Supplementary Material

A. Additional Preliminaries

We start by introducing a few additional notation and lemmas that will be used in our proofs.

Definition 13. *The* ε-hockey stick divergence *of two (discrete) distributions* $\mathcal{D}, \mathcal{D}'$ *is defined as*

$$d_\varepsilon(\mathcal{D}\|\mathcal{D}') := \sum_{v \in \mathrm{supp}(\mathcal{D})} \left[\Pr_{x\sim\mathcal{D}}[x = v] - e^\varepsilon \cdot \Pr_{x\sim\mathcal{D}'}[x = v]\right]_+$$

where $[y]_+ := \max\{y, 0\}$.

The following is a (well-known) simple restatement of DP in terms of hockey stick divergence:

Lemma 14. *An algorithm* $\mathcal{A}$ *is* (ε, δ)*-DP iff for any neighboring datasets* X *and* X'*, it holds that* $d_\varepsilon(\mathcal{A}(X), \mathcal{A}(X')) \le \delta$.

We will also use the following result in (Ghazi et al., 2020b, Lemma 21), which allows us to easily compute differential privacy parameters noise addition algorithms for Δ-summation.

Lemma 15 (Ghazi et al. (2020b))**.** *An* $\mathcal{D}$*-noise addition mechanism for* Δ*-summation is* (ε, δ)*-DP iff* $d_\varepsilon(\mathcal{D}\|k + \mathcal{D}) \le \delta$ *for all* $k \in \{-\Delta, \dots, +\Delta\}$.

Finally, we will also use the following lemma, which can be viewed as an alternative formulation of the basic composition theorem.

Lemma 16. *Let* $\mathcal{D}_1, \dots, \mathcal{D}_k, \mathcal{D}'_1, \dots, \mathcal{D}'_k$ *be any distributions and* $\varepsilon_1, \dots, \varepsilon_k$ *any non-negative real numbers. Then, we have* $d_{\varepsilon_1+\cdots+\varepsilon_k}(\mathcal{D}_1 \times \cdots \times \mathcal{D}_k \| \mathcal{D}'_1 \times \cdots \times \mathcal{D}'_k) \le \sum_{i\in[k]} d_{\varepsilon_i}(\mathcal{D}_i\|\mathcal{D}'_i)$.

B. On the Near Optimality of (Discrete) Laplace Mechanism

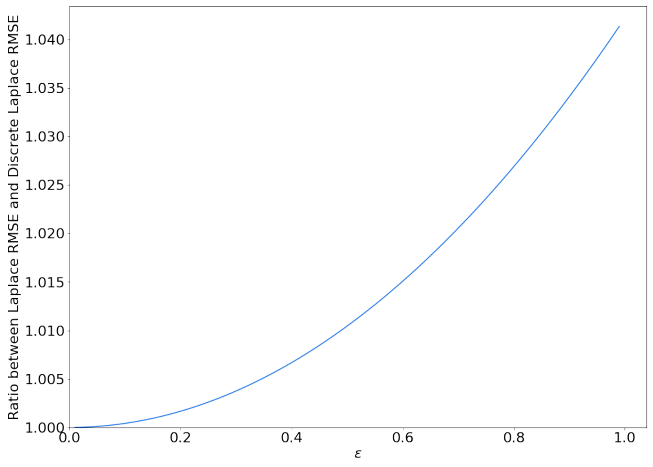


Figure 2: Comparison between RMSE of the Laplace Mechanism and RMSE of the Discrete Laplace Mechanism when varing ε.

In this section, we briefly discuss the near-optimality of Laplace mechanism in the central model. First, we recall that it follows from Ghosh et al. (2012) that the smallest MSE of any ε-DP algorithm (in the central model) for binary summation (i.e., 1-summation) is at least that of the Discrete Laplace mechanism minus a term that converges to zero as $n \to \infty$. In other words, RMSE of the Discrete Laplace mechanism, up to an $o(1)$ additive factor, forms a lower bound for binary summation. Notice that a lower bound on the RMSE for binary summation also yields a lower bound on the RMSE for real summation, since the latter is a more general problem. We plot the comparison between RMSE of the Discrete Laplace mechanism and RMSE of the Laplace mechanism in Figure 2 for $\varepsilon \in [0, 1]$. Since the former is a lower bound on RMSE of any ε-DP algorithm for real summation, this plot shows how close the Laplace mechanism is to the lower bound. Specifically, we have found that, in this range of $\varepsilon < 1$, the Laplace mechanism is within 5% of the optimal error (assuming n is sufficiently large). Finally, note that our real summation algorithm (Theorem 1) can achieve accuracy arbitrarily close to the Laplace mechanism, and hence our algorithm can also get to within 5% of the optimal error for any $\varepsilon \in [0, 1]$.

C. Missing Proofs from Section 3

Below we provide (simple) proofs of the generic form of the error (Observation 5) and the communication cost (Observation 6) of our protocol.

Proof of Observation 5. Consider each user i. The sum of its output is not affected by the noise from $\mathcal{S}$ since the sum of their messages are zero; in other words, the sum is simply $x_i + Z_i^{+1} - Z_i^{-1}$. As a result, the total error is exactly equal to $\left(\sum_{i\in[n]} Z_i^{+1}\right) - \left(\sum_{i\in[n]} Z_i^{-1}\right)$. Since each of Z_i^{+1}, Z_i^{-1} is an i.i.d. random variable distributed as $\mathcal{D}^{\text{central}}_{/n}$, the error is distributed as $\mathcal{D}^{\text{central}} - \mathcal{D}^{\text{central}}$. □

Proof of Observation 6. Each user sends at most one message corresponding to its input; in addition to this, the expected number of messages sent in the subsequent steps is equal to

$$\begin{aligned}\mathbb{E}\left[Z_i^{+1} + Z_i^{-1} + \sum_{\mathbf{s}\in\mathcal{S}}\sum_{m\in\mathbf{s}} Z_i^{\mathbf{s}}\right] &= \mathbb{E}[\mathcal{D}^{\text{central}}_{/n}] + \mathbb{E}[\mathcal{D}^{\text{central}}_{/n}] + \sum_{\mathbf{s}\in\mathcal{S}}\sum_{m\in\mathbf{s}} \mathbb{E}[\mathcal{D}^{\mathbf{s}}_{/n}] \\ &= 2\mathbb{E}[\mathcal{D}^{\text{central}}_{/n}] + \sum_{\mathbf{s}\in\mathcal{S}} |\mathbf{s}| \cdot \mathbb{E}[\mathcal{D}^{\mathbf{s}}_{/n}] \\ &= \frac{1}{n}\left(2\mathbb{E}[\mathcal{D}^{\text{central}}] + \sum_{\mathbf{s}\in\mathcal{S}} |\mathbf{s}| \cdot \mathbb{E}[\mathcal{D}^{\mathbf{s}}]\right)\end{aligned}$$

Finally, since each message is a number from $[-\Delta, \Delta]$, it can be represented using $\lceil \log_2 \Delta \rceil + 1$ bits as desired. □

D. Missing Proofs from Section 4

D.1. From Shuffle Model to Central Model

Let us restate Algorithm 3 in a slightly different notation where each user's input $x_1, \dots, x_n$ is represented explicitly in the algorithm (instead of the histogram as in Algorithm 3):

Algorithm 4 Central Algorithm

```
1: procedure CORRNOISE($x_1, \dots, x_n$)
2:     $u_{-\Delta}, \dots, u_{-1} \leftarrow 0$
3:   for $j \in [\Delta]$
4:     $u_j \leftarrow |\{i \in [n] \mid x_i = j\}|$
5:   Sample $z^{+1} \sim \mathcal{D}^{\text{central}}$
6:   $u_{+1} \leftarrow u_{+1} + Z^1$
7:   Sample $z^{-1} \sim \mathcal{D}^{\text{central}}$
8:   $u_{-1} \leftarrow u_{-1} + Z^{-1}$
9:   for $\mathbf{s} \in \mathcal{S}$
10:    Sample $z^{\mathbf{s}} \sim \mathcal{D}^{\mathbf{s}}$
11:    for $m \in \mathbf{s}$
12:      $u_m \leftarrow u_m + z^{\mathbf{s}}$
13:  return $(u_{-\Delta}, \dots, u_{-1}, u_{+1}, \dots, u_{+\Delta})$.
```

Observation 17. CORRNOISERANDOMIZER *is (ε, δ)-DP in the shuffle model iff* CORRNOISE *is (ε, δ)-DP in the central model.*

Proof. Consider the analyzer's view for CORRNOISERANDOMIZER; all the analyzer sees is a number of messages $-\Delta, \dots, -1, +1, \dots, +\Delta$. Let u_i denote the number of messages i, for $i \in \{-\Delta, \dots, -1, +1, \dots, +\Delta\}$. We will argue that $(u_{-\Delta}, \dots, u_{-1}, u_{+1}, \dots, u_{+\Delta})$ has the same distribution as the output of CORRNOISE, from which the observation follows. To see this, notice that the total number of noise atom $\mathbf{s}$ in CORRNOISERANDOMIZER across all users is $z_1^{\mathbf{s}} + \cdots + z_n^{\mathbf{s}}$ which is distributed as $\mathcal{D}^{\mathbf{s}}$. Similarly, the number of additional -1 messages and that of

$+1$ messages (from Line 5 of CORRNOISERANDOMIZER) are both distributed as (independent) $\mathcal{D}^{\text{central}}$. As a result, $(u_{-\Delta}, \dots, u_{-1}, u_{+1}, \dots, u_{+\Delta})$ has the same distribution as the output of CORRNOISE. □

Observation 7 then immediately follows from Observation 17 and the fact that the two algorithms are equivalent.

D.2. Flooding Central Noise: Proof of Lemma 9

Proof of Lemma 9. Consider any two input neighboring datasets $\mathbf{h}, \mathbf{h}'$. Let $\varepsilon := \varepsilon^* + \varepsilon_1$. From Lemma 14, it suffices to show that $d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}')) \leq \delta_1$. Let $\kappa = \langle \mathbf{v}, \mathbf{h}\rangle - \langle \mathbf{v}, \mathbf{h}'\rangle$; note that by definition of neighboring datasets, we have $-\Delta \leq \kappa \leq \Delta$. We may simplify $d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}'))$ as

$$\begin{aligned}
&d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}')) \\
&= \sum_{a,b \in \mathbb{Z}} \Big[\Pr[z^{+1} - z^{-1} + \langle \mathbf{v}, \mathbf{h}\rangle = a, z^{-1} + \tilde{z}^{-1,+1} = b] - e^{\varepsilon} \Pr[z^{1} - z^{-1} + \langle \mathbf{v}, \mathbf{h}'\rangle = a, z^{-1} + \tilde{z}^{-1,+1} = b]\Big]_+ \\
&= \sum_{c,b \in \mathbb{Z}} \Big[\Pr[z^{+1} - z^{-1} = c - \kappa, z^{-1} + \tilde{z}^{-1,+1} = b] - e^{\varepsilon} \Pr[z^{+1} - z^{-1} = c, z^{-1} + \tilde{z}^{-1,+1} = b]\Big]_+, \qquad (1)
\end{aligned}$$

where the second equality comes from simply substituting $c = a - \langle \mathbf{v}, \mathbf{h}'\rangle$.

For convenience, let $\lambda_c := \max\{0, \kappa - c\}$. We may expand the first probability above as

$$\begin{aligned}
&\Pr[z^{+1} - z^{-1} = c - \kappa, z^{-1} + \tilde{z}^{-1,+1} = b] \\
&= \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \Pr[\tilde{z}^{\{-1,+1\}} = b - d]. \qquad (2)
\end{aligned}$$

Similarly, let $\gamma_c := \max\{0, c\}$. We may expand the second probability above as

$$\begin{aligned}
&\Pr[z^{+1} - z^{-1} = c, z^{-1} + \tilde{z}^{-1,+1} = b] \\
&= \sum_{d'=\gamma_c}^{\infty} \Pr[z^{+1} = c + d'] \Pr[z^{-1} = d'] \Pr[\tilde{z}^{\{-1,+1\}} = b - d'] \\
&= \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c + d - (\lambda_c - \gamma_c)] \Pr[z^{-1} = d - (\lambda_c - \gamma_c)] \Pr[\tilde{z}^{\{-1,+1\}} = b - d + (\lambda_c - \gamma_c)]. \qquad (3)
\end{aligned}$$

Next, notice that for $d \geq \lambda_c$, we have

$$\begin{aligned}
&\frac{\Pr[z^{+1} = c + d - (\lambda_c - \gamma_c)] \Pr[z^{-1} = d - (\lambda_c - \gamma_c)]}{\Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d]} \\
&= \frac{\exp\left(-\frac{\varepsilon^*}{\Delta}(c + d - (\lambda_c - \gamma_c))\right) \exp\left(-\frac{\varepsilon^*}{\Delta}(d - (\lambda_c - \gamma_c))\right)}{\exp\left(-\frac{\varepsilon^*}{\Delta}(c - \kappa + d)\right) \exp\left(-\frac{\varepsilon^*}{\Delta}(d)\right)} \\
&= \exp\left(-\frac{\varepsilon^*}{\Delta}(\kappa - 2(\lambda_c - \gamma_c))\right) \\
&\geq \exp(-\varepsilon^*),
\end{aligned}$$

where the last inequality employs a simple observation that $(\lambda_c - \gamma_c) \leq \kappa \leq \Delta$.

Plugging the above back into (3), we get

$$\begin{aligned}
&\Pr[z^{+1} - z^{-1} = c, z^{-1} + \tilde{z}^{-1,+1} = b] \qquad (4) \\
&\geq e^{-\varepsilon^*} \cdot \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \Pr[\tilde{z}^{\{-1,+1\}} = b - d + (\lambda_c - \gamma_c)]. \qquad (5)
\end{aligned}$$

Then, plugging both (2) and (5) into (1), we arrive at

$$\begin{aligned}
&d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}')) \\
&\leq \sum_{c,b \in \mathbb{Z}} \left[\sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \left(\Pr[\tilde{z}^{\{-1,+1\}} = b - d] - e^{\varepsilon_1} \Pr[\tilde{z}^{\{-1,+1\}} = b - d + (\lambda_c - \gamma_c)]\right)\right]_+
\end{aligned}$$

$$\leq \sum_{c,b\in\mathbb{Z}} \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \cdot \left[\Pr[\tilde{z}^{\{-1,+1\}} = b - d] - e^{\varepsilon_1} \Pr[\tilde{z}^{\{-1,+1\}} = b - d + (\lambda_c - \gamma_c)]\right]_+$$
$$= \sum_{c\in\mathbb{Z}} \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \left(\sum_{b\in\mathbb{Z}} \left[\Pr[\tilde{z}^{\{-1,+1\}} = b - d] - e^{\varepsilon_1} \Pr[\tilde{z}^{\{-1,+1\}} = b - d + (\lambda_c - \gamma_c)]\right]_+\right)$$
$$= \sum_{c\in\mathbb{Z}} \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d] \cdot d_{\varepsilon_1}(\hat{\mathcal{D}} \| \hat{\mathcal{D}} + (\gamma_c - \lambda_c)).$$

Now, recall that $|\gamma_c - \lambda_c| \leq |\kappa| \leq \Delta$. Thus, from our assumption that the $\hat{\mathcal{D}}$-mechanism is $(\varepsilon_1, \delta_1)$-DP for Δ-summation and Lemma 15, we can conclude that $d_{\varepsilon_1}(\hat{\mathcal{D}} \| \hat{\mathcal{D}} + (\gamma_c - \lambda_c)) \leq \delta_1$. Hence, we can further bound $d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}'))$ as

$$d_\varepsilon(\mathcal{M}_{\text{cor}}(\mathbf{h}), \mathcal{M}_{\text{cor}}(\mathbf{h}')) \leq \delta_1 \cdot \left(\sum_{c\in\mathbb{Z}} \sum_{d=\lambda_c}^{\infty} \Pr[z^{+1} = c - \kappa + d] \Pr[z^{-1} = d]\right) = \delta_1.$$

This implies that $\mathcal{M}_{\text{cor}}$ is $(\varepsilon^* + \varepsilon_1, \delta_1)$-DP as desired. □

D.3. Finding a Right Inverse: Proof of Theorem 12

Proof of Theorem 12. We first select $\mathcal{S}$ to be the collections $\{-1, +1\}$ and all multisets of the form $\{i, -\lfloor i/2 \rfloor, -\lceil i/2 \rceil\}$ for all $i \in \{-\Delta, \dots, -2, +2, \dots, +\Delta\}$.

Throughout this proof, we write $\mathbf{c}_i$ to denote the ith column of $\mathbf{C}$. We construct a right inverse $\mathbf{C}$ by constructing its columns iteratively as follows:

- First, we let $\mathbf{c}_{-1} = \mathbf{1}_{\{-1,+1\}}$ denote the indicator vector of coordinate $\mathbf{s} = \{-1, +1\}$.
- For convenience, we let $\mathbf{c}_1 = \mathbf{0}$ although this column is not present in the final matrix $\mathbf{C}$.
- For each $i \in \{2, \dots, \Delta\}$:
  - Let $i_1 = \lfloor i/2 \rfloor$ and $i_2 = \lceil i/2 \rceil$.
  - Let $\mathbf{c}_i$ be $\mathbf{1}_{\{i,-i_1,-i_2\}} - \mathbf{c}_{-i_1} - \mathbf{c}_{-i_2}$.
  - Let $\mathbf{c}_{-i}$ be $\mathbf{1}_{\{-i,i_1,i_2\}} - \mathbf{c}_{i_1} - \mathbf{c}_{i_2}$.

We will now show by induction that $\mathbf{C}$ as constructed above indeed satisfies $\tilde{\mathbf{A}}\mathbf{C} = \mathbf{I}$. Let $\tilde{\mathbf{A}}_{i,*}$ denote the ith row of $\tilde{\mathbf{A}}$. Equivalently, we would like to show that[11]

$$\tilde{\mathbf{A}}_{i',*}\mathbf{c}_i = \mathbf{1}[i' = i], \tag{6}$$

for all $[-\Delta, \Delta]_{-1}$. We will prove this by induction on $|i|$.

(Base Case) We will show that this holds for $i = -1$. Since $\mathbf{c}_1 = \mathbf{1}_{\{-1,1\}}$, we have

$$\tilde{\mathbf{A}}_{i',*}\mathbf{c}_i = \tilde{\mathbf{A}}_{i,\{-1,1\}} = \mathbf{1}[i \in \{-1, 1\}] = \mathbf{1}[i = -1].$$

(Inductive Step) Now, suppose that (6) holds for all i such that $|i| < m$ for some $m \in \mathbb{N} \setminus \{1\}$. We will now show that it holds for $i = m, -m$ as well. Let $m_1 = \lfloor m/2 \rfloor$ and $m_2 = \lceil m/2 \rceil$. For $i = m$, let $\mathbf{s} = \{m, -m_1, -m_2\}$; we have

$$\begin{aligned}
\mathbf{A}_{i',*}\mathbf{c}_i &= \mathbf{A}_{i',*}\left(\mathbf{1}_\mathbf{s} - \mathbf{c}_{-m_1} - \mathbf{c}_{-m_2}\right) \\
\text{(From inductive hypothesis)} &= \mathbf{A}_{i',*}\mathbf{1}_\mathbf{s} - \mathbf{1}[i' = -m_1] - \mathbf{1}[i' = -m_2] \\
&= \mathbf{s}_{i'} - \mathbf{1}[i' = -m_1] - \mathbf{1}[i' = -m_2] \\
&= \mathbf{1}[i = i'],
\end{aligned}$$

where the last inequality follows from the definition of $\mathbf{s}$.

The case $i = -m$ is similar. Specifically, letting $\mathbf{s} = \{-m, m_1, m_2\}$, we have

$$\mathbf{A}_{i',*}\mathbf{c}_i = \mathbf{A}_{i',*}\left(\mathbf{1}_\mathbf{s} - \mathbf{c}_{m_1} - \mathbf{c}_{m_2}\right)$$

[11] We use $\mathbf{1}[E]$ to denote the indicator of condition E, i.e., $\mathbf{1}[E] = 1$ if E holds and $\mathbf{1}[E] = 0$ otherwise.

$$\begin{aligned}
\text{(From inductive hypothesis)} &= \mathbf{A}_{i',*}\mathbf{1}_{\mathbf{s}} - \mathbf{1}[i' = m_1] - \mathbf{1}[i' = m_2] \\
&= \mathbf{s}_{i'} - \mathbf{1}[i' = m_1] - \mathbf{1}[i' = m_2] \\
&= \mathbf{1}[i = i'].
\end{aligned}$$

Let $\Gamma = \Delta \cdot \lceil 1 + \log \Delta \rceil$. Let $\mathbf{t} \in \mathbb{Z}^{\mathcal{S}}$ be defined by

$$\mathbf{t}_{\mathbf{s}} = \begin{cases} \Gamma & \text{if } \mathbf{s} = \{-1, 1\} \\ \lceil \Gamma/m \rceil & \text{if } \mathbf{s} = \{m, -\lceil m/2 \rceil, -\lfloor m/2 \rfloor\} \text{ or } \mathbf{s} = \{-m, \lceil m/2 \rceil, \lfloor m/2 \rfloor\} \text{ for some } m \in \{2, \ldots, \Delta\}. \end{cases}$$

Since $\|\mathbf{s}\|_1 \leq 3$ for all $\mathbf{s} \in \mathcal{S}$, we have

$$\|\mathbf{t}\|_{\mathcal{S}} \leq 3 \cdot \|\mathbf{t}\|_1 = \Gamma + \sum_{m=2,\ldots,\Delta} \lceil \Gamma/m \rceil = \Gamma \cdot O(\log \Delta) = O(\Delta \log^2 \Delta),$$

as desired.

Finally, we will show via induction on $|i|$ that

$$\sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},i}|}{\mathbf{t}_{\mathbf{s}}} \leq \frac{|i| \cdot \lceil 1 + \log |i| \rceil}{\Gamma}, \tag{7}$$

which, from our choice of Γ, implies the last property.

(Base Case) We have

$$\sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},-1}|}{\mathbf{t}_{\mathbf{s}}} = \frac{1}{\mathbf{t}_{\{-1,+1\}}} = \frac{1}{\Gamma}.$$

(Inductive Step) For convenience, in the derivation below, we think of $\mathbf{C}_{\mathbf{s},1}$ as 0 for all $\mathbf{s} \in \mathcal{S}$; note that in fact, row 1 does not exist for $\mathbf{C}$.

Suppose that (7) holds for all $i \in [-\Delta, \Delta]_{-1}$ such that $|i| < m$ for some $m \in \mathbb{N} \setminus \{1\}$. Now, consider $i = m$; letting $m_1 = \lceil m/2 \rceil, m_2 = \lfloor m/2 \rfloor$ from the definition of $\mathbf{c}_i$, we have

$$\begin{aligned}
\sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},i}|}{\mathbf{t}_{\mathbf{s}}} &\leq \frac{1}{\mathbf{t}_{\{m,-m_1,-m_2\}}} + \sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},m_1}|}{\mathbf{t}_{\mathbf{s}}} + \sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},m_2}|}{\mathbf{t}_{\mathbf{s}}} \\
\text{(From inductive hypothesis)} &\leq \frac{1}{\mathbf{t}_{\{m,-m_1,-m_2\}}} + \frac{m_1 \lceil 1 + \log m_1 \rceil}{\Gamma} + \frac{m_2 \lceil 1 + \log m_2 \rceil}{\Gamma} \\
&\leq \frac{1}{\mathbf{t}_{\{m,-m_1,-m_2\}}} + \frac{m_1 \lceil \log m \rceil}{\Gamma} + \frac{m_2 \lceil \log m \rceil}{\Gamma} \\
&= \frac{1}{\mathbf{t}_{\{m,-m_1,-m_2\}}} + \frac{m \lceil \log m \rceil}{\Gamma} \\
\text{(From our choice of } \mathbf{t}_{\{m,-m_1,-m_2\}}) &\leq \frac{m}{\Gamma} + \frac{m \lceil \log m \rceil}{\Gamma} \\
&= \frac{m \lceil 1 + \log m \rceil}{\Gamma}.
\end{aligned}$$

Similarly, for $i = -m$, we have

$$\begin{aligned}
\sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},i}|}{\mathbf{t}_{\mathbf{s}}} &\leq \frac{1}{\mathbf{t}_{\{-m,m_1,m_2\}}} + \sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},-m_1}|}{\mathbf{t}_{\mathbf{s}}} + \sum_{\mathbf{s} \in \mathcal{S}} \frac{|\mathbf{C}_{\mathbf{s},-m_2}|}{\mathbf{t}_{\mathbf{s}}} \\
\text{(From inductive hypothesis)} &\leq \frac{1}{\mathbf{t}_{\{-m,m_1,m_2\}}} + \frac{m_1 \lceil 1 + \log m_1 \rceil}{\Gamma} + \frac{m_2 \lceil 1 + \log m_2 \rceil}{\Gamma} \\
&\leq \frac{1}{\mathbf{t}_{\{-m,m_1,m_2\}}} + \frac{m_1 \lceil \log m \rceil}{\Gamma} + \frac{m_2 \lceil \log m \rceil}{\Gamma} \\
&= \frac{1}{\mathbf{t}_{\{-m,m_1,m_2\}}} + \frac{m \lceil \log m \rceil}{\Gamma} \\
\text{(From our choice of } \mathbf{t}_{\{-m,m_1,m_2\}}) &\leq \frac{m}{\Gamma} + \frac{m \lceil \log m \rceil}{\Gamma}
\end{aligned}$$

$$= \frac{m\lceil 1 + \log m\rceil}{\Gamma}. \qquad \square$$

D.4. Negative Binomial Mechanism for Linear Queries: Proof of Corollary 11

Proof of Corollary 11. We denote the $(\mathcal{D}^i)_{i\in\mathcal{I}}$-noise addition mechanism by $\mathcal{M}$. Furthermore, we write $\boldsymbol{\mathcal{D}}$ to denote the noise distribution (i.e., distribution of $\mathbf{z}$ where $z_i \sim \mathcal{D}^i$).

Consider two neighboring datasets $\mathbf{h}, \mathbf{h}'$. We have

$$\begin{aligned} d_\varepsilon(\mathcal{M}(\mathbf{h})\|\mathcal{M}(\mathbf{h}')) &= d_\varepsilon\left(\mathbf{Q}\mathbf{h} + \boldsymbol{\mathcal{D}} \| \mathbf{Q}\mathbf{h}' + \boldsymbol{\mathcal{D}}\right) \\ &= d_\varepsilon\left(\boldsymbol{\mathcal{D}} \| \mathbf{Q}(\mathbf{h}' - \mathbf{h}) + \boldsymbol{\mathcal{D}}\right) \\ &= d_\varepsilon\left(\prod_{i\in\mathcal{I}} \mathcal{D}^i \middle\| \prod_{i\in\mathcal{I}} ((\mathbf{Q}(\mathbf{h}' - \mathbf{h}))_i + \mathcal{D}^i)\right). \end{aligned}$$

Now, notice that $\mathbf{Q}(\mathbf{h}' - \mathbf{h})$ is dominated by $2\mathbf{t}$. Let $\varepsilon_i := \varepsilon \cdot \frac{|(\mathbf{Q}(\mathbf{h}'-\mathbf{h}))_i|}{2\mathbf{t}_i}$ for all $i \in \mathcal{I}$; we have $\sum_{i\in\mathcal{I}} \varepsilon_i \leq \varepsilon$. As a result, applying Lemma 16, we get

$$\begin{aligned} d_\varepsilon(\mathcal{M}(\mathbf{h})\|\mathcal{M}(\mathbf{h}')) &\leq \sum_{i\in\mathcal{I}} d_{\varepsilon_i}(\mathcal{D}^i \| (\mathbf{Q}(\mathbf{h}' - \mathbf{h}))_i + \mathcal{D}^i) \\ \text{(From Theorem 10 and our choice of } \varepsilon_i, p_i, r_i) &\leq \sum_{i\in\mathcal{I}} \delta/|\mathcal{I}| \\ &= \delta, \end{aligned}$$

which concludes our proof. $\square$

D.5. From Integers to Real Numbers: Proof of Theorem 1

We now prove Theorem 1. The proof follows the discretization via randomized rounding of (Balle et al., 2020).

Proof of Theorem 1. The algorithm for real summation works as follows:

- Let $\Delta = \lceil \frac{\varepsilon}{2} \cdot \sqrt{n/\zeta} \rceil$.
- Each user randomly round the input $x_i \in [0, 1]$ to $y_i \in \{0, \ldots, \Delta\}$ such that
$$y_i = \begin{cases} \lfloor x_i\Delta \rfloor & \text{with probability } 1 - (x_i - \lfloor x_i\Delta \rfloor), \\ \lfloor x_i\Delta \rfloor + 1 & \text{with probability } x_i - \lfloor x_i\Delta \rfloor. \end{cases}$$
- Run Δ-summation algorithm from Theorem 2 on the inputs $y_1, \ldots, y_n$ with $\gamma = \zeta/2$ to get an estimated sum s.
- Output s/Δ.

Using the communication guarantee of Theorem 2, we can conclude that, in the above algorithm, each user sends

$$1 + \left(\frac{\Delta}{\gamma\varepsilon n}\right) \cdot O\left(\log(1/\delta) + \log^2 \Delta \cdot \log(\Delta/\delta)\right) = 1 + \left(\frac{1}{\zeta\sqrt{\zeta n}}\right) O\left(\log(1/\delta) + \log^2(n/\zeta) \cdot \log(n/(\zeta\delta))\right)$$

messages in expectation and that each message contains

$$\lceil \log \Delta \rceil + 1 = \left\lceil \log\left(\frac{\varepsilon}{2} \cdot \sqrt{n/\zeta}\right) \right\rceil + 1 \leq 0.5(\log n + \log(1/\zeta)) + O(1)$$

bits.

We will next analyze the MSE of the protocol. As guaranteed by Theorem 2, s can be written as $y_1 + \cdots + y_n + e$ where e is distributed as $\mathrm{DLap}((1-\gamma)\varepsilon/\Delta)$. Since $e, \Delta x_1 - y_1, \ldots, \Delta x_n - y_n$ are all independent and have zero mean, the MSE can be rearranged as

$$\begin{aligned} \mathbb{E}[(x_1 + \cdots + x_n - s/\Delta)^2] &= \frac{1}{\Delta^2} \cdot \mathbb{E}[((\Delta x_1 - y_1) + \cdots + (\Delta x_n - y_n) - e)^2] \\ &= \frac{1}{\Delta^2}\left(\mathbb{E}[e^2] + \sum_{i=1}^n \mathbb{E}[(\Delta x_i - y_i)^2]\right). \end{aligned}$$

Recall that $\mathbb{E}[e^2]$ is simply the variance of the discrete Laplace distribution with parameter $(1-\gamma)\varepsilon/\Delta$, which is at most $\frac{2\Delta^2}{(1-\gamma)^2\varepsilon^2}$. Moreover, for each i, $\mathbb{E}[(\Delta x_i - y_i)^2] = (x_i - \lfloor x_i \Delta \rfloor)(1 - (x_i - \lfloor x_i \Delta \rfloor)) \leq 1/4$. Plugging this to the above equality, we have

$$\mathbb{E}[(x_1 + \cdots + x_n - s/\Delta)^2] = \frac{1}{\Delta^2}\left(\frac{2\Delta^2}{(1-\gamma)^2\varepsilon^2} + \frac{n}{4}\right) \leq \frac{2}{(1-\zeta)^2\varepsilon^2},$$

where the inequality follows from our parameter selection. The right hand side of the above inequality is indeed the MSE of Laplace mechanism with parameter $(1-\zeta)\varepsilon$. This concludes our proof. □

E. From Real Summation to 1-Sparse Vector Summation: Proof of Corollary 3

In this section, we prove our result for 1-Sparse Vector Summation (Corollary 3). The idea, formalized below, is to run our real-summation randomizer in each coordinate independently.

Proof of Corollary 3. Let $v^i \in \mathbb{R}^d$ denote the (1-sparse) vector input to the ith user. The randomizer simply runs the real summation randomizer on each coordinate of v^i, which gives the messages to send. It then attaches to each message the coordinate, as to distinguish the different coordinates. We stress that the randomizer for each coordinate is run with privacy parameters $(\varepsilon/2, \delta/2)$ instead of (ε, δ).

Algorithm 5 1-Sparse Vector Randomizer

```
1: procedure SparseVectorCorrNoiseRandomizer$_n(v^i)$
2:   for $j \in [d]$
3:     $S_j \leftarrow$ CorrNoiseRandomizer$^{\varepsilon/2,\delta/2}(v^i_j)$
4:     for $m \in S_j$
5:       Send $(i, m)$
```

The analyzer simply separates the messages based on the coordinates attached to them. Once this is done, messages corresponding to each coordinate are then plugged into the real summation randomizer (that just sums them up), which gives us the estimate of that coordinate. (Note that, similar to R, each R_j is a multiset.)

Algorithm 6 1-Sparse Vector Randomizer

```
1: procedure SparseVectorCorrNoiseAnalyzer
2:   $R \leftarrow$ multiset of messages received
3:   for $j \in [d]$
4:     $R_j \leftarrow \{m \mid (j, m) \in R_j\}$
5:   return (CorrNoiseAnalyzer$(R_1), \ldots,$ CorrNoiseAnalyzer$(R_d)$)
```

The accuracy claim follows trivially from that of the real summation accuracy in Theorem 1 with $(\varepsilon/2, \delta/2)$-DP. In terms of the communication complexity, since v^i is non-zero in only one coordinate and that the expected number of messages sent for CorrNoiseRandomizer with zero input is only $\tilde{O}_{\zeta,\varepsilon}\left(\frac{\log(1/\delta)}{\sqrt{n}}\right)$, the total expected number of messages sent by SparseVectorCorrNoiseRandomizer is

$$1 + d \cdot \tilde{O}_{\zeta,\varepsilon}\left(\frac{\log(1/\delta)}{\sqrt{n}}\right)$$

as desired. Furthermore, each message is the real summation message, which consists of only $\frac{1}{2}\log n + O(\log \frac{1}{\zeta})$ bits, appended with a coordinate, which can be represented in $\lceil \log d \rceil$ bits. Hence, the total number of bits required to represent each message of SparseVectorCorrNoiseRandomizer is $\log d + \frac{1}{2}\log n + O(\log \frac{1}{\zeta})$.

We will finally prove that the algorithm is $(\varepsilon/2, \delta/2)$-DP. Consider any two neighboring input datasets $X = (v_1, \ldots, v_n)$ and $X' = (v_1, \ldots, v_{n-1}, v'_n)$ that differs only on the last user's input. Let $J \subseteq [d]$ denote the set of coordinates that v_n, v'_n differ on. Since v_n, v'_n are both 1-sparse, we can conclude that $|J| \leq 2$. Notice that the view of the analyzer is $R_1, \ldots, R_d$; let $\mathcal{D}_1, \ldots, \mathcal{D}_d$ be the distributions of $R_1, \ldots, R_d$ respectively for the input dataset X, and $\mathcal{D}'_1, \ldots, \mathcal{D}'_d$ be the respective distributions for the input dataset X'. We have

$$\begin{aligned}
d_\varepsilon(\mathcal{D}_1 \times \cdots \times \mathcal{D}_d \| \mathcal{D}'_1 \times \cdots \times \mathcal{D}'_d) &= d_\varepsilon\left(\prod_{j \in J} \mathcal{D}_j \middle\| \prod_{j \in J} \mathcal{D}'_j\right) \\
\text{(From Lemma 16 and } |J| \le 2) &\le \sum_{j \in J} d_{\varepsilon/2}\left(\mathcal{D}_j \| \mathcal{D}'_j\right) \\
\text{(Since CorrNoiseRandomizer is } (\varepsilon/2, \delta/2)\text{-DP)} &\le \sum_{i \in J} \delta/2 \\
&\le \delta.
\end{aligned}$$

From this and since $R_1, \ldots, R_d$ are independent, we can conclude that the algorithm is (ε, δ)-DP as claimed. □

F. Concrete Parameter Computations

In this section, we describe modifications to the above proofs that result in a more practical set of parameters. As explained in the next section, these are used in our experiments to get a smaller amount of noise than the analytic bounds.

F.1. Tight Bound for Matrix-Specified Linear Queries

By following the definition of differential privacy, one can arrive at a compact description of when the mechanism is differentially private:

Lemma 18. *The $\boldsymbol{\mathcal{D}}$-noise addition mechanism is (ε, δ)-DP for $\mathbf{Q}$-linear query problem if the following holds for every pair of columns $j, j' \in [\Delta]$ of $\mathbf{Q}$:*

$$d_\varepsilon\left(\prod_{i \in \mathcal{I}} (\mathcal{D}^i + Q_{i,j} - Q_{i,j'}) \middle\| \prod_{i \in \mathcal{I}} \mathcal{D}^i\right) \le \delta.$$

Proof. This follows from the facts that the output distribution is $\prod_{i \in \mathcal{I}} (\mathcal{D}^i + \mathbf{Q}_{i,*}\mathbf{h})$, d_ε is shift-invariant, and Lemma 14. □

Plugging the above into the negative binomial mechanism, we get:

Lemma 19. *Suppose that $\mathcal{D}^i = \mathrm{NB}(r_i, p_i)$ for each $i \in \mathcal{I}$. The $(\mathcal{D}^i)_{\mathcal{I}}$-mechanism is (ε, δ)-DP for $\mathbf{Q}$-linear query problem if the following holds for every pair of columns $i, i' \in [q]$ of $\mathbf{Q}$:*

$$d_\varepsilon\left(\prod_{i \in \mathcal{I}} (\mathrm{NB}(r_i, p_i) + \mathbf{Q}_{i,j} - \mathbf{Q}_{i,j'}) \middle\| \prod_{i \in \mathcal{I}} \mathrm{NB}(r_i, p_i)\right) \le \delta.$$

G. Refined Analytic Bounds

We use the following analytic bound, which is a more refined version of Corollary 11. Notice below that we may select $\mathbf{t}' = 2\mathbf{t}$ where $\mathbf{t}$ is as in Corollary 11 and satisfies the requirement. However, the following bound allows us greater flexibility in choosing $\mathbf{t}'$. By optimizing for $\mathbf{t}'$, we get 30-50% reduction in the noise magnitude compared to using Corollary 11 together with the explicitly constructed $\mathbf{t}$ from the proof of Theorem 12.

Lemma 20. *Suppose that, for every pair of columns $\mathbf{q}_j, \mathbf{q}_{j'}$ of $\mathbf{Q} \in \mathbb{Z}^{\mathcal{I} \times [\Delta]}$, $\mathbf{q}_j - \mathbf{q}_{j'}$ is dominated by $\mathbf{t}' \in \mathbb{R}_+^{\mathcal{I}}$. For each $i \in \mathcal{I}$, let $p_i = e^{-0.2\varepsilon/t'_i}$, $r_i = 3(1 + \log(|\mathcal{I}|/\delta))$ and $\mathcal{D}^i = \mathrm{NB}(r_i, p_i)$. Then, $(\mathcal{D}^i)_{i \in \mathcal{I}}$-noise addition mechanism is (ε, δ)-DP for $\mathbf{Q}$-linear query.*

Proof. We denote the $(\mathcal{D}^i)_{i \in \mathcal{I}}$-noise addition mechanism by $\mathcal{M}$. Furthermore, we write $\boldsymbol{\mathcal{D}}$ to denote the noise distribution (i.e., distribution of $\mathbf{z}$ where $z_i \sim \mathcal{D}^i$).

Consider two neighboring datasets $\mathbf{h}, \mathbf{h}'$. We have

$$\begin{aligned}
d_\varepsilon(\mathcal{M}(\mathbf{h}) \| \mathcal{M}(\mathbf{h}')) &= d_\varepsilon\left(\mathbf{Q}\mathbf{h} + \boldsymbol{\mathcal{D}} \| \mathbf{Q}\mathbf{h}' + \boldsymbol{\mathcal{D}}\right) \\
&= d_\varepsilon\left(\boldsymbol{\mathcal{D}} \| \mathbf{Q}(\mathbf{h}' - \mathbf{h}) + \boldsymbol{\mathcal{D}}\right)
\end{aligned}$$

$$= d_\varepsilon\left(\prod_{i \in \mathcal{I}} \mathcal{D}^i \middle\| \prod_{i \in \mathcal{I}} ((\mathbf{Q}(\mathbf{h}' - \mathbf{h}))_i + \mathcal{D}^i)\right).$$

Now, notice that our assumption implies that $\mathbf{Q}(\mathbf{h}' - \mathbf{h})$ is dominated by $\mathbf{t}'$. Let $\varepsilon_i := \varepsilon \cdot \frac{|(\mathbf{Q}(\mathbf{h}'-\mathbf{h}))_i|}{t'_i}$ for all $i \in \mathcal{I}$; we have $\sum_{i \in \mathcal{I}} \varepsilon_i \leq \varepsilon$. As a result, applying Lemma 16, we get

$$\begin{aligned} d_\varepsilon(\mathcal{M}(\mathbf{h}) \| \mathcal{M}(\mathbf{h}')) &\leq \sum_{i \in \mathcal{I}} d_{\varepsilon_i}(\mathcal{D}^i \| (\mathbf{Q}(\mathbf{h}' - \mathbf{h}))_i + \mathcal{D}^i) \\ \text{(From Theorem 10 and our choice of } \varepsilon_i, p_i, r_i) &\leq \sum_{i \in \mathcal{I}} \delta / |\mathcal{I}| \\ &= \delta, \end{aligned}$$

which concludes our proof. □

H. Experimental Evaluation: Additional Details and Results

In this section, we give additional details on how to set the parameters and provide additional experimental results, including the effects of ε, δ on the Δ-summation protocols and an experiment on the real summation problem on a census dataset.

H.1. Noise parameter settings.

Our Correlated Noise Protocol. Recall that the notions of $\mathcal{D}^{\mathrm{central}}, \hat{\mathcal{D}}, \tilde{\mathcal{D}}^{\mathbf{s}}, \varepsilon^*, \varepsilon_1, \varepsilon_2, \delta_1, \delta_2$ from Theorem 2. We always set ε^* beforehand: in the Δ-summation experiments (where the entire errors come from the Discrete Laplace noises), ε^* is set to be 0.9ε as to minimize the noise. On the other hand, for real summation experiments where most of the errors are from discretization, we set ε^* to be 0.1ε; this is also to help reduce the number of messages as there is now more privacy budget allocated for the $\hat{\mathcal{D}}$ and $\tilde{\mathcal{D}}^{\mathbf{s}}$ noise. Once ε^* is set, we split the remaining $\varepsilon - \varepsilon^*$ among ε_1 and ε_2; we give between 50% and 90% to ε_1 depending on the other parameters, so as to minimize the number of messages.

Once the privacy budget is split, we first use Lemma 20 to set $\hat{\mathcal{D}} = \mathrm{NB}(\hat{r}, \hat{p}), \tilde{\mathcal{D}}^{\mathbf{s}} = \mathrm{NB}(r^{\mathbf{s}}, p^{\mathbf{s}})$; the latter also involves an optimization problem over $\mathbf{t}'$, which turns out to be solvable in a relatively short amount of time for moderate values of Δ. This gives us the "initial" parameters for $\mathbf{r}, \hat{p}, r^{\mathbf{s}}, p^{\mathbf{s}}$. We then formulate a generic optimization problem, together with the condition in Lemma 19, and use it to further search for improved parameters. This last search for parameters works well for small values of Δ and results in as much as $40 - 80\%$ reduction in the expected number of messages. However, as Δ grows, this gain becomes smaller since the number of parameters grows linearly with Δ and the condition in Lemma 19 also takes more resources to compute; thus, the search algorithm fails to yield parameters significantly better than the initial parameters.

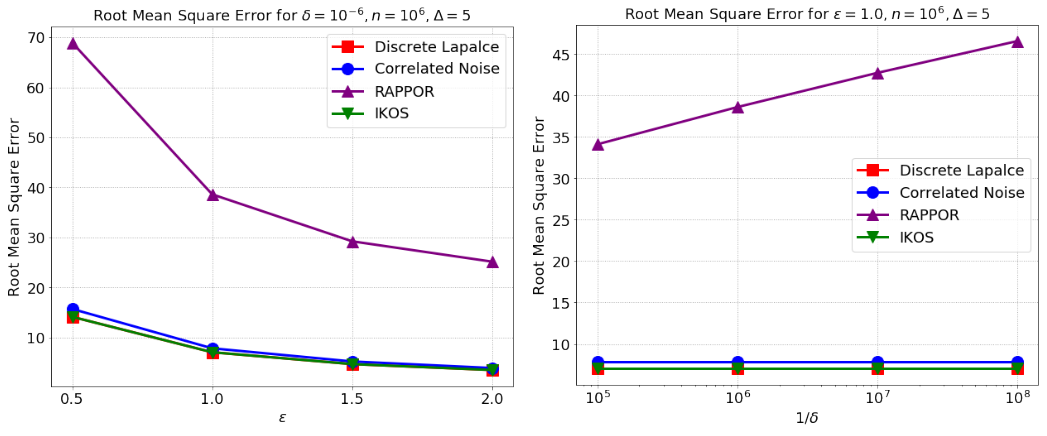
Fragmented RAPPOR. We use the same approach as Ghazi et al. (2020b) to set the parameter (i.e., flip probability) of Fragmented RAPPOR. At a high level, we set a parameter in an "optimistic" manner, meaning that the error and expected number of messages reported might be even lower than the true flip probability that is (ε, δ)-DP. Roughly speaking, when we would like to check whether a flip probability is feasible, we take two explicit input datasets X, X' and check whether $d_\varepsilon(\mathcal{M}(X) \| \mathcal{M}(X')) \leq \delta$. This is "optimistic" because, even if the parameter passes this test, it might still not be (ε, δ)-DP due to some other pair of datasets. For more information, please refer to Appendix G.3 of Ghazi et al. (2020b).

H.2. Effects of ε and δ on Δ-Summation Results

Recall that in Section 5, we consider the RMSEs and the communication complexity as Δ changes, and for the latter also as n changes. In this section, we extend the study to the effect of ε, δ. Here, we fix $n = 10^6$ and $\Delta = 5$ throughout. When δ varies, we fix $\varepsilon = 1$; when ε varies, we fix $\delta = 10^{-6}$.

Error. In terms of the error, RAPPOR's error slowly increases as δ decreases, while other algorithms' errors remain constant. As ε decreases, all algorithms' errors also increase. These are shown in Figure 3.

Communication. The number of bits required for the central DP algorithm remains constant (i.e., $\lceil \log \Delta \rceil$) regardless of the values of δ or ε. In all other algorithms, the expected number of bits sent per user decreases when δ or ε increases. These are demonstrated in Figure 4. We remark that this also holds for RAPPOR, even thought it might be hard to see from



(a) RMSEs for varying ε (b) RMSEs for varying δ

Figure 3: Error of our "correlated noise" real-summation protocol compared to other protocols on the rent dataset when varying ε or δ.

the plots because, in our regime of parameters, RAPPOR is within 0.1% of the central DP algorithm.

H.3. Experiments on Real Summation

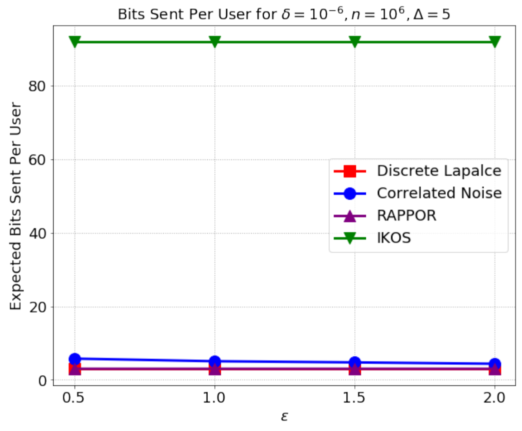
We evaluate our algorithm on the public 1940 US Census IPUMS dataset (Ruggles et al., 2020). We consider the *household rent* feature in this dataset; and consider the task of computing the average rent. We restrict to the rents below \$400; this reduced the number of reported rents from $68, 290, 222$ to $66, 994, 267$, preserving more than 98% of the data. We use randomized rounding (see the proof of Theorem 1) with number of discretization levels[12] $\Delta = 20, 50, 100, 200$. Note that 200 is the highest "natural" discretization that is non-trivial in our dataset, since the dataset only contains integers between 0 and 400.

We remark that, for all algorithms, the RMSE is input-dependent but there is still an explicit formula for it; the plots shown below use this formula. On the other hand, the communication bounds do not depend on the input dataset.

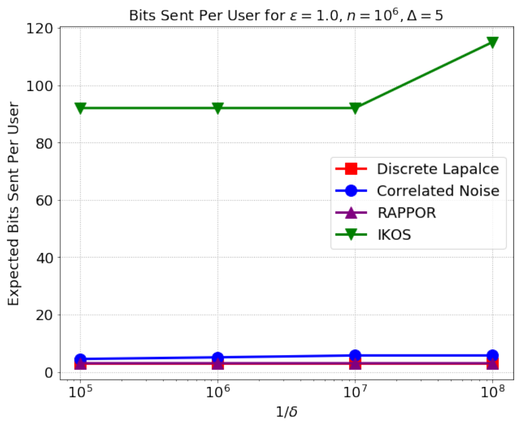
Error. In terms of the errors, unfortunately all of the shuffle DP mechanisms incur significantly higher errors than the Laplace mechanism in central DP. The reason is that the discretization (i.e., randomized rounding) error is already very large, even for the largest discretization level of $\Delta = 200$. Nonetheless, the errors for both our algorithm and IKOS decrease as the discretization level increases. Specifically, when $\Delta = 200$, the RMSEs for both of these mechanisms are less than half the RMSE of RAPPOR for *any* number of discretization levels. Notice also that the error for RAPPOR increases for large Δ; the reason is that, as shown in Section 5, the error of RAPPOR grows with Δ even without discretization. These results are shown in Figure 5a.

Communication. In terms of communication, we compare to the *non-private* "Discretized" algorithm that just discretizes the input into Δ levels and sends it to the analyzer; it requires only one message of $\lceil \log \Delta \rceil$ bits. RAPPOR incurs little (less than 1%) overhead, in expected number of bits sent, compared to this baseline. When Δ is small, our correlated noise algorithm incurs a small amount of overhead (less than 20% for $\Delta = 25$), but this overhead grows moderately as Δ grows (i.e., to less than 60% for $\Delta = 200$). Finally, we note that, since the number n of users is very large, the message length of IKOS (which must be at least $\log n\Delta$) is already large and thus the number of bits sent per user is very large — more than 10 times the baseline.

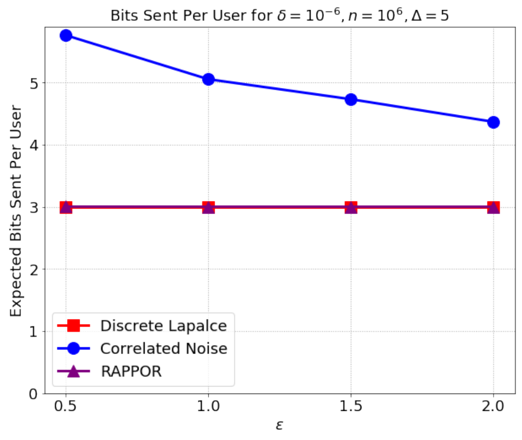
[12]This means that we first use randomized rounding as in the proof of Theorem 1, then run the Δ-summation protocol, and the analyzer just multiplies the estimate by $400/\Delta$.



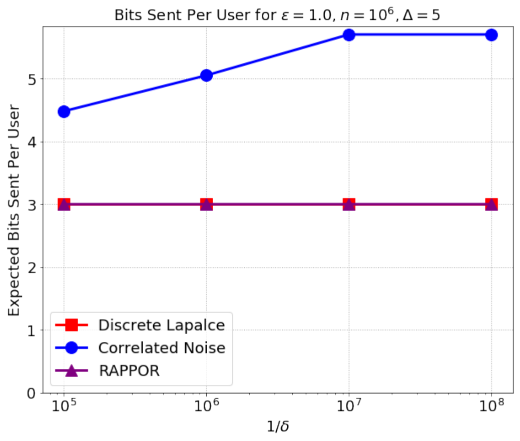
(a) Expected bits sent for varying ε



(b) Expected bits sent for varying δ

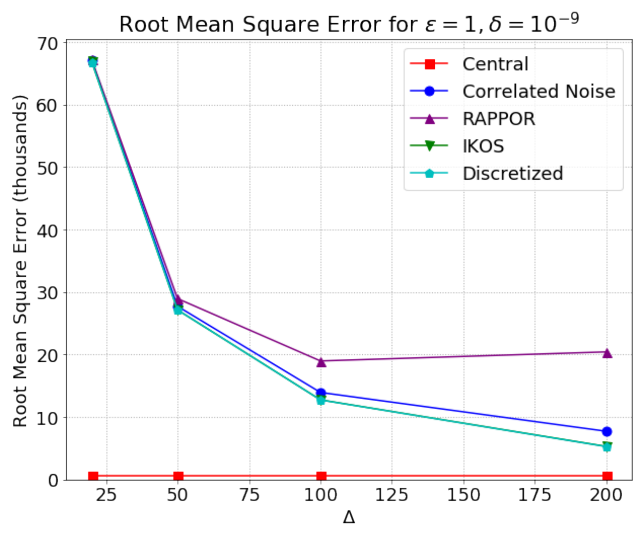


(c) Expected bits sent for varying ε (without IKOS)

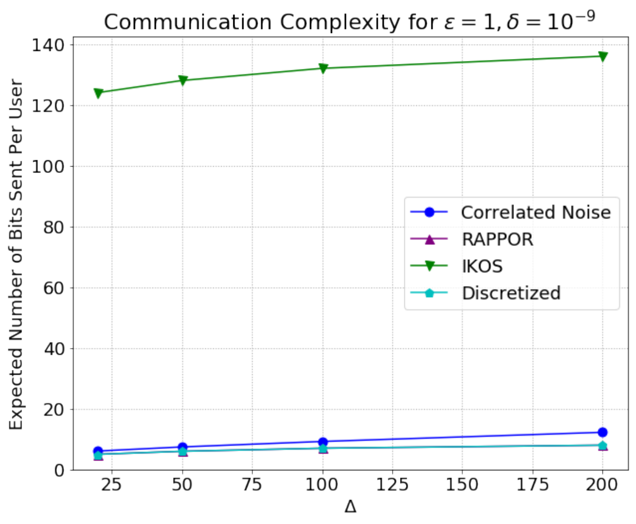


(d) Expected bits sent for varying δ (without IKOS)

Figure 4: Communication complexity of our "correlated noise" real-summation protocol compared to other protocols on the rent dataset when varying ε or δ. Since the number of bits sent in the IKOS algorithm is usually very large, it is hard to distinguish the other three lines in the plots where it is included; so we provide the plots without the IKOS algorithm as well.



(a) RMSEs for varying Δ



(b) Expected bits sent for varying Δ

Figure 5: Error and communication complexity of our "correlated noise" real-summation protocol compared to other protocols on the rent dataset. The Laplace mechanism is included for comparison in the RMSE plot, even though it is not implementable in the shuffle model; it is not included in the communication plot since the communication is not well-defined. For both plots, we also include "Discretized" which is a *non-private* algorithm that just discretizes the input and sends it to the analyzer, which then sums up all the incoming messages.