Anonymized Histograms in Intermediate Privacy Models

Badih Ghazi **Pritish Kamath** **Ravi Kumar** **Pasin Manurangsi**
Google Research
Mountain View, CA, US
badihghazi@gmail.com, pritish@alum.mit.edu,
ravi.k53@gmail.com, pasin@google.com

Abstract

We study the problem of privately computing the *anonymized histogram* (a.k.a. *unattributed histogram*), which is defined as the histogram without item labels. Previous works have provided algorithms with ℓ_1 and ℓ_2^2-errors of $O_\varepsilon(\sqrt{n})$ in the central model of differential privacy (DP).

In this work, we provide an algorithm with a nearly matching error guarantee of $\widetilde{O}_\varepsilon(\sqrt{n})$ in the shuffle DP and pan-private models. Our algorithm is very simple: it just post-processes the discrete Laplace-noised histogram! Using this algorithm as a subroutine, we show applications in privately estimating symmetric properties of distributions such as entropy, support coverage, and support size.

1 Introduction

Computing histograms is among the most well-studied tasks in data analytics and machine learning. Suppose that there is a domain $[D] := \{1, \ldots, D\}$, where the ith user's input is $z_i \in [D]$. The *histogram* of the users' inputs $\{z_1, \ldots, z_n\}$ is defined as $\boldsymbol{h} := (h_1, \ldots, h_D)$ where $h_j := |\{i \in [n] \mid z_i = j\}|$, i.e., the number of users who contribute item $j \in [D]$. For many tasks, however, the different items themselves are not important and it instead suffices to consider the *anonymized histogram* (a.k.a. *unattributed histogram*) corresponding to $\boldsymbol{h}$, which is defined as $\boldsymbol{n}_{\boldsymbol{h}} := (n^{(1)}, \ldots, n^{(D)})$ where $n^{(\ell)}$ denotes the ℓth largest element among the h_j's. Whenever $\boldsymbol{h}$ is clear from context, we will skip the subscript and denote the anonymized histogram as simply $\boldsymbol{n}$.

Anonymized histograms have several applications including estimating symmetric properties of discrete distributions [10, 2, 15, 30], privately releasing the degree-distributions in social networks [33, 32, 39], and anonymizing password frequency lists [12, 13]. For more details, we refer the reader to [43] and the references therein.

In this work, we study private anonymized histograms. The notion of privacy we study is differential privacy (DP) [23, 22], which has emerged as a very popular notion of private data analysis leading to numerous practical deployments [26, 42, 29, 7, 20, 1, 35, 40].

Multiple works have studied the problem of computing private anonymized histogram, with the focus so far being on the central model of DP.[1] Moreover, two measures of error have been studied: ℓ_1-error and ℓ_2^2-error[2]. For the ℓ_2^2 case, Hay et al. [33] give an ε-DP algorithm with an expected error

[1] In the *central model* of DP, a trusted curator has access to the raw inputs of the users, and is supposed to output a DP estimate of the desired function, in this case, the anonymized histogram.

[2] Defined as $\|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_1 = \sum_{j \in [D]} |n^{(j)} - \hat{n}^{(j)}|$ and $\|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_2^2 = \sum_{j \in [D]} (n^{(j)} - \hat{n}^{(j)})^2$, where $\hat{\boldsymbol{n}}$ denotes the output anonymized histogram.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

of $\widetilde{O}(\sqrt{n}/\varepsilon^2)$[3]. As for the ℓ_1-error, Blocki et al. [12] observed that the exponential mechanism yields an expected ℓ_1-error of $O(\sqrt{n}/\varepsilon)$ since there are at most $\exp(O(\sqrt{n}))$ anonymized histograms in total [31]; recently, this bound was improved to $O(\sqrt{n\log(1/\varepsilon)}/\varepsilon)$ by Suresh [43]. On the lower bound front, Alda and Simon [5] proved a bound of $\Omega(\sqrt{n/\varepsilon})$ for the expected ℓ_1-error; recently, this was improved to $\Omega(\sqrt{n\log(1/\varepsilon)/\varepsilon})$ by Manurangsi [36], matching the aforementioned upper bound of [43] to within a constant factor. The latter lower bound also applies to (ε, δ)-DP algorithms for any sufficiently small δ (depending only on ε).

The anonymized histogram problem generalizes the COUNT-DISTINCT problem, which asks for the number of items j such that $h_j > 0$. COUNT-DISTINCT can be easily solved in the central DP model by applying the discrete Laplace mechanism. In the (non-interactive) local DP setting, Chen et al. [16] proved a lower bound of $\Omega_\varepsilon(n)$, which means that one cannot asymptotically beat the trivial algorithm that always outputs zero. The strong lower bounds on the error incurred by protocols in the local setting generally motivate the study of intermediate models of privacy including the pan-private [24] and shuffle DP [11, 25, 17] models. In these models, it turns out that COUNT-DISTINCT can be solved to within $\widetilde{O}_\varepsilon(\sqrt{n})$-error while lower bounds of $\Omega_\varepsilon(\sqrt{n})$ are known [37, 8].

In this work, we show that, surprisingly, the anonymized histogram problem (which seems much harder than COUNT-DISTINCT) can in fact be solved with essentially the same asymptotic error of $O_\varepsilon(\sqrt{n})$ as COUNT-DISTINCT, in both the pan-private and shuffle DP models. On the other hand, the aforementioned lower bound [16] for COUNT-DISTINCT also implies an $\Omega_\varepsilon(n)$ lower bound for the expected ℓ_1-error of anonymized histogram in the more challenging local DP model. In other words, in the typical case where ε is an absolute constant, it is impossible to achieve any asymptotic advantage over the trivial algorithm that always outputs the all-zeros histogram.

1.1 Our Results

A prominent approach for computing private histograms in the central DP model is to add discrete[4] Laplace noise to each histogram entry. We show that there is a post-processing algorithm that takes such a noised histogram and produces an accurate estimate of the anonymized histogram:

Theorem 1 (Informal; Theorems 9 and 21)**.** *There is an algorithm that takes in a noisy histogram, where an independent discrete Laplace noise of parameter $1/\varepsilon$ is added to each entry, and outputs an approximate anonymized histogram such that the expected ℓ_1 and ℓ_2^2-errors are*[5] $\widetilde{O}_\varepsilon(\sqrt{n+D})$.

Note that there is a dependency of $\sqrt{D}$ in the error bound in Theorem 1; when the domain size is large, this can dominate the $\sqrt{n}$ term. Fortunately, we show that this can be overcome by first randomly hashing into B buckets before computing the histogram. By picking B to be $O(n)$, we show that one can achieve an error that is $\widetilde{O}_\varepsilon(\sqrt{n})$ *without* any dependency on D:

Theorem 2 (Informal; Corollary 13)**.** *There is an algorithm that takes in a noisy hashed histogram, where an independent discrete Laplace noise of parameter $1/\varepsilon$ is added to each bucket after hashing, and outputs an approximate anonymized histogram such that the expected ℓ_1 and ℓ_2^2-errors are $\widetilde{O}_\varepsilon(\sqrt{n})$.*

Random hashing and computing discrete Laplace-noised histograms can be implemented in the pan-private[6] and shuffle DP settings [28, 9], where in the latter case we have to concede $\delta > 0$ in the privacy parameter. Thus, the theorem above yields:

Corollary 3. *For any $\varepsilon > 0, \delta \in (0, 1]$, there is an ε-DP algorithm for anonymized histogram in the pan-private model and an (ε, δ)-DP algorithm in the shuffle DP model, with expected ℓ_1 and ℓ_2^2-errors of $\widetilde{O}_\varepsilon(\sqrt{n})$.*

[3] In fact, Hay et al. [33] proved a slightly stronger expected error bound of $O(d\log^3 n/\varepsilon^2)$, where d denotes the number of unique values appearing in $\boldsymbol{n}$. Since there are at most $\sqrt{n}$ different values in the worst case, this guarantee yields an $O(\sqrt{n}\log^3 n/\varepsilon^2)$ bound for any instance.

[4] Our results can be adapted to continuous Laplace noise [23] with only a constant factor overhead in the error. We use discrete noise since it can be applied in the shuffle DP model (which is discrete in nature).

[5] $\widetilde{O}(f)$ denotes $O(f\log^c f)$ for some constant $c > 0$ and $O_\varepsilon(f)$ denotes $O(g(\varepsilon)f)$ for some function $g(\cdot)$.

[6] This is done by starting with D i.i.d. discrete Laplace r.v.'s and incrementing each entry for the next item.

As an immediate application of the above, we get algorithms for estimating symmetric properties of distributions; a distribution property is said to be *symmetric* if it remains unchanged under relabeling of the domain symbols. For any (non-private) symmetric estimator with low sensitivity, we get a private estimator in the pan-private and shuffle DP models.

Theorem 4 (Informal version of Theorem 17)**.** *For all $\varepsilon > 0, \delta \in (0, 1]$, and distributions $\mathcal{D}$, for any symmetric distribution property f, and any symmetric estimator $\hat{f}$, there exists an ε-DP mechanism $\mathcal{M}$ in the pan-private model and an (ε, δ)-DP mechanism $\mathcal{M}$ in the shuffle DP model, such that $\mathcal{M}$ outputs an α-approximation to $f(\mathcal{D})$ with high probability. The sample complexity of the mechanism $\mathcal{M}$ is given as $C_{\hat{f}}(f, \alpha) + D_{\hat{f}}(\alpha, \varepsilon)$, where the first term is the non-private sample complexity of $\hat{f}$ and the second term depends on the sensitivity of $\hat{f}$.*

In particular, in Section 5, we apply the above result to estimate the Shannon entropy and support coverage of discrete distributions, which to the best of our knowledge, is the first such sample complexity bound in the pan-private and shuffle DP models.

1.2 Overview of Techniques

We now describe the high-level ideas of our algorithms and proofs. For ease of exposition, will be intentionally informal; all details are formalized in later sections. To describe our algorithm, we need definitions of prevalence and cumulative prevalence.

Definition 5 (Prevalence and Cumulative Prevalence)**.** The *prevalence* of a histogram $\boldsymbol{h}$ is defined as $\boldsymbol{\varphi}^{\boldsymbol{h}} := (\varphi_0^{\boldsymbol{h}}, \ldots, \varphi_n^{\boldsymbol{h}})$, where $\varphi_r^{\boldsymbol{h}} := |\{j \in [D] : h_j = r\}|$ is the number of entries with value r. The *cumulative prevalence* of a histogram $\boldsymbol{h}$ is defined as $\boldsymbol{\varphi}_{\geq}^{\boldsymbol{h}} := (\varphi_{\geq 1}^{\boldsymbol{h}}, \ldots, \varphi_{\geq n}^{\boldsymbol{h}})$, where $\varphi_{\geq r}^{\boldsymbol{h}} := |\{j \in [D] : h_j \geq r\}|$ is the number of entries with value at least r.

Prevalence and cumulative prevalence can be similarly defined for an anonymized histogram $\boldsymbol{n}$; note that $\boldsymbol{\varphi}^{\boldsymbol{h}} = \boldsymbol{\varphi}^{\boldsymbol{n}_{\boldsymbol{h}}}$ and $\boldsymbol{\varphi}_{\geq}^{\boldsymbol{h}} = \boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}_{\boldsymbol{h}}}$. An important property of cumulative prevalence is that it preserves the ℓ_1-distance.

Observation 6. For all anonymized histograms $\boldsymbol{n}, \hat{\boldsymbol{n}}$, it holds that $\|\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}} - \boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}}\|_1 = \|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_1$.

We stress that, while cumulative prevalence has been used before in DP algorithms for computing anonymized histograms [43, 36], these algorithms require access to the true anonymized histogram first and therefore will only work in the central DP model.

Algorithm for $\boldsymbol{\ell_1}$-error. For each $j \in [D]$, using the discrete Laplace-noised count, we produce an unbiased estimate for whether $h_j \geq r$ for each $r \in [n]$. Adding these up over all $j \in [D]$ gives an unbiased estimate for $\varphi_{\geq r}^{\boldsymbol{h}}$. We then "project" the estimated $\boldsymbol{\varphi}_{\geq}$ back so that it corresponds to a valid anonymized histogram. It can be seen that this last step can at most double the error.

While the algorithm described above is simple, it is unclear why it incurs an error of $\widetilde{O}_{\varepsilon}(\sqrt{n + D})$. The analysis turns out to be subtle and delicate. The key is that the unbiased estimator we use has variance that decreases exponentially with $|h_j - r|$. (In other words, the uncertainty is only when h_j is close to r.) Roughly speaking, this means that the total error is dominated by the error in the case where $h_j = r$. Suppose for simplicity that we only focus on this case. Since there are $\varphi_r^{\boldsymbol{h}}$ entries satisfying the condition, the expected ℓ_1-error for $\boldsymbol{\varphi}_{\geq r}$ will be $\widetilde{O}_{\varepsilon}(\sqrt{\varphi_r^{\boldsymbol{h}}})$. Thus, in total, the ℓ_1-error of the estimated cumulative prevalence is dominated by $\widetilde{O}_{\varepsilon}(\sum_{r \in [n]} \sqrt{\varphi_r^{\boldsymbol{h}}})$. We can now apply the Cauchy–Schwarz inequality to yield $\sum_{r \in [n]} \sqrt{\varphi_r^{\boldsymbol{h}}} \leq \sqrt{\sum_{r \in [n]} 1/r} \cdot \sqrt{\sum_{r \in [n]} r \cdot \varphi_r^{\boldsymbol{h}}} = \Theta(\sqrt{n \log n})$, where the last equality follows from the fact that $\sum_{r \in [n]} r \cdot \varphi_r^{\boldsymbol{h}}$ is simply the total counts in the histogram. See Section 3 for details.

Handling Large Domain Sizes. When $D \gg n$, we randomly hash into $B = \widetilde{O}(n)$ buckets and compute the noisy "reduced" histogram on these B buckets. We can use our approach above to compute the anonymized histogram on these B buckets with ℓ_1-error at most $\widetilde{O}_{\varepsilon}(\sqrt{n})$. While this is a reasonable approach, it does not yet give a good estimate for the original anonymized histogram: the reason is that there could be as many as $\widetilde{\Omega}(n)$ collisions due to hashing. To handle this, we define

a function that "inverts" the reduced anonymized histogram to the original anonymized histogram. We then show that (i) this inverse has "sensitivity" $O(1)$ and (ii) the reduced histogram is concentrated around its mean with an ℓ_1-deviation of $\widetilde{O}(\sqrt{n})$. Combining these two allows us to conclude that the inverse of the noisy anonymized histogram has expected ℓ_1-error of $\widetilde{O}_\varepsilon(\sqrt{n})$ as desired. See Section 4 for details.

Privately Computing Symmetric Properties of Distributions. A large body of work starting with [2] has shown that plug-in estimators using the so-called profile maximum likelihood (PML) distribution achieve nearly optimal sample complexity for many symmetric distribution properties. At the core of these works is an important fact that many symmetric distribution properties have estimators (based on the anonymized histogram) with low sensitivity. Our algorithms then apply these estimators on our private anonymized histogram. The sensitivity of the estimator, together with the ℓ_1-error bound we have shown for our anonymized histogram, immediately yield bounds on the errors of the estimators. See Section 5 for details.

Algorithm for ℓ_2^2-error. Adapting the algorithm for ℓ_2^2-error proceeds as follows: recall (e.g., by Hölder's inequality) that $\|\boldsymbol{n}-\hat{\boldsymbol{n}}\|_2^2 \leq \|\boldsymbol{n}-\hat{\boldsymbol{n}}\|_1 \cdot \|\boldsymbol{n}-\hat{\boldsymbol{n}}\|_\infty$. Notice also that the concentration of the noise implies that each discrete Laplace-noised count is within $O(\log D/\varepsilon)$ of the true value. Due to this, we may only change the last (i.e., "projection") step with an extra constraint that each entry of the output estimated anonymized histogram is within $O(\log D/\varepsilon)$ of the corresponding entry of the noisy histogram. This way we have ensured that $\|\boldsymbol{n}-\hat{\boldsymbol{n}}\|_\infty \leq O(\log D/\varepsilon)$. Combining this with our earlier bound on the expected ℓ_1-error immediately yields the desired bound on the ℓ_2^2-error. See Appendix B for details.

1.3 Other Related Work

The original paper on the pan-private model [24] also studies the problem of estimating $\varphi_r^{\boldsymbol{n}}$. However, their focus is on algorithms with small space complexity, and, if one were to sum up their error bounds for all r directly, it would yield a trivial bound of $\Omega(n)$ on the total error in the anonymized histogram.

Several recent works have studied the testing/estimation/learning of symmetric and other properties of distributions under privacy constraints, mostly in the central DP model [3, 4, 6, 14, 19, 34] and some in the local DP model [21]. In particular, Acharya et al. [3] study privately computing symmetric distribution properties in the central model. Indeed, they also exploit the fact that the estimators have low sensitivity. However, in the central DP model, low sensitivity allows one to get a private estimate by adding Laplace noise directly to the non-private estimate, whereas we need to compute the estimator from our approximate anonymized histogram.

2 Differential Privacy

In this section, we review the basics of differential privacy (DP) and the shuffle DP and the pan-private models. Let $[n]$ denote the set $\{1, \ldots, n\}$ and let $\mathbf{1}[\cdot]$ denote the binary indicator function.

Two datasets $S = \{z_1, \ldots, z_n\}$ and $S' = \{z'_1, \ldots, z'_n\}$ are said to be *neighboring*, denoted $S \sim S'$, if there is an index $i \in [n]$ such that $z_j = z'_j$ if and only if $j \in [n] \setminus \{i\}$. We recall the following definition [23, 22]:

Definition 7 (Differential Privacy (DP))**.** Let $\varepsilon > 0$ and $\delta \in [0, 1]$. A randomized algorithm $\mathcal{M} : \mathcal{Z}^n \to \mathcal{R}$ is (ε, δ)-*differentially private* ((ε, δ)-*DP*) if, for all $S \sim S'$ and all (measurable) outcomes $E \subseteq \mathcal{R}$, we have that $\Pr[\mathcal{M}(S) \in E] \leq e^\varepsilon \cdot \Pr[\mathcal{M}(S') \in E] + \delta$.

We denote $(\varepsilon, 0)$-DP as ε-DP or *pure*-DP. The case when $\delta > 0$ is referred to as *approximate*-DP.

In the *central DP* model, all the inputs are stored and processed by an analyzer and the privacy is enforced only on the output of the analyzer.

Shuffle DP [11, 25, 17]. In the *shuffle DP* model, there are three algorithms, namely, a local randomizer $\mathcal{R}$, a shuffler $\mathcal{S}$, and an analyzer $\mathcal{A}$. Let $S = \{z_1, \ldots, z_n\}$ be the input dataset. The randomizer $\mathcal{R}$ takes $z \in S$ as input and outputs a set of messages. The shuffler $\mathcal{S}$ takes the multiset

of messages obtained from $\mathcal{R}$ applied to each $z \in S$ and permutes them randomly. The analyzer $\mathcal{A}$ takes this permuted multiset and computes the final output. The privacy in the shuffle model is enforced on the output of the shuffler $\mathcal{S}$, when a single input is changed.

Pan-privacy [24]. In the pan-private model, there is an algorithm that takes in a data stream of unbounded length consisting of elements in the domain. It is required that the internal state of the algorithm after any number of steps satisfies ε-DP over the data stream prefix until that step.

Histograms in Central DP. For a distribution D, let $x \sim D$ denote that the random variable x is chosen from D. For $p \in (0, 1)$, the *discrete Laplace distribution* (aka symmetric geometric distribution), denoted by $\mathrm{DLap}(p)$, is the distribution supported on $\mathbb{Z}$ whose probability mass at $i \in \mathbb{Z}$ is $\frac{1-p}{1+p} \cdot p^{|i|}$.

We use the following well-known fact about central DP and histograms.

Fact 8. *The algorithm that adds $\mathrm{DLap}(e^{-\varepsilon/2})$ noise to each entry of a histogram is ε-DP in the central model.*

We refer to the output of this algorithm as the *discrete Laplace-noised* histogram.

For a histogram $\boldsymbol{h}$, we use $\boldsymbol{n_h}$ to denote the anonymized histogram corresponding to $\boldsymbol{h}$, often dropping the subscript whenever $\boldsymbol{h}$ is clear from context.

3 Post-Processing Noised Histogram

In this section we describe our main algorithm, which obtains an anonymized histogram by suitably post-processing a noised histogram. We first define the following function $f : \mathbb{Z} \to \mathbb{R}$, which is used in our post-processing method described in Algorithm 1.

$$f(m) = \begin{cases} 1 & \text{if } m > 0, \\ 1 + \frac{p}{(1-p)^2} & \text{if } m = 0, \\ -\frac{p}{(1-p)^2} & \text{if } m = -1, \\ 0 & \text{if } m < -1. \end{cases} \tag{1}$$

Algorithm 1 Anonymized Histogram Estimator w.r.t. ℓ_1 loss.

Input: Discrete Laplace-noised histogram $\boldsymbol{h}'$, i.e., $h'_j \sim h_j + \mathrm{DLap}(p)$

for $r \in [n]$ **do**
  $\hat{\varphi}_{\geq r} \leftarrow \sum_{j \in [D]} f(h'_j - r)$, where f is defined in (1)
return $\hat{\boldsymbol{n}}$ that minimizes $\|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1$.

We give an efficient implementation of the above algorithm in Algorithm 5 (Appendix D.1), which runs in time $O(D + n \log n)$. The main idea is to compute $\hat{\boldsymbol{n}}$ that minimizes $\|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1$ using ℓ_1-isotonic regression. We now state the main guarantee of the post-processing method.

Theorem 9. *For all histograms $\boldsymbol{h}$ with $\|\boldsymbol{h}\|_1 = n$, the estimate $\hat{\boldsymbol{n}}$ returned by Algorithm 1 satisfies*

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] \;\leq\; O\left(\sqrt{C_p(n + D)\log n}\right) .$$

where, $C_p := \frac{p^2}{(1-p)^5} + \frac{p}{1-p}$.

From Fact 8, for ε-DP, we may set $p = e^{-\varepsilon/2}$. Theorem 9 then gives a bound of $O(\sqrt{(n + D)\log n}/\varepsilon^{2.5})$.

The crucial result needed in our analysis is the guarantee of the individual estimator f. We show that $f(h'_j - r)$ is an unbiased estimator of $\mathbf{1}[h_j \geq r]$ and furthermore its variance decreases (exponentially) as $|h_j - r|$ increases.

Lemma 10. *For all $h, r \in \mathbb{N} \cup \{0\}$, if $h' \sim h + \mathrm{DLap}(p)$, then it holds that*

$$\mathbb{E}[f(h' - r)] \;=\; \mathbf{1}[h \geq r], \quad \textit{and} \tag{2}$$

$$\mathrm{Var}[f(h' - r)] \;\leq\; 4p^{|h-r|+1}\left(\tfrac{p}{(1-p)^3} + (1-p)\right). \tag{3}$$

Proof. Let $\tau := h - r, \tau' := h' - r$, and $x := p/(1-p)^2$. Consider $g : \mathbb{Z} \to \mathbb{R}$ defined by $g(m) := f(m) - 1/2$. Notice that (2) is equivalent to $\mathbb{E}[g(\tau')] = \mathbf{1}[\tau \geq 0] - 1/2$. Due to symmetry, it suffices to consider the case where $\tau \geq 0$. In this case, we have

$$\mathbb{E}[g(\tau')] = \mathbb{E}_{Z\sim\mathrm{DLap}(p)}[g(\tau + Z)] = \Pr[Z > -\tau] \cdot (1/2) + \Pr[Z \leq -\tau] \cdot \mathbb{E}[g(\tau + Z) \mid Z \leq -\tau].$$

The last term can be expanded as follows:

$$\begin{aligned}
&\mathbb{E}[g(\tau + Z) \mid Z \leq -\tau] \\
&= (1/2 + x) \cdot \Pr[Z = -\tau \mid Z \leq -\tau] - (1/2 + x) \cdot \Pr[Z = -\tau - 1 \mid Z \leq -\tau] \\
&\quad - (1/2) \cdot \Pr[Z < -\tau - 1 \mid Z \leq -\tau] \\
&= (1/2 + x) \cdot (1 - p) - (1/2 + x) \cdot p(1 - p) - (1/2) \cdot p^2 = 1/2.
\end{aligned}$$

Combining the two equations above, we arrive at $\mathbb{E}[g(\tau')] = \Pr[Z > -\tau] \cdot (1/2) + \Pr[Z \leq -\tau] \cdot (1/2) = 1/2$, thereby proving (2).

To prove (3), notice again that $\mathrm{Var}[f(\tau')] = \mathrm{Var}[g(\tau')]$. Again, due to symmetry, we may only consider the case $\tau \geq 0$. Here, we have

$$\begin{aligned}
\mathrm{Var}[g(\tau')] \;&=\; \mathbb{E}_{Z\sim\mathrm{DLap}(p)}[(g(\tau + Z) - 1/2)^2] \\
&=\; \Pr[Z \leq -\tau] \cdot \mathbb{E}[(g(\tau + Z) - 1/2)^2 \mid Z \leq -\tau] \\
&\leq\; p^\tau \cdot \mathbb{E}[(g(\tau + Z) - 1/2)^2 \mid Z \leq -\tau].
\end{aligned}$$

Similar to before, we can expand the last term as

$$\begin{aligned}
&\mathbb{E}[(g(\tau + Z) - 1/2)^2 \mid Z \leq -\tau] \\
&= x^2 \cdot \Pr[Z = -\tau \mid Z \leq -\tau] + (1 + x)^2 \cdot \Pr[Z = -\tau - 1 \mid Z \leq -\tau] \\
&\quad + 1 \cdot \Pr[Z < -\tau - 1 \mid Z \leq -\tau] \\
&= x^2 \cdot (1 - p) + (1 + x)^2 \cdot p(1 - p) + p^2 \\
&\leq p^2/(1 - p)^3 + 2(1 + x^2) \cdot p(1 - p) + p^2 \\
&\leq 4p^2/(1 - p)^3 + 2p(1 - p).
\end{aligned}$$

Plugging this into the above, we get $\mathrm{Var}[g(\tau')] \leq 4p^{\tau+1}(p/(1-p)^3 + (1-p))$ as desired. □

The following is an immediate consequence of Lemma 10, by summing over all $j \in [D]$.

Observation 11. For all $r \in [n]$, and $\kappa := 4p\left(\frac{p}{(1-p)^3} + (1-p)\right)$ it holds that

$$\mathbb{E}[\hat{\varphi}_{\geq r}] \;=\; \varphi^{\boldsymbol{n}}_{\geq r} \quad \text{and} \quad \mathrm{Var}[\hat{\varphi}_{\geq r}] \;\leq\; \textstyle\sum_{\ell=0}^{n} \kappa \cdot p^{|\ell - r|} \cdot \varphi^{\boldsymbol{n}}_{\ell}\,.$$

Proof of Theorem 9. The error is bounded as

$$\begin{aligned}
&\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 \;=\; \|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \boldsymbol{\varphi}^{\boldsymbol{n}}_{\geq}\|_1 \;\leq\; \|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1 + \|\hat{\boldsymbol{\varphi}}_{\geq} - \boldsymbol{\varphi}^{\boldsymbol{n}}_{\geq}\|_1 \;\leq\; 2 \cdot \|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1 \\
\implies\; &\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 \;\leq\; 2 \cdot \textstyle\sum_{r\in[n]} |\varphi^{\hat{\boldsymbol{n}}}_{\geq r} - \hat{\varphi}_{\geq r}|,
\end{aligned} \tag{4}$$

where we use that $\|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1 \leq \|\hat{\boldsymbol{\varphi}}_{\geq} - \boldsymbol{\varphi}^{\boldsymbol{n}}_{\geq}\|_1$ by our choice of $\hat{\boldsymbol{n}}$. From Observation 11, we have

$$\mathbb{E}\left[|\varphi^{\hat{n}}_{\geq r} - \hat{\varphi}_{\geq r}|\right] \;\leq\; \sqrt{\mathrm{Var}[\hat{\varphi}_{\geq r}]} \;\leq\; \sqrt{\kappa} \cdot \sqrt{\textstyle\sum_{\ell=0}^{n} p^{|\ell-r|} \cdot \varphi^{\boldsymbol{n}}_{\ell}}.$$

Combining this with (4), we have

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] \;\leq\; 2 \cdot \textstyle\sum_{r\in[n]} \mathbb{E}\left[|\varphi^{\hat{n}}_{\geq r} - \hat{\varphi}_{\geq r}|\right]$$

$$
\begin{aligned}
&\le 2\sqrt{\kappa}\cdot\left(\sum_{r\in[n]}\sqrt{\sum_{\ell=0}^{n} p^{|\ell-r|}\cdot\varphi_\ell^{\boldsymbol{n}}}\right)\\
&\le 2\sqrt{\kappa}\cdot\sqrt{\sum_{r\in[n]}\tfrac{1}{r}}\cdot\sqrt{\sum_{r\in[n]} r\cdot\left(\sum_{\ell=0}^{n} p^{|\ell-r|}\cdot\varphi_\ell^{\boldsymbol{n}}\right)} \quad \text{(Cauchy–Schwarz)}\\
&\le 2\sqrt{\kappa}\cdot O(\sqrt{\log n})\cdot\sqrt{\sum_{\ell=0}^{n}\varphi_\ell^{\boldsymbol{n}}\cdot\left(\sum_{r\in[n]} r\cdot p^{|\ell-r|}\right)}\\
&\le 2\sqrt{\kappa}\cdot O(\sqrt{\log n})\cdot\sqrt{\sum_{\ell=0}^{n}\varphi_\ell^{\boldsymbol{n}}\cdot 2(\ell+1)\cdot\left(\sum_{t=0}^{\infty}(t+1)\cdot p^t\right)}\\
&= 2\sqrt{\kappa}\cdot O(\sqrt{\log n})\cdot\sqrt{\sum_{\ell=0}^{n}\varphi_\ell^{\boldsymbol{n}}\cdot 2(\ell+1)\cdot(1/(1-p)^2)}\\
&= 2\sqrt{\kappa/(1-p)^2}\cdot O(\sqrt{\log n})\cdot\sqrt{\left(\sum_{\ell=0}^{n}\ell\cdot\varphi_\ell^{\boldsymbol{n}}\right)+\left(\sum_{\ell=0}^{n}\varphi_\ell^{\boldsymbol{n}}\right)}\\
&= 2\sqrt{\kappa/(1-p)^2}\cdot O(\sqrt{\log n})\cdot\sqrt{n+D}. \qquad \square
\end{aligned}
$$

4 Reducing Domain Size via Hashing

In this section we propose an algorithm to handle the case where $D \gg n$. The approach in this case is to hash the domain into something smaller. Let $B \in \mathbb{N}$ be the number of hash values. The distribution of the anonymized histogram produced after random hashing into B buckets is equivalent to the following process:

- Let $\boldsymbol{n} = (n^{(1)}, \ldots, n^{(D)})$ be the starting anonymized histogram.
- Pick a uniformly random hash function $H : [D] \to [B]$.
- Let $\boldsymbol{h}^{\mathrm{red}} := (h_1^{\mathrm{red}}, \ldots, h_B^{\mathrm{red}})$ denote the reduced histogram given by $h_i^{\mathrm{red}} = \sum_{j\in H^{-1}(i)} n^{(j)}$.
- Let $\boldsymbol{n}^{\mathrm{red}}$ denote the corresponding anonymized histogram.

Let Γ^B be the mapping from $\boldsymbol{n}$ to $\mathbb{E}[\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}}]$ where $\boldsymbol{n}^{\mathrm{red}}$ is generated as above, and the expectation is over the choice of random hash functions H. With this notation, we present Algorithm 2.

Algorithm 2 Anonymized Histogram Estimator w.r.t. ℓ_1 loss, for large domains.

Input: Discrete Laplace-noised histogram $\tilde{\boldsymbol{h}}^{\mathrm{red}}$, i.e., $\tilde{h}_j^{\mathrm{red}} \sim h_j^{\mathrm{red}} + \mathrm{DLap}(p)$

Compute an estimate $\hat{\boldsymbol{n}}^{\mathrm{red}}$ of $\boldsymbol{n}^{\mathrm{red}}$ using Algorithm 1

return $\hat{\boldsymbol{n}}$ that minimizes $\|\Gamma^B(\hat{\boldsymbol{n}}) - \boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}^{\mathrm{red}}}\|_1$, s.t. $\|\hat{\boldsymbol{n}}\|_1 = n$

We give an efficient implementation of (a variant of) the above algorithm in Algorithm 7 (Appendix D.2), which runs in time $\widetilde{O}_\varepsilon(D + n\log n)$. The main result of this section is the following:

Theorem 12. *For all $B > 4n$ and histograms $\boldsymbol{h}$ with $\|\boldsymbol{h}\|_1 \le n$, the estimate $\hat{\boldsymbol{n}}$ returned by Algorithm 2 satisfies*

$$
\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] &\le O(\mathbb{E}[\|\hat{\boldsymbol{n}}^{\mathrm{red}} - \boldsymbol{n}^{\mathrm{red}}\|_1] + n\log n/\sqrt{B})\\
&\le O\left(\sqrt{C_p(n+B)\log n} + n\log n/\sqrt{B}\right). \qquad \textit{(using Theorem 9)}
\end{aligned}
$$

By setting $B = n\sqrt{\log n}$, we get the following corollary.

Corollary 13. *For all $\varepsilon > 0$, Algorithm 2 for $p = e^{-\varepsilon/2}$ and $B = n\sqrt{\log n}$ is an ε-DP algorithm, and achieves an expected ℓ_1-error of $E(n,\varepsilon) = O(\sqrt{n}(\log n)^{3/4}/\varepsilon^{2.5})$.*[7]

We describe the main steps in the proof of Theorem 12.

Lipschitzness of Inverse of Γ^B. We start by showing that the "inverse" of Γ^B is $O(1)$-Lipschitz:

Lemma 14. *[Proof in Appendix A.1] For $B > 4n$ and all anonymized histograms $\boldsymbol{n}, \boldsymbol{n}'$ with $\|\boldsymbol{n}\|_1, \|\boldsymbol{n}'\|_1 \le n$,*

$$\|\Gamma^B(\boldsymbol{n}) - \Gamma^B(\boldsymbol{n}')\|_1 \ge \|\boldsymbol{n} - \boldsymbol{n}'\|_1/4\,.$$

[7] By choosing $B = n\sqrt{\varepsilon^{2.5}\log n}$, the bound in Corollary 13 could be improved to $O(\sqrt{n}((\log n)^{3/4}/\varepsilon^{1.25} + (\log n)^{1/2}/\varepsilon^{2.5}))$; we state the simpler bound for brevity.

Concentration of $\boldsymbol{n}^{\mathrm{red}}$. We now bound the expected ℓ_1-distance between $\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}}$ and its expectation $\Gamma^B(\boldsymbol{n})$.

Lemma 15. *Assume that $B \geq 2n$. Then, $\mathbb{E}[\|\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}} - \Gamma^B(\boldsymbol{n})\|_1] \leq O(n \log n/\sqrt{B})$.*

Proof. We have

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}} - \Gamma^B(\boldsymbol{n})\|_1] &= \textstyle\sum_{r\in[n]} \mathbb{E}[|\varphi_{\geq r}^{\boldsymbol{n}^{\mathrm{red}}} - \Gamma^B(\boldsymbol{n})_r|] \\
&= \textstyle\sum_{r\in[n]} \mathbb{E}[|\varphi_{\geq r}^{\boldsymbol{n}^{\mathrm{red}}} - \mathbb{E}[\varphi_{\geq r}^{\boldsymbol{n}^{\mathrm{red}}}]|] \\
&\leq \textstyle\sum_{r\in[n]} \sqrt{\mathrm{Var}[\varphi_{\geq r}^{\boldsymbol{n}^{\mathrm{red}}}]}. \qquad (5)
\end{aligned}$$

We next use the following bound on the variance.

Lemma 16. *[Proof in Appendix A.2] For all $r \in [n]$, $\mathrm{Var}[\varphi_{\geq r}^{\boldsymbol{n}^{\mathrm{red}}}] \leq \frac{16n}{B} \cdot \left(\frac{n}{r^2} + \sum_{t\in[r-1]} \frac{t\cdot\varphi_t^{\boldsymbol{n}}}{r(r-t)}\right)$.*

Plugging Lemma 16 into (5), we have

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}} - \Gamma^B(\boldsymbol{n})\|_1] &\leq O\left(\sqrt{n/B}\right) \cdot \textstyle\sum_{r\in[n]} \sqrt{\frac{n}{r^2} + \sum_{t\in[r-1]} \frac{t\cdot\varphi_t^{\boldsymbol{n}}}{r(r-t)}} \\
\text{(Cauchy–Schwarz)} \quad &\leq O\left(\sqrt{n/B}\right) \cdot \sqrt{\textstyle\sum_{r\in[n]} \frac{1}{r}} \sqrt{\textstyle\sum_{r\in[n]} r \cdot \left(\frac{n}{r^2} + \sum_{t\in[r-1]} \frac{t\cdot\varphi_t^{\boldsymbol{n}}}{r(r-t)}\right)} \\
&= O\left(\sqrt{n\log(n)/B}\right) \sqrt{\textstyle\sum_{r\in[n]} \left(\frac{n}{r} + \sum_{t\in[r-1]} \frac{t\cdot\varphi_t^{\boldsymbol{n}}}{r-t}\right)} \\
&= O\left(\sqrt{n\log(n)/B}\right) \sqrt{O(n\log n) + \textstyle\sum_{t\in[n-1]} \sum_{\ell\in[n-t]} \frac{t\cdot\varphi_t^{\boldsymbol{n}}}{\ell}} \\
&\leq O\left(\sqrt{n\log(n)/B}\right) \sqrt{O(n\log n) + O(\log n) \cdot \textstyle\sum_{t\in[n-1]} t\cdot\varphi_t^{\boldsymbol{n}}} \\
&= O\left(\sqrt{n\log(n)/B}\right) \sqrt{O(n\log n)} \\
&= O(n\log n/\sqrt{B})\,. \qquad \square
\end{aligned}$$

Putting things together. With all the components ready, we can now prove Theorem 12.

Proof of Theorem 12. By Lemma 14, we have

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_1] &\leq 4 \cdot \mathbb{E}[\|\Gamma^B(\boldsymbol{n}) - \Gamma^B(\hat{\boldsymbol{n}})\|_1] \\
&\leq 4 \cdot \left(\mathbb{E}[\|\Gamma^B(\boldsymbol{n}) - \boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}^{\mathrm{red}}}\|_1 + \|\boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}^{\mathrm{red}}} - \Gamma^B(\hat{\boldsymbol{n}})\|_1]\right) \\
&\leq 8 \cdot \left(\mathbb{E}[\|\Gamma^B(\boldsymbol{n}) - \boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}^{\mathrm{red}}}\|_1]\right) \qquad \text{(By definition of } \hat{\boldsymbol{n}}) \\
&\leq 8 \cdot \left(\mathbb{E}[\|\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}} - \boldsymbol{\varphi}_{\geq}^{\hat{\boldsymbol{n}}^{\mathrm{red}}}\|_1 + \|\Gamma^B(\boldsymbol{n}) - \boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^{\mathrm{red}}}\|_1]\right) \\
&\leq O(\mathbb{E}[\|\boldsymbol{n}^{\mathrm{red}} - \hat{\boldsymbol{n}}^{\mathrm{red}}\|_1] + n\log n/\sqrt{B}) \quad \text{(From Lemma 15).} \qquad \square
\end{aligned}$$

5 Estimating Symmetric Properties of Discrete Distributions

In this section we show how to use Theorem 12 for the task of estimating symmetric properties of discrete distributions over $[k]$. Here, a distribution property is *symmetric* if it remains unchanged under relabeling of the domain symbols. For example, a notable such property is the Shannon entropy of a distribution $\mathcal{D}$ defined as $H(\mathcal{D}) := \sum_x \mathcal{D}(x) \log \frac{1}{\mathcal{D}(x)}$, a central object in information theory, machine learning, and statistics. If the support is unbounded, estimating $H(\mathcal{D})$ is impossible with any finite number of samples. Our goal is to estimate the entropy of a distribution $\mathcal{D} \in \Delta_k$ up to an additive $\pm\alpha$ error, where Δ_k denotes the set of all distributions over $[k]$.

One of the key ideas in the literature (e.g., [3]) is to design *low sensitivity* estimators $\hat{f} : \mathcal{X}^n \to \mathbb{R}$ for the desired symmetric distribution property $f : \Delta_k \to \mathbb{R}$. The (non-private) *sample complexity* of a property $f : \Delta_k \to \mathbb{R}$ using estimator $\hat{f}$, denoted by $C_{\hat{f}}(f, \alpha)$, is the smallest number of samples n needed to estimate $f(\mathcal{D})$ upto accuracy α with probability at least 0.9, that is, $\Pr[|\hat{f}(S) - f(\mathcal{D})| > \alpha] < 0.1$.[8] The *sensitivity* of an estimator $\hat{f}$ is $\Delta_{n,\hat{f}}$, which is the smallest value for which it holds for adjacent datasets $S \sim S'$ each with n elements, that $|\hat{f}(S) - \hat{f}(S')| \leq \Delta_{n,\hat{f}}$. Let $D_{\hat{f}}(\alpha, \varepsilon) := \min\{n \mid \Delta_{n,\hat{f}} \leq 0.1\alpha/E(n, \varepsilon)\}$, for $E(n, \varepsilon)$ defined in Corollary 13.

We will only consider *symmetric* estimators $\hat{f}$, for which we will abuse notation to use $\hat{f}(\boldsymbol{n})$ to denote $\hat{f}(S)$ for any dataset S that corresponds to the anonymized histogram $\boldsymbol{n}$.

Theorem 17. *For all $\varepsilon > 0, \delta \in (0, 1]$, for any symmetric distribution property f, and any symmetric estimator $\hat{f}$, there exists an ε-DP mechanism in the pan-private model and (ε, δ)-DP mechanism in the shuffle DP model, such that $\Pr_{S\sim\mathcal{D}^n}[|\mathcal{M}(S) - f(\mathcal{D})| > \alpha] < 0.2$ with sample complexity*

$$O\left(C_{\hat{f}}\left(f, \frac{\alpha}{2}\right) + D_{\hat{f}}\left(\frac{\alpha}{2}, \varepsilon\right)\right).$$

Proof. Let $\boldsymbol{n}$ denote the anonymized histogram corresponding to the sampled dataset S. The mechanism $\mathcal{M}$ simply outputs $\hat{f}(\hat{\boldsymbol{n}})$ for $\hat{\boldsymbol{n}}$ returned by Algorithm 2. Clearly the mechanism $\mathcal{M}$ is DP due to the post-processing property.

We have that for a suitable $n = O\left(C_{\hat{f}}\left(f, \frac{\alpha}{2}\right) + D_{\hat{f}}\left(\frac{\alpha}{2}, \varepsilon\right)\right)$, it holds that

$$\Pr\left[\left|\hat{f}(\boldsymbol{n}) - f(\mathcal{D})\right| > \frac{\alpha}{2}\right] \leq 0.1 \qquad \text{and} \qquad \Pr\left[\left|\hat{f}(\hat{\boldsymbol{n}}) - \hat{f}(\boldsymbol{n})\right| > \frac{\alpha}{2}\right] \leq 0.1,$$

where the first inequality holds by definition of $C_{\hat{f}}(f, \alpha/2)$, and the second inequality holds because $|\hat{f}(\hat{\boldsymbol{n}}) - \hat{f}(\boldsymbol{n})| \leq \Delta_{n,\hat{f}} \cdot \|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1$ and by the guarantee of Theorem 12 and Markov's inequality $\Pr[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 > 10E(n, \varepsilon)] \leq 0.1$. By a union bound, we have $\Pr[|\hat{f}(\hat{\boldsymbol{n}}) - \hat{f}(\boldsymbol{n})| > \alpha] \leq 0.44$. □

We get the following sample complexity bounds for private estimation of Shannon entropy in the pan-private and shuffle DP models, as an immediate application of Theorem 17.

Corollary 18 (Proof in Appendix C)**.** *For all $\varepsilon > 0, \delta \in (0, 1]$, there exists an ε-DP mechanism in the pan-private model and (ε, δ)-DP mechanism in the shuffle DP model, that can estimate the entropy of $\mathcal{D} \in \Delta_k$ up to an additive error of $\pm\alpha$ with a sample complexity of*

$$\min_{\lambda \in (0, \frac{1}{2})} \left\{ O\left(\frac{k}{\alpha} + \frac{\log^2 k}{\alpha^2} + \frac{\log^4(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right), \quad O\left(\frac{k}{\lambda^2 \alpha \log k} + \frac{\log^2 k}{\alpha^2} + \left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right)^{1/(1-2\lambda)}\right) \right\}.$$

These bounds have the same dependence on k as in the work of Acharya et al. [3]. The dependence on α and ε is slightly worse due to a higher cost of sensitivity in our setting, and the worse dependence on ε in our guarantees in Corollary 13.

We also derive sample complexity bounds for private estimation of *support coverage* and *support size*. The *support coverage* of a distribution $\mathcal{D}$ and an integer m is defined as $S_m(\mathcal{D}) := \sum_{x \in \text{supp}(\mathcal{D})}(1 - (1 - \mathcal{D}(x))^m)$, i.e., the expected number of distinct elements seen if we draw m i.i.d. samples from $\mathcal{D}$. Here we would like to estimate $S_m(\mathcal{D})$ to within an additive error of $\pm\alpha m$. The *support size* of a discrete distribution $\mathcal{D}$ is the number of atoms x such that $\mathcal{D}(x) > 0$. In general, this is impossible with finite sample since some atom of $\mathcal{D}$ might have an arbitrarily small probability mass. To avoid this, we follow prior work and consider only probability distributions in $\Delta_{\geq 1/K} := \{\mathcal{D} \mid \forall x \in \text{supp}(\mathcal{D}), \mathcal{D}(x) \geq 1/K\}$, i.e., those with non-zero mass of at least $1/K$ at every atom, for some K. We defer the details of the exact sample complexity bounds and the proofs to Appendix C.

[8] The choice of 0.1 is arbitrary; using the "median trick", we can boost the success probability to $1 - \beta$ with an additional multiplicative $\log(1/\beta)$ more samples.

6 Conclusions and Future Directions

In this paper we give simple algorithms for privately computing anonymized histograms. Our algorithms can be implemented in the shuffle and pan-private models. There are several immediate open questions as discussed below.

Our upper bounds have a dependency of $O(1/\varepsilon^{2.5})$ in the ℓ_1-error case and $O(1/\varepsilon^{3.5})$ in the ℓ_2^2-error case; it is unclear if these are tight. Similarly, there is a lower order multiplicative term of $O(\log n)$ and $O(\log^2 n)$ in our ℓ_1 and ℓ_2^2-error bounds respectively. Closing these gaps would be an interesting next step; these would also lead to improvements to the sample complexity bounds on private estimation of symmetric distribution properties, such as the Shannon entropy (Corollary 18).

While our analysis of the large domain case relies on the hash function being uniformly random, it is conceivable that this is not required for the approach to work. It will be interesting if this approach, perhaps with some modifications, can also work with a weaker family of hash functions such as a pairwise independent hash family.

Note also that our shuffle DP algorithm has $\delta > 0$, i.e., approximate-DP. This may not be necessary: there are a couple of recent algorithms for computing histogram with shuffle DP with $\delta = 0$ [27, 18]. Our post-processing approach does not immediately apply to these algorithms because the noise to each count is not a discrete Laplace noise (and in fact is not even an independent additive noise). Adapting our approach to their setting is another interesting research direction.

References

[1] J. M. Abowd. The US Census Bureau adopts differential privacy. In *KDD*, pages 2867–2867, 2018.

[2] J. Acharya, H. Das, A. Orlitsky, and A. T. Suresh. A unified maximum likelihood approach for estimating symmetric properties of discrete distributions. In *ICML*, pages 11–21, 2017.

[3] J. Acharya, G. Kamath, Z. Sun, and H. Zhang. INSPECTRE: privately estimating the unseen. *J. Priv. Confidentiality*, 10(2), 2020.

[4] J. Acharya, Z. Sun, and H. Zhang. Differentially private testing of identity and closeness of discrete distributions. In *NeurIPS*, pages 6879–6891, 2018.

[5] F. Aldà and H. U. Simon. A lower bound on the release of differentially private integer partitions. *IPL*, 129:1–4, 2018.

[6] M. Aliakbarpour, I. Diakonikolas, and R. Rubinfeld. Differentially private identity and equivalence testing of discrete distributions. In *ICML*, pages 169–178, 2018.

[7] Apple Differential Privacy Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 2017.

[8] V. Balcer, A. Cheu, M. Joseph, and J. Mao. Connecting robust shuffle privacy and pan-privacy. In *SODA*, pages 2384–2403, 2021.

[9] B. Balle, J. Bell, A. Gascón, and K. Nissim. Private summation in the multi-message shuffle model. In *CCS*, pages 657–676, 2020.

[10] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *FOCS*, pages 259–269, 2000.

[11] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnés, and B. Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, pages 441–459, 2017.

[12] J. Blocki, A. Datta, and J. Bonneau. Differentially private password frequency lists. In *NDSS*, 2016.

[13] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *S & P*, pages 538–552, 2012.

[14] B. Cai, C. Daskalakis, and G. Kamath. Priv'it: Private and sample efficient identity testing. In *ICML*, pages 635–644, 2017.

[15] M. Charikar, K. Shiragur, and A. Sidford. Efficient profile maximum likelihood for universal symmetric property estimation. In *STOC*, pages 780–791, 2019.

[16] L. Chen, B. Ghazi, R. Kumar, and P. Manurangsi. On distributed differential privacy and counting distinct elements. In *ITCS*, pages 56:1–56:18, 2021.

[17] A. Cheu, A. D. Smith, J. R. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019.

[18] A. Cheu and C. Yan. Pure differential privacy from secure intermediaries. *CoRR*, abs/2112.10032, 2021.

[19] I. Diakonikolas, M. Hardt, and L. Schmidt. Differentially private learning of structured discrete distributions. In *NIPS*, pages 2566–2574, 2015.

[20] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *NeurIPS*, pages 3571–3580, 2017.

[21] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Minimax optimal procedures for locally private estimation. *JASA*, 2017.

[22] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[23] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[24] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin. Pan-private streaming algorithms. In *ICS*, pages 66–80, 2010.

[25] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pages 2468–2479, 2019.

[26] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, pages 1054–1067, 2014.

[27] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, R. Pagh, and A. Velingker. Pure differentially private summation from anonymous messages. In *ITC*, pages 15:1–15:23, 2020.

[28] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pages 3505–3514, 2020.

[29] A. Greenberg. Apple's "differential privacy" is about collecting your data – but not your data. *Wired, June*, 13, 2016.

[30] Y. Hao and A. Orlitsky. The broad optimality of profile maximum likelihood. In *NeurIPS*, pages 10989–11001, 2019.

[31] G. H. Hardy and S. Ramanujan. Asymptotic Formulaæ in Combinatory Analysis. *Proc. London Math. Soc.*, s2-17(1):75–115, 01 1918.

[32] M. Hay, C. Li, G. Miklau, and D. D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, pages 169–178, 2009.

[33] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *VLDB*, 3(1):1021–1032, 2010.

[34] V. Karwa and S. Vadhan. Finite sample differentially private confidence intervals. In *ITCS*, 2018.

[35] K. Kenthapadi and T. T. L. Tran. Pripearl: A framework for privacy-preserving analytics and reporting at linkedin. In *CIKM*, pages 2183–2191, 2018.

[36] P. Manurangsi. Tight bounds for differentially private anonymized histograms. In *SOSA*, pages 203–213, 2022.

[37] D. J. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright. Pan-private algorithms via statistics on sketches. In *PODS*, pages 37–48, 2011.

[38] A. Orlitsky, A. T. Suresh, and Y. Wu. Optimal prediction of the number of unseen species. *PNAS*, 113(47):13283–13288, 2016.

[39] S. Raskhodnikova and A. Smith. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. *FOCS*, 2016.

[40] R. Rogers, S. Subramaniam, S. Peng, D. Durfee, S. Lee, S. K. Kancha, S. Sahay, and P. Ahammad. Linkedin's audience engagements API: A privacy preserving data analytics system at scale. *J. Priv. Confiden.*, 11(3), 2021.

[41] G. Rote. Isotonic Regression by Dynamic Programming. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, pages 1:1–1:18, 2019.

[42] S. Shankland. How Google tricks itself to protect Chrome user privacy. *CNET, October*, 2014.

[43] A. T. Suresh. Differentially private anonymized histograms. In *NeurIPS*, pages 7969–7979, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] *We discuss limitations and scope for future improvements in Section 6.*
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] *Since this is a purely theoretical paper regarding private algorithms for approximately computing anonymized histograms, we do not foresee any immediate potential negative impacts.*
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] *The proof overviews are contained in the main body; with proofs of key lemmas included in the supplementary material.*
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Missing Proofs from Section 4

A.1 Proof of Lemma 14

Lemma 14. *[Proof in Appendix A.1] For $B > 4n$ and all anonymized histograms $\boldsymbol{n}, \boldsymbol{n}'$ with $\|\boldsymbol{n}\|_1, \|\boldsymbol{n}'\|_1 \leq n$,*

$$\|\Gamma^B(\boldsymbol{n}) - \Gamma^B(\boldsymbol{n}')\|_1 \geq \|\boldsymbol{n} - \boldsymbol{n}'\|_1/4\,.$$

We start with the following observation that $\Gamma^B(\boldsymbol{n})_\ell := \mathbb{E}[\varphi_{\geq \ell}^{\boldsymbol{n}^{\mathrm{red}}}]$ only depends on $\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell}^{\boldsymbol{n}}$. This is simply because changing an entry of value at least ℓ to some other value at least ℓ does not change whether, after hashing, the corresponding hash bucket exceeds ℓ.

Observation 19. $\Gamma^B(\boldsymbol{n})_\ell$ does *not* depend on $\varphi_{\geq r}^{\boldsymbol{n}}$ for any $r > \ell$.

Before can prove Lemma 14, we will also need to show the following lemma, which is even stronger than Lipschitzness.

Lemma 20. *Let $\bar{\boldsymbol{n}}, \tilde{\boldsymbol{n}}$ be anonymized histograms such that $\|\tilde{\boldsymbol{n}} - \bar{\boldsymbol{n}}\|_1 = 1$, $\|\bar{\boldsymbol{n}}\|_1 = n$, $\|\tilde{\boldsymbol{n}}\|_1 = n+1$, and $r \in [n]$ be such that $\varphi_{\geq r}^{\tilde{\boldsymbol{n}}} = \varphi_{\geq r}^{\bar{\boldsymbol{n}}} + 1$. Then, we have*

- $\Gamma^B(\tilde{\boldsymbol{n}})_\ell \;=\; \Gamma^B(\bar{\boldsymbol{n}})_\ell$ *for all* $\ell < r$.
- $\Gamma^B(\tilde{\boldsymbol{n}})_r \;\geq\; \Gamma^B(\bar{\boldsymbol{n}})_r + 1 - n/B$.
- $\sum_{\ell > r} |\Gamma^B(\tilde{\boldsymbol{n}})_\ell - \Gamma^B(\bar{\boldsymbol{n}})_\ell| \;<\; n/B$.

Specifically, these also imply that $\|\Gamma^B(\tilde{\boldsymbol{n}}) - \Gamma^B(\bar{\boldsymbol{n}})\|_1 \leq 1$.

Proof. Let $\bar{\boldsymbol{n}}^{\mathrm{red}}$ and $\tilde{\boldsymbol{n}}^{\mathrm{red}}$ be the random variables as defined at the beginning of Section 4, but with starting anonymized histograms $\bar{\boldsymbol{n}}, \tilde{\boldsymbol{n}}$ respectively. Furthermore, let j be the entry such that $\tilde{n}^{(j)} = \bar{n}^{(j)} + 1$.

Recall that H is our random hash function. Consider $\bar{\boldsymbol{n}}^{\mathrm{red}}(H)$ and $\tilde{\boldsymbol{n}}^{\mathrm{red}}(H)$ resulting from the same hash function H. By definition, we have $\Gamma^B(\tilde{\boldsymbol{n}}) - \Gamma^B(\bar{\boldsymbol{n}}) = \mathbb{E}_H[\boldsymbol{\varphi}_{\geq}^{\tilde{\boldsymbol{n}}^{\mathrm{red}}(H)} - \boldsymbol{\varphi}_{\geq}^{\bar{\boldsymbol{n}}^{\mathrm{red}}(H)}]$.

Furthermore, we have $\boldsymbol{\varphi}_{\geq}^{\tilde{\boldsymbol{n}}^{\mathrm{red}}(H)} - \boldsymbol{\varphi}_{\geq}^{\bar{\boldsymbol{n}}^{\mathrm{red}}(H)} = \mathbf{1}_{\tilde{n}_{H(j)}^{\mathrm{red}}}$, where $\tilde{n}_{H(j)}^{\mathrm{red}}$ is the value of the bucket $H(j)$ to which j gets hashed. Note that with probability at least $1 - n/B$, there is no collision with j and therefore $\tilde{n}_{H(j)}^{\mathrm{red}} = r$. Otherwise, with probability at most n/B, there is a collision and $\tilde{n}_{H(j)}^{\mathrm{red}} > r$. This implies the desired bounds. □

We are now ready to prove Lemma 14.

Proof of Lemma 14. Let $\boldsymbol{n}^{\min}$ be such that $\varphi_{\geq r}^{\boldsymbol{n}^{\min}} = \min\{\varphi_{\geq r}^{\boldsymbol{n}}, \varphi_{\geq r}^{\boldsymbol{n}'}\}$ for all $r \in \mathbb{N}$. Note that $\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1 + \|\boldsymbol{n}^{\min} - \boldsymbol{n}'\|_1 = \|\boldsymbol{n} - \boldsymbol{n}'\|_1$. Assume w.l.o.g. that $\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1 \geq \|\boldsymbol{n}^{\min} - \boldsymbol{n}'\|_1$. The previous inequality implies that $\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1 \geq \|\boldsymbol{n} - \boldsymbol{n}'\|_1/2$.

Now, let $S := \{r \mid \varphi_{\geq r}^{\boldsymbol{n}} > \varphi_{\geq r}^{\boldsymbol{n}^{\min}}\}$. Consider the following hybrids for $j = 0, \ldots, n$: let $\boldsymbol{n}^j$ be such that $\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}^j} = (\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq j}^{\boldsymbol{n}}, \varphi_{\geq j+1}^{\boldsymbol{n}^{\min}}, \ldots, \varphi_{\geq n}^{\boldsymbol{n}^{\min}})$. By Lemma 20 and the definition of S, we have

$$\sum_{r \in S} \Gamma^B(\boldsymbol{n}^{j+1})_r - \Gamma^B(\boldsymbol{n}^j)_r \geq \frac{3}{4}\left(\varphi_{\geq j+1}^{\boldsymbol{n}^{j+1}} - \varphi_{\geq j+1}^{\boldsymbol{n}^j}\right),$$

since $\|\boldsymbol{n}^{\min}\|_1 \leq \|\boldsymbol{n}\|_1 \leq n$. By summing the above over all $j = 0, \ldots, n-1$, we have

$$\sum_{r \in S} \Gamma^B(\boldsymbol{n})_r - \Gamma^B(\boldsymbol{n}^{\min})_r \geq \frac{3}{4}\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1. \tag{6}$$

Similarly, we can consider the following hybrids for $j = 0, \ldots, n$: let $\boldsymbol{n}'^j$ be such that $\boldsymbol{\varphi}_{\geq}^{\boldsymbol{n}'^j} = (\varphi_{\geq 1}^{\boldsymbol{n}'}, \ldots, \varphi_{\geq j}^{\boldsymbol{n}'}, \varphi_{\geq j+1}^{\boldsymbol{n}^{\min}}, \ldots, \varphi_{\geq n}^{\boldsymbol{n}^{\min}})$. By Lemma 20 and the definition of S (which implies that

$\varphi^{\boldsymbol{n}'}_{\geq r} = \varphi^{\boldsymbol{n}^{\min}}_{\geq r}$ for all $r \in S$), we have

$$\sum_{r \in S} \Gamma^B(\boldsymbol{n}'^{j+1})_r - \Gamma^B(\boldsymbol{n}'^{j})_r \leq \frac{1}{4}\left(\varphi^{\boldsymbol{n}'^{j+1}}_{\geq j+1} - \varphi^{\boldsymbol{n}'^{j}}_{\geq j+1}\right).$$

Again, by summing the above over all $j = 0, \ldots, n-1$, we have

$$\sum_{r \in S} \Gamma^B(\boldsymbol{n}')_r - \Gamma^B(\boldsymbol{n}^{\min})_r \leq \frac{1}{4}\|\boldsymbol{n}^{\min} - \boldsymbol{n}'\|_1. \tag{7}$$

Subtracting (7) from (6), we complete the proof as follows:

$$\begin{aligned}\|\Gamma^B(\boldsymbol{n}) - \Gamma^B(\boldsymbol{n}')\|_1 \;&\geq\; \sum_{r \in S} \Gamma^B(\boldsymbol{n})_r - \Gamma^B(\boldsymbol{n}')_r \;\;\geq\;\; \frac{3}{4}\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1 - \frac{1}{4}\|\boldsymbol{n}^{\min} - \boldsymbol{n}'\|_1 \\ &\geq\; \frac{1}{2}\|\boldsymbol{n}^{\min} - \boldsymbol{n}\|_1 \;\;\geq\;\; \frac{1}{4}\|\boldsymbol{n} - \boldsymbol{n}'\|_1. \qquad \square\end{aligned}$$

A.2 Proof of Lemma 16

Lemma 16. *[Proof in Appendix A.2] For all $r \in [n]$, $\mathrm{Var}[\varphi^{\boldsymbol{n}^{\mathrm{red}}}_{\geq r}] \leq \frac{16n}{B} \cdot \left(\frac{n}{r^2} + \sum_{t \in [r-1]} \frac{t \cdot \varphi^{\boldsymbol{n}}_t}{r(r-t)}\right)$.*

Proof of Lemma 16. Note that $\varphi^{\boldsymbol{n}^{\mathrm{red}}}_{\geq r}$ is the number of hash buckets whose total value is at least r. Let X denote the number of hash buckets whose maximum value hashed to it is at least r, i.e., $X := |\{i \in [B] \mid \max_{j \in H^{-1}(i)} n^{(j)} \geq r\}|$. Furthermore, for $t \in [r-1]$, let Y_t denote the number of buckets whose total value is at least r and the maximum value hashed to it is equal to t, i.e., $Y_t := |\{i \in [B] \mid \max_{j \in H^{-1}(i)} n^{(j)} = t, \sum_{j \in H^{-1}(i)} n^{(j)} \geq r\}|$.

It is easy to see that $X, Y_1, \ldots, Y_{r-1}$ are negatively correlated. Therefore,

$$\mathrm{Var}[\varphi^{\boldsymbol{n}^{\mathrm{red}}}_{\geq r}] \;=\; \mathrm{Var}[X + Y_1 + \cdots + Y_{r-1}] \;\leq\; \mathrm{Var}[X] + \sum_{t \in [r-1]} \mathrm{Var}[Y_t].$$

We will next bound each term in the RHS above.

Bounding $\mathrm{Var}[X]$. To bound $\mathrm{Var}[X]$, further note that $X = \varphi^{\boldsymbol{n}}_{\geq r} - \sum_{i \in [B]} Z_i$ where $Z_i := \max\{0, |\{j \in H^{-1}(i) \mid n^{(j)} \geq r\}| - 1\}$. It is again easy to see that Z_i's are negatively correlated. Therefore,

$$\mathrm{Var}[X] = \mathrm{Var}\left[\sum_{i \in [B]} Z_B\right] \leq \sum_{i \in [B]} \mathrm{Var}[Z_i] \leq \sum_{i \in [B]} \mathbb{E}[Z_i^2].$$

It is also easy to verify that, for any $k \in \mathbb{N}$, $\Pr[Z_i = k] = \binom{\varphi^{\boldsymbol{n}}_{\geq r}}{k+1}(1/B)^{k+1}(1 - 1/B)^{\varphi^{\boldsymbol{n}}_{\geq r} - k - 1} \leq (\varphi^{\boldsymbol{n}}_{\geq r}/B)^{k+1}/(k+1)!$. Therefore, we have

$$\begin{aligned}\mathbb{E}[Z_i^2] \leq \sum_{k \in \mathbb{N}} k^2 \cdot (\varphi^{\boldsymbol{n}}_{\geq r}/B)^{k+1}/(k+1)! \;&\leq\; 2 \cdot \sum_{k \in \mathbb{N}} (\varphi^{\boldsymbol{n}}_{\geq r}/B)^{k+1} \\ \leq 4\left(\frac{\varphi^{\boldsymbol{n}}_{\geq r}}{B}\right)^2 \;&\leq\; 4\left(\frac{n}{Br}\right)^2 \;=\; \frac{4}{r^2} \cdot \left(\frac{n}{B}\right)^2.\end{aligned}$$

Plugging this back into the previous inequality, we have

$$\mathrm{Var}[X] \;\leq\; \frac{4}{r^2} \cdot \frac{n^2}{B}.$$

Bounding $\mathbf{Var}[\boldsymbol{Y_t}]$. We may write Y_t as $\sum_{i\in[B]} W_i$ where $W_i := \mathbf{1}[\max_{j\in H^{-1}(i)} n^{(j)} = t, \sum_{j\in H^{-1}(i)} n^{(j)} \geq r]$. Again, the W_i's are negatively correlated. Therefore, we have

$$\mathrm{Var}[Y_t] = \mathrm{Var}\left[\sum_{i\in[B]} W_i\right] \leq \sum_{i\in[B]} \mathrm{Var}[W_i] \leq \sum_{i\in[B]} \Pr[W_i = 1], \tag{8}$$

where the last inequality uses the fact that W_i is a Bernoulli r.v.

For each t, let $S_t \subseteq \mathbb{N}$ denote the set of indices j for which $n^{(j)} = t$. Furthermore, let $S_{\leq t} := S_1 \cup \cdots \cup S_t$. Notice that

$$\begin{aligned}
\Pr[W_i = 1] &= \Pr\left[\max_{j\in H^{-1}(i)} n^{(j)} = t \text{ and } \sum_{j\in H^{-1}(i)} n^{(j)} \geq r\right] \\
&= \Pr\left[\max_{j\in H^{-1}(i)} n^{(j)} = t \text{ and } \sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r\right] \\
\text{(Union bound)} &\leq \Pr\left[\max_{j\in H^{-1}(i)} n^{(j)} \geq t \text{ and } \sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r\right] \\
&= \Pr\left[\exists j^* \in S_t \text{ s.t } H(j^*) = i \text{ and } \sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r\right] \\
&\leq \sum_{j^*\in S_t} \Pr\left[H(j^*) = i \text{ and } \sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r\right] \\
&= \sum_{j^*\in S_t} \Pr\left[H(j^*) = i \text{ and } \sum_{j\in H^{-1}(i)\cap (S_{\leq t}\setminus\{j^*\})} n^{(j)} \geq r - t\right] \\
\text{(Independence of hash values)} &= \sum_{j^*\in S_t} \Pr[H(j^*) = i] \Pr\left[\sum_{j\in H^{-1}(i)\cap (S_{\leq t}\setminus\{j^*\})} n^{(j)} \geq r - t\right] \\
&= \sum_{j^*\in S_t} \frac{1}{B} \cdot \Pr\left[\sum_{j\in H^{-1}(i)\cap (S_{\leq t}\setminus\{j^*\})} n^{(j)} \geq r - t\right] \\
&\leq \sum_{j^*\in S_t} \frac{1}{B} \cdot \Pr\left[\sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r - t\right] \\
&= \frac{\varphi_t^{\boldsymbol{n}}}{B} \cdot \Pr\left[\sum_{j\in H^{-1}(i)\cap S_{\leq t}} n^{(j)} \geq r - t\right] \\
&= \frac{\varphi_t^{\boldsymbol{n}}}{B} \cdot \Pr\left[\sum_{j\in S_{\leq t}} n^{(j)} \cdot U_j \geq r - t\right],
\end{aligned} \tag{9}$$

where U_j denote the indicator $\mathbf{1}[H(j) = i]$.

We will next bound the last term based on two cases:

- Case I: $t \geq r/2$. Note that

$$\mathbb{E}\left[\sum_{j\in S_{\leq t}} n^{(j)} \cdot U_j\right] = \left(\sum_{j\in S_{\leq t}} n^{(j)}\right) \cdot \frac{1}{B} \leq \frac{n}{B}.$$

Therefore, we may apply Markov's inequality to get

$$\Pr\left[\sum_{j\in S_{\le t}} n^{(j)}\cdot U_j \ge r-t\right] \le \frac{n}{B(r-t)} \le \frac{2n}{B}\cdot\frac{t}{r(r-t)},$$

where the latter inequality follows from $t \ge r/2$.

- Case II: $t < r/2$. Since U_j's are independent, we can also compute the variance of the sum as

$$\begin{aligned}\mathrm{Var}\left[\sum_{j\in S_{\le t}} n^{(j)}\cdot U_j\right] &= \sum_{j\in S_{\le t}} \mathrm{Var}[n^{(j)}\cdot U_j] \quad\le\quad \sum_{j\in S_{\le t}} \frac{(n^{(j)})^2}{B}\\ &= \frac{1}{B}\sum_{\ell\in[t]} \ell^2\cdot\varphi^{\boldsymbol{n}}_\ell \quad\le\quad \frac{t}{B}\sum_{\ell\in[t]}\ell\cdot\varphi^{\boldsymbol{n}}_\ell \quad\le\quad \frac{tn}{B}.\end{aligned}$$

Note also that we have $r-t-n/B > (r-t)/2 \ge r/4$, where we used $t < r/2$. We may now apply Chebyshev's inequality to get

$$\Pr\left[\sum_{j\in S_{\le t}} n^{(j)}\cdot U_j \ge r-t\right] \le \frac{tn/B}{(r/4)^2} \le \frac{16n}{B}\cdot\frac{t}{r(r-t)}.$$

Therefore, in both cases, we have $\Pr\left[\sum_{j\in S_{\le t}} n^{(j)}\cdot U_j \ge r-t\right] \le \frac{16n}{B}\cdot\frac{t}{n-t}$. Combining this with (8) and (9), we get

$$\mathrm{Var}[Y_t] \ge B\cdot\frac{\varphi^{\boldsymbol{n}}_t}{B}\cdot\frac{16n}{B}\cdot\frac{t}{r(r-t)} = \frac{16n}{B}\cdot\frac{t\cdot\varphi^{\boldsymbol{n}}_t}{r(r-t)}.$$

By summing up our bounds on X and Y_t's, we arrive at the desired bound

$$\mathrm{Var}[\varphi^{\boldsymbol{n}^{\mathrm{red}}}_{\ge r}] \le \frac{16n}{B}\cdot\left(\frac{n}{r^2} + \sum_{t\in[r-1]}\frac{t\cdot\varphi^{\boldsymbol{n}}_t}{r(r-t)}\right).$$

□

B Extension to ℓ_2^2-error

A different error measure used in [33] is the ℓ_2^2-error, defined as $\|\boldsymbol{n}-\hat{\boldsymbol{n}}\|_2^2 = \sum_{j\in[D]}(n^{(j)}-\hat{n}^{(j)})^2$. In this section, we show the flexibility of our method by showing that we can get an error of $\widetilde{O}_\varepsilon(\sqrt{n})$, nearly matching that of [33].

B.1 Small Domain Case

We start with the setting $D \le \widetilde{O}(n)$ which does not require hashing.

The idea is to constrain our estimate anonymized histogram such that the ℓ_∞-distance from the Laplace histogram is not too large. A full description is presented in Algorithm 3.

Algorithm 3 Anonymized Histogram Estimator w.r.t. ℓ_2^2 loss.

Input: Discrete Laplace-noised histogram $\boldsymbol{h}'$, i.e., $h'_j \sim h_j + \mathrm{DLap}(p)$
Parameter: $\gamma = 10\log(2nD)/\log(1/p)$

for $r \in [n]$ **do**
  $\hat{\varphi}_{\ge r} \leftarrow \sum_{j\in[D]} f(h'_j - r)$, where f is defined in (1)
return $\hat{\boldsymbol{n}}$ that minimizes $\|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\ge} - \hat{\boldsymbol{\varphi}}_{\ge}\|_1$ subject to $\|\boldsymbol{n}_{\boldsymbol{h}'} - \hat{\boldsymbol{n}}\|_\infty \le \gamma$ and $\|\hat{\boldsymbol{n}}\|_1 = n$. (If no such $\hat{\boldsymbol{n}}$ exists, output the all-zeros histogram.)

Theorem 21. *Let $\hat{\boldsymbol{n}}$ be the output of Algorithm 3. We have*

$$\mathbb{E}[\|\hat{\boldsymbol{n}}-\boldsymbol{n}\|_2^2] \le O\left(\frac{\log(nD)}{\log(1/p)}\cdot\sqrt{C_p(n+D)\log n}\right) + O(1).$$

Plugging in $p = e^{-\varepsilon/2}$ (sufficient for ε-DP), we have the bound of $O(\sqrt{(n+D)\log n} \cdot \log(nD)/\varepsilon^{3.5})$ for $\varepsilon \le 1$.

Proof. First, the standard concentration of the noise implies that

$$\Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty > \gamma] \le 0.1/n^2.$$

We thus have

$$\begin{aligned}\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2] &\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] \cdot \mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2 \mid \|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] + \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty > \gamma] \cdot n^2 \\ &\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] \cdot \mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2 \mid \|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] + O(1).\end{aligned}$$

Now, recall that $\|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_2^2 \le \|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_\infty \cdot \|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_1$. Plugging this into the above, we get

$$\begin{aligned}\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2] &\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] \cdot \mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 \cdot \|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_\infty \mid \|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] + O(1) \\ &\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] \cdot \mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 \cdot 2\gamma \mid \|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] + O(1) \\ &\le 2\gamma \cdot \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] \cdot \mathbb{E}[2\|\boldsymbol{\varphi}^{\boldsymbol{n}}_{\ge} - \hat{\boldsymbol{\varphi}}_{\ge}\|_1 \mid \|\boldsymbol{n} - \boldsymbol{n}_{\boldsymbol{h}'}\|_\infty \le \gamma] + O(1) \\ &\le 4\gamma \cdot \mathbb{E}[\|\boldsymbol{\varphi}^{\boldsymbol{n}}_{\ge} - \hat{\boldsymbol{\varphi}}_{\ge}\|_1] + O(1) \\ &\le 4\gamma \cdot O\left(\sqrt{C_p(n+D)\log n}\right) + O(1), \\ &= O\left(\frac{\log(nD)}{\log(1/p)} \cdot \sqrt{C_p(n+D)\log n}\right) + O(1),\end{aligned}$$

where the second inequality follows from the constraint on $\hat{\boldsymbol{n}}$ and the last inequality follows from our analysis in the ℓ_1-error case (Theorem 9). □

B.2 Large Domain Case

We next move on to the case $D \gg n$, which will require random hashing.

B.2.1 Barrier to Extending to the ℓ_2^2-error

It turns out that, unlike the ℓ_1-error case, using a single noisy hashed histogram with $B = \widetilde{O}(n)$ is not sufficient for us to get $\widetilde{O}(\sqrt{n})$ ℓ_2^2-error. Before we provide the fix for this, let us briefly sketch why this is the case.

Let $c = \lceil 10\sqrt{B} \rceil$ and $q = \lfloor n/c \rfloor$. Consider the following two datasets (before hashing):

- There are c items, each with value q.
- There are $c - 2$ items each with value q and one additional item with value $2q$.

It is not hard to see (from birthday paradox) that, after randomly hashing into B buckets, it is impossible to distinguish the two cases with advantage more than 0.1 (even without any discrete Laplace noise). This implies that the expected ℓ_2^2-error must be at least $\Omega(q^2) = \Omega(n^2/B)$. For $B = \widetilde{O}(n)$, this is at least $\widetilde{\Omega}(n)$.

B.2.2 Two-Hash Approach

We now sketch an approach for large D. Here we would use two hashes. The first is with $B_1 = \widetilde{O}(n)$ buckets similar to before and but the second with a much larger, say, $B_2 = O(n^4)$ buckets. We then use the ℓ_1 approach on the first hash with the additional ℓ_∞ constraint on the second hash. The main point here is that w.h.p. there would be no collision at all in the second hash; therefore, the ℓ_∞ constraint will be valid. A similar analysis to the small domain case shows that this only adds $O_\varepsilon(\log n)$ multiplicative overhead on the expected ℓ_2^2-error (compared to the ℓ_1-error).

More precisely, our approach is presented in Algorithm 4.

Theorem 22. *Let $\hat{\boldsymbol{n}}$ be the output of Algorithm 4. We have*

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2] \le O\left(\frac{\log(nB_2)}{\log(1/p)} \cdot \left(\sqrt{C_p(n+B_1)\log n} + \frac{n\log n}{\sqrt{B_1}}\right)\right) + O\left(\frac{n^4}{B_2} + 1\right).$$

Algorithm 4 Anonymized Histogram Estimator w.r.t. ℓ_2^2 loss, for large domains.

Input: Two discrete Laplace-noised histograms $\tilde{\boldsymbol{h}}^{1,\text{red}}$ and $\tilde{\boldsymbol{h}}^{2,\text{red}}$ given as
- $\tilde{h}_j^{1,\text{red}} \sim h_j^{1,\text{red}} + \text{DLap}(p)$ where the random hash has B_1 buckets, and
- $\tilde{h}_j^{2,\text{red}} \sim h_j^{2,\text{red}} + \text{DLap}(p)$ where the random hash has B_2 buckets.

Parameter: $\gamma = 10\log(2nB_2)/\log(1/p)$

Compute an estimate $\hat{\boldsymbol{n}}^1$ by running Algorithm 2 on $\tilde{\boldsymbol{h}}^1$.
return $\hat{\boldsymbol{n}}$ that minimizes $\|\hat{\boldsymbol{n}} - \hat{\boldsymbol{n}}^1\|_1$ subject to $\|\boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}} - \hat{\boldsymbol{n}}\|_\infty \le \gamma$ and $\|\hat{\boldsymbol{n}}\|_1 = n$. (If no such $\hat{\boldsymbol{n}}$ exists, just output the all-zero histogram.)

Plugging in $p = e^{-\varepsilon/4}$ (sufficient for ε-DP), $B_1 = n\sqrt{\log n}$ and $B_2 = n^4$, we have the bound of $O(\sqrt{n}(\log n)^{7/4}/\varepsilon^{3.5})$ for $\varepsilon \le 1$.

Proof. The probability that there is any collision in the second hash is at most n^2/B_2. Furthermore, standard concentration inequality of the noise implies that the noise added to any of $h_j^{2,\text{red}}$ is greater than γ is at most $0.1/n^2$. Therefore, by a union bound, we have

$$\Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty > \gamma] \le n^2/B_2 + 0.1/n^2.$$

The remaining analysis is now similar to that of Theorem 21. Specifically, we have

$$\begin{aligned}
&\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_2^2] \\
&\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty \le \gamma] \cdot \mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1 \cdot \|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_\infty \mid \|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty \le \gamma] \\
&\qquad + \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty > \gamma] \cdot n^2 \\
&\le \Pr[\|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty \le \gamma] \cdot \mathbb{E}[2\|\hat{\boldsymbol{n}}^1 - \boldsymbol{n}\|_1 \cdot (2\gamma) \mid \|\boldsymbol{n} - \boldsymbol{n}_{\tilde{\boldsymbol{h}}^{2,\text{red}}}\|_\infty \le \gamma] + n^4/B_2 + 0.1 \\
&\le (4\gamma) \cdot \mathbb{E}[\|\hat{\boldsymbol{n}}^1 - \boldsymbol{n}\|_1] + n^4/B_2 + 0.1 \\
&\le O\left(\gamma \cdot \left(\sqrt{C_p(n+B_1)\log n} + \sqrt{n^2/B_1}\log n\right) + n^4/B_2 + 1\right),
\end{aligned}$$

where the last inequality follows from Theorem 12. □

C Missing Proof from Section 5

In this section, we prove sample complexity bounds for estimating symmetric distribution properties. In addition to entropy (Corollary 18), we also show sample complexity bounds for support coverage (Corollary 23) and support size (Corollary 24).

C.1 Entropy

Proof of Theorem 17. Acharya et al. [3] consider two estimators for entropy estimation.

- The first estimator $\hat{H}$ they study is the entropy of the empirical distribution. This has a non-private sample complexity of $C_{\hat{H}}(H,\alpha) = O\left(\frac{k}{\alpha} + \frac{\log^2 k}{\alpha^2}\right)$. The sensitivity of this estimator is $O(\log n/n)$. Thus, we have $D_{\hat{H}}(\alpha,\varepsilon)$ is the smallest n that satisfies
$$\frac{\log n}{n} \cdot \frac{\sqrt{n}\log n}{\varepsilon^{2.5}} \lesssim \alpha.$$
And so, $D_{\hat{H}}(\alpha,\varepsilon) \le O(\frac{\log^4(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5})$. Thus, plugging this into Theorem 17 gives rise to a sample complexity of $O\left(\frac{k}{\alpha} + \frac{\log^2 k}{\alpha^2} + \frac{\log^4(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right)$.
- The second estimator $\hat{H}$ they study is based on the prior work of [2], which for any $\lambda \in (0,1)$ has a non-private sample complexity of $C_{\hat{H}}(H,\alpha) = O\left(\frac{k}{\lambda^2\alpha\log k} + \frac{\log^2 k}{\alpha^2}\right)$. The sensitivity of this estimator is $1/n^{1-\lambda}$. Thus, we have $D_{\hat{H}}(\alpha,\varepsilon)$ is the smallest n that satisfies
$$\frac{1}{n^{1-\lambda}} \cdot \frac{\sqrt{n}\log n}{\varepsilon^{2.5}} \lesssim \alpha.$$

And so, $D_{\hat{H}}(\alpha,\varepsilon) \leq O\left(\left(\frac{\log^2(1/(\alpha\varepsilon)}{\alpha^2\varepsilon^5}\right)^{1/(1-2\lambda)}\right)$. Thus, plugging this into Theorem 17 gives rise to a sample complexity of $O\left(\frac{k}{\lambda^2\alpha\log k} + \frac{\log^2 k}{\alpha^2} + \left(\frac{\log^2(1/(\alpha\varepsilon)}{\alpha^2\varepsilon^5}\right)^{1/(1-2\lambda)}\right)$. □

C.2 Support Coverage

The *support coverage* of a distribution $\mathcal{D}$ and an integer m is defined as $S_m(\mathcal{D}) := \sum_{x\in\text{supp}(\mathcal{D})}(1-(1-\mathcal{D}(x))^m)$, i.e., the expected number of distinct elements seen if we draw m i.i.d. samples from $\mathcal{D}$. Here we would like to estimate $S_m(\mathcal{D})$ to within an additive error of $\pm\alpha m$.

Our bounds are stated below. Note that, in the case of large m, our sample complexity is asymptotically the same as that of [3] but our dependency on α, ε is worse in the case of smaller m.

Corollary 23 (Support Coverage)**.** *There exists an (ε,δ)-DP mechanism, in the pan-private and shuffle DP models, that can estimate $S_m(\mathcal{D})$ up to an additive error of $\pm\alpha m$ with a sample complexity of*

$$\begin{cases} O\left(\frac{m\log(1/\alpha)}{\log(\alpha m\varepsilon)}\right) & \text{if } m \geq \Omega\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right), \\ O\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right) & \text{if } m \leq O\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right). \end{cases}$$

Proof. Throughout the proof, it will be simpler to think of estimating $\frac{S_m(\mathcal{D})}{m}$ to within an additive error of $\pm\alpha$. Acharya et al. [3] consider two estimators for support coverage estimation.

- **Sparse Case.** Assuming that $m \geq \Omega\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right)$. When $m \geq 2n$, Acharya et al. [3] (based on an earlier work [38]) gave an estimator $\hat{S}_m$, parameterized by $r = \log(3/\alpha)$ and $t = m/n - 1$, that has a non-private sample complexity of $C_{\hat{S}_n}(S_m,\alpha) = O\left(\frac{m\log(1/\alpha)}{\log m}\right)$. The sensitivity of this estimator is $\frac{1+e^{r(t-1)}}{m} = \frac{1+e^{\log(3/\alpha)(m/n-2)}}{m}$. Thus, we have $D_{\hat{S}_m}(\alpha,\varepsilon)$ is the smallest n that satisfies
$$\frac{1+e^{\log(3/\alpha)(m/n-2)}}{m}\cdot\frac{\sqrt{n}\log n}{\varepsilon^{2.5}} \leq \alpha$$
And so, $D_{\hat{S}_m}(\alpha,\varepsilon) \leq O\left(\frac{m\log(1/\alpha)}{\log(\alpha m\varepsilon)}\right)$. Thus, plugging this into Theorem 17 gives rise to a sample complexity of $O\left(\frac{m\log(1/\alpha)}{\log(\alpha m\varepsilon)}\right)$.
- **Dense Case.** The second estimator $\hat{S}_m$ they study works when n is a multiple of m. We remark that, although the estimator as described in [3] does not seem to fit our setting, it can be viewed in our framework as follows. The examples are divided in to n/m batches each of size m. We then build a histogram $\boldsymbol{h}$ on $\text{supp}(\mathcal{D}) \times [n/m]$ where each example in batch i appends i to its original item. The final estimate is then $\varphi^{\boldsymbol{h}}_{\geq 1}/n$. It was shown in [3] that the non-private sample complexity is $C_{\hat{S}_m}(S_m,\alpha) = O\left(1/\alpha^2\right)$.
The sensitivity of this estimator is $1/n$. Thus, we have $D_{\hat{S}_m}(\alpha,\varepsilon)$ is the smallest n that satisfies
$$\frac{1}{n}\cdot\frac{\sqrt{n}\log n}{\varepsilon^{2.5}} \leq \alpha$$
And so, $D_{\hat{S}_m}(\alpha,\varepsilon) \leq O\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right)$. Thus, plugging this into Theorem 17 gives rise to a sample complexity of $O\left(\max\left\{m, \frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right\}\right)$, where the m part comes from our constraint that $n \geq m$. □

C.3 Support Size

Finally, we consider the problem of estimating the support size of $\mathcal{D}$. In general, this is impossible with finite sample since some atom of $\mathcal{D}$ might have an arbitrarily small probability mass. To avoid this, we follow prior work and consider only probability distributions in $\Delta_{\geq 1/K} := \{\mathcal{D} \mid \forall x \in \text{supp}(\mathcal{D}), \mathcal{D}(x) \geq 1/K\}$, i.e., those with non-zero mass of at least $1/K$ at every atom, for some K.

Orlitsky et al. [38] proved that the support coverage with $m \geq \Omega(K \cdot \log(3/\alpha))$ is a good estimate to the support size of any $\mathcal{D} \in \Delta_{\geq 1/K}$ to within an error of $\pm\alpha K$. In particular, they showed that if $m \geq K \log(3/\alpha)$, then for any $\mathcal{D} \in \Delta_{1/K}$, it holds that $|S_m(\mathcal{D}) - S(\mathcal{D})| \leq \alpha K/3$. Combining this with Corollary 24, we immediately get the following bounds for the latter.

Corollary 24 (Support Size)**.** *There exists an (ε, δ)-DP mechanism, in the pan-private and shuffle DP settings, that can estimate the support size of $\mathcal{D} \in \Delta_{\geq 1/K}$ up to an additive error of $\pm\alpha K$ with a sample complexity of*

$$\begin{cases} O\left(\frac{K \log^2(1/\alpha)}{\log(\alpha K \varepsilon)}\right) & \text{if } K \geq \Omega\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2 \varepsilon^5 \log(1/\alpha)}\right), \\ O\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2\varepsilon^5}\right) & \text{if } K \leq O\left(\frac{\log^2(1/(\alpha\varepsilon))}{\alpha^2 \varepsilon^5 \log(1/\alpha)}\right). \end{cases}$$

D Fast Post-Processing Algorithms

So far we have not focused on the running time of our post-processing algorithms. Nonetheless, it is not hard to see that our algorithms run in polynomial time. Below, we show that our algorithms can even be made to run in $\widetilde{O}_\varepsilon(D+n)$-time for the ℓ_1-error case.

D.1 Small Domain Case

We start with how to implement Algorithm 1 in $O(D + n \log n)$ time. The first observation is that the expression for each $\hat{\varphi}_{\geq r}$ can be easily computed in constant time if we precompute the number of buckets above a certain threshold beforehand. The second observation is that the last step is exactly the same as the so-called ℓ_1-*isotonic regression* problem (by viewing the cumulative prevalence $\varphi_{\geq 1}, \ldots, \varphi_{\geq n}$ as the variables), for which an $O(n \log n)$-algorithm is known [41]. A full description is given in Algorithm 5.

Algorithm 5 Efficient Anonymized Histogram Estimator w.r.t. ℓ_1 loss.

Input: Discrete Laplace-noised histogram $\boldsymbol{h}'$, i.e., $h'_j \sim h_j + \mathrm{DLap}(p)$

$c_0, \ldots, c_n \leftarrow 0$ {Counts for hash values}
for $j \in [D]$ **do**
  $v \leftarrow \max\{\min\{h'_j, n\}, 0\}$ {Clip so that the value is between $0, n$}
  $c_v \leftarrow c_v + 1$
$c_{\geq 0}, \ldots, c_{\geq n+1} \leftarrow 0$ {Counts for hash values above a certain threshold.}
for $r = n, \ldots, 0$ **do**
  $c_{\geq r} \leftarrow c_{\geq r+1} + c_r$
for $r \in [n]$ **do**
  $\hat{\varphi}_{\geq r} \leftarrow c_{\geq r+1} + \left(1 + \frac{p}{(1-p)^2}\right) c_r + \left(-\frac{p}{(1-p)^2}\right) c_{r-1}$
Find $\hat{\boldsymbol{n}}$ that minimizes $\|\boldsymbol{\varphi}^{\hat{\boldsymbol{n}}}_{\geq} - \hat{\boldsymbol{\varphi}}_{\geq}\|_1$ using ℓ_1-isotonic regression algorithm
return $\hat{\boldsymbol{n}}$

D.2 Large Domain Case

The large domain case is more complicated, as solving the optimization problem $\|\Gamma^B(\hat{\boldsymbol{n}}) - \boldsymbol{\varphi}^{\hat{\boldsymbol{n}}^{\mathrm{red}}}_{\geq}\|_1$ does not seem to be as simple as isotonic regression. Nonetheless, we give a modified algorithm below that takes $\widetilde{O}_\varepsilon(D + n \log n)$ time and has a similar error bound (within a polylogarithmic factor). The idea is to use two noisy histograms in a similar manner as we did for ℓ_2^2 loss (Algorithm 4): One noisy histogram (without hashing) is used to determine the high-value counts, whereas the other noisy histogram, with hashing, is used to determine the low-value counts.

D.2.1 Computing Γ^B

Recall from Observation 19 that $\Gamma^B(\boldsymbol{n})_\ell$ does not depend on $\varphi^{\boldsymbol{n}}_{\geq r}$ for any $r > \ell$. Therefore, we may view $\Gamma^B(\boldsymbol{n})_\ell$ as a function of $\varphi^{\boldsymbol{n}}_{\geq 1}, \ldots, \varphi^{\boldsymbol{n}}_{\geq \ell}$. We overload the notion and write $\Gamma^B(\varphi^{\boldsymbol{n}}_{\geq 1}, \ldots, \varphi^{\boldsymbol{n}}_{\geq \ell})_\ell$ to denote $\Gamma^B(\boldsymbol{n})_\ell$ of a corresponding histogram $\boldsymbol{n}$. We start by giving an algorithm for computing

such a value. Recall that $\Gamma^B(\boldsymbol{n})_\ell$ is defined as the expected number of hash buckets whose total value is at least ℓ. Let $\Xi_{\geq i}(\boldsymbol{n})$ denote the probability that a fixed hash bucket has total value at least i. By linearity of expectation, we have $\Gamma^B(\boldsymbol{n})_\ell = B \cdot \Xi_{\geq \ell}(\boldsymbol{n})$. Below, we keep updating $\Xi_{\geq i}$ as we include one more histogram value, using the fact that the hash function puts this histogram value in a random hash bucket.

Algorithm 6 Computation of $\Gamma^B(\varphi_{\geq 1}, \ldots, \varphi_{\geq \ell})_\ell$.

Input: $\varphi_{\geq 1}, \ldots, \varphi_{\geq \ell}$, cumulative prevalence of a histogram

$\varphi_{\geq \ell+1} \leftarrow 0$. {For notational convenience.}
$\Xi_{\geq 0} \leftarrow 1$ and $\Xi_{\geq 1}, \ldots, \Xi_{\geq \ell} \leftarrow 0$ {Initialize Ξ values}
for $r = 1, \ldots, \ell$ **do**
  for $j = 1, \ldots, \varphi_{\geq r} - \varphi_{\geq r+1}$ **do** {Add a new histogram entry of value r}
    for $k = \ell, \ldots, 0$ **do**
      $\Xi_{\geq k} \leftarrow \frac{B-1}{B} \cdot \Xi_{\geq k} + \frac{1}{B} \cdot \Xi_{\geq \max\{k-r,0\}}$
return $B \cdot \Xi_{\geq \ell}$

Since $\varphi^{\boldsymbol{n}}_{\geq 1} \leq n$, there are at most n pairs (r, j) considered by the algorithm. Therefore, the total runtime of Algorithm 6 is $O(n\ell)$.

D.2.2 Fast Post-processing Algorithm

Our fast post-processing algorithm for large domains is stated below (Algorithm 7). For convenience, we use the convention that $\varphi^{\boldsymbol{n}}_{\geq 0} = n$.

Algorithm 7 Efficient Anonymized Histogram Estimator w.r.t. ℓ_1 loss, for large domains.

Input: Two discrete Laplace-noised histograms $\tilde{\boldsymbol{h}}^1$ and $\tilde{\boldsymbol{h}}^{2,\mathrm{red}}$ given as
- $\tilde{h}^1_j \sim h_j + \mathrm{DLap}(p)$ (without any hashing), and
- $\tilde{h}^{2,\mathrm{red}}_j \sim h^{2,\mathrm{red}}_j + \mathrm{DLap}(p)$ where the random hash has B buckets.

Parameter: $m = \lceil 10 \log D / \log(1/p) \rceil$

Compute an estimate $\hat{\boldsymbol{n}}^1$ of $\boldsymbol{n}$ using Algorithm 1
Compute an estimate $\hat{\boldsymbol{n}}^{2,\mathrm{red}}$ of $\boldsymbol{n}^{2,\mathrm{red}}$ using Algorithm 1
$\hat{\varphi}_{\geq 0} \leftarrow n$
for $r \in [m]$ **do**
  $\mathsf{lb} = 0, \mathsf{ub} = \hat{\varphi}_{\geq r-1}$ {Binary search to find $\hat{\varphi}_{\geq r}$}
  while $\mathsf{ub} > \mathsf{lb} + 1$ **do**
    $\mathsf{mid} \leftarrow \lfloor (\mathsf{ub} + \mathsf{lb})/2 \rfloor$
    $v \leftarrow \Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq r-1}, \mathsf{mid})_r$ {See Algorithm 6}
    if $v \geq \varphi^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}_{\geq r}$ **then**
      $\mathsf{ub} \leftarrow \mathsf{mid}$
    else
      $\mathsf{lb} \leftarrow \mathsf{mid}$
  $v_{\mathsf{ub}} \leftarrow \Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq r-1}, \mathsf{ub})_r$
  $v_{\mathsf{lb}} \leftarrow \Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq r-1}, \mathsf{lb})_r$
  if $|v_{\mathsf{lb}} - \varphi^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}_{\geq r}| \leq |v_{\mathsf{ub}} - \varphi^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}_{\geq r}|$ **then**
    $\hat{\varphi}_{\geq r} \leftarrow \mathsf{lb}$
  else
    $\hat{\varphi}_{\geq r} \leftarrow \mathsf{ub}$
Find $\hat{\boldsymbol{n}}$ that minimizes $\sum_{r=1}^{m} |\hat{\varphi}_{\geq r} - \varphi^{\hat{\boldsymbol{n}}}_{\geq r}| + \sum_{r=m+1}^{n} |\varphi^{\hat{\boldsymbol{n}}^1}_{\geq r} - \varphi^{\hat{\boldsymbol{n}}}_{\geq r}|$ using ℓ_1-isotonic regression algorithm
return $\hat{\boldsymbol{n}}$

Run Time Analysis. Recall that Algorithm 1 runs in $O(D + n \log n)$ time. In binary search, we call Algorithm 6 for a total of at most $O(m \log n) = O(\log D \log n/\varepsilon)$ times; since each call runs in

$O(nr) = O(n \log D/\varepsilon)$ time, the total running time of this step is $O(n \log n (\log D/\varepsilon)^2)$. Finally, note that the last step is again an ℓ_1-isotonic regression problem and therefore can be solved in $O(n \log n)$ time. Thus, in total the running time is $\widetilde{O}_\varepsilon(D+n)$ as desired. Note that we may assume without loss of generality that $D \le n^{O(1)}$ as otherwise we may first hash to, e.g., $B' = O(n^2)$ buckets, which results in an increase in the expected ℓ_1-error of at most $O(1)$.

Error Analysis. We now prove the error guarantee of the algorithm, which is stated precisely below.

Theorem 25. *Suppose that $B > 3nm$. Then, we have*

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] \le O(\sqrt{n^2/B} \cdot \log n) + O(\sqrt{C_p B \log n}) + O(\sqrt{C_p n \log n}).$$

Note that, since taking $p = e^{-\varepsilon/4}$ makes the algorithm ε-DP, we get the following:

Corollary 26. *For all $\varepsilon > 0$, Algorithm 7 for $p = e^{-\varepsilon/4}$ and $B = 10nm$ is ε-DP, and achieves an expected ℓ_1-error of $O(\sqrt{n}(\log(nD))/\varepsilon^3)$.*

The quantitative bound above matches Corollary 13 to within a factor of $(\log n)^{1/4}/\varepsilon^{0.5}$ (recall that we can assume without loss of generality that $D \le n^{O(1)}$).

To prove Theorem 25, we will need the following auxiliary lemma, whose proof is given later in this section.

Lemma 27. *Suppose that $B > 3nm$. Then,*

$$\sum_{r=1}^{m} |\hat{\varphi}_{\ge r} - \hat{\varphi}^{\boldsymbol{n}}_{\ge r}| \le O\left(\|\Gamma^B(\boldsymbol{n}) - \varphi^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}_{\ge}\|_1\right). \tag{10}$$

We are now ready to prove Theorem 25.

Proof of Theorem 25. First, the expected ℓ_1-error can be written as

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] = \mathbb{E}\left[\sum_{r=1}^{n} |\varphi^{\hat{\boldsymbol{n}}}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}|\right] \le O\left(\mathbb{E}\left[\sum_{r=1}^{m} |\hat{\varphi}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}| + \sum_{r=m+1}^{n} |\varphi^{\hat{\boldsymbol{n}}^1}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}|\right]\right),$$

where the last step follows from the triangle inequality and our choice of $\hat{\boldsymbol{n}}$.

We now plug in Lemma 27, which yields:

$$\begin{aligned}\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] &\le O\left(\mathbb{E}\left[\|\Gamma^B(\boldsymbol{n}) - \varphi^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}_{\ge}\|_1\right] + \mathbb{E}\left[\sum_{r=m+1}^{n} |\varphi^{\hat{\boldsymbol{n}}^1}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}|\right]\right) \\ &\overset{\text{Lemma 15, Theorem 9}}{\le} O(\sqrt{n^2/B} \cdot \log n + \sqrt{C_p B \log n}) + \mathbb{E}\left[\sum_{r=m+1}^{n} |\varphi^{\hat{\boldsymbol{n}}^1}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}|\right]\end{aligned}$$

To bound the last term, we use Observation 11, which gives:

$$\begin{aligned}\mathbb{E}\left[\sum_{r=m+1}^{n} |\varphi^{\hat{\boldsymbol{n}}^1}_{\ge r} - \varphi^{\boldsymbol{n}}_{\ge r}|\right] &\le \sum_{r=m+1}^{n} \sqrt{\mathrm{Var}[\varphi^{\hat{\boldsymbol{n}}^1}_{\ge r}]} \le \sum_{r=m+1}^{n} \sqrt{\kappa} \cdot \sqrt{\sum_{\ell=0}^{n} p^{|\ell-r|} \cdot \varphi^{\boldsymbol{n}}_{\ell}} \\ &\le \sqrt{\kappa} \cdot \sqrt{\sum_{r=m+1}^{n} \frac{1}{r}} \cdot \sqrt{\sum_{r=m+1}^{n} r \cdot \left(\sum_{\ell=0}^{n} p^{|\ell-r|} \cdot \varphi^{\boldsymbol{n}}_{\ell}\right)} \quad \text{(Cauchy–Schwarz)} \\ &\le \sqrt{\kappa} \cdot O(\sqrt{\log n}) \cdot \sqrt{\sum_{\ell=0}^{n} \varphi^{\boldsymbol{n}}_{\ell} \cdot \left(\sum_{r=m+1}^{n} r \cdot p^{|\ell-r|}\right)} \\ &\le \sqrt{\kappa} \cdot O(\sqrt{\log n}) \cdot \sqrt{\left(\varphi^{\boldsymbol{n}}_0 \cdot \sum_{r=m+1}^{n} r \cdot p^r\right) + \left(\sum_{\ell=1}^{n} \varphi^{\boldsymbol{n}}_{\ell} \cdot \left(\sum_{r=1}^{n} r \cdot p^{|\ell-r|}\right)\right)}.\end{aligned}$$

The second term $\sum_{\ell=1}^{n} \varphi_\ell^{\boldsymbol{n}} \cdot \left(\sum_{r=1}^{n} r \cdot p^{|\ell-r|}\right)$ can be bounded in the exact same way as in Theorem 9, which gives an upper bound of $O(\sqrt{n/(1-p)^2})$. The first term $\left(\varphi_0^{\boldsymbol{n}} \cdot \sum_{r=m+1}^{n} r \cdot p^r\right)$ can be bounded as follows:

$$\varphi_0^{\boldsymbol{n}} \cdot \sum_{r=m+1}^{n} r \cdot p^r = D\left(p^{m+1} \cdot \sum_{i=0}^{n}(i+m+1)p^i\right) \quad \leq \quad D\left(\frac{p}{D^{10}} \cdot \left(\frac{m+1}{1-p} + \frac{1}{(1-p)^2}\right)\right)$$
$$\leq O\left(\frac{1}{(1-p)^2}\right).$$

Putting the four inequalities above together, we arrive at

$$\mathbb{E}[\|\hat{\boldsymbol{n}} - \boldsymbol{n}\|_1] \leq O(\sqrt{n^2/B} \cdot \log n + \sqrt{C_p B \log n}) + O(\sqrt{\kappa/(1-p)^2}) \cdot O(\sqrt{n \log n}). \qquad \square$$

Proof of Lemma 27. Let $\sigma := B/n$. We will prove by induction that

$$\sum_{r=1}^{\ell} |\hat{\varphi}_{\geq r} - \hat{\varphi}_{\geq r}^{\boldsymbol{n}}| \leq 3\left(1 + \frac{3}{\sigma}\right)^{\ell} \left(\sum_{r=1}^{\ell} |\Gamma^B(\boldsymbol{n})_r - \varphi_{\geq r}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}|\right), \tag{11}$$

for all $\ell = 0, \ldots, m$. The base case $\ell = 0$ is trivial.

We will next prove the inductive step. Suppose that the bound (11) holds for $\ell - 1$. We will now show that it also holds for ℓ. Recall from Lemma 20 that $\Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell})_\ell$ is increasing in $\hat{\varphi}_{\geq \ell}$ when $\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell-1}$ are held fixed. This means that our binary search algorithm finds the optimum, which implies

$$|\Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell-1}, \hat{\varphi}_{\geq \ell})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| \leq |\Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell-1}, \varphi_{\geq \ell}^{\boldsymbol{n}})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}|. \tag{12}$$

Applying the third inequality in Lemma 20 repeatedly, we have

$$\begin{aligned}
& |\Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \hat{\varphi}_{\geq \ell})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| \\
\text{(Lemma 20)} \quad & \leq |\Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell-1}, \hat{\varphi}_{\geq \ell})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{\sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma} \\
\text{(From (12))} \quad & \leq |\Gamma^B(\hat{\varphi}_{\geq 1}, \ldots, \hat{\varphi}_{\geq \ell-1}, \varphi_{\geq \ell}^{\boldsymbol{n}})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{\sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma} \\
\text{(Lemma 20)} \quad & \leq |\Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \varphi_{\geq \ell}^{\boldsymbol{n}})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{2 \cdot \sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma} \\
& = |\Gamma^B(\boldsymbol{n})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{2 \cdot \sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma}.
\end{aligned} \tag{13}$$

Again applying the second inequality in Lemma 20, we get

$$\begin{aligned}
& |\hat{\varphi}_{\geq \ell} - \varphi_{\geq \ell}^{\boldsymbol{n}}| \\
& \leq \frac{1}{1 - 1/\sigma} \cdot |\Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \hat{\varphi}_{\geq \ell})_\ell - \Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \varphi_{\geq \ell}^{\boldsymbol{n}})_\ell| \\
& \leq \frac{1}{1 - 1/\sigma} \cdot \left(|\Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \hat{\varphi}_{\geq \ell})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + |\Gamma^B(\varphi_{\geq 1}^{\boldsymbol{n}}, \ldots, \varphi_{\geq \ell-1}^{\boldsymbol{n}}, \varphi_{\geq \ell}^{\boldsymbol{n}})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}|\right) \\
& \overset{(13)}{\leq} \frac{1}{1 - 1/\sigma} \left(|\Gamma^B(\boldsymbol{n})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{2 \cdot \sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma} + |\Gamma^B(\boldsymbol{n})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}|\right) \\
& \leq 3|\Gamma^B(\boldsymbol{n})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \frac{3 \cdot \sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|}{\sigma} \qquad \left(\text{using } \frac{1}{1 - 1/\sigma} \leq \frac{3}{2}\right).
\end{aligned}$$

Therefore, we have

$$\sum_{r=1}^{\ell} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}| \leq 3|\Gamma^B(\boldsymbol{n})_\ell - \varphi_{\geq \ell}^{\hat{\boldsymbol{n}}^{2,\mathrm{red}}}| + \left(1 + \frac{3}{\sigma}\right) \cdot \left(\sum_{r=1}^{\ell-1} |\varphi_{\geq r}^{\boldsymbol{n}} - \hat{\varphi}_{\geq r}|\right).$$

Plugging in the inductive hypothesis, we can conclude that (12) also holds for ℓ.

Finally, plugging in $\ell = m$ and using the fact that $\sigma \geq 3\ell$, we arrive at the claimed bound. $\square$