noise power spectrum. It is worth noting that both theory and experiment show that $P_0 = 0$ for KKR, but since we are using a logarithmic representation of Figure 3, the point $P_0 = 0$ is not plotted in Figure 3.

Convergence of the IIW approach. We experimentally verify the convergence of the IIW approach with $|\frac{\mathbf{G}}{\mathbf{H}}|^2$ as the initial value of $\widetilde{P}_{\mathbf{f}}$ on each dataset. Because the IIW-G approach with



(a) Taxi (b) Taxi (c) RP Taxi (d) RP Taxi

Fig. 7: MSE results on full-domain with $\varepsilon = 4$, varying d.

$|\mathbf{G}|^2$ as the initial value has a huge performance improvement of the KRR approach on the smooth dataset, we additionally verified the convergence of the IIW-G approach on the Taxi dataset.

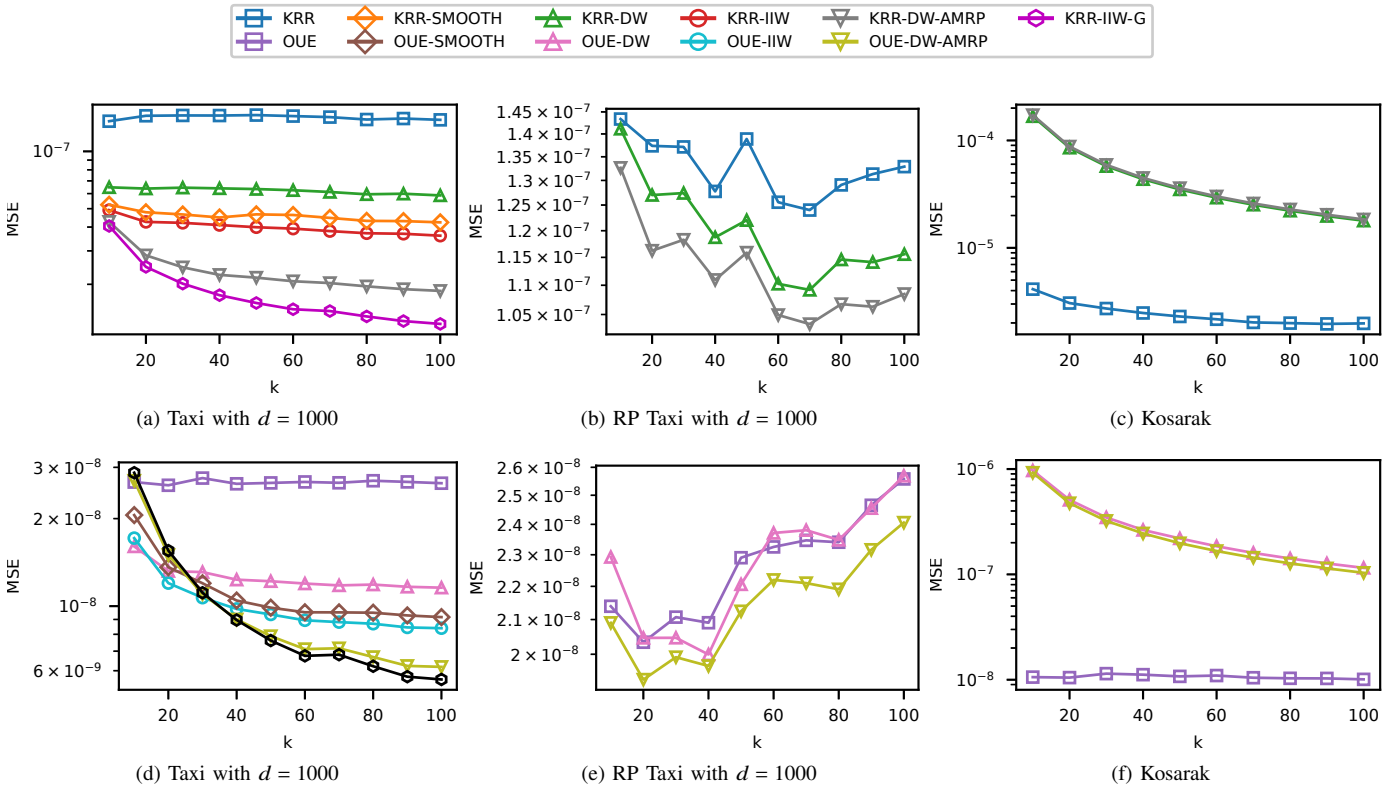
Figure 4 demonstrates the convergence of IIW approach for all datasets. We find that the IIW approach converges quickly, for KRR after a few iterations, while for OUE it can be considered converged after at most 30 iterations. Because the results of the 1st iteration of IIW are equivalent to the DW approach, we can see that the IIW approach significantly improves the MSE of DW on the smooth dataset (i.e., Taxi dataset). However, for non-smooth datasets, the IIW approach is not always working, and may even worsen the performance of DW. Therefore, we run DW and AMRP approaches on the non-smooth datasets without the IIW approach in the following experiments.

Figure 5 describes the convergence of IIW and IIW-G approaches for the Taxi dataset with different ε. The convergence rate of IIW in the experiment is unaffected by the change in ε, but the convergence rate of KRR-IIW-G slows down as ε becomes smaller. In particular, when $\varepsilon = 1$, KRR-IIW-G is stuck at the initial value even after 200 iterations. This is because when the noise amplitude is too large, the signal can be considered as a Gaussian random process with mean $1/d$ after superimposing the noise, while the initial value of g tends to be $1/d$. As a result, IIW-G is stuck and over-fitted to the noise. For $\varepsilon >= 2$, KRR-IIW-G may eventually converge to a result similar to KKR-IIW, it can obtain a much lower MSE than KRR-IIW, or even lower than OUE-IIW, during the iterative process. However, the number of iterations of the KRR-IIW-G approach to obtain the optimal MSE is not stable and is mainly influenced by the ratio of the noise to the average signal amplitude, i.e., $\frac{\sqrt{\text{Var}^*}}{1/d} = d\sqrt{\text{Var}^*}$. The ratio is 10.93, 2.95, 0.99, and 0.36 when ε=1, 2, 3, and 4 for KRR, respectively. Therefore, to ensure validity, this ratio should be lower than 1 when using KRR-IIW-G, and the number of iterations can be set to about 10 to achieve a desirable MSE.

C. Full-domain Evaluation

Different privacy budgets ε. Theoretically, the larger the privacy budget ε, the less added noise in LDP protocols, the weaker the privacy protection, and the more accurate the frequency estimation. We experiment with all protocols and approaches and varying ε from 1 to 5. The experimental results are showed in Figure 6. For the smooth Taxi dataset, we can see that the KRR-IIW-G perform significantly better than the others, including the OUE-based approaches. However, according to our experiments on the convergence of KRR-IIW-G in Figure 5, the estimates are trustworthy only when ε is greater than 2.25, and when ε is between 1 and 2.25, KRR-IIW-G is stuck at the initial value. The performance of the smoothing approach is very similar to that of the DW approach but not as good as that of the IIW approach for both KRR and OUE. When $\varepsilon = 5$, after appending to the KRR or OUE, the IIW approach reduces the MSE to one-third to one-quarter of the original, and the noise reduction effect is very stable. For both non-smooth datasets, the effect of DW noise reduction decreases with increasing ε. This is because our estimated expectation of the noise spectrum is still accurate, and the larger the noise is, the more pronounced the noise reduction effect is. The additional AMRP approach havs some performance improvement, but the improvement is weak.

Different domain size d. For the same dataset with fixed privacy budgets ε, the greater the domain size d, the higher the MSE of KRR becomes, while the MSE of OUE remains essentially the same as shown in (4) and (5). Because the domain size of the Kosarak dataset is fixed, we only experiment on the Taxi dataset and the RP Taxi dataset with varying d. Figure 7 illustrates the MSE results of compared LDP protocols and approaches on these two datasets with varying d from 100 to 1000. For the smooth dataset, KRR-IIW-G still achieves the best denoising effect. Theoretically, KRR only performs better than OUE when $d < 3e^{\varepsilon} + 2 = 166$, and KRR performs worse as d increases. However, after appending the IIW and IIW-G approaches, KRR-IIW can perform better than OUE at d = 800, and KRR-IIW-G performs even better than OUE-IIW. The performance of the smoothing approach is similar to that



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

(d) Taxi with $d = 1000$ (e) RP Taxi with $d = 1000$ (f) Kosarak

Fig. 8: MSE results on top k frequent values (first row: KRR, second row: OUE, and KRR-IIW-G in both (a) and (d)) with $\varepsilon = 4$, varying k.

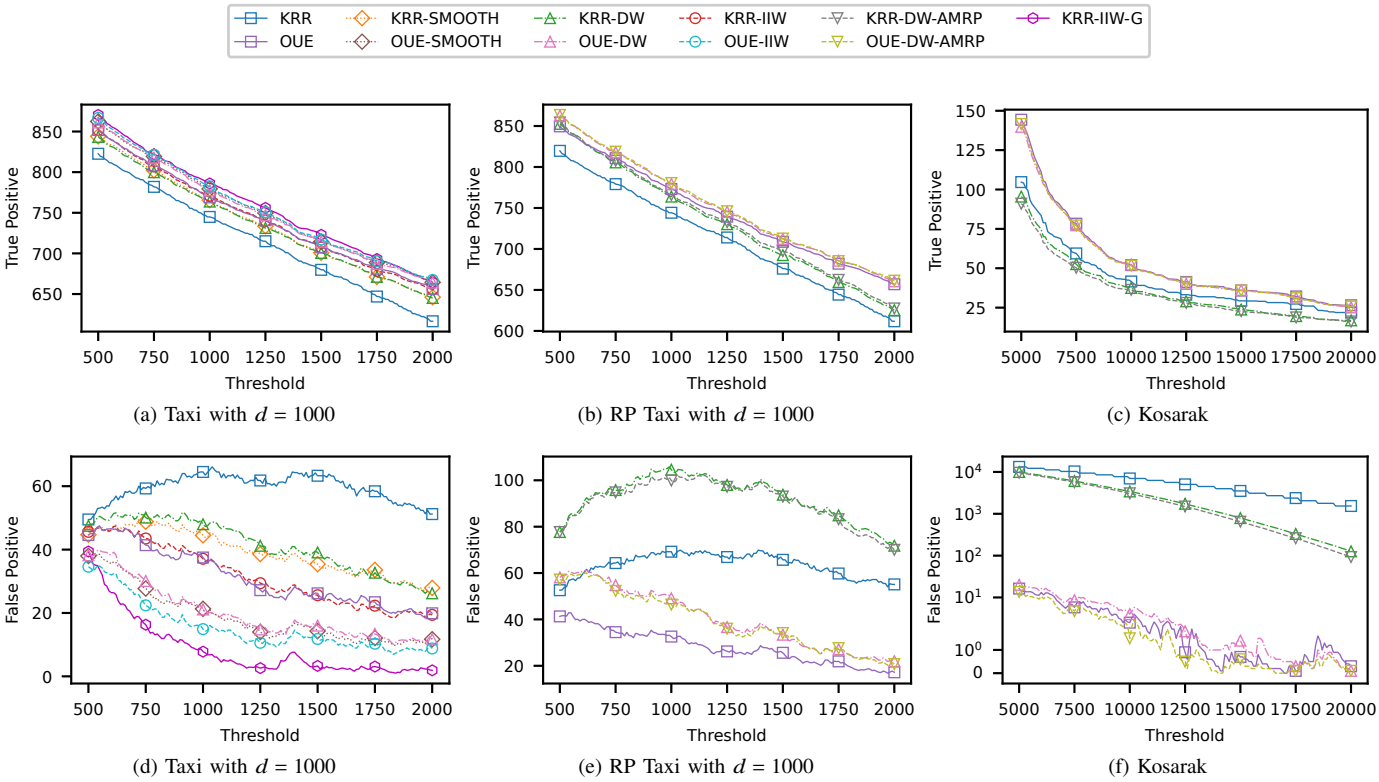
of the DW approach but worse than that of the IIW approach for both KRR and OUE. We can see that for both smooth and non-smooth datasets, the noise reduction performance of DW and IIW approaches improves with increasing d. And the AMRP approach has performance improvement to the DW approach on the RP Taxi dataset.

D. Frequent-value Evaluation

For frequent-value evaluation, we only consider the MSE of the top k values among the domain. We measure the MSE of the k most frequent value with varying k from 10 to 100, fixing $\varepsilon = 4$. The experimental results are illustrated in Figure 8. We observe that our proposed IIW algorithm performs best among all algorithms (except KRR-IIW-G) for the Taxi dataset. The performance of the smoothing approach is still similar to that of the DW approach. When $k < 30$, the performance of KRR-IIW-G on k top value is slightly weaker than OUE, but still much better than KRR. For the RP Taxi dataset, the DW approach is much less effective than the Taxi dataset, and in particular, the OUE-DW approach is even worse than the OUE at some points. However, the addition of the AMRP approach guarantees a certain enhancement for both KRR and OUE. For the Kosara dataset, both the appending DW and DW-AMRP approaches to KRR or OUE are worse than no appending for the top k value. We believe this is because the actual power spectrum $P_{\mathbf{n}}$ is found to fluctuate up and down from our estimates and eventually lead to an over-suppression of extremely large frequency values. However, these extreme points are easy to identify and we can just leave these extreme values unchanged after filtering to address this issue.

E. Distinguish True Counts from Noise

Figure 9 shows the number of true (false) positives for different approaches and datasets with $\varepsilon = 4$. For clarity, here we use count c_i as threshold instead of frequency f_i, and we have $c_i = mf_i$. For the Taxi and RP taxi datasets, since we have $m \cdot \mathrm{ST}_{\mathrm{OUE}} = 763$, $m \cdot \mathrm{ST}_{\mathrm{KRR}} = 1676$, the threshold range in our experiment is set from 500 to 2000. For the Kosara dataset, we just set the threshold range from 5000 to 20000 as in the previous study [9]. The number of true (false) counts is an average of 10 experiments. From figure 9, we can find that all approaches perform similarly to the previous full-domain evaluation experiments on the Taxi dataset. For this smooth dataset, KRR outputs the least true positives and the most false positives. In contrast, KRR-IIW-G achieves the best performance. For extremely non-smooth datasets, AMRP has improved performance over DW for both KRR and OUE. Specifically, for the RP Taxi dataset, DW and DW-AMRP output more true positives while also more false positives than the original approaches. For the Kosarak dataset, KRR-DW and KRR-DW-AMRP output about ten fewer true positives and thousands of fewer false positives, and OUE-DW-AMRP



(a) Taxi with $d = 1000$ (b) RP Taxi with $d = 1000$ (c) Kosarak

(d) Taxi with $d = 1000$ (e) RP Taxi with $d = 1000$ (f) Kosarak

Fig. 9: Number of true(false) positives with $\varepsilon = 4$

has improved performance over OUE by outputting fewer false positives.

In summary, our proposed approaches have improved accuracy for both smooth and non-smooth datasets. For smooth datasets, we recommend using the IIW approach since it is very stable and has a clear accuracy improvement. Even though the KRR-IIW-G approach may have better accuracy, it can only be used when $d\sqrt{\text{Var}^*} < 1$. For extremely non-smooth datasets, the DW-AWRP approach can be used, but the improvement in accuracy is relatively weak. To use our proposed approaches, the aggregator should know the structure of the h-vector. In most cases, the aggregator should know this as long as the aggregator can estimate the data using the equation (1). However, if the aggregator is uncertain about the structure of the h-vector, we give two possible ways to obtain the unknown h-vector for blind deconvolution in Appendix B.

VI. Conclusion

In this paper, we introduce a convolution framework for locally differentially private frequency estimation. This framework enables us to analyze and calibrate the estimated frequencies of different LDP protocols in the form of convolutions. By transforming the encoding and perturbation steps into a circular convolution process, the problem of estimating frequencies by the data collector is converted into a deconvolution problem. We propose two Wiener filter-based deconvolution approaches to boost the accuracy of existing LDP protocols. Experimental results demonstrate that the proposed approaches significantly improve the accuracy of state-of-the-art protocols on the real-world smooth dataset. For extremely non-smooth datasets, our approaches are still effective although the improvement is limited.

VII. Acknowledgment

This work is supported by the National Key R&D Program of China (2020YFE0200600). The authors also thank the reviewers and shepherd for their constructive feedback.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[2] C. T. Kenny, S. Kuriwaki, C. McCartan, E. T. Rosenman, T. Simko, and K. Imai, "The use of differential privacy for census data and its impact on redistricting: The case of the 2020 us census," *Science advances*, vol. 7, no. 41, p. eabk3283, 2021.

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

[4] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 127–135.

[5] A. D. P. Team. (2017) Learning with privacy at scale. [Online]. Available: https://machinelearning.apple.com/research/learning-with-privacy-at-scale

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

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

[8] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," in *International Conference on Machine Learning*. PMLR, 2016, pp. 2436–2444.
[9] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 729–745.
[10] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under local differential privacy," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 759–763.
[11] 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*, 2014, pp. 1054–1067.
[12] R. Bassily, K. Nissim, U. Stemmer, and A. Guha Thakurta, "Practical locally private heavy hitters," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[13] T. Wang, M. Lopuhaa-Zwakenberg, Z. Li, B. Skoric, and N. Li, "Locally differentially private frequency estimation with consistency," in *NDSS'20: Proceedings of the NDSS Symposium*, 2020.
[14] W. K. Pratt, "Generalized wiener filtering computation techniques," *IEEE Transactions on Computers*, vol. 100, no. 7, pp. 636–641, 1972.
[15] T. G. Stockham, T. M. Cannon, and R. B. Ingebretsen, "Blind deconvolution through digital signal processing," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 678–692, 1975.
[16] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 666–681, 2020.
[17] J. Dong, S. Roth, and B. Schiele, "Deep wiener deconvolution: Wiener meets deep learning for image deblurring," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1048–1059, 2020.
[18] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1218–1234, 2006.
[19] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 192–203.
[20] J. Jia and N. Z. Gong, "Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2008–2016.
[21] 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*, 2020, pp. 621–635.
[22] G. Cormode, T. Kulkarni, and D. Srivastava, "Answering range queries under local differential privacy," *Proceedings of the VLDB Endowment*, vol. 12, no. 10, pp. 1126–1138, 2019.
[23] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, and S. Jha, "Answering multi-dimensional analytical queries under local differential privacy," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 159–176.
[24] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.
[25] G. Cormode, S. Maddock, and C. Maple, "Frequency estimation under local differential privacy," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2046–2058, 2021.
[26] V. Feldman, J. Nelson, H. Nguyen, and K. Talwar, "Private frequency estimation via projective geometry," in *International Conference on Machine Learning*. PMLR, 2022, pp. 6418–6433.
[27] 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)*. IEEE, 2019, pp. 638–649.
[28] L. Lyu, Y. Li, X. He, and T. Xiao, "Towards differentially private text representations," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1813–1816.
[29] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 3, no. 1, 2010.
[30] N. Wang, X. Xiao, Y. Yang, T. D. Hoang, H. Shin, J. Shin, and G. Yu, "Privtrie: Effective frequent term discovery under local differential privacy," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 2018, pp. 821–832.
[31] R. Bassily, "Linear queries estimation with local differential privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 721–729.
[32] A. Hiller and R. T. Chin, "Iterative wiener filters for image restoration," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 1901–1904.
[33] N. Y. C. Taxi and L. Commission. (2021) Tlc trip record data. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[34] Kosarak. [Online]. Available: http://fimi.uantwerpen.be/data/
[35] D. Nychka, "Some properties of adding a smoothing step to the em algorithm," *Statistics & probability letters*, vol. 9, no. 2, pp. 187–193, 1990.

Appendix A
Derivation of Formula (1) and (2)

According to the definition of pure LDP, we have the expectation of $\sum_{k=1}^{m} \mathbb{1}_{\text{Support}(y_k)}(i)$ as

$$\begin{aligned} \mathrm{E}(\textstyle\sum_{k=1}^{m} \mathbb{1}_{\text{Support}(y_k)}(i)) &= m f_i p^* + m(1-f_i)q^* \\ &= m f_i (p^* - q^*) + m q^* \end{aligned} \tag{26}$$

Move mq^* to the left, then divide both sides by $m(p^* - q^*)$, we have:

$$\mathrm{E}\left(\frac{\sum_{k=1}^{m}(\mathbb{1}_{\text{Support}(y_k)}(i) - mq^*}{m(p^* - q^*)}\right) = f_i \tag{27}$$

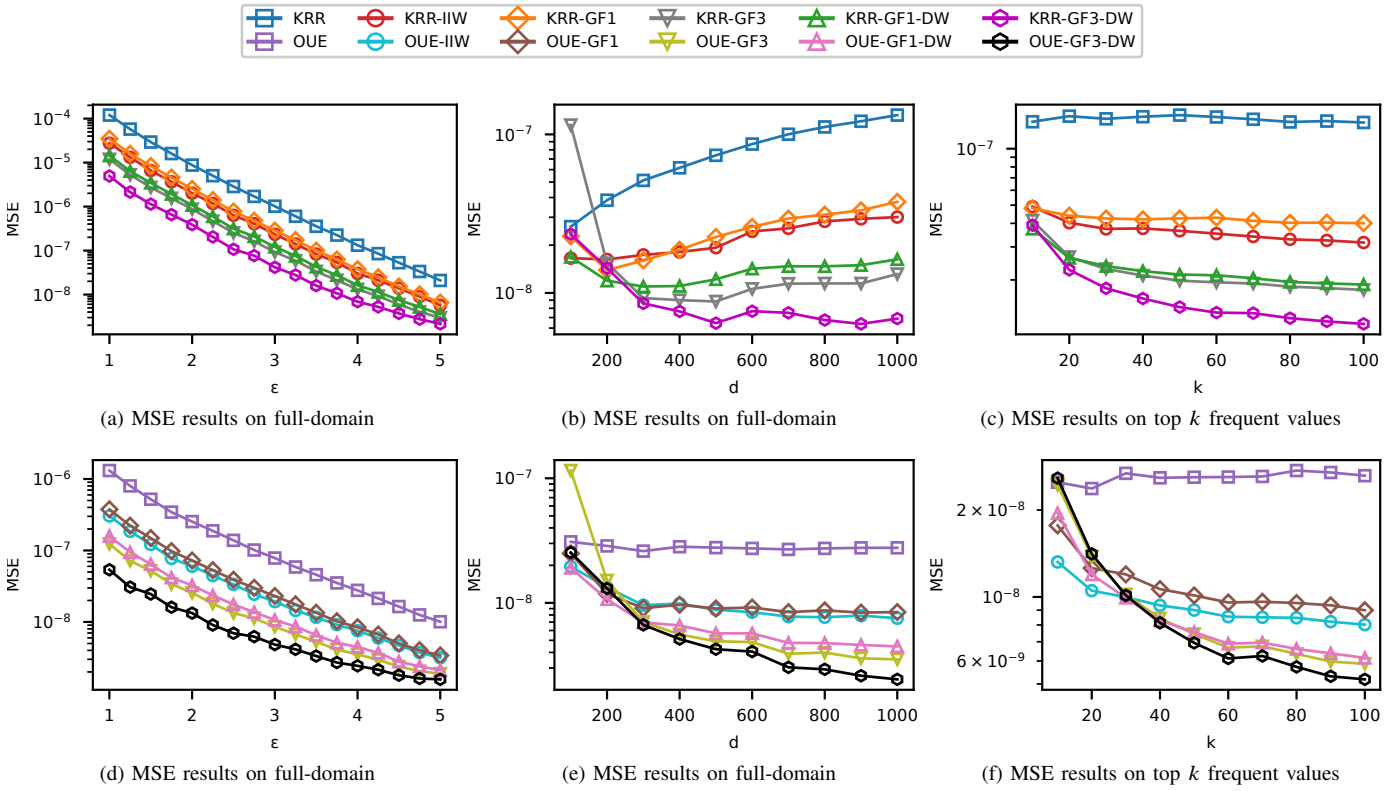
With equation (27), we get the formula (1) to estimate f_i. And to calculate the variance of $\widetilde{f_i}$, which is to calculate the variance of $\sum_{k=1}^{m} \mathbb{1}_{\text{Support}(y_k)}(i)$. We can find it can be viewed as the summation of mf_i(resp. $m(1-f_i)$) independent random variables drawn from the Bernoulli distribution with parameter p^*(resp. q^*). So we can derive $\mathrm{Var}[\widetilde{f_i}]$ as follows:

$$\begin{aligned} \mathrm{Var}[\widetilde{f_i}] &= \mathrm{Var}\left[\frac{\sum_{k=1}^{m} \mathbb{1}_{\text{Support}(y_k)}(i) - mq^*}{m(p^* - q^*)}\right] \\ &= \frac{\mathrm{Var}[\sum_{k=1}^{m} \mathbb{1}_{\text{Support}(y_k)}(i)]}{m^2(p^* - q^*)^2} \\ &= \frac{m f_i p^*(1-p^*) + m(1-f_i)q^*(1-q^*)}{m^2(p^* - q^*)^2} \\ &= \frac{q^*(1-q^*)}{m(p^* - q^*)^2} + \frac{f_i(1-p^*-q^*)}{m(p^* - q^*)} \end{aligned} \tag{28}$$

Appendix B
Blind deconvolution

Blind deconvolution is the data aggregator or the adversary inverses to convolution without knowing the structure of the h-vector. Here we give two possible ways to obtain the unknown h-vector for blind deconvolution.

1) As the perturbation of the LDP protocol is run locally, it is possible that the adversary can reuse the same input locally to obtain various outputs. After enough outputs are obtained, the adversary can estimate the h-vector by counting the frequency of each item supported by all these outputs.
2) For pure LDP protocols in which one output supports multiple inputs, the expected number of inputs supported by outputs is $m(d-1)q^* + mp^*$ with m inputs. Since the outputs are known, the number of supported inputs is also



(a) MSE results on full-domain (b) MSE results on full-domain (c) MSE results on top k frequent values

(d) MSE results on full-domain (e) MSE results on full-domain (f) MSE results on top k frequent values

Fig. 10: MSE results(first row: KRR, second row: OUE), (a) and (d) with $d = 1000$, (b) and (e) with $\varepsilon = 4$, (c) and (f) with $d = 1000$ and $\varepsilon = 4$

known, and both p^* and q^* are related to ε. Once the type of the LDP protocol used is known, we can estimate the ε and thus obtain the h-vector.

Appendix C
Smoothing Baseline with a Larger Width

We use the Gaussian filter approach with a larger smooth width for the taxi dataset, where the sigma parameters of the Gaussian filter are 1 and 3 (namely GF1 and GF3), respectively. Besides, we use the output of the Gaussian to estimate $P_{\mathbf{f}}$ and pass it to the DW method (namely GF1-DW and GF3-DW). Figure 10 gives the experimental results. For the taxi dataset, larger width can greatly improve performance when the noise is high, but when the noise is low(i.e., small domain size and large epsilon), details are lost and thus performance degrades, especially when d = 100, GF-3 performs even worse than without Gaussian filtering. Overall, GF1 performs weaker than IIW while GF3 performs better than IIW, but the best-performing method is GF3-DW, which combines Gaussian filtering with Wiener deconvolution. Since the performance of Wiener deconvolution is related to the accuracy of the initial estimate $\widetilde{P}_{\mathbf{f}}$, using the output of the Gaussian filter as the initial estimate can significantly improve the denoising ability of Wiener deconvolution for smooth datasets. And compared to GF3, GF3-DW can better preserve the details of the estimation results while removing noise. Moreover, other post-processing approaches can also be combined with Wiener deconvolution to improve performance.