Individualized Privacy Accounting via Subsampling with Applications in Combinatorial Optimization

Badih Ghazi [1] **Pritish Kamath** [1] **Ravi Kumar** [1] **Pasin Manurangsi** [2] **Adam Sealfon** [3]

Abstract

In this work, we give a new technique for analyzing individualized privacy accounting via the following simple observation: if an algorithm is one-sided add-DP, then its subsampled variant satisfies two-sided DP. From this, we obtain several improved algorithms for private combinatorial optimization problems, including decomposable submodular maximization and set cover. Our error guarantees are asymptotically tight and our algorithm satisfies *pure*-DP while previously known algorithms (Gupta et al., 2010; Chaturvedi et al., 2021) are *approximate*-DP. We also show an application of our technique beyond combinatorial optimization by giving a pure-DP algorithm for the shifting heavy hitter problem in a stream; previously, only an approximate-DP algorithm was known (Kaplan et al., 2021; Cohen & Lyu, 2023).

1. Introduction

In combinatorial optimization, we typically wish to select a discrete object to minimize or maximize certain objective functions subject to certain constraints. In several settings, such objective functions or constraints may depend on sensitive information of users. For example, clustering and facility location tasks may involve taking users' location information as part of the objectives or constraints. Similarly, data summarization may require user-produced examples as part of the objective (Mirzasoleiman et al., 2016). Due to this, several works have considered studying combinatorial optimization problems under *differential privacy (DP)* (Dwork et al., 2006b;a)—a widely-used and rigorous notion to quantify privacy properties of an algorithm. To state the definition, we use $\mathcal{X}$ to denote the domain of each user's data. Two datasets $D, D' \subseteq \mathcal{X}^*$ are said to be *add-remove neighbors* if D is a result of adding an element to D' or removing an element from D'.

Definition 1.1 (Add-remove DP, Dwork et al. (2006b))**.** A randomized algorithm $\mathcal{M} : \mathcal{X}^* \to \mathcal{O}$ is (ε, δ)-*DP* if, for every add-remove neighboring datasets D, D' and every set $S \subseteq \mathcal{O}$ of outcomes, we have

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in S] + \delta.$$

When $\delta > 0$, we say that the algorithm satisfies *approximate-DP*; when $\delta = 0$, we say that the algorithm satisfies *pure-DP*. The latter is preferable as it provides stronger privacy protection; more specifically, it does not allow for "catastrophic failure" where the sensitive input data is leaked in the clear.

Submodular Maximization. In submodular maximization, we are given a submodular[1] set function $F : 2^{[m]} \to \mathbb{R}$, where $[m] = \{1, \ldots, m\}$ is the universe. The goal is to find a subset $T \subseteq [m]$ that maximizes $F(T)$ under certain constraints. In this work, we consider two types of constraints:

- *Cardinality Constraint*: Here we want $|T| = k$.
- *Matroid Constraint*: Given a rank-k matroid $M = ([m], \mathcal{I})$, we want T to be an independent set of the matroid (i.e., $T \in \mathcal{I}$). Note that this generalizes the cardinality constraint (when M is the uniform matroid).

Submodular maximization is among the most well-studied problems in combinatorial optimization; several algorithms date back to the 70's (Nemhauser et al., 1978) and many variants continue to be studied to this day (e.g., Duetting et al. (2023); Banihashem et al. (2023)).

In this work, we consider the 1-decomposable[2] monotone submodular maximization problem[3]. Here each $x \in \mathcal{X}$ is associated with a monotone submodular function $f_x : 2^{[m]} \to [0, 1]$ and the goal is to maximize $F_D := \sum_{x \in D} f_x$.

For DP submodular maximization under cardinality con-

*Equal contribution [1]Google Research, Mountain View, USA [2]Google Research, Thailand [3]Google Research, New York, USA. Correspondence to: Pasin Manurangsi <pasin@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]A set function F is *submodular* if, for every $S \subseteq T \subseteq [m]$ and $v \in [m]$, we have $F(S \cup \{v\}) - F(S) \geq F(T \cup \{v\}) - F(T)$.

[2]More generally, λ-*decomposable* refers to the same definition but with $f_x : 2^{[m]} \to [0, \lambda]$. All results discussed in this paper applied to λ-decomposable functions as well by appropriately scaling the functions.

[3]Aka the *Combinatorial Public Project* problem (Papadimitriou et al., 2008).

straint, Gupta et al. (2010) gave a polynomial-time (ε, δ)-DP algorithm that achieves[4] $\left(1 - \frac{1}{e}\right)$-approximation and $O\left(\frac{k \log m \log(1/\delta)}{\varepsilon}\right)$-error. They also show a lower bound of $\Omega\left(\frac{k \log m}{\varepsilon}\right)$ on the error. Since $\left(1 - \frac{1}{e} + o(1)\right)$-approximation is NP-hard (Feige, 1998), Gupta et al.'s result is tight up to the $O(\log(1/\delta))$ factor in the error. The matroid constraint case was first studied by Mitrovic et al. (2017), who gave an efficient (ε, δ)-DP $\frac{1}{2}$-approximation and the same error bound. Recently, Chaturvedi et al. (2021) improved the approximation ratio to $\left(1 - \frac{1}{e} - \eta\right)$ for any constant $\eta > 0$ while retaining the same error bound. Again, this is tight up to a factor of $O\left(\log(1/\delta)\right)$ in the error.

Set Cover. In the *Set Cover* problem, we are given a set system $(\mathcal{U}, \mathcal{S} = (S_1, \ldots, S_m))$. The goal is to output as few sets as possible that cover the universe, i.e., $S_{i_1}, \ldots, S_{i_k}$ such that $S_{i_1} \cup \cdots \cup S_{i_k} = \mathcal{U}$. We use $\mathrm{SetCov}(\mathcal{U}, \mathcal{S})$ to denote the optimal size of the set cover. Set Cover can be viewed as a "dual" version of submodular maximization under cardinality constraint, since the coverage function[5] $F(I) = |\bigcup_{i \in I} S_i|$ is submodular. Set Cover is a classic combinatorial optimization problem, being one of the 21 original NP-complete problems (Karp, 1972).

For private Set Cover, DP is w.r.t. adding or removing an item from the universe (and all the sets)[6]. Unfortunately, Gupta et al. (2010) show that no non-trivial approximation is possible for the setting where we output the indices $i_1, \ldots, i_k$ directly. Instead, they proposed what is now sometimes referred to as the *open set* setting, where we instead output a permutation $\pi : [m] \to [m]$. Each element $x \in \mathcal{U}$ then chooses the first set containing it in the sequence (i.e., $\min\{i \mid x \in S_{\pi(i)}\}$). The cost is then defined as the number of sets that are chosen; we use $\mathrm{CostSetCov}_{\mathcal{U},\mathcal{S}}(\pi)$ to denote the cost of π. We work in this model throughout the paper. Under this model, they provide an (ε, δ)-DP algorithm with an expected approximation ratio $O\left(\ln n + \frac{\ln m \log(1/\delta)}{\varepsilon}\right)$ for the problem where n denotes the size of the input dataset $|\mathcal{U}|$. This is nearly tight as it is NP-hard to achieve an $o(\ln n)$-approximation (Dinur & Steurer, 2014; Moshkovitz, 2015), and Gupta et al. (2010) show that no ε-DP algorithm can achieve an $o\left(\frac{\ln m}{\varepsilon}\right)$-approximation.

Metric k-Means and k-Median. In the *metric* (k, q)-*clustering* problem, there is a metric space $([m], d)$ whose diameter[7] is at most one, each user input is a point in this metric space (i.e., $\mathcal{X} = [m]$), and the goal is to output a subset $S \subseteq [m]$ of size k that minimizes $\mathrm{cost}^q(S; D) := \sum_{x \in D} \min_{c \in S} d(c, x)^q$. When $q = 1$ and $q = 2$, this problem is referred to as *metric k-median* and *metric k-means* respectively. In the non-private setting, even though constant-factor approximation algorithms for these problems have long been known (Charikar et al., 1999), tight (hardness of) approximation ratios are not yet known and this remains a challenging and active area of research (see, e.g., Anand & Lee (2024) and the references therein).

For private k-median, Gupta et al. (2010) gave an ε-DP algorithm with approximation ratio[8] $(5+\eta)$ for any constant $\eta > 0$ with error[9] $O\left(\frac{k^2 \log^2(mn)}{\varepsilon}\right)$. On the lower bound front, they showed that any ε-DP algorithm must incur error at least $\Omega(k \log n)$. Later, the error bound was improved by Jones et al. (2021) to $O\left(\frac{k \log(mn) \log(1/\delta)}{\varepsilon}\right)$, albeit with an (ε, δ)-DP algorithm. The approximation ratio of Jones et al. (2021) is a constant, but for simplicity is not explicit.

Shifting Heavy Hitters. Our framework will also apply to the ostensibly unrelated *shifting heavy hitters* problem (Kaplan et al., 2021). Here, each user i's data $\mathbf{x}_i$ is a stream $(x_{i,1}, \ldots, x_{i,T}) \in \mathcal{Y}^T$ where $x_{i,t}$ is the "bucket" that the user contributes to at time t. For $\tau \geq 0, t \in [T]$, a *τ-heavy hitter* at time t is an element $y \in \mathcal{Y}$ that appears at least τ times, i.e., $w_t(y) \geq \tau$ where $w_t(y) := |\{i \in [n] \mid x_{i,t} = y\}|$. Following Kaplan et al. (2021), an algorithm is said to have error τ with probability (w.p.) $1 - \beta$ if the following holds w.p. $1 - \beta$ for all $t \in [T]$:

- Every reported element x satisfies $w_t(x) > 0$.
- Every τ-heavy hitter is reported.

Without any additional assumption, the best error one can achieve with (ε, δ)-DP is $\tilde{O}(\sqrt{T \log(1/\delta)})$. The main result of Kaplan et al. (2021) is that, under the assumption that each user contributes to at most k heavy hitters, the error can be reduced to $\tilde{O}(\sqrt{k} \cdot \log(1/\delta) \log T)$.

All state-of-the-art algorithms we have discussed so far are *approximate-DP*. Meanwhile, known pure-DP algorithms have significantly worse error guarantees; in fact, for some problems such as private set cover and shifting heavy hitters, no non-trivial pure-DP algorithms are known. This leads us to the main question of this work: *Are there pure-DP algorithms that achieve similar (or even better) bounds?*

[4] An output T^* is said to achieve *α-approximation* and *κ-error* if $F(T^*) \geq \alpha \cdot \mathrm{OPT} - \kappa$, where OPT is the optimum.

[5] Maximizing the coverage function among k-size I is known as the *max k-Coverage* problem, which is a special case of submodular maximization with cardinality constraint and is also well studied in the literature.

[6] More precisely, we can define $\mathcal{X} = \{0, 1\}^m$, where $x \in \mathcal{X}$ belongs to all S_i such that $x_i = 1$.

[7] The diameter is defined as $\max_{a,b \in [m]} d(a, b)$.

[8] They only claim an approx. ratio of 6 but it is straightforward to see that this can be extended to any approx. ratio greater than 5.

[9] We note that in both (Gupta et al., 2010) and (Jones et al., 2021), it was assumed that $n \leq m$.

1.1. Our Results

We answer the above question positively by giving pure-DP algorithms for all above problems via a unified framework. As explained below, our error bounds are all nearly tight. In fact, for the optimization problems, we even improve on the error bounds from previous approximate-DP algorithms.

We note that all of our algorithms run in polynomial time and we will not state this explicitly below for brevity.

Monotone Submodular Maximization. For monotone submodular maximization with a cardinality constraint, we can get an approximation ratio arbitrarily close to $1 - \frac{1}{e}$ while having an error $O\left(\frac{k \log m}{\varepsilon}\right)$, as stated more precisely below. The former matches (Gupta et al., 2010) where the latter improves on their bound by a factor of $O(\log(1/\delta))$ and is tight due to their $\Omega\left(\frac{k \log m}{\varepsilon}\right)$ lower bound.

Theorem 1.2. *For any $0 < \varepsilon, \beta, \eta < 1$, there is an ε-DP algorithm for monotone submodular maximization under a cardinality constraint that achieves $\left(1 - \frac{1}{e} - \eta\right)$-approximation and $O\left(\frac{k \log(m/\beta)}{\eta\varepsilon}\right)$-error w.p. $1 - \beta$.*

For a matroid constraint, we get almost the same bound except for a slightly worse dependency on the parameter η:

Theorem 1.3. *For any $0 < \varepsilon, \beta, \eta < 1$, there is an ε-DP algorithm for monotone submodular maximization under matroid submodular maximization that achieves $\left(1 - \frac{1}{e} - \eta\right)$-approximation and $O\left(\frac{k \log\left(\frac{m}{\eta\beta}\right)}{\eta\varepsilon}\right)$-error w.p. $1 - \beta$.*

Set Cover. For the private set cover problem, we give a pure-DP algorithm that improves the approximation ratio from Gupta et al. (2010) by a factor of $O(\log(1/\delta))$. This is tight due to the aforementioned $\Omega\left(\frac{\log m}{\varepsilon}\right)$ from Gupta et al. (2010) and the NP-hardness of factor $\Omega(\log n)$ (Dinur & Steurer, 2014; Moshkovitz, 2015).

Theorem 1.4. *For any $0 < \varepsilon, \beta < 1$, there is an ε-DP algorithm for Set Cover that achieves $O\left(\log n + \frac{\log(m/\beta)}{\varepsilon}\right)$-approximation w.p. $1 - \beta$.*

Metric k-Means and k-Median For private metric k-means and k-median, we provide a pure-DP algorithm with approximation ratio $O(1)$ and error $O\left(\frac{k \log(mn)}{\varepsilon}\right)$, as stated below. The error bound improves upon the approximate-DP algorithm of Jones et al. (2021) by a factor of $O(\log(1/\delta))$. For $n \leq m^{O(1)}$, our error bound is tight due to the aforementioned lower bound $\Omega\left(\frac{k \log m}{\varepsilon}\right)$ from Gupta et al. (2010). Similar to (Jones et al., 2021), we choose to keep our analysis simple, and thus we do not compute the approximation ratio explicitly. As we mentioned earlier, the tight approximation ratio is not known even in the non-private setting.

Theorem 1.5. *For any $0 < \varepsilon, \beta < 1$, there is an ε-DP algorithm for metric k-median and k-means that achieves an $O(1)$-approximation and $O\left(\frac{k \log(mn/\beta)}{\varepsilon}\right)$-error w.p. $1 - \beta$.*

Shifting Heavy Hitters. We provide a pure-DP algorithm for the problem, which is stated informally below; for a formal version, see Theorem 5.6.

Theorem 1.6 (Informal)**.** *Assuming that each user contributes to $\leq k$ heavy hitters, there is an ε-DP shifting heavy hitter algorithm with error $O\left(\frac{k \log(T|\mathcal{Y}|/\beta)}{\varepsilon}\right)$ w.p. $1 - \beta$.*

In comparison, the error bound of Kaplan et al. (2021) is $\tilde{O}\left(\frac{\sqrt{k} \log(1/\delta) \log(T|\mathcal{Y}|/\beta)}{\varepsilon}\right)$, which can be smaller for large k. However, their algorithm is approximate-DP and it can be easily seen (via a packing lower bound) that our bound is the best possible for pure-DP; see Appendix B.

1.2. Technical Overview

There are two slightly different settings to which our techniques apply: *repeated exponential mechanism* and *repeated above threshold*. In this overview, we will focus on the repeated exponential mechanism and only briefly mention the repeated above threshold mechanism at the end.

Repeated Exponential Mechanism. Gupta et al. (2010) proposed the following algorithm for Set Cover: repeatedly use the ε_0-DP exponential mechanism (McSherry & Talwar, 2007)[10] to find the next set that covers the maximum number of (uncovered) element. Since the exponential mechanism is applied m times here, if we were to apply a composition theorem, the error would grow polynomially in m (either m for basic composition or $\sqrt{m}$ for advanced composition (Dwork et al., 2010)). Perhaps surprisingly, they instead show that this algorithm is (ε, δ)-DP for $\varepsilon = O(\varepsilon_0 \cdot \log(1/\delta))$, i.e., independent of m.

The intuition behind this is roughly that an element "causes" only a single set to be picked: the first one in the permutation that contains it. The sets picked before this set have their scores (of the exponential mechanism) completely independent from the element. On the other hand, for the sets picked after this set, the element is already covered and does not factor into the scores at all. Thus, we should be able to "charge the privacy budget" only once when this particular set is picked. While Gupta et al. (2010) show that such an *individualized privacy accounting* works for approximate-DP, it unfortunately fails for pure-DP: this mechanism is not ε-DP for any $\varepsilon = o(m\varepsilon_0)$. (See Appendix C.)

One-Sided DP. This is a notion of DP where the "neighboring relationship" can be asymmetric (Kotsogiannis et al., 2020). (See also (Takagi et al., 2022), who call this *asym-*

[10] See Section 2 for more detail.

metric DP.) In particular, we consider the following notion of one-sided DP with respect to adding an element[11].

Definition 1.7 (One-Sided DP, Kotsogiannis et al. (2020))**.** A mechanism $\mathcal{M}$ is said to be *ε-add-DP* iff, for every pair D, D' of datasets such that D results from adding an item to D' and every possible subset S of outputs, we have $\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in S]$.

To emphasize the differences, we will refer to add-remove DP (Definition 1.1) as "two-sided DP".

It turns out that, while it fails for two-sided DP, the above intuition does apply to one-sided DP: the repeated exponential mechanism is ε_0-add-DP. The proof of this fact is also relatively straightforward (see Section 4).

From One-Sided to Two-Sided DP. While the above observation is nice, we have not accomplished our goal yet, since we wish to design a two-sided DP mechanism, not a one-sided DP one. This brings us to our second observation: we can turn any one-sided DP mechanism into a two-sided one by subsampling with probability $p = 1 - e^{-\varepsilon}$ (see Lemma 3.1). With these two ingredients, we arrive at our result by just using the subsampled version of the existing—e.g., Gupta et al. (2010)'s—algorithm!

Repeated Above Threshold. Our technique also applies to a slightly different setting where we consider a stream and wish to detect if a query is above a certain threshold at each time step. Again, we achieve an "individualized privacy accounting", where the amount of noise required to achieve one-sided DP scales only with the number of times an individual contributes to above threshold results. The subsampled version of this then satisfies two-sided DP and provides our algorithm for the shifting heavy hitter problem.

Lastly, we note that, while we use the repeated exponential mechanism for submodular maximization results, we actually do *not* use it for Set Cover. This is due to a technical barrier that only allows us to get $O(\log n \log m/\varepsilon)$ via this approach. (See Appendix F for more details.) For Theorem 1.4, we actually use the repeated above threshold mechanism applied to a (non-private) streaming approximation algorithm for Set Cover (Kumar et al., 2015).

2. Preliminaries

Subsampling. Subsampling is a standard technique in DP (Balle et al., 2018; Wang et al., 2020). We will use the so-called Poisson subsampling where each user is kept with probability p. More precisely, we write $\mathcal{S}^p : \mathcal{X}^* \to \mathcal{X}^*$ as a subsampling operator, i.e., $\mathcal{S}_p(D)$ outputs a random subset of D such that each user is kept independently with probability p. For any mechanism $\mathcal{M}$, we write $\mathcal{M}^{\mathcal{S}_p}$ to denote the mechanism $D \mapsto \mathcal{M}(\mathcal{S}_p(D))$.

Concentration Inequalities. It will be convenient to use a version of the Chernoff bound that includes both multiplicative and additive terms, as stated below.

Theorem 2.1 (Chernoff bound; Corollary 2.11 of Chaturvedi et al. 2020)**.** *Let $Z_1, \ldots, Z_m$ be i.i.d. random variables such that $Z_i \in [0,1]$ and let $\mu = \mathbb{E}[Z_1 + \cdots + Z_m]$. Then, for $\alpha \in [0,1]$ and $\zeta \geq 0$, we have*

$$\Pr[Z_1 + \cdots + Z_m < (1-\alpha)\mu - \zeta] \leq \exp\left(-\alpha\zeta\right),$$
$$\Pr[Z_1 + \cdots + Z_m > (1+\alpha)\mu + \zeta] \leq \exp\left(-\alpha\zeta/3\right).$$

Exponential Mechanism. Given a candidate set $\mathcal{C}$ and a scoring function $q : \mathcal{C} \times \mathcal{X}^* \to \mathbb{R}$, the *exponential mechanism* $\textsc{ExpMech}_{\varepsilon}(\mathcal{C}, q; D)$ outputs each candidate $c \in \mathcal{C}$ with probability proportional to $\exp\left(\varepsilon \cdot q(c; D)\right)$. Its guarantee is as follows:

Theorem 2.2 (McSherry & Talwar (2007))**.** *If q has sensitivity at most Δ (w.r.t. D), then the exponential mechanism is $(2\varepsilon\Delta)$-DP. Furthermore, w.p. $1-\beta$, the output c^* satisfies $q(c^*; D) \geq \max_{c \in \mathcal{C}} q(c; D) - O(\log(|\mathcal{C}|/\beta)/\varepsilon)$.*

3. One-Sided DP and Subsampling

We start by proving our main observation that subsampling a one-sided DP mechanism makes it two-sided DP. We remark that, while there is a large literature on privacy amplification by subsampling, we are not aware of such a connection between one-sided and two-sided DP before.

Lemma 3.1. *For any $p \in [0,1)$ and $\varepsilon_0 > 0$, if $\mathcal{M}$ is an ε_0-add-DP mechanism, then $\mathcal{M}^{\mathcal{S}_p}$ is ε-DP for $\varepsilon = \ln\left(\max\left\{\frac{1}{1-p}, 1 + p(e^{\varepsilon_0} - 1)\right\}\right)$.*

Proof. Consider any datasets D, D' such that $D = D' \cup \{x\}$, and any possible output o. On the one hand, we have

$$\begin{aligned}
&\Pr[\mathcal{M}^{\mathcal{S}_p}(D) = o] \\
&= \sum_{D_s \subseteq D} \Pr[\mathcal{M}(D_s) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s] \\
&\geq \sum_{D_s \subseteq D'} \Pr[\mathcal{M}(D_s) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s] \\
&\overset{(\star)}{=} \sum_{D_s \subseteq D'} \Pr[\mathcal{M}(D_s) = o] \cdot (1-p) \Pr[\mathcal{S}_p(D') = D_s] \\
&= (1-p) \cdot \Pr[\mathcal{M}^{\mathcal{S}_p}(D') = o],
\end{aligned}$$

where $(\star)$ uses the fact that $\mathcal{S}_p(D)$ is the same as $\mathcal{S}_p(D')$ when conditioned on x not being selected.

On the other hand, we have[12]

$$\Pr[\mathcal{M}^{\mathcal{S}_p}(D) = o]$$

[11] Kotsogiannis et al. (2020) actually define one-sided DP with respect to *replacing* a sensitive record. However, we are defining it with respect to adding an element.

[12] The following sequence of inequalities is standard in

$$
\begin{aligned}
&= \sum_{D_s \subseteq D'} \Big(\Pr[\mathcal{M}(D_s) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s] \\
&\quad + \Pr[\mathcal{M}(D_s \cup \{x\}) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s \cup \{x\}] \Big) \\
&\overset{(\bigstar)}{\leq} \sum_{D_s \subseteq D'} \Big(\Pr[\mathcal{M}(D_s) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s] \\
&\quad + e^{\varepsilon_0} \cdot \Pr[\mathcal{M}(D_s) = o] \cdot \Pr[\mathcal{S}_p(D) = D_s \cup \{x\}] \Big) \\
&\overset{(\blacklozenge)}{=} \sum_{D_s \subseteq D'} \Big(\Pr[\mathcal{M}(D_s) = o] \cdot (1-p) \cdot \Pr[\mathcal{S}_p(D') = D_s] \\
&\quad + e^{\varepsilon_0} \cdot \Pr[\mathcal{M}(D_s) = o] \cdot p \cdot \Pr[\mathcal{S}_p(D') = D_s] \Big) \\
&= (1 + p(e^{\varepsilon_0} - 1)) \cdot \Pr[\mathcal{M}^{\mathcal{S}_p}(D') = o],
\end{aligned}
$$

where ★ follows from the ε_0-add-DP property of $\mathcal{M}$ and ♦ follows from the fact that x is included in D_s with probability p independently of other items.

Thus, the algorithm is ε-DP as claimed. □

We note that Lemma 3.1 can be extended to approximate-DP or Rényi-DP by replacing the second half of the proof by the corresponding amplification-by-subsampling proofs.

The following corollary, which is an immediate consequence of plugging in $\varepsilon_0 = \ln(2)$ and $p = 1 - e^{-\varepsilon}$ into Lemma 3.1, will be more convenient to work with throughout the remainder of the paper. It is useful to note that $p = \Theta(\varepsilon)$ for $\varepsilon \leq 1$ while $\varepsilon_0 = \ln(2)$ is independent of ε.

Corollary 3.2. *For any $\varepsilon > 0$, if $\mathcal{M}$ is $\ln(2)$-add-DP, then $\mathcal{M}^{\mathcal{S}_p}$ is ε-DP for $p = 1 - e^{-\varepsilon}$.*

4. Algorithm I: Repeated EM

In the first setting, the interaction proceeds in L rounds. In round i, the algorithm is given a candidate set $\mathcal{C}_i$ and a scoring function $q_i : \mathcal{C}_i \times \mathcal{X}^* \to \mathbb{R}$ (which can depend on the output of previous rounds). The goal is to output $c_i^* \in \mathcal{C}_i$ which achieves an approximately maximum score $q_i(c; D)$. The algorithm we use (Algorithm 1)—originally from Gupta et al. (2010)—simply applies the exponential mechanism at each step.

To analyze the algorithm, we need a couple of assumptions.

Assumption 4.1 (Monotonicity)**.** For every i, c, adding an element to D does not decrease $q_i(c; D)$.

Assumption 4.2 (Bounded Realized Sensitivity)**.** For every add-remove neighbors D, D' and every possible output $(c_1^*, \ldots, c_L^*)$, $\sum_{i \in [L]} |q_i(c_i^*; D) - q_i(c_i^*; D')| \leq \Delta$.

amplification-by-subsampling literature (e.g., (Li et al., 2012)); we only include it here for completeness.

Algorithm 1 REPEATED-EM$_{\varepsilon_0, \mathcal{A}}$ (Gupta et al., 2010)

Parameters: $\varepsilon_0 > 0$, an algorithm $\mathcal{A}$ for selecting a candidate set and a scoring function
Input: Dataset $D \in \mathcal{X}^*$
for $i = 1, \ldots, L$ **do**
  $(\mathcal{C}_i, q_i) \leftarrow \mathcal{A}(c_1^*, \ldots, c_{i-1}^*)$
  $c_i^* \leftarrow \text{EXPMECH}_{\frac{\varepsilon_0}{\Delta}}(\mathcal{C}_i, q_i; D)$
return $(c_1^*, \ldots, c_L^*)$

Under these assumptions, the algorithm is add-DP:

Theorem 4.3. *Under Assumptions 4.1 and 4.2,* REPEATED-EM$_{\varepsilon_0}$ *(Algorithm 1) is ε_0-add-DP.*

Proof. Consider any D, D' such that $D = D' \cup \{x\}$ for some x, and any output $o = (o_1, \ldots, o_L)$. We will write $\mathcal{M}$ as a shorthand for REPEATED-EM$_{\varepsilon_0, \mathcal{A}}$. We have

$$
\begin{aligned}
\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]} &= \prod_{i \in [L]} \frac{\Pr[\text{EXPMECH}_{\frac{\varepsilon_0}{\Delta}}(\mathcal{C}_i, q_i; D) = o_i]}{\Pr[\text{EXPMECH}_{\frac{\varepsilon_0}{\Delta}}(\mathcal{C}_i, q_i; D') = o_i]} \\
&= \prod_{i \in [L]} \frac{\frac{\exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(o_i; D)\right)}{\sum_{c \in \mathcal{C}_i} \exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(c; D)\right)}}{\frac{\exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(o_i; D')\right)}{\sum_{c \in \mathcal{C}_i} \exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(c; D')\right)}}.
\end{aligned}
$$

Assumption 4.1 implies that $\sum_{c \in \mathcal{C}_i} \exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(c; D)\right) \geq \sum_{c \in \mathcal{C}_i} \exp\left(\frac{\varepsilon_0}{\Delta} \cdot q_i(c; D')\right)$. Thus, we have

$$
\begin{aligned}
\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]} &\leq \exp\left(\frac{\varepsilon_0}{\Delta} \cdot \sum_{i \in [L]} (q_i(o_i; D) - q_i(o_i; D'))\right) \\
&\leq \exp(\varepsilon_0),
\end{aligned}
$$

where the second inequality follows from Assumption 4.2.

As a result, $\mathcal{M}$ is ε_0-add-DP, concluding our proof. □

By applying Corollary 3.2, we immediately have that its subsampled variant is ε-DP:

Theorem 4.4. *Under Assumptions 4.1 and 4.2,* REPEATED-EM$^{\mathcal{S}_p}_{\ln(2), \mathcal{A}}$ *is ε-DP for $p = 1 - e^{-\varepsilon}$.*

4.1. Applications

4.1.1. Monotone Submodular Maximization under Cardinality Constraint

The algorithm in Gupta et al. (2010) for monotone submodular maximization under cardinality constraint is based on the classic greedy algorithm that runs in k rounds, each round finding an element that maximizes the marginal gain. More precisely, the algorithm—which we call DPSUBMODGREEDY$_{\varepsilon_0, F_D}$—is exactly REPEATED-EM$_{\varepsilon_0, \mathcal{A}}$ where $L = k$ and the candidate sets and scoring functions are as follows:

- $\mathcal{C}_i$ is the set of remaining elements $[m] \setminus \{c_1^*, \dots, c_{i-1}^*\}$
- $q_i(c; D)$ is the marginal gain $F_D(\{c_1^*, \dots, c_{i-1}^*, c\}) - F_D(\{c_1^*, \dots, c_{i-1}^*\})$

They proved the following utility guarantee:

Theorem 4.5 (Gupta et al. 2010)**.** *For any $\varepsilon_0, \beta > 0$, DPSUBMODGREEDY$_{\varepsilon_0}$ achieves $(1-1/e)$-approximation and $O\left(\frac{k\log(m/\beta)}{\varepsilon_0}\right)$-error with probability $1-\beta$.*

Proof of Theorem 1.2. We simply run the subsampled version of Algorithm 1. More precisely, we use the algorithm DPSUBMODGREEDY$^{\mathcal{S}_p}_{\ln(2),F_D}$ where $p = 1 - e^{-\varepsilon}$. The privacy analysis follows from the straightforward observation that $\mathcal{C}_i, q_i$ in DPSUBMODGREEDY satisfies Assumptions 4.1 and 4.2 and Theorem 4.4.

For the utility, let $D_{\mathrm{s}} \sim \mathcal{S}_p(D)$ denote the subsampled dataset that is fed as an input to DPSUBMODGREEDY$_{\ln(2)}$ and let $T^* := \{c_1^*, \dots, c_k^*\}$ denote the output set. From Theorem 4.5, w.p. $1-\beta/2$, we have

$$F_{D_{\mathrm{s}}}(T^*) \geq \left(1-\frac{1}{e}\right) \cdot \max_{T\subseteq\mathcal{S},|T|=k} F_{D_{\mathrm{s}}}(T) - O\left(k\log\left(\frac{m}{\beta}\right)\right). \tag{1}$$

Furthermore, applying the Chernoff bound (Theorem 2.1) with $Z_x := f_x(T)\cdot\mathbf{1}[x\in D_{\mathrm{s}}], \mu = p\cdot F_D(T), \alpha = 0.01\eta, \zeta = \frac{2000k\log(m/\beta)}{\eta}$ and a union bound over all sets $T \in \binom{\mathcal{U}}{\leq k}$, we can conclude that the following hold simultaneously for all $T \in \binom{\mathcal{U}}{\leq k}$ w.p. at least $1-\beta/2$:

$$F_{D_{\mathrm{s}}}(T) \geq (1-\alpha)p\cdot F_D(T) - \zeta, \tag{2}$$
$$F_{D_{\mathrm{s}}}(T) \leq (1+\alpha)p\cdot F_D(T) + \zeta. \tag{3}$$

When (1), (2), and (3) all hold, we have

$$\begin{aligned}
&F_D(T^*)\\
&\overset{(3)}{\geq} \frac{1}{(1+\alpha)p}F_{D_{\mathrm{s}}}(T^*) - \zeta/p\\
&\overset{(1)}{\geq} \frac{1}{(1+\alpha)p}\cdot\left(1-\frac{1}{e}\right)\cdot\max_{T\subseteq\mathcal{S},|T|=k}F_{D_{\mathrm{s}}}(T)\\
&\quad - O\left(\frac{k\log(m/\beta)}{p}\right) - \zeta/p\\
&\overset{(2)}{\geq} \frac{1-\alpha}{1+\alpha}\cdot\left(1-\frac{1}{e}\right)\cdot\max_{T\subseteq\mathcal{S},|T|=k}F_D(T)\\
&\quad - O\left(\frac{k\log(m/\beta)}{p}\right) - 2\zeta/p\\
&\geq \left(1-\frac{1}{e}-\eta\right)\cdot\max_{T\subseteq\mathcal{S},|T|=k}F_D(T) - O\left(\frac{k\log(m/\beta)}{\eta\varepsilon}\right),
\end{aligned}$$

which concludes our proof. □

4.1.2. Monotone Submodular Maximization under Matroid Constraint

For maximization over a matroid, the greedy algorithm is *not* known to achieve $1-1/e$ approximation ratio. Instead, one has to resort to the so-called *continuous greedy algorithm* of Călinescu et al. (2011) (which is in turn based on an earlier algorithmic idea by Vondrák (2008)). Chaturvedi et al. (2021) followed this route and privatized the continuous greedy algorithm, albeit only achieving approximate-DP. Similarly to the above, we show that this algorithm in fact satisfies one-sided DP and, using the subsampled version of it, we prove Theorem 1.3. Due to space constraints, we defer the full proof to Appendix D.

4.1.3. Metric k-Means and k-Median

For metric k-means/median, we first use the repeated exponential mechanism to select $O(k\log n)$ points that form a *bicriteria* $O(1)$-approximate solution (where "bicriteria" refers to the fact that the set has size larger than k). To turn a bicriteria solution to an actual solution, we use a standard technique in private clustering (Jones et al., 2021; Ghazi et al., 2020): we snap each input point to the closest point in the bicriteria solution and add noise to the counts to create "private synthetic data". We can then run any non-private approximation algorithm on this synthetic data to get our ultimate solution. As described, this algorithm only satisfies one-sided DP. To achieve two-sided DP, we again use the subsampled version of it, similar to Theorem 1.2. The full proof is deferred to Appendix G.

5. Algorithm II: Repeated Above Threshold

Again, the interaction proceeds in L rounds. In round i, the algorithm is given a function $h_i : \mathcal{X}^* \to \mathbb{R}$ together with a threshold τ_i (which can depend on the outputs from the previous rounds). The goal is to decide whether $h_i(D) \geq \tau_i$. The algorithm we consider is one that repeatedly applies a variant of the AboveThreshold algorithm (Dwork et al., 2009)[13], where we only add noise to the function (but not to the threshold as in (Dwork et al., 2009)) and the noise is drawn from the exponential distribution[14] (rather than the Laplace distribution); see Algorithm 2. We remark that our algorithm is also different from both previous works of Kaplan et al. (2021) (who need add another noise term to make their analysis work) and of Cohen & Lyu (2023) (who simply use the Laplace mechanism in each round).

We analyze the DP guarantee of Algorithm 2 under the following assumptions (similar to the ones in Section 4).

Assumption 5.1 (Monotonicity)**.** For every i, adding an element to D does not decrease $h_i(D)$.

[13] See also Algorithm 1 in Dwork & Roth (2014).

[14] Recall that $\mathrm{Exp}(\lambda)$ has CDF $1-e^{-\lambda x}$ for $x \in [0,\infty)$.

Assumption 5.2 (Bounded Realized Sensitivity)**.** For every add-remove neighbors D, D' and all possible sequences $(h_1, \dots, h_L)$ of functions and outputs $(o_1, \dots, o_L)$, $\sum_{\substack{i\in[L]\\ o_i=\top}} |h_i(D) - h_i(D')| \le \Delta$.

Algorithm 2 REPEATED-AT$_{\varepsilon_0,\mathcal{A}}$

Parameters: $\varepsilon_0 > 0$, an algorithm $\mathcal{A}$ for selecting candidate and scoring functions
Input: Dataset $D \in \mathcal{X}^*$
  for $i = 1, \dots, L$ **do**
    $(h_i, \tau_i) \leftarrow \mathcal{A}(c_1^*, \dots, c_{i-1}^*)$
    $\theta_i \sim \mathrm{Exp}(\varepsilon_0/\Delta)$
    if $h_i(D) + \theta_i > \tau_i$ **then**
      $c_i^* = \top$
    else
      $c_i^* = \perp$
  return $(c_1^*, \dots, c_L^*)$

Theorem 5.3. *Under Assumptions 5.1 and 5.2,* REPEATED-AT$_{\varepsilon_0,\mathcal{A}}$ *(Algorithm 2) is ε_0-add-DP.*

Proof. Consider any D, D' such that $D = D' \cup \{x\}$ for some x, and any output $o = (o_1, \dots, o_L)$. We will write $\mathcal{M}$ as a shorthand for REPEATED-AT$_{\varepsilon_0,\mathcal{A}}$. Let $\mathcal{I}_\top := \{i \in L \mid o_i = \top\}$ and $\mathcal{I}_\perp := \{i \in L \mid o_i = \perp\}$. We have

$$\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]} = \prod_{i\in\mathcal{I}_\top} \frac{\Pr[h_i(D) + \theta_i > \tau_i]}{\Pr[h_i(D') + \theta_i > \tau_i]} \cdot \prod_{i\in\mathcal{I}_\perp} \frac{\Pr[h_i(D) + \theta_i \le \tau_i]}{\Pr[h_i(D') + \theta_i \le \tau_i]}.$$

Since $h_i(D') \le h_i(D)$ (Assumption 5.1), we have $\frac{\Pr[h_i(D)+\theta_i\le\tau_i]}{\Pr[h_i(D')+\theta_i\le\tau_i]} \le 1$. Meanwhile, from the definition of $\mathrm{Exp}\left(\frac{\varepsilon_0}{\Delta}\right)$, we have $\frac{\Pr[h_i(D)+\theta_i\le\tau_i]}{\Pr[h_i(D')+\theta_i\le\tau_i]} \le \exp\left(\frac{\varepsilon_0}{\Delta} \cdot (h_i(D) - h_i(D'))\right)$. Thus, we have

$$\frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]} \le \exp\left(\frac{\varepsilon_0}{\Delta} \cdot \sum_{i\in\mathcal{I}_\top} (h_i(D) - h_i(D'))\right) \le \exp(\varepsilon_0),$$

where the second inequality is from Assumption 5.2. □

By applying Corollary 3.2, we immediately have that its subsampled variant is ε-DP:

Theorem 5.4. *Under Assumptions 5.1 and 5.2,* REPEATED-AT$^{\mathcal{S}_p}_{\ln(2),\mathcal{A}}$ *is ε-DP for $p = 1 - e^{-\varepsilon}$.*

5.1. Applications

5.1.1. SHIFTING HEAVY HITTERS

To formalize our result, we need to first formalize the assumption that "each user contributes to at most k heavy hitters". To do this, let us first define $\tau^*(k) := \frac{4000k\log(T|\mathcal{Y}|/\beta)}{\varepsilon}$ to be (half of) our target error. The assumption we work with is the following, which is the same as that of Kaplan et al. (2021) (but with different $\tau^*(k)$).

Assumption 5.5. For every user $i \in [n]$, we have $|\{t \in [T] \mid w_t(x_i) > \tau^*(k)\}| \le k$.

Our theorem can now be formalized as follows.

Theorem 5.6. *For any $0 < \varepsilon \le 1$, under Assumption 5.5 for $\tau^*(k) = \frac{1000k\log(T|\mathcal{Y}|/\beta)}{\varepsilon}$, there is a shifting heavy hitters algorithm with error $2\tau^*(k)$ w.p. $1 - \beta$.*

Note that Assumption 5.5 is required only for utility; privacy guarantee holds for all input datasets, as is standard in DP.

Our THRESHMONITOR$_{\varepsilon_0,\tau,k}$ algorithm (which is a simplification of the algorithm from Kaplan et al. (2021)) is presented in Algorithm 3. Here, we keep the counter C_i for the number of times the user i has contributed to the heavy hitters. When this hits k, we simply drop this user and never include this user in the counts in the subsequent rounds.

Algorithm 3 THRESHMONITOR$_{\varepsilon_0,\tau,k}$

Parameters: $\varepsilon_0 > 0$, τ the desired heavy hitter threshold, $k \in \mathbb{N}$ limit on the number of contributions
Input: Input data stream $D \in (\mathcal{Y}^T)^n$
  $\mathcal{I} \leftarrow [n]$
  for $i = 1, \dots, n$ **do**
    $C_i \leftarrow 0$
  for $t = 1, \dots, T$ **do**
    for $y \in \mathcal{Y}$ **do**
      $w_t^{\mathcal{I}}(y; D) \leftarrow |\{i \in \mathcal{I} \mid x_{i,t} = y\}|$
      $\theta_{t,y} \sim \mathrm{Exp}(\varepsilon_0/k)$
      if $w_t^{\mathcal{I}}(y; D) + \theta_{t,y} > \tau_i$ **then**
        Report y for time step t
        for $i \in \mathcal{I}$ such that $x_{i,t} = y$ **do**
          $C_i \leftarrow C_i + 1$
          if $C_i = k$ **then**
            $\mathcal{I} \leftarrow \mathcal{I} \setminus \{i\}$

We are now ready to prove Theorem 5.6.

Proof of Theorem 5.6. We use the subsampled version of THRESHMONITOR, i.e., THRESHMONITOR$^{\mathcal{S}_p}_{\ln(2),\tau,k}$ for $p = 1 - e^{-\varepsilon}$ and $\tau = 1.5p \cdot \tau^*(k)$. It is not hard to see that THRESHMONITOR is an instantiation of Algorithm 2 with $h^{\mathcal{I}}_{t,x}(D) := w^{\mathcal{I}}_t(x)$ that satisfies Assumption 5.1 and Assumption 5.2 with $\Delta = k$. Thus, Theorem 5.4 implies that THRESHMONITOR$^{\mathcal{S}_p}_{\ln(2),\tau,k}$ is ε-DP as desired.

To see the utility guarantee, let D_{s} denote the subsampled dataset used as the input to THRESHMONITOR$_{\ln(2),\tau,k}$. Note also that $\tau^*(k)\cdot p \ge 0.5\tau^*(k)\cdot\varepsilon = 2000k\ln(T|\mathcal{Y}|/\beta)$. We first apply the Chernoff bound (Theorem 2.1) with

$Z_i := \mathbf{1}[i \in D_s] \cdot \mathbf{1}[x_{i,t} = y], \mu = p \cdot w_t(y; D), \alpha = 0.1, \zeta = 100 \cdot \ln(2T|\mathcal{Y}|/\beta) \le 0.1p \cdot \tau^*(k)$ together with a union bound over all $t \in [T], y \in \mathcal{Y}$, we can conclude that the following hold simultaneously for all $t \in [T], y \in \mathcal{Y}$ with probability at least $1 - \beta/2$:

$$w_t(y; D_s) \ge 0.9p \cdot w_t(y; D) - 0.1p \cdot \tau^*(k), \quad (4)$$
$$w_t(y; D_s) \le 1.1p \cdot w_t(y; D) + 0.1p \cdot \tau^*(k). \quad (5)$$

Furthermore, by tail bounds for exponential noise, the following holds for all $t \in [T], x \in \mathcal{X}$ with probability $1-\beta/2$:

$$\theta_{t,y} \le k \log(2T|\mathcal{Y}|/\beta) \le 0.01\tau^*(k). \quad (6)$$

We will continue the remainder of the analysis assuming that (4), (5), and (6) hold for all $t \in [T], y \in \mathcal{Y}$; by a union bound, this occurs with probability at least $1 - \beta$.

By (5) and (6), if y is reported at time t, then we must have

$$w_t(y; D) \ge \frac{\tau - 0.1p \cdot \tau^*(k)}{1.1p} > \tau^*(k).$$

This satisfies the first part of the accuracy requirement. Furthermore, this and Assumption 5.5 imply that each data record i in D_s contributes to at most k reported heavy hitters. Thus, $\mathcal{I}$ remains the entire dataset for the entire run. As a result, for each $(2\tau^*(k))$-heavy hitter y at time step t,

$$w_t^{\mathcal{I}}(y; D) \overset{(4)}{\ge} 0.9p \cdot 2\tau^*(k) - 0.1p \cdot \tau^*(k) > \tau.$$

Since $\theta_{t,y} \ge 0$, this implies that y is reported at time t. Thus, the algorithm satisfies the claimed accuracy bounds. □

5.1.2. Set Cover

We use the so-called *greedy scaling* algorithm from Kumar et al. (2015). Their algorithm works in $O(\log n)$ rounds. In each round, we iteratively keep all sets that cover at least a certain number of elements; this threshold is geometrically decreased across different rounds until every element is covered. We adapt this algorithm to the DP setting using our Repeated-AT algorithm (Algorithm 2) to find the sets to be picked in each round. Note that we also resample the dataset D_s in each round to avoid having to do a union bound over too large a number of events. This requires us to carefully split the privacy budget in each round (which we assigns in geometrically increasing manners). The complete description is presented in Algorithm 4. Due to space constraints, we defer the full proof to Appendix E.

Algorithm 4 DPGreedyScaling$_\varepsilon$

Parameters: $\varepsilon > 0$
Input: Input universe $D \in \mathcal{X}^*$, subsets $S_1, \dots, S_m \subseteq D$
  $\mathcal{I} \leftarrow [m]$
  $n_{\min} \leftarrow 100 \log m$
  $\tilde{n} \leftarrow \max\{n + \mathrm{Lap}(0.5/\varepsilon), n_{\min}\}$
  $j \leftarrow 1$
  $R \leftarrow \lfloor \log(\tilde{n}/n_{\min}) \rfloor$
  for $r = 1, \dots, R$ **do**
    $\tau_r \leftarrow 1000 \cdot \frac{\tilde{n}}{2^r}$
    $\varepsilon_r \leftarrow \frac{\varepsilon}{4 \cdot 2^{R-r}}$
    $p_r \leftarrow 1 - e^{-\varepsilon_r}$
    $D_{s,r} \sim \mathcal{S}^{p_r}(D)$
    for $i \in \mathcal{I}$ **do**
      $\theta_{r,i} \sim \mathrm{Exp}(\ln(2))$
      if $\left|(S_i \cap D_{s,r}) \setminus \left(\bigcup_{j'<j} S_{\pi(j')}\right)\right| + \theta_{r,i} > p_r \cdot \tau_r$ **then**
        $\pi(j) \leftarrow i$
        $\mathcal{I} \leftarrow \mathcal{I} \setminus \{i\}$
        $j \leftarrow j + 1$
  for $i \in \mathcal{I}$ **do**
    $\pi(j) \leftarrow i$
    $j \leftarrow j + 1$
  return $(\pi(1), \dots, \pi(m))$

6. Conclusion and Open Questions

In this work, we make a simple observation that subsampling a one-sided DP mechanism makes it two-sided DP. Applying this observation to the repeated exponential mechanism and the repeated above threshold mechanism, we obtain novel pure-DP algorithms for several combinatorial optimization problems and for the shifting heavy hitters problem. It remains interesting to explore the applications of this framework further. One clear barrier of the current approach is that it requires monotonicity (Assumptions 4.1 and 5.1). This prevents us from applying this to the non-monotone submodular maximization problems; meanwhile (Chaturvedi et al., 2021) show that a Gupta et al.-like analysis still works for approximate-DP. In particular, they achieve $\left(\frac{1}{e} - \eta\right)$-approximation and $O\left(\frac{k \log\left(\frac{m}{\eta\beta}\right) \log(1/\delta)}{\eta\varepsilon}\right)$-error for non-monotone submodular maximization under matroid constraint. A concrete question here is whether we can achieve a similar guarantee under pure-DP.

Impact Statement

This work advances the area of optimization and data analytics with privacy. There might be potential societal consequences of our work, none which we feel is significant enough to be highlighted.

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning

with differential privacy. In *CCS*, pp. 308–318, 2016.

Anand, A. and Lee, E. Separating k-median from the supplier version. In *IPCO*, 2024.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3): 544–562, 2004.

Balle, B., Barthe, G., and Gaboardi, M. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *NIPS*, pp. 6280–6290, 2018.

Banihashem, K., Biabani, L., Goudarzi, S., Hajiaghayi, M., Jabbarzade, P., and Monemizadeh, M. Dynamic constrained submodular optimization with polylogarithmic update time. In *ICML*, pp. 1660–1691, 2023.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: the SuLQ framework. In *PODS*, pp. 128–138, 2005.

Brown, D. G. How I wasted too long finding a concentration inequality for sums of geometric variables, 2011. https://uwspace.uwaterloo.ca/bitstream/handle/10012/17210/negbin.pdf.

Călinescu, G., Chekuri, C., Pál, M., and Vondrák, J. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

Chang, A., Ghazi, B., Kumar, R., and Manurangsi, P. Locally private k-means in one round. In *ICML*, pp. 1441–1451, 2021.

Charikar, M., Guha, S., Tardos, É., and Shmoys, D. B. A constant-factor approximation algorithm for the *k*-median problem (extended abstract). In *STOC*, pp. 1–10, 1999.

Chaturvedi, A., Nguyen, H. L., and Zakynthinou, L. Differentially private decomposable submodular maximization. *CoRR*, abs/2005.14717, 2020.

Chaturvedi, A., Nguyen, H. L., and Zakynthinou, L. Differentially private decomposable submodular maximization. In *AAAI*, pp. 6984–6992, 2021.

Chaturvedi, A., Nguyen, H. L., and Nguyen, T. D. Streaming submodular maximization with differential privacy. In *ICML*, volume 202, pp. 4116–4143, 2023.

Chekuri, C., Vondrák, J., and Zenklusen, R. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*, pp. 575–584, 2010.

Cohen, E. and Lyu, X. The target-charging technique for privacy accounting across interactive computations. In *NeurIPS*, 2023.

Dinur, I. and Steurer, D. Analytical approach to parallel repetition. In *STOC*, pp. 624–633, 2014.

Duetting, P., Fusco, F., Lattanzi, S., Norouzi-Fard, A., and Zadimoghaddam, M. Fully dynamic submodular maximization over matroids. In *ICML*, pp. 8821–8835, 2023.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

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

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

Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. P. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pp. 381–390, 2009.

Dwork, C., Rothblum, G. N., and Vadhan, S. P. Boosting and differential privacy. In *FOCS*, pp. 51–60, 2010.

Feige, U. A threshold of ln *n* for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

Feldman, D., Fiat, A., Kaplan, H., and Nissim, K. Private coresets. In *STOC*, pp. 361–370, 2009.

Ghazi, B., Kumar, R., and Manurangsi, P. Differentially private clustering: Tight approximation ratios. In *NeurIPS*, 2020.

Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. Differentially private combinatorial optimization. In *SODA*, pp. 1106–1125, 2010.

Jones, M., Nguyen, H. L., and Nguyen, T. D. Differentially private clustering via maximum coverage. In *AAAI*, pp. 11555–11563, 2021.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.

Kaplan, H., Mansour, Y., and Stemmer, U. The sparse vector technique, revisited. In *COLT*, pp. 2747–2776, 2021.

Karp, R. M. Reducibility among combinatorial problems. In *Symposium on the Complexity of Computer Computations*, pp. 85–103, 1972.

Kotsogiannis, I., Doudalis, S., Haney, S., Machanavajjhala, A., and Mehrotra, S. One-sided differential privacy. In *ICDE*, pp. 493–504, 2020.

Kumar, R., Moseley, B., Vassilvitskii, S., and Vattani, A. Fast greedy algorithms in mapreduce and streaming. *ACM Trans. Parallel Comput.*, 2(3):14:1–14:22, 2015.

Li, G. Z., Nguyen, D., and Vullikanti, A. Differentially private partial set cover with applications to facility location. In *IJCAI*, pp. 4803–4811, 2023.

Li, N., Qardaji, W. H., and Su, D. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *ASIACCS*, pp. 32–33, 2012.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *FOCS*, pp. 94–103, 2007.

Mirzasoleiman, B., Badanidiyuru, A., and Karbasi, A. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pp. 1358–1367, 2016.

Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. Differentially private submodular maximization: Data summarization in disguise. In *ICML*, pp. 2478–2487, 2017.

Moshkovitz, D. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory Comput.*, 11:221–235, 2015.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

Nguyen, H. L., Chaturvedi, A., and Xu, E. Z. Differentially private k-means via exponential mechanism and max cover. In *AAAI*, pp. 9101–9108, 2021.

Nissim, K. and Stemmer, U. Clustering algorithms for the centralized and local models. In *ALT*, pp. 619–653, 2018.

Papadimitriou, C. H., Schapira, M., and Singer, Y. On the hardness of being truthful. In *FOCS*, pp. 250–259, 2008.

Rafiey, A. and Yoshida, Y. Fast and private submodular and k-submodular functions maximization with matroid constraints. In *ICML*, pp. 7887–7897, 2020.

Sadeghi, O. and Fazel, M. Differentially private monotone submodular maximization under matroid and knapsack constraints. In *AISTATS*, pp. 2908–2916, 2021.

Stemmer, U. and Kaplan, H. Differentially private k-means with constant multiplicative error. In *NeurIPS*, pp. 5436–5446, 2018.

Takagi, S., Kato, F., Cao, Y., and Yoshikawa, M. Asymmetric differential privacy. In *BigData*, pp. 1576–1581, 2022.

Vondrák, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pp. 67–74, 2008.

Wang, Y., Balle, B., and Kasiviswanathan, S. P. Subsampled Rényi differential privacy and analytical moments accountant. *J. Priv. Confidentiality*, 10(2), 2020.

A. Additional Related Work

Below we provide additional discussion on related work.

AboveThreshold with Individualized Privacy Loss. As stated earlier, Kaplan et al. (2021) provide a modification of the sparse vector technique (SVT) (Dwork et al., 2009) that can be used for individualized privacy loss. Their algorithm is similar to Algorithm 2 presented in our work except that (i) they use Laplace noise (since they want two-sided DP), (ii) they add noise to the threshold (similar to standard SVT) and (iii) they also add another "noise of noise" term to the query value. The last one is to help with the analysis but results in an increase of $O(\log(\log(1/\delta)/\varepsilon))$ in their error bound. Very recently, Cohen & Lyu (2023) gave a different algorithm for the task that gets rid of this bound. Their algorithm is essentially the same as our Algorithm 2 with the exception that they use Laplace noise instead of Exponential noise. The framework of Cohen & Lyu (2023) is extremely generic and individualized privacy loss accounting is only one of the applications of their framework. However, their results only apply for approximate-DP. It remains interesting to see if our framework can be used for any of the applications in their paper.

Private Submodular Maximization. Although our paper assumes that the function is decomposable, DP submodular maximization has also been studied under other assumptions. For example, Mitrovic et al. (2017) also study the setting where the function F_D is only assumed to have low-sensitivity, i.e., $|F_D(S) - F_{D'}(S)| \leq \Delta$ for all neighboring datasets D, D'. This is a more relaxed assumption than decomposability. Our techniques do not seem to apply here, both because (i) it is not clear how to related the subsampled function value to the function value on the entire dataset and (ii) the individualized privacy accounting is not known to be applicable here even for approximate-DP. It remains an interesting question whether one can get improved bound under this weaker assumption. Note that, for 1-sensitive monotone submodular functions, the best ε-DP algorithm under cardinality constraint is still from (Mitrovic et al., 2017) and achieves approximation ratio $\left(1 - \frac{1}{e}\right)$ and error $O\left(\frac{k^2 \log n}{\varepsilon}\right)$. For matroid constraint, the best algorithm is that of Rafiey & Yoshida (2020), which achieves the same approximation ratio but with error $O\left(\frac{k^7 \log n}{\varepsilon^3}\right)$.

Another related work on the topic is by Sadeghi & Fazel (2021), who gave an improved approximation algorithm if the *total curvature* of the function is smaller than one and also proposed an algorithm for the online setting. A recent work of Chaturvedi et al. (2023) also studied private submodular maximization in the streaming setting. Finally, we note that the aforementioned work also considered other settings including non-monotone functions. Our techniques do not apply in this case; we provide more discussion regarding this in Section 6.

Partial Set Cover. Li et al. (2023) initiated a study of private partial set cover, where it suffices to cover a fraction (not all) of the elements, and apply it as a subroutine to the DP k-supplier with outlier problem. Since they also use Gupta et al.'s algorithm, our technique can be applied to their setting to achieve improved bounds as well.

Private Clustering. We remark that in (Gupta et al., 2010; Jones et al., 2021), the metric k-median problem is defined in so that each point in the metric can appear in the dataset only once. This implies $n \leq m$ and their bound thus only depends on m. We choose to state the more general formulation and bounds here, which is why we also have the dependency on n.

While we focus our attention on discrete finite metric, DP clusterings have also been studied in other infinite metrics, such as the ℓ_p-metric (Blum et al., 2005; Feldman et al., 2009; Nissim & Stemmer, 2018; Stemmer & Kaplan, 2018; Ghazi et al., 2020; Jones et al., 2021; Nguyen et al., 2021; Chang et al., 2021). Some of these works, e.g., (Jones et al., 2021; Nguyen et al., 2021), uses the repeated exponential mechanism as a subroutine. Therefore, our techniques can be applied to reduce the errors in these bounds.

Amplification by Subsampling. There is, by now, a large body of literature on DP amplification by subsampling (e.g., (Li et al., 2012; Balle et al., 2018; Wang et al., 2020; Abadi et al., 2016)). However, we are not aware of any result that allows us to goes from an approximate-DP guarantee to a pure-DP guarantee, which is what Theorem 4.4 achieves (albeit with a one-sided DP requirement). It remains an intriguing question to further explore the power of amplification by subsampling.

B. Lower Bound for Shifting Heavy Hitters

We prove the following lower bound for the shifting heavy hitters problem, which shows that the bound on $\tau^*(k)$ we achieved in Theorem 5.6 is essentially tight.

Theorem B.1. *For any sufficiently large $k, T \in \mathbb{N}$ such that $T \geq k, \varepsilon \in (0,1)$ and $|\mathcal{Y}| \geq 5$, there is no ε-DP algorithm that can achieve $2\tau^*(k)$ error w.p. 0.5 under Assumption 5.5 for $\tau^*(k) = 0.001k \log\left(\frac{|\mathcal{Y}|T}{k}\right)$.*

Proof. Assume w.l.o.g. that $\mathcal{Y} = [r+3]$ for $r \in \mathbb{N}$. Suppose for the sake of contradiction that there exists an ε-DP algorithm $\mathcal{M}$ that achieve $2\tau^*(k)$ error w.p. 0.5 under Assumption 5.5 for $\tau^*(k) = 0.001k \log\left(\frac{|\mathcal{Y}|T}{k}\right)$. Let $\mathcal{X}'$ denote the set of all vectors in $\{0, \dots, r\}^T$ whose Hamming weight is at most k. For each $\mathbf{x} = (x_1, \dots, x_T) \in \mathcal{X}'$, let $D_{\mathbf{x}}$ denote dataset where there are $n = 2\tau^*(k) + 1$ users and, for each $i \in [n]$, user i's input $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,T})$ is defined as follows:

$$x_{i,t} = \begin{cases} x_t & \text{if } x_t \neq 0 \\ r + 1 + \lceil 3(i-1)/n \rceil & \text{otherwise.} \end{cases}$$

for all $t \in [T]$. Notice here that $r+1, r+2, r+3$ are never $\tau^*(k)$-heavy hitters. Thus, since the Hamming norm of $\mathbf{x}$ is at most k, Assumption 5.5 is satisfied.

Now, let $O_{\mathbf{x}}$ be the set of outcomes where x_t is reported at time t for all $t \in [T]$ such that $x_t \neq 0$ and no element in $[r] \setminus \{x_t\}$ is reported. From the utility guarantee of $\mathcal{M}$, we have $\Pr[\mathcal{M}(D_{\mathbf{x}}) \in O_{\mathbf{x}}] \geq 1/2$. From ε-DP, we have $\Pr[\mathcal{M}(\emptyset) \in O_{\mathbf{x}}] \geq e^{-\varepsilon \cdot n}/2$. Meanwhile, since $O_{\mathbf{x}}$ are disjoint for all $\mathbf{x} \in \mathcal{X}'$, we have

$$\begin{aligned} 1 &\geq \sum_{\mathbf{x} \in \mathcal{X}'} \Pr[\mathcal{M}(\emptyset) \in O_{\mathbf{x}}] \\ &\geq |\mathcal{X}'| \cdot e^{-\varepsilon \cdot n}/2 \\ &\geq r^{\lceil 0.1k \rceil} \binom{T}{\lceil 0.1k \rceil} \frac{e^{-\varepsilon \cdot n}}{2} \\ &\geq \left(\frac{rT}{k}\right)^{0.01k} \frac{e^{-\varepsilon \cdot n}}{2} \\ &> 1, \end{aligned}$$

where the last inequality follows from our choice of $n = 2\tau^*(k) + 1$. This completes our proof. □

C. Counterexample for Pure-DP without Subsampling

Recall that, if we apply the (basic) composition theorem to the L calls of exponential mechanisms in REPEATED-EM$_{\varepsilon,\mathcal{A}}$ (Algorithm 1), we get that the algorithm is $2L\varepsilon$-DP. (Note that this is *two-sided* DP.) The lemma below shows that we cannot hope to do much better than this bound. This lemma also gives a justification to our subsampling framework since we can achieve ε-DP with subsampling (Theorem 4.4).

Lemma C.1. *There exists $\mathcal{A}$ that satisfies Assumption 4.1 and Assumption 4.2 but that* REPEATED-EM$_{\varepsilon,\mathcal{A}}$ *is not ε'-DP for any $\varepsilon' < L\varepsilon$.*

Proof. In fact, we will use the instantiation for 1-decomposable submodular function as in Section 4.1.1. Let $m = \left\lceil 1 + \frac{(e^\varepsilon - 1)L}{e^\varepsilon - e^{\varepsilon'/L}} \right\rceil$, $D' = \emptyset$, and $D = \{x\}$, where f_x is defined as $f_x(S) = \min\{1, |S \cap [m-L]|\}$ for $S \subseteq [m]$. Let $\mathcal{M}$ denote the mechanism DPSUBMODGREEDY$_{\varepsilon,F_D}$ from Section 4.1.1 (which is an instantiation of REPEATED-EM$_{\varepsilon,\mathcal{A}}$). Now, consider the output $o = (m-L+1, \dots, m)$. Then, we have

$$\begin{aligned} \frac{\Pr[\mathcal{M}(D') = o]}{\Pr[\mathcal{M}(D) = o]} &= \prod_{i \in [L]} \frac{\Pr[\text{EXPMECH}_{\frac{\varepsilon_0}{\Delta}}(\mathcal{C}_i, q_i; D') = o_i]}{\Pr[\text{EXPMECH}_{\frac{\varepsilon_0}{\Delta}}(\mathcal{C}_i, q_i; D) = o_i]} \\ &= \prod_{i \in [L]} \frac{\frac{1}{m-i+1}}{\frac{1}{e^\varepsilon(m-L)+L-i+1}} \end{aligned}$$

$$< \prod_{i\in[L]} e^{\varepsilon'/L}$$
$$= e^{\varepsilon'},$$

where the inequality is due to our choice of parameter m. Thus, the algorithm is not ε'-DP. □

D. Monotone Submodular Maximization over Matroid Constraint

As mentioned earlier, we will use the private version of the continuous greedy algorithm due to Chaturvedi et al. (2021). To describe the algorithm, we need several additional definitions.

Definition D.1 (Multilinear Extension)**.** For a given set function $F : 2^{[m]} \to \mathbb{R}$, its multilinear extension $F^{\text{mult}} : [0,1]^m \to \mathbb{R}$ defined by

$$F^{\text{mult}}(\mathbf{y}) = \sum_{S\subseteq[m]} F(S) \prod_{i\in S} y_i \prod_{i\in([m]\setminus S)} (1-y_i).$$

For a given matroid $\mathcal{M} = ([m], \mathcal{I})$, we write $\mathcal{P}(\mathcal{M})$ to denote the *matroid polytope*, which is the convex hull of all characteristics of all independent sets in $\mathcal{M}$. The continuous greedy algorithm finds a vector in $\mathcal{P}(\mathcal{M})$ with a large multilinear extension value. Since computing the multilinear extension directly is inefficient, it is only computed approximately. In (Chaturvedi et al., 2021), this was done by randomly sampling $z^1, \dots, z^s$ uniformly and independently from $[0,1]^m$, and then computing

$$G^{\mathbf{z}}(y) := \frac{1}{s} \sum_{j\in[s]} F(\{u \in [m] \mid z_u^j < y_u\}),$$

and using it as a proxy for the multilinear extension. The full algorithm is presented in Algorithm 5.

Algorithm 5 $\text{PrivContGreedy}_{\varepsilon_0,\eta,s}$ (Chaturvedi et al., 2021)

Parameters: $\varepsilon_0 > 0$, step size η, number of draws s
Input: Dataset $D \in \mathcal{X}^*$
  $T \leftarrow \lceil 1/\eta \rceil$
  $k \leftarrow \text{rank}(\mathcal{M})$
  $z^1, \dots, z^s \sim [0,1]^m$
  for $t = 1, \dots, T$ **do**
    $B^{t,0} \leftarrow \emptyset$
    for $i = 1, \dots, k$ **do**
      $\mathcal{C}^{t,i} \leftarrow \{u \in [m] \setminus B^{t,i-1} \mid B^{t,i-1} \cup \{u\} \in \mathcal{I}\}$
      if $\mathcal{C}^{t,i} = \emptyset$ **then**
        $y^{t,i} \leftarrow y^{t,i-1}$
      else
        $q^{t,i}(u; D) \leftarrow G_D^{\mathbf{z}}(y^{t,i-1} + \eta \cdot \mathbf{1}_u) - G_D^{\mathbf{z}}(y^{t,i-1})$ for all $u \in \mathcal{C}^{t,i}$
        $u^{t,i} \leftarrow \text{ExpMech}_{\varepsilon_0}(\mathcal{C}^{t,i}, q^{t,i}; D)$
        $y^{t,i} \leftarrow y^{t,i-1} + \eta \cdot \mathbf{1}_{u^{t,i}}$
        $B^{t,i} \leftarrow B^{t,i-1} \cup \{u^{t,i}\}$
    $y^{t+1,0} \leftarrow y^{t,k}$
  return $y^{T,k}$

Let $g_x^{\mathbf{z}} := \frac{1}{s} \sum_{j\in[s]} f_x(\{u \in [m] \mid z_u^j < y_u\})$. Notice that $G_D^{\mathbf{z}} = \sum_{x\in D} g_x^{\mathbf{z}}$. The utility guarantee of Algorithm 5 was shown in (Chaturvedi et al., 2021); we state this below.

Theorem D.2 (Chaturvedi et al. 2021)**.** *For any $\beta > 0$, let $s = 6k^2T^4 \cdot \log(m/\beta)$, then with probability $1-\beta$, the output $y = y^{T,k}$ satisfies*

$$F_D^{\text{mult}}(y) \geq \left(1 - \frac{1}{e} - \eta\right) \max_{S\in\mathcal{I}}\{F_D(S)\} - O\left(\frac{k}{\eta\varepsilon} \log \frac{m}{\eta\beta}\right).$$

It is simple to see that $q^{t,i}$ is monotone. Moreover, as observed in (Chaturvedi et al., 2021), the realized marginal sensitivity of the score $G_D^{\mathbf{z}}$ is bounded:

Observation D.3 (Chaturvedi et al. 2021). For any $z^1, \dots, z^s \in [0,1]^m$, any neighboring datasets D' and $D = D' \cup \{x\}$ and any selection of $(u^{t,i})_{t\in[T],i\in[k]}$, we have

$$\begin{aligned}
&\sum_{t\in[T]}\sum_{i\in[k]} \left|q^{t,i}(u^{t,i};D) - q^{t,i}(u^{t,i};D')\right| \\
&= \sum_{t\in[T]}\sum_{i\in[k]} \left[\left(G_D^{\mathbf{z}}(y^{t,i}) - G_D^{\mathbf{z}}(y^{t,i-1})\right) - \left(G_{D'}^{\mathbf{z}}(y^{t,i}) - G_{D'}^{\mathbf{z}}(y^{t,i-1})\right)\right] \\
&= \sum_{t\in[T]}\sum_{i\in[k]} \left(g_x^{\mathbf{z}}(y^{t,i}) - g_x^{\mathbf{z}}(y^{t,i-1})\right) \\
&= g_x^{\mathbf{z}}(y^{T,k}) - g_x^{\mathbf{z}}(y^{1,0}) \le 1.
\end{aligned}$$

Combining these observations together with Theorem 4.3 immediately yields the following:

Lemma D.4. *For any $0 < \varepsilon, \eta, \beta < 1$, there is an ε-add-DP algorithm that for matroid submodular maximization that with probability $1-\beta$ outputs y such that*

$$F_D^{\text{mult}}(y) \ge \left(1 - \frac{1}{e} - \eta\right) \max_{S\in\mathcal{I}}\{F_D(S)\} - O\left(\frac{k}{\eta\varepsilon}\log\frac{m}{\eta\beta}\right).$$

Rounding. To get from a fractional solution to an integral solution (i.e., a set), we recall the following rounding algorithm due to Chekuri et al. (2010). It should be noted that this rounding algorithm only depends on y (and the matroid $\mathcal{M}$) and does *not* depend on the function F.

Definition D.5 (Chekuri et al. 2010)**.** Let $\mathcal{M} = ([m], \mathcal{I})$ be any matroid. There exists a randomized rounding algorithm SwapRound that takes in $y \in \mathcal{P}(\mathcal{M})$ and output a set $S \in \mathcal{I}$ such that, for any submodular function F and any $y \in [0,1]^m$, we have

$$\mathbb{E}_{S\sim\text{SwapRound}(\mathbf{y})}[F(S)] \ge F^{\text{mult}}(y).$$

We use a slightly different rounding procedure compared to (Chaturvedi et al., 2021): while they apply SwapRound once, we apply it multiple times and use the exponential mechanism to pick the best of them. This allows us to get high probability bound (compared to expected bound) on the approximation ratio and error. This is summarized below.

Lemma D.6. *For any $0 < \varepsilon, \eta, \beta < 1$, there is an ε-add-DP algorithm that for matroid submodular maximization that with probability $1-\beta$ outputs $S^* \in \mathcal{I}$ such that*

$$F_D(S^*) \ge \left(1 - \frac{1}{e} - \eta\right) \max_{S\in\mathcal{I}}\{F_D(S)\} - O\left(\frac{k}{\eta\varepsilon}\log\frac{m}{\eta\beta}\right).$$

Proof. The algorithm works as follows:

- Run the algorithm from Lemma D.4 with parameters $\varepsilon/2, \eta/2, \beta/3$ to get $y \in \mathcal{P}(\mathcal{M})$.
- Run $\text{SwapRound}(y)$ $h = \lceil 10\log(3/\beta)/\eta\rceil$ times to arrive at sets $S_1^*, \dots, S_h^*$.
- Use $(\varepsilon/2)$-DP exponential mechanism based on the score $q(S_i^*; D) := F_D(S_i^*)$ to select S^* from $S_1^*, \dots, S_h^*$ to (approximately) maximize $F_D(S^*)$.

Since the second step is just a post-processing of the result from the first step, we can apply composition across the two add-DP mechanisms in the first and last steps to conclude that this is ε-add-DP.

As for the utility, recall from Lemma D.4 that with probability $1 - \beta/3$, we have

$$F_D^{\text{mult}}(y) \ge \left(1 - \frac{1}{e} - \frac{\eta}{2}\right) \max_{S\in\mathcal{I}}\{F_D(S)\} - O\left(\frac{k}{\eta\varepsilon}\log\frac{m}{\eta\beta}\right). \tag{7}$$

For fixed y, let $\theta := F_D^{\text{mult}}(y)$ and $\text{OPT} := \max_{S\in\mathcal{I}} F_D(S)$. Since $\mathbb{E}_{S\sim\text{SwapRound}(y)}[F_D(S)] = F_D^{\text{mult}}(y) = \theta$ and $F_D(S) \le \text{OPT}$ for all $S \in \mathcal{I}$, Markov inequality implies that

$$\begin{aligned}\Pr_{S\sim\text{SwapRound}(y)}[F_D(S) \ge \theta - \eta\cdot\text{OPT}/2] &= 1 - \Pr_{S\sim\text{SwapRound}(y)}[\text{OPT} - F_D(S) > \text{OPT} + \eta\cdot\text{OPT}/2 - \theta] \\ &\ge 1 - \frac{\text{OPT}-\theta}{\text{OPT}+\eta\cdot\text{OPT}/2-\theta} \\ &= \frac{\eta\cdot\text{OPT}/2}{\text{OPT}+\eta\cdot\text{OPT}/2-\theta} \\ &\ge \eta/4.\end{aligned}$$

Thus, by our choice of h, the following holds with probability at least $1-\beta/3$:

$$\max\{F_D(S_1^*),\dots,F_D(S_h^*)\} \ge F_D^{\text{mult}}(y) - \eta\cdot\text{OPT}/2. \tag{8}$$

Finally, by the utility of the exponential mechanism (Theorem 2.2), with probability at least $1-\beta/3$, the following holds:

$$F_D(S^*) \ge \max\{F_D(S_1^*),\dots,F_D(S_h^*)\} - O\left(\frac{\log(h/\beta)}{\varepsilon}\right). \tag{9}$$

When (7),(8), and (9) all hold (which happens with probability at least $1-\beta$ due to the union bound), we have

$$F_D(S^*) \ge \left(1-\frac{1}{e}-\eta\right)\max_{S\in\mathcal{I}} F_D(S) - O\left(\frac{k}{\eta\varepsilon}\log\frac{m}{\eta\beta}\right),$$

which concludes our proof. □

We are now ready to prove the main theorem here (Theorem 1.3) by appealing to concentration bounds (similar to the proof of Theorem 1.2).

Proof of Theorem 1.3. Let $\mathcal{A}_{\varepsilon,\eta,\beta}$ denote the algorithm from Lemma D.6. We simply run the subsampled version of this algorithm. More precisely, we use the algorithm $\mathcal{A}^{\mathcal{S}_p}_{\ln(2),\eta/2,\beta/2}$ where $p = 1-e^{-\varepsilon}$. The privacy guarantee immediately follows from Theorem 4.3.

To analyze the utility, let $D_{\text{s}} \sim \mathcal{S}_p(D)$ denote the subsampled dataset that is fed as an input to $\mathcal{A}_{\ln(2),\eta/2,\beta/2}$ and let S^* denote the output set. By the utility guarantee of $\mathcal{A}$ (Lemma D.6), with probability $1-\beta/2$, we have

$$F_{D_{\text{s}}}(S^*) \ge \left(1-\frac{1}{e}-\frac{\eta}{2}\right)\cdot\max_{S\subseteq I} F_{D_{\text{s}}}(S) - O\left(\frac{k}{\eta}\log\frac{m}{\eta\beta}\right). \tag{10}$$

Furthermore, applying the Chernoff bound (Theorem 2.1) with $Z_x := f_x(S)\cdot\mathbf{1}[x\in D_{\text{s}}], \mu = p\cdot F_D(S), \alpha = 0.1\eta, \zeta = \frac{100k\log(m/\beta)}{\eta}$ together with a union bound over all sets $S \in \binom{\mathcal{U}}{\le k}$, we can conclude that the following holds simultaneously for all $S \in \binom{\mathcal{U}}{\le k}$ with probability at least $1-\beta/2$:

$$F_{D_{\text{s}}}(S) \ge (1-\alpha)p\cdot F_D(S) - \zeta, \tag{11}$$
$$F_{D_{\text{s}}}(S) \le (1+\alpha)p\cdot F_D(S) + \zeta. \tag{12}$$

When (10), (11), and (12) all hold, we have

$$\begin{aligned}F_D(S^*) &\overset{(12)}{\ge} \frac{1}{(1+\alpha)p}F_{D_{\text{s}}}(S^*) - \zeta/p \\ &\overset{(10)}{\ge} \frac{1}{(1+\alpha)p}\cdot\left(1-\frac{1}{e}-\frac{\eta}{2}\right)\cdot\max_{S\in\mathcal{I}} F_{D_{\text{s}}}(S) - O\left(\frac{k\log\left(\frac{m}{\eta\beta}\right)}{\eta p}\right) - \zeta/p \\ &\overset{(11)}{\ge} \frac{1-\alpha}{1+\alpha}\cdot\left(1-\frac{1}{e}-\frac{\eta}{2}\right)\cdot\max_{S\in\mathcal{I}} F_D(S) - O\left(\frac{k\log\left(\frac{m}{\eta\beta}\right)}{\eta p}\right) - 2\zeta/p\end{aligned}$$

$$\geq \left(1 - \frac{1}{e} - \eta\right) \cdot \max_{S \in \mathcal{I}} F_D(S) - O\left(\frac{k \log\left(\frac{m}{\eta\beta}\right)}{\eta\varepsilon}\right),$$

where the last inequality follows from our choice of parameters. □

E. Set Cover: Proof of Theorem 1.4

Proof of Theorem 1.4. We use DPGREEDYSCALING$_{\varepsilon_0}$. To see the privacy guarantee of the algorithm, note that the computation of $\tilde{n}$ is 0.5ε-DP. Meanwhile, the rth round is (a post-processing of) an instantiation of REPEATED-AT$^{S_{p_r}}_{\ln(2),\mathcal{A}}$ with $h_{r,i}(D) = \left|(S_i \cap D) \setminus \left(\bigcup_{j'<j} S_{\pi(j')}\right)\right|$. It is simple to verify that this satisfies Assumptions 5.1 and 5.2. Thus, Theorem 5.4 implies that the rth round is ε_r-DP. Applying (basic) composition theorem, we get that the entire algorithm is ε'-DP for

$$\varepsilon' = 0.5\varepsilon + \sum_{r \in \mathbb{N}} \varepsilon_r \leq \varepsilon,$$

as desired.

Next, we analyze the approximation guarantee. If $n \leq \frac{1000 \log(m/\beta)}{\varepsilon}$, then any output will yield an approximation ratio at most $n \leq O\left(\frac{\log(m/\beta)}{\varepsilon}\right)$. Hence, we may assume w.l.o.g. that $n > \frac{1000 \log(m/\beta)}{\varepsilon}$. In this case, standard tail bounds for Laplace noise shows that with probability $1 - \beta/3$, we have

$$0.5\tilde{n} \leq n \leq 2\tilde{n}. \tag{13}$$

We will condition on this event happening for the remainder of the analysis.

We write $S_{\pi(<j)}$ as a shorthand for $\bigcup_{j'<j} S_{\pi(j')}$. For all $r = \{0, \ldots, R\}$, let $\beta_r = \beta/2^{R-r}$ and let j_r denote the value of j at the end of round r. Furthermore, let $q = 100 \cdot \mathrm{SetCov}(\mathcal{C}, \mathcal{S})$.

Let us fix $r \in [R]$. By tail bounds for exponential noise, the following holds for all $i \in \mathcal{I}$ with probability $1 - \beta_r/6$:

$$\theta_{r,i} \leq \log(6m/\beta_r). \tag{14}$$

Furthermore, applying the Chernoff bound (Theorem 2.1) with $Z_x := \mathbf{1}\left[x \in (D \setminus S_{T \cup \pi(<j_{r-1})})\right] \cdot \mathbf{1}[x \in D_{\mathrm{s},r}], \mu = p \cdot |D \setminus S_{T \cup \pi(<j_{r-1})}|, \alpha = 0.3, \zeta_r = 10 \cdot \ln(12m^q/\beta_r)$ together with a union bound over all sets $T \in \binom{[m]}{\leq q}$, we can conclude that the following hold simultaneously for all $T \in \binom{[m]}{\leq q}$ with probability at least $1 - \beta_r/6$:

$$|D_{\mathrm{s},r} \setminus S_{T \cup \pi(<j_{r-1})}| \geq 0.7p_r |D \setminus S_{T \cup \pi(<j_{r-1})}| - \zeta_r, \tag{15}$$

$$|D_{\mathrm{s},r} \setminus S_{T \cup \pi(<j_{r-1})}| \leq 1.3p_r |D \setminus S_{T \cup \pi(<j_{r-1})}| + \zeta_r. \tag{16}$$

Note that, by a union bound, the probability that (13) holds and (14), (15), and (16) hold for all $r \in [R]$ is at least

$$1 - \frac{\beta}{3} - \sum_{r \in [R]} \frac{2\beta_r}{6} \geq 1 - \beta.$$

For the remainder of the analysis, we will assume that (13) holds and (14), (15), and (16) hold for all $r \in [R]$. A crucial claim used to bound the approximation ratio is stated below.

Claim E.1. For all $r \in [R]$, we have

$$j_r - j_{r-1} \leq q, \tag{17}$$

and

$$|D \setminus S_{\pi(<j_r)}| \leq \frac{0.1q \cdot \tau_r}{p_r}. \tag{18}$$

Before we prove Claim E.1, let us show how to use this to finish the proof of the approximation guarantee. The number of total sets chosen is at most

$$\begin{aligned} j_R + |D \setminus S_{\pi(<j_R)}| &\leq R \cdot q + \frac{0.1q \cdot \tau_R}{p_R} \\ &\leq O\left(\left(\log n + \frac{\log m}{\varepsilon}\right)\right) \cdot \mathrm{SetCov}(\mathcal{C}, \mathcal{S}), \end{aligned}$$

where the first inequality is due to Claim E.1 and the second inequality is due to our choice of parameters. Thus, we can conclude that the approximation ratio is $O\left(\log n + \frac{\log m}{\varepsilon}\right)$ as desired. □

We now prove Claim E.1.

Proof of Claim E.1. Before we proceed, let us state a few inequalities that will be subsequently useful. First, note that $\tau_r = \frac{\tilde{n}}{2^r} \geq n_{\min} \cdot 2^{R-r}$. From this, we can derive

$$\theta_{r,i} \overset{(14)}{\leq} \log(6m/\beta_r) = \log(m/\beta) + (R - r + 3) \leq 0.005\tau_r, \tag{19}$$

and

$$\zeta_r = 10 \cdot \ln(12m^q/\beta_r) \leq 10q \ln(m/\beta_r) \leq 0.05q\tau_r. \tag{20}$$

For brevity, we call the two statements in the claim $P(r)$. We will prove $P(r)$ by induction. For convenience, we also define $P(0)$ where we let $j_{-1} = j_0, \tau_0 = \tilde{n}, \varepsilon_0 = \frac{\varepsilon}{4 \cdot 2^R}$ and $p_0 = 1 - e^{-\varepsilon_0}$.

Base case. For $r = 0$, (17) trivially holds. Meanwhile, (18) follows immediately from (13).

Inductive Step. Suppose that $P(r-1)$ holds for some $r \in \mathbb{N}$. To see that (17) holds, note that (18) from $P(r-1)$ and (16) with $T = \emptyset$ implies that

$$\begin{aligned} |D_{\mathrm{s},r} \setminus S_{\pi(<j_{r-1})}| &\overset{(16)}{\leq} 1.3p_r |D \setminus S_{\pi(<j_{r-1})}| + \zeta_r \\ &\overset{(18)}{\leq} 1.3p_r \cdot \frac{0.1q\tau_{r-1}}{p_{r-1}} + \zeta_r. \end{aligned}$$

Note that $\frac{p_r}{p_{r-1}} = \frac{1-e^{-\varepsilon_r}}{1-e^{-\varepsilon_r/2}} = 1 + e^{-\varepsilon_r/2} \leq 2$ and that $\tau_{r-1} = 2\tau_r$. Plugging these together with (20) into the above, we get

$$|D_{\mathrm{s},r} \setminus S_{\pi(<j_{r-1})}| \leq 0.52q\tau_r + 0.05q\tau_r \leq 0.6\tau_r.$$

From (19), every set chosen in round r covers at least $0.995\tau_r$ additional uncovered elements in $D_{\mathrm{s},r}$. As a result, the number of sets chosen in round r (which is equal to $j_r - j_{r-1}$) is at most

$$\frac{|D_{\mathrm{s},r} \setminus S_{\pi(<j_{r-1})}|}{0.995\tau_r} \leq \frac{0.6q\tau_r}{0.995\tau_r} < q,$$

proving (17).

Next, we will prove (18). First, observe that at the end of the rounds, since $\theta_{r,i}$ are all non-negatives, the remaining sets $i \notin \mathcal{I}$ must satisfy $|D_{\mathrm{s},r} \cap S_i| < \tau_r$. This implies that the number of remaining elements $|D_{\mathrm{s},r} \cap S_{\pi(<j_r)}|$ is at most $\tau_r \cdot \mathrm{SetCov}(\mathcal{C}, \mathcal{S}) = 0.01\tau_r \cdot q$. From (17), we have $j_r - j_{r-1} \leq q$. Thus, we may apply (15) with $T = \{\pi(j_{r-1}), \ldots, \pi(j_r - 1)\}$ to arrive at

$$|D \setminus S_{\pi(<j_r)}| < \frac{|D_{\mathrm{s},r} \setminus S_{\pi(<j_r)}| + \zeta_r}{0.7p_r} \leq \frac{0.01\tau_r \cdot q + 0.05q\tau_r}{0.7p_r} < \frac{0.1q \cdot \tau_r}{p_r},$$

proving (18).

Thus, $P(r)$ holds for all $r \in [R]$. This completes our proof. □

F. On Set Cover via Subsampled Repeated EM

The algorithm of Gupta et al. (2010) for Set Cover is exactly the same as their algorithm for submodular maximization (DPSUBMODGREEDY$_{\varepsilon_0}$) except with $k = m$ and $F_D^{\mathrm{SetCov}}(T) := |\bigcup_{i\in T} S_i|$. Finally, the output permutation is just $\pi = (c_1^*, \ldots, c_m^*)$. Similar to Section 4.1.1, it is possible to use the subsampled version of this algorithm for Set Cover. Unfortunately, here we can only show an approximation ratio of $O\left(\frac{\log n \log m}{\varepsilon}\right)$ instead of the optimal $O\left(\log n + \frac{\log m}{\varepsilon}\right)$ that we presented in Section 5.1.2:

Theorem F.1. *For any $0 < \varepsilon, \beta, \eta \le 1$, $\mathrm{DPSUBMODGREEDY}^{\mathcal{S}_p}_{\ln(2),F_D^{\mathrm{SetCov}}}$ where $p = 1 - e^{-\varepsilon}$ is a polynomial-time ε-DP algorithm for Set Cover that achieves $O\left(\frac{\log n \log(m/\beta)}{\varepsilon}\right)$-approximation with probability $1 - \beta$.*

We state the utility guarantee of DPSUBMODGREEDY from (Gupta et al., 2010) in a fine-grained fashion below.

Theorem F.2 (Gupta et al. 2010)**.** *With probability $1 - \beta$, the output π from $\mathrm{DPSUBMODGREEDY}_{\varepsilon_0,F_D^{\mathrm{SetCov}}}$ satisfies the following: there exists $r = O(\mathrm{SetCov}(\mathcal{U}, \mathcal{S}) \cdot \log n)$ such that $\left|S_i \setminus \left(\bigcup_{j\in[r]} S_{\pi(j)}|\right)\right| \le O\left(\frac{\log(m/\beta)}{\varepsilon_0}\right)$ for all $i \in [m]$.*

Proof of Theorem F.1. The privacy guarantee follows in a similar manner as in the proof of Theorem 1.2.

To analyze the utility, let $D_{\mathrm{s}} \sim \mathcal{S}_p(D)$ denote the subsampled dataset that is fed as an input to $\mathrm{DPSUBMODGREEDY}_{\ln(2),F_D^{\mathrm{SetCov}}}$; note that we view D_{s} as the universe $\mathcal{U}$ and let $S_i' := D_{\mathrm{s}} \cap S_i$ for all $i \in [m]$. Furthermore, let us abbreviate $\left(\bigcup_{j\in T} S_j\right)$ as S_T, $\left(\bigcup_{j\in T} S_j'\right)$ as S_T'; furthermore, we abbreviate $S_{\{\pi(1),\ldots,\pi(r)\}}$ as $S_{\pi(\le r)}$ and similarly define $S'_{\pi(\le r)}$.By the utility guarantee of DPSUBMODGREEDY (Theorem F.2), with probability $1 - \beta/2$, there exists $r = O(\mathrm{SetCov}(D_{\mathrm{s}}, \mathcal{S}') \cdot \log n) \le O(\mathrm{SetCov}(D, \mathcal{S}) \cdot \log n)$ where

$$|S_i' \setminus S'_{\pi(\le r)}| \le O(\log(m/\beta)) \qquad \forall i \in [m]. \tag{21}$$

This means that each S_i' can cover at most $O(\log(m/\beta))$ elements from $D_{\mathrm{s}} \setminus S'_{\pi(\le r)}$, which implies

$$\Omega\left(\frac{|D_{\mathrm{s}} \setminus S'_{\pi(\le r)}|}{\log(m/\beta)}\right) \le \mathrm{SetCov}(D_{\mathrm{s}}, \mathcal{S}') \le \mathrm{SetCov}(D, \mathcal{S}). \tag{22}$$

Furthermore, applying the Chernoff bound (Theorem 2.1) with $Z_x := \mathbf{1}\left[x \in (D \setminus S_T)\right] \cdot \mathbf{1}[x \in D_{\mathrm{s}}], \mu = p \cdot |D \setminus S_T|, \alpha = 0.1, \zeta = 2000r\log(m/\beta)$ together with a union bound over all sets $T \in \binom{\mathcal{U}}{\le r}$, we can conclude that the following holds simultaneously for all $T \in \binom{\mathcal{U}}{\le r}$ with probability at least $1 - \beta/2$:

$$|D_{\mathrm{s}} \setminus S_T'| \ge (1 - \alpha)p|D \setminus S_T| - \zeta. \tag{23}$$

When (22) and (23) both hold, we have

$$\begin{aligned}
|D \setminus S_{\pi(\le r)}| &\overset{(23)}{\le} \frac{1}{(1-\alpha)p}\left(\zeta + |D_{\mathrm{s}} \setminus S'_{\pi(\le r)}|\right) \\
&\overset{(22)}{\le} \frac{1}{(1-\alpha)p}\left(\zeta + \log(m/\beta) \cdot \mathrm{SetCov}(D, \mathcal{S})\right) \\
&\le O\left(\frac{\log n \log(m/\beta)}{\varepsilon} \cdot \mathrm{SetCov}(D, \mathcal{S})\right),
\end{aligned}$$

where the last inequality follows from our choice of r, p, α.

Finally, observe that the number of sets that are chosen after r is at most $|D \setminus S_{\pi(\le r)}|$. Thus, in total the number of sets chosen is at most $r + |D \setminus S_{\pi(\le r)}| \le O\left(\frac{\log n \log(m/\beta)}{\varepsilon} \cdot \mathrm{SetCov}(D, \mathcal{S})\right)$. □

G. Metric k-Means and k-Median

We write $\mathrm{OPT}_k^q(D)$ to denote $\min_{\substack{S\subseteq[m]\\|S|=k}} \mathrm{cost}^q(S; D)$. Furthermore, for notational convenience, we let $\mathrm{cost}^q(\emptyset; D) = n$ (i.e., we think of $\min_{c\in S} d(c, x)^q$ as being 1 when S is empty).

G.1. One-Sided DP Algorithms

G.1.1. Bicriteria Approximation

It will be more convenient to start from a one-sided DP algorithm. In fact, we will first give a bicriteria approximation algorithm where the output set size is $O(k \log n)$, as stated below. Note that the algorithm is fairly similar to that of Jones et al. (2021). However, ours is simpler since we can apply the repeated exponential mechanism (Algorithm 1) directly, while their algorithm uses maximum k-coverage algorithm as a black-box (and thus they have to handle different distance scales explicitly).

Theorem G.1. *For any $0 < \varepsilon, \beta < 1$, there is an ε-add-DP algorithm that, with probability $1 - \beta$, outputs a set $T \subseteq [m]$ of size at most $O(k \log n)$ such that $\mathrm{cost}^q(T; D) \le \mathrm{OPT}_k^q(D) + O\left(\frac{k \cdot \log(m/\beta)}{\varepsilon}\right)$.*

Proof. If $2k \ln n \ge m$, then the bound is trivial by outputting $T = [m]$. Otherwise, we use the Repeated-EM$_{\varepsilon,\mathcal{A}}$ algorithm with $L = \lceil 2k \ln n \rceil$ and the candidate sets and scoring functions selected as follows:

- $\mathcal{C}_i$ is the set of remaining elements $[m] \setminus \{c_1^*, \dots, c_{i-1}^*\}$
- $q_i(c; D)$ is the reduction in the cost after adding c, i.e., $\mathrm{cost}^q(\{c_1^*, \dots, c_{i-1}^*\}; D) - \mathrm{cost}^q(\{c_1^*, \dots, c_{i-1}^*, c\}; D)$.

It is simple to see that Assumption 4.1 and Assumption 4.2 are satisfied and thus the algorithm is ε-add-DP by Theorem 4.3.

To analyze its utility, let $T_i = \{c_1^*, \dots, c_i^*\}$ denote the solution set maintained by the algorithm at time step i. Invoking Theorem 2.2 together with a union bound, we can conclude that, with probability $1 - \beta$, the following holds for all $i \in [L]$.

$$(\mathrm{cost}^q(T_i; D) - \mathrm{cost}^q(T_{i+1}; D)) \ge \max_{c \in \mathcal{C}_i} (\mathrm{cost}^q(T_i; D) - \mathrm{cost}^q(T_i \cup \{c\}; D)) - \kappa \cdot \frac{\log(m/\beta)}{\varepsilon}, \tag{24}$$

where $\kappa > 0$ is a constant. For the remainder of the proof, we will assume that this event holds.

Let $\Psi_i = \mathrm{cost}^q(T_i; d) - \mathrm{OPT}_k^q(D)$ denote the difference between the cost of the solution at the ith step and the optimal solution (among k-size subsets). Let S^* denote the optimal solution, i.e., $|S^*| = k$ such that $\mathrm{cost}^q(S^*; D) = \mathrm{OPT}_k^q(D)$. If $\Psi_i \ge 0$, then it is simple to see that $\sum_{c \in S^*} (\mathrm{cost}^q(T_i; D) - \mathrm{cost}^q(T_i \cup \{c\}; D)) \ge \Psi_i$. This implies that $\max_{c \in \mathcal{C}_i} (\mathrm{cost}^q(T_i; D) - \mathrm{cost}^q(T_i \cup \{c\}; D)) \ge \frac{\Psi_i}{k}$. Thus, if $\Psi_i \ge k \cdot \left(2\kappa \cdot \frac{\log(m/\beta)}{\varepsilon}\right)$, then we can apply (24) to conclude that $(\mathrm{cost}^q(T_i; D) - \mathrm{cost}^q(T_{i+1}; D)) \ge \frac{\Psi_i}{2k}$. Rearranging, this gives

$$\Psi_{i+1} \le \left(1 - \frac{1}{2k}\right) \Psi_i, \tag{25}$$

Thus, either we have $\Psi_i < k \cdot \left(2\kappa \cdot \frac{\log(m/\beta)}{\varepsilon}\right)$ for some $i \in [L]$ which immediately satisfies the desired accuracy bound, or we can repeatedly apply (25) to arrive at

$$\Psi_L \le \left(1 - \frac{1}{2k}\right)^L \Psi_0 \le e^{-0.5L/k} \cdot n \le 1,$$

where the last inequality is due to our choice of L. As such, the accuracy guarantee also holds in this case. □

G.1.2. From Bicriteria to True Approximation

We can then go from bicriteria approximation to true approximation (i.e., output set size k) via standard techniques.

Theorem G.2. *For any $0 < \varepsilon, \beta < 1$, there is an ε-add-DP algorithm for metric k-median and metric k-means that achieves $O(1)$-approximation and $O\left(\frac{k \log(mn/\beta)}{\varepsilon}\right)$-error w.p. $1 - \beta$.*

Proof. The algorithm works as follows.

- Run the $\varepsilon/2$-add-DP algorithm from Theorem G.1 to obtain $T \subseteq [m]$ of size $O(k \log n)$.
- Create a histogram $(\tilde{h}_t)_{t \in T}$ as follows. First, let $\hat{h}_t = |\{x \in D \mid t = \mathrm{argmin}_{t' \in T} d(t', x)\}|$ (where ties are broken arbitrarily in argmin). Then, sample $\theta_t \sim \mathrm{Exp}(\varepsilon/2)$ independently and let $\tilde{h}_t = \hat{h}_t + \theta_t$.

- Let $\tilde{D}$ be the dataset that, for each $t \in T$, contains $\tilde{h}_t$ copies of t. Run any (non-private) $O(1)$-approximation algorithm (e.g., (Arya et al., 2004) for k-median and (Kanungo et al., 2004) for k-means) on $\tilde{D}$ to produce a solution S. Then, output S.

Since the second step is an application of the exponential-noise mechanism, it satisfies $(\varepsilon/2)$-add-DP. As a result, by applying the basic composition theorem and viewing the last step as a post-processing, we can conclude that the entire algorithm is ε-add-DP.

By Theorem G.1, with probability $1 - \beta/2$, we have

$$\mathrm{cost}^q(T; D) \le \mathrm{OPT}_k^q(D) + O\left(\frac{k \cdot \log(m/\beta)}{\varepsilon}\right) \tag{26}$$

Furthermore, by standard concentration of sum of exponential random variables (see e.g., (Brown, 2011)), the following holds with probability $1 - \beta/2$:

$$\sum_{t \in T} \theta_t \le O\left(\frac{k \log(n/\beta)}{\varepsilon}\right). \tag{27}$$

Henceforth, we will assume that (26) and (27) hold.

Let S^* denote the optimal solution in the original dataset D, i.e., $|S^*| = k$ such that $\mathrm{cost}^q(S^*; D) = \mathrm{OPT}_k^q(D)$. Furthermore, let $\hat{D}$ be the dataset that, for each $t \in T$, contains $\hat{h}_t$ copies of t. By the guarantee of the non-private approximation algorithm, we have

$$\begin{aligned}
\mathrm{cost}^q(S; \tilde{D}) &\le O(1) \cdot \mathrm{OPT}^q(\tilde{D}) \\
&\le O(1) \cdot \mathrm{cost}^q(S^*; \tilde{D}) \\
&\overset{(27)}{\le} O(1) \cdot \mathrm{cost}^q(S^*; \hat{D}) + O\left(\frac{k \log(n/\beta)}{\varepsilon}\right) \\
&\overset{(\clubsuit)}{\le} O(1) \cdot (\mathrm{cost}^q(S^*; D) + \mathrm{cost}^q(T; D)) + O\left(\frac{k \log(n/\beta)}{\varepsilon}\right) \\
&\overset{(26)}{\le} O(1) \cdot \mathrm{OPT}_k^q(D) + O\left(\frac{k \log(mn/\beta)}{\varepsilon}\right),
\end{aligned} \tag{28}$$

where ($\clubsuit$) follows from the qth power triangle inequality, i.e., $d(u, v)^q \le 2^q(d(u, w)^q + d(w, v)^q)$ for all $u, v, w \in [m]$.

Finally, we have

$$\begin{aligned}
\mathrm{cost}^q(S; D) &\overset{(\spadesuit)}{\le} O(1) \cdot \left(\mathrm{cost}^q(S; \hat{D}) + \mathrm{cost}^q(T; D)\right) \\
&\overset{(27),(26)}{\le} O(1) \cdot \left(\mathrm{cost}^q(S; \tilde{D}) + \frac{k \log(mn/\beta)}{\varepsilon}\right) \\
&\overset{(28)}{\le} O(1) \cdot \left(\mathrm{OPT}_k^q(D) + \frac{k \log(mn/\beta)}{\varepsilon}\right),
\end{aligned}$$

where ($\spadesuit$) again follows from the qth power triangle inequality. □

G.2. From One-Sided to Two-Sided DP

Finally, we go from one-sided to two-sided DP using Theorem 4.4 similar to the previous analyses.

Proof of Theorem 1.5. Let $\mathcal{A}$ denote a $\ln(2)$-add-DP algorithm from Theorem G.2. We use the algorithm subsampled version of this algorithm, i.e., $\mathcal{A}^{\mathcal{S}_p}$ for $p = 1 - e^{-\varepsilon}$. Theorem 4.4 immediately implies that this algorithm is ε-DP as desired.

For the utility, let $D_{\mathrm{s}} \sim \mathcal{S}_p(D)$ denote the subsampled dataset that is fed as an input to $\mathcal{A}$ and let T^* denote the output set. From Theorem G.2, w.p. $1 - \beta/2$, we have

$$\mathrm{cost}^q(T^*; D_{\mathrm{s}}) \le O(1) \cdot \mathrm{OPT}_k^q(D_{\mathrm{s}}) + O\left(k \log\left(\frac{mn}{\beta}\right)\right). \tag{29}$$

Furthermore, applying the Chernoff bound (Theorem 2.1) with $Z_x := (\min_{c\in T} d(c,x)^q) \cdot \mathbf{1}[x \in D_{\mathrm{s}}], \mu = p \cdot \mathrm{cost}^q(T;D), \alpha = 0.1, \zeta = 20000k\log(m/\beta)$ together with a union bound over all sets $T \in \binom{[m]}{\leq k}$, we can conclude that the following hold simultaneously for all $T \in \binom{[m]}{\leq k}$ with probability at least $1-\beta/2$:

$$\mathrm{cost}^q(T;D_{\mathrm{s}}) \geq (1-\alpha)p \cdot \mathrm{cost}^q(T;D) - \zeta, \tag{30}$$

$$\mathrm{cost}^q(T;D_{\mathrm{s}}) \leq (1+\alpha)p \cdot \mathrm{cost}^q(T;D) + \zeta. \tag{31}$$

When (29), (30), and (31) all hold, we have

$$\begin{aligned}
\mathrm{cost}^q(T^*;D) &\overset{(30)}{\leq} \frac{1}{(1-\alpha)p}\mathrm{cost}^q(T^*;D_{\mathrm{s}}) + \frac{\zeta}{(1-\alpha)p} \\
&\overset{(29)}{\leq} O\left(\frac{1}{(1-\alpha)p}\right)\cdot \mathrm{OPT}_k^q(D_{\mathrm{s}}) + O\left(\frac{k\log\left(\frac{mn}{\beta}\right)}{p(1-\alpha)}\right) + \frac{\zeta}{(1-\alpha)p} \\
&\overset{(31)}{\leq} O\left(\frac{1+\alpha}{1-\alpha}\right)\cdot \mathrm{OPT}_k^q(D) + O\left(\frac{k\log\left(\frac{mn}{\beta}\right)}{p(1-\alpha)}\right) + O\left(\frac{\zeta}{(1-\alpha)p}\right) \\
&\leq O(1)\cdot \mathrm{OPT}_k^q(D) + O\left(\frac{k\log\left(\frac{mn}{\beta}\right)}{\varepsilon}\right),
\end{aligned}$$

which concludes our proof. □