Fast and Private Max-Sum Diversification

Ron Zadicario
Tel Aviv University
ronzadicario@mail.tau.ac.il

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

ABSTRACT

Result diversification is crucial for generating informative, non-redundant data summaries and query outputs. Although its various formulations have been extensively studied across an array of data-driven disciplines, existing methods fail to address the privacy concerns that arise when the underlying data is sensitive. In this work, we initiate the study of result diversification under *differential privacy*, focusing on the *max-sum diversification* (MSD) problem, a widely adopted model with the objective of maximizing a linear combination of a submodular function, quantifying relevance, and the sum of pairwise distances between selected items, quantifying diversity. We propose differentially private algorithms for MSD under both cardinality and matroid constraints, achieving nearly optimal utility guarantees. At the same time, we design more efficient algorithms that maintain strong guarantees. Notably, the proposed algorithms are faster than existing non-private methods, making them appealing even in non-private settings. Experimental evaluations on real-world datasets demonstrate that the proposed approach achieves utility comparable to that of non-private baselines even under strong privacy guarantees, and significantly improves execution times for cardinality constraints.

PVLDB Reference Format:
Ron Zadicario and Tova Milo. Fast and Private Max-Sum Diversification.
PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/ronzadi/Differentially-Private-Max-Sum-Diversification.

1 INTRODUCTION

Result diversification is the problem of selecting a subset of items that is both relevant and diverse, subject to feasibility constraints. It underlies numerous data summarization and selection tasks, where capturing data variety, avoiding redundancy, and providing informative outputs are crucial. Consequently, its various formulations have been extensively studied across multiple domains, including databases, machine learning, and operations research. Among these formulations, the *max-sum diversification* (MSD) problem is one of the most well-studied and widely adopted diversity models in the database literature. It has been applied to a wide range of tasks, including query result diversification [3, 8, 20, 29, 55], graph summarization [32, 53], rule mining [7, 34], spatial keyword search [60], as well as recommender systems [11, 18, 47] and feature selection [30, 58]. In this problem, we are given a set of items, referred to as the *ground set*, together with pairwise distances and a monotone submodular set function f that quantifies the *relevance* of each subset. The objective is to select a subset S that maximizes a linear combination of $f(S)$ and the sum of pairwise distances among the items in S, subject to given feasibility constraints [8]. Prior work has primarily focused on cardinality constraints, which limit the size of the selected subset, and more generally on *matroid* constraints (defined in Section 3), which capture a broad range of practical feasibility criteria. A notable special case of this model is that of diverse Top-k queries [31], where each item v is associated with a score $q(v)$, the relevance of a subset S is given by $f(S) = \sum_{v \in S} q(v)$, and the constraint is cardinality $|S| \leq k$. By allowing arbitrary monotone submodular relevance functions and matroid constraints, MSD supports a substantially more expressive class of data summarization tasks, as illustrated in Example 1.1.

Yet, many compelling applications of result diversification, and of MSD in particular, involve sensitive individual data, such as purchase histories, locations, and search patterns. Improper handling of such data can pose significant privacy risks [56, 59], and therefore the maximization procedure must provide formal privacy guarantees for the individuals contributing to the dataset.

Example 1.1. An online retailer wishes to select a set S of k popular products from a dataset filtered by `category` = "Health Care", for purposes such as catalog design, promotion, or trend analysis. Selecting the Top-k most purchased items may primarily reflect the preferences of a narrow group of highly active users. To instead maximize user reach, the retailer can maximize the coverage function $f(S) = \frac{1}{m} |\bigcup_{v \in S} \mathcal{U}(v)|$, where $\mathcal{U}(v)$ denotes the set of users who purchased item v and m is the total number of users. This objective is monotone submodular and often used for summarization [19]. However, optimizing coverage alone may still yield a redundant set of items: Figure 1a shows results on the Amazon Reviews dataset [37], where three of the six selected items (indicated by shaded rows) are essential oils. Selecting overly similar products may neglect users' varied interests and reduce long-term satisfaction [12].

To encourage diversity, each product is associated with a set of subcategories, and pairwise distances are defined using the Jaccard distance [39] between these sets. The retailer can then maximize a linear combination of $f(S)$ and the sum of pairwise distances among items in S, producing a relevant and diverse set of products. Additional constraints can be incorporated naturally. For example, limiting the number of selected products within each price category induces a matroid constraint.

Critically, the selection procedure must also preserve users' privacy. Since the objective is derived from individual transaction

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

Product	Subcategory
Muscle Cream	Treatments
Cassia Oil	Essential Oils
Oil Diffuser	Diffusers
Heating Pad	Hot & Cold Therapies
Garlic Oil	Essential Oils
Rosemary Oil	Essential Oils
Coverage: 0.15	**Diversity:** 0.20

(a) Relevance-only selection. Three of six items are from the "Essential Oils" subcategory (shaded).

Product	Subcategory
Muscle Cream	Treatments
Cassia Oil	Essential Oils
Oil Diffuser	Diffusers
Heating Pad	Hot & Cold Therapies
Shoe Insoles	Foot Health
Ear Otoscope	Ear Care
Coverage: 0.145	**Diversity:** 0.31

(b) Selection balancing relevance and diversity. Shaded items are newly added.

Figure 1: Representative products from the Amazon Health Care category. Product and subcategory names are simplified for readability.

logs, an exact solution could inadvertently reveal sensitive information. For instance, the inclusion of a niche product could signal the presence of a user with a rare medical condition. Preventing such leakage necessitates a mechanism that provides formal privacy guarantees. Using the algorithms proposed in this work, the retailer can generate a relevant, diverse, and privacy-preserving summary, as illustrated in Figure 1b, where the shaded rows indicate two products added by the private and diversity-aware algorithm.

Differential Privacy [22, 24] is a rigorous notion that has become the gold standard for analyzing sensitive data under strong privacy guarantees, and has been adopted by multiple companies [2, 23] and governmental organizations [21, 26, 54]. Intuitively, the output distribution of a differentially private (DP) algorithm changes only minimally when a single individual's data is modified, thereby ensuring that individual-level information is not disclosed (see Section 3 for the formal definition). Privacy is typically achieved by injecting noise into the computation process, which may result in some degradation in utility. In our setting, achieving stronger privacy often means computing a set of items with a slightly lower quality score. Although DP submodular maximization has received considerable attention in recent years [16, 33, 43, 48], the design of DP algorithms for the more general MSD problem has yet to be addressed. Existing techniques for DP submodular maximization do not directly apply to MSD, as its objective function is not submodular.

In this work, we study the MSD problem under differential privacy. We propose DP algorithms that provide near-optimal utility guarantees under both cardinality and general matroid constraints. Our proposed algorithms also improve time complexity compared to existing non private methods, making them valuable even in non private settings. Experimental evaluation on two real-world applications, Amazon product and Uber pickup location summarization, shows that our approach achieves utility comparable to non private baselines while significantly reducing the number of objective evaluations. We now state our contributions, with a summary of the main results in Table 1. To provide context, Section 2 reviews relevant lower bounds from prior work.

1.1 Our Results

In the DP setting, the MSD objective function depends on a sensitive dataset D and is a linear combination of a submodular relevance term and a diversity term. It is called *decomposable* if it can be expressed as a sum of functions, each depending on a single record of D. The formal definitions are provided in Section 4.

We begin with a direct DP adaptation of the greedy algorithm of Borodin et al. [8], which serves as a natural baseline. Building on this, our main contribution is a faster DP algorithm whose analysis goes well beyond the original work, simultaneously addressing subsampling, privacy, and the absence of submodularity.

Greedy algorithm for cardinality constraints. Cardinality constraints, which capture the fundamental Top-k query class, are among the most commonly studied constraint types. Borodin et al. [8] proposed a *non-oblivious*[1] greedy algorithm for cardinality-constrained MSD, achieving a tight $1/2$-approximation. We extend their analysis to account for the error introduced by replacing greedy selections with privatized ones. For decomposable objectives, we show that the technique of Gupta et al. [33] provides improved privacy guarantees. The precise results, corresponding to row *(i)* of Table 1, are discussed in Section 5. However, this algorithm makes $O(nk)$ value oracle calls[2], where k is the cardinality bound and n is the ground set size. For large k (e.g., $k = \Omega(n)$), this complexity reaches $\Omega(n^2)$ oracle calls, underscoring the need for more efficient algorithms that better suit interactive settings.

Faster greedy algorithm via subset sampling. Algorithms for submodular maximization are often accelerated by "sample greedy" methods, where the next element is chosen from a smaller random subset rather than the entire ground set, improving efficiency at the cost of a slightly weaker approximation guarantee [10, 42, 43].Yet, these analyses rely on the submodularity of the objective function, which does not hold in our setting. Nevertheless, we propose an efficient "sample greedy" algorithm for the DP MSD problem with cardinality constraints. *To our knowledge, this is the fastest known algorithm for MSD, making it valuable even in non-private settings.*

We present two instantiations of the algorithm, each illustrating a distinct trade-off between complexity and utility. Furthermore, we extend the technique of Gupta et al. [33] to account for the sampling step, and prove stronger guarantees for decomposable objectives. The precise results, corresponding to rows *(ii)* and *(iii)* of Table 1, are given in Section 6.

Local search for matroid constraints. Some data summarization tasks require more general constraints than simple cardinality, and matroids are among the most studied such constraints. Since the greedy paradigm does not provide any constant-factor approximation for the MSD problem under matroid constraints, Borodin et al. [8] showed that a natural single-swap local search algorithm achieves a $(1/2 - \gamma)$-approximation. However, checking for local optimality or ensuring the selection of improving swaps is infeasible under differential privacy. To address this, we propose a DP local search algorithm for MSD under matroid constraints that performs a predefined number of local search steps and selects swaps privately. The analysis bounds the loss incurred by allowing

[1] An optimization algorithm is non-oblivious if it selects the next element with respect to an auxiliary function rather than the true objective.

[2] Number of evaluations of the objective function; see Section 3.

Table 1: Expected utility guarantees of (ε, δ)-DP algorithms and their query complexity. Here, $\tilde{O}$ omits poly-logarithmic factors in δ^{-1} and γ^{-1}. For general Δ-sensitive functions with cardinality constraints, additive errors incur an additional $\sqrt{k}$ factor. Non-private algorithms are implied by dropping the additive term (i.e., $\varepsilon = \infty$).

Constraint	Approx.	Additive Error (Δ-decomposable)	Oracle Calls	Algorithm	
k-Cardinality	$1/2$	$\tilde{O}(\Delta k \varepsilon^{-1} \log n)$	$O(nk)$	`DP-Greedy` (1)	(i)
	$1/2 - \gamma$	$\tilde{O}(\Delta k \varepsilon^{-1} \log n)$	$O(n \log k \log \gamma^{-1})$	`DP-NOSG` (2)	(ii)
	$1 - 2/e - \gamma - 1/k$	$\tilde{O}(\Delta k \varepsilon^{-1} \log n)$	$O(n \log \gamma^{-1})$	`DP-OSG` (2)	(iii)
Matroid (rank k)	$1/2 - \gamma$	$\tilde{O}\left(\frac{\Delta k^{1.5} \sqrt{\log k} \log n}{\varepsilon \sqrt{\gamma}}\right)$	$O(\gamma^{-1} nk \log k)$	`DP-SLS` (3)	(iv)

non-improving swaps, yielding near-optimal guarantees. We further improve efficiency by sampling a subset of candidate swaps in each iteration, reducing to $O(\gamma^{-1} k \log k)$ oracle calls, a substantial improvement over the $O(n^2 + \gamma^{-1} nk^2 \log k)$ by Borodin et al. [8]. *To our knowledge, this algorithm achieves the best known complexity for MSD under matroid constraints.*

Importantly, even without privacy, the algorithm achieves (in expectation) the same approximation ratio as Borodin et al. [8] with improved complexity. The precise statements of our results, corresponding to row *(iv)* of Table 1, are given in Section 7.

2 RELATED WORK

We provide a brief survey of relevant literature, starting with non-private settings, followed by lower bounds, and finally covering additional privacy-related contributions.

Non-private result diversification. The MSD problem generalizes the *max-sum dispersion* problem (also called *remote-clique*), which aims to maximize $\sum_{\{u,v\} \subseteq S} d(u, v)$ and has attracted sustained attention for over three decades [27, 35, 36, 49]. Gollapudi and Sharma [31] first studied MSD for modular relevance and cardinality constraints, obtaining a 1/2-approximation by reduction to max-sum dispersion, which was later extended to matroid constraints by Abbassi et al. [1]. Borodin et al. [8] substantially generalized this framework to submodular relevance and matroid constraints, providing a greedy algorithm for cardinality constraints and a local search algorithm for general matroid constraints with $1/2$ and $(1/2 - \gamma)$ approximations, respectively. For Euclidean distances with modular relevance, Indyk et al. [38] introduced composable coresets for streaming and distributed settings. Ceccarello et al. [14] later generalized these constructions to metrics with bounded doubling dimension, and Ceccarello et al. [13] extended the framework to matroid constraints. A fully dynamic variant for doubling metrics was studied by Pellizzoni et al. [46]. Agarwal et al. [4] proposed near-linear-time algorithms using efficient indexes for euclidean distances. Additionally, Cevallos et al. [15] studied MSD under negative-type distances, a setting that is incomparable to general metric distances. Despite the extensive literature, none of these works provides formal privacy guarantees.

Hardness and lower bounds. Cardinality-constrained max-sum dispersion is known to be NP-hard even with metric distances [35]; hence, the more general MSD problem is also NP-hard. Moreover, Borodin et al. [8] showed that the approximation factor of $1/2$ is tight under the assumption that the *planted-clique problem* [5] is hard. Additionally, the DP MSD problem, which we formulate in Section 4, is a generalization of DP monotone submodular maximization, and therefore all previously established lower bounds for the latter apply in our setting. Gupta et al. [33] show that any ε-DP algorithm for maximizing a submodular function subject to a cardinality constraint k over a ground set of size n must incur an expected additive error of $\Omega(k \log(n/k)/\varepsilon)$. Chaturvedi et al. [17] extended this lower bound from the $(\varepsilon, 0)$ to the (ε, δ) setting, and proved that any (ε, δ)-DP algorithm that achieves a multiplicative approximation factor of c must incur additive error $\Omega(kc \log(\varepsilon/\delta)/\varepsilon)$, assuming $n \geq k(e^{\varepsilon} - 1)/\delta$ and $c \geq 4\delta/(e^{\varepsilon} - 1)$. In both lower bounds, the hard instance uses a 1-decomposable objective function. This implies that our additive error terms for decomposable objectives under cardinality constraints are optimal up to logarithmic terms.

DP submodular maximization. To the best of our knowledge, the MSD problem has not been studied in the context of differential privacy. We therefore review the most closely related line of work, which focuses on DP submodular maximization. The work of Gupta et al. [33] was the first to address DP submodular maximization, focusing on decomposable objective functions under cardinality constraints. Mitrovic et al. [43] considered the more general case of bounded-sensitivity submodular objectives, proposing algorithms for both monotone and non-monotone functions under matroid and p-extendable systems constraints. Chaturvedi et al. [16] revisited decomposable objectives, extending the results of Gupta et al. [33] from cardinality to matroid constraints and from monotone to non-monotone functions, improving upon Mitrovic et al. [43]. Rafiey and Yoshida [48] studied private submodular and k-submodular maximization under matroid constraints, achieving improved running time at the cost of larger additive error. Other notable advances include Salazar and Cummings [51], who addressed the online setting under cardinality constraints, Sadeghi and Fazel [50], who studied bounded-curvature objectives under matroid and knapsack constraints in both offline and online settings, and Chaturvedi et al. [17], who proposed a streaming algorithm for cardinality constraints. Yet, all of these approaches are applicable to submodular objectives and do not directly apply for MSD.

3 PRELIMINARIES

In this section, we present the basic definitions and notation, introduce submodular set functions and matroids, and review key concepts from differential privacy relevant to this work.

Notation and Set Functions. Let V denote a finite ground set of size n. For a set function $f : 2^V \to \mathbb{R}$, the *marginal contribution* of an element $u \in V$ to a set $S \subseteq V$ is $f(u \mid S) = f(S \cup \{u\}) - f(S)$. A function $d : V \times V \to \mathbb{R}_{\geq 0}$ is a *pseudometric* if it is symmetric, satisfies the triangle inequality, and $d(u, u) = 0$ for all $u \in V$. Extend d to sets via $d(S) = \sum_{\{u,v\} \subseteq S} d(u, v)$ and $d(S, T) = \sum_{u \in S, v \in T} d(u, v)$ for disjoint sets $S, T \subseteq V$. For $u \in V$, write $d(u \mid S) = d(S \cup \{u\}) - d(S) = \sum_{v \in S} d(u, v)$.

Submodular Functions and Matroids. A set function $f : 2^V \to \mathbb{R}$ is *monotone* if $f(S) \leq f(T)$ for all $S \subseteq T$, *non-negative* if $f(S) \geq 0$ for all $S \subseteq V$, and *submodular* [45] if $f(u \mid S) \geq f(u \mid T)$ for all $S \subseteq T$ and $u \in V \setminus T$. A *matroid* [57] is a pair $(V, \mathcal{I})$, where $\mathcal{I} \subseteq 2^V$ satisfies *(i)* if $A \subseteq B$ and $B \in \mathcal{I}$ then $A \in \mathcal{I}$, and *(ii)* for $A, B \in \mathcal{I}$ with $|A| < |B|$, there exists $u \in B \setminus A$ such that $A \cup \{u\} \in \mathcal{I}$. Subsets in $\mathcal{I}$ are called *independent sets*, and inclusion-wise maximal independent sets are called *bases*. All bases have the same cardinality, known as the *rank* of the matroid. We assume access to f, d, and $\mathcal{I}$ via *oracles*: a *value oracle* returns $f(S)$ or $d(S)$ for a given S, and an *independence oracle* determines whether $S \in \mathcal{I}$. The total number of oracle calls commonly serves as a proxy for algorithmic complexity. See Section A for additional background and examples of matroids.

Differential Privacy. A dataset D is a finite multiset of records from a domain $\mathcal{X}$. The number of records in D is denoted by $m = |D|$. We let $\mathcal{D}$ denote the space of all possible datasets over this domain. Two datasets D and D' are called *neighboring* (denoted $D \sim D'$) if they differ in one record. Intuitively, differential privacy ensures that the distribution of outputs of a randomized algorithm does not significantly change when the data of a single individual is changed.

Definition 3.1 (Differential Privacy [22, 24]). A randomized algorithm $\mathcal{A}$ is (ε, δ)-differentially private (DP) if for any neighboring datasets $D \sim D'$ and any set of possible outputs $S \subseteq Range(\mathcal{A})$,

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

If $\delta = 0$, we say that $\mathcal{A}$ is ε-DP.

In our privacy analysis, we use the following standard composition results, which capture how privacy loss accumulates when multiple DP algorithms are applied sequentially. A more general statement appears in [24].

Theorem 3.2 (Composition [24, 25]). *Let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be ε-DP algorithms. Their k-fold adaptive composition $\mathcal{A}_{[k]}$, which outputs $y_i = \mathcal{A}_i(D, y_1, \dots, y_{i-1})$ for $i = 1, \dots, k$, satisfies:*

(i) $k\varepsilon$-DP (Basic),

(ii) $(\sqrt{2k \log(1/\delta)}\varepsilon + k\varepsilon(e^{\varepsilon} - 1),\ \delta)$-DP for any $\delta > 0$ (Advanced).

In particular, to achieve (ε, δ)-DP for $\varepsilon \in (0, 1)$, it suffices that each $\mathcal{A}_i$ satisfies $\frac{\varepsilon}{2\sqrt{2k \log(1/\delta)}}$-DP.

DP algorithms must be calibrated to the sensitivity of the function of interest with respect to single-record modifications of the input dataset, defined formally as follows.

Definition 3.3 (Sensitivity [24]). The *sensitivity* of a function $q_D : \mathcal{R} \to \mathbb{R}$ that depends on a dataset D is defined as $\Delta_q = \sup_{r \in \mathcal{R}} \sup_{D \sim D'} |q_D(r) - q_{D'}(r)|$.

The exponential mechanism [41] is a primitive for privately selecting an approximately highest-scoring element from a candidate set according to a quality function that depends on the sensitive dataset.

Definition 3.4 (The Exponential Mechanism [41]). Given $D \in \mathcal{D}$, a finite set of candidates $\mathcal{R}$, a quality function $q_D : \mathcal{R} \to \mathbb{R}$, and a privacy parameter ε, the exponential mechanism $\mathsf{EM}(D, q, \mathcal{R}, \varepsilon)$ outputs $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\varepsilon \cdot q_D(r)}{2\Delta_q}\right)$.

We assume that the sensitivity bound Δ_q is provided alongside q and do not consider it as a separate input.

Theorem 3.5 ([6, 41]). *The exponential mechanism is ε-DP. Moreover,*

$$\mathbb{E}\left[\mathsf{EM}(D, q, \mathcal{R}, \varepsilon)\right] \geq \max_{r \in \mathcal{R}} q_D(r) - \frac{2\Delta_q \log |\mathcal{R}|}{\varepsilon}.$$

The utility guarantee in expectation, which we use in this work, is due to [6]. The privacy guarantee is due to [41], as is the "with high probability" utility guarantee, which we omit here.

4 PROBLEM FORMULATION

The DP max-sum diversification (MSD) problem studied in this work extends the setting of Borodin et al. [8] to incorporate differential privacy requirements, and is formalized below.

Definition 4.1 (DP max-sum diversification). An instance of DP MSD is a tuple $\langle D, V, \mathcal{I}, f_D, d_D \rangle$ where:

- D is a sensitive dataset.
- V is a (public) dataset of items, called the *ground set.*
- $\mathcal{I} \subseteq 2^V$ is a collection of feasible subsets.
- $f_D : 2^V \to \mathbb{R}^+$ is a monotone submodular function that depends on the sensitive dataset D.
- $d_D : V \times V \to \mathbb{R}$ is a pseudometric (possibly depending on D). and we write $d_D(S) \triangleq \sum_{\{u,v\} \subseteq S} d_D(u, v)$.

The goal is to find $S \in \mathcal{I}$ that approximately maximizes

$$\phi_D(S) \triangleq f_D(S) + \lambda \cdot d_D(S)$$

while satisfying differential privacy with respect to D.

The parameter $\lambda \geq 0$ controls the trade-off between relevance and diversity. Since $\lambda \cdot d_D$ remains a pseudometric, we simplify notation by writing $\phi_D(S) = f_D(S) + d_D(S)$ throughout the paper, and omitting the subscript D when it is clear from context.

Example 4.1. In the setting of Example 1.1, the sensitive dataset D consists of purchase records where each entry specifies the products purchased by a user. The ground set V contains Amazon products along with attributes such as product name, price, and category. The relevance function is defined as $f_D(S) = \frac{1}{m}\left|\bigcup_{v \in S} \mathcal{U}_D(v)\right|$, where $\mathcal{U}_D(v)$ is the set of users in D who purchased item v. Let $C(v)$ denote the set of categories of item $v \in V$, diversity quantified by the Jaccard distance $d(v, v') = 1 - \frac{|C(v) \cap C(v')|}{|C(v) \cup C(v')|}$ between category sets of products v and v'.

Decomposable objective functions. A set function f_D is called Δ-*decomposable* [33, 43] if $f_D(S) = \frac{1}{m} \sum_{x \in D} f_x(S)$, where each $f_x : 2^V \to [0, \Delta]$ depends only on the individual record x, and $m = |D|$. A Δ-decomposable function has sensitivity $\frac{\Delta}{m}$. Prior work has studied DP maximization of decomposable submodular functions [16, 17, 33], as in some cases they allow stronger privacy guarantees compared to the general case. We extend this notion to our setting by defining ϕ_D as Δ-*decomposable* if $\phi_D(S) = \frac{1}{m}\sum_{x\in D}\phi_x(S)$, where each $\phi_x : 2^V \to [0,\Delta]$ depends only on $x \in D$. Our algorithms for cardinality constraints provide stronger privacy guarantees for decomposable objectives.

Example 4.2. In the setting of Example 1.1, associate each user $x \in D$ with a monotone, submodular utility $f_x(S) = \min\{|S \cap \mathcal{P}(x)|, 1\}$, where $\mathcal{P}(x)$ is the set of products purchased by user x. This function indicates whether user x is covered by S and depends solely on the data of that individual. For $\lambda \in [0,1]$, define the per-user objective

$$\phi_x(S) = (1-\lambda)f_x(S) + \frac{2\lambda}{k(k-1)}d(S).$$

As both relevance and scaled diversity are bounded in $[0,1]$,we have $\phi_x(S) \in [0,1]$. The total objective is the average of these individual contributions:

$$\phi(S) = \frac{1}{|D|}\sum_{x\in D}\phi_x(S) = \frac{1-\lambda}{|D|}\sum_{x\in D}f_x(S) + \frac{2\lambda}{k(k-1)}d(S).$$

It follows that ϕ is 1-decomposable.

5 GREEDY ALGORITHM FOR CARDINALITY CONSTRAINTS

We start by presenting DP-Greedy, a DP adaptation of the greedy algorithm from [8], replacing each selection step with the exponential mechanism. Although it achieves high utility, as also observed experimentally in Section 8, it scales poorly as k increases. *Addressing this limitation constitutes a primary contribution of this work* and is the focus of the subsequent section. DP-Greedy, stated as Algorithm 1, is *non-oblivious*: it selects elements according to the auxiliary function $\phi'_D(S) = \frac{1}{2}f_D(S) + d_D(S)$ rather than the objective ϕ_D.

Algorithm 1 DP-Greedy

```
Input: Dataset $D$, ground set $V$, monotone submodular $f_D$, pseudometric $d_D$, cardinality $k$, privacy parameter $\varepsilon_0$.
Output: Size $k$ subset of $V$
Define $\phi'_D : 2^V \to \mathbb{R}^+$ via $\phi'_D(S) = \frac{1}{2}f_D(S) + d_D(S)$
Set $S_0 \leftarrow \emptyset$
for $i = 1, \dots, k$ do
    Let $N_i \leftarrow V \setminus S_{i-1}$
    For all $u \in N_i$, define $q^i_D(u) = \phi'_D(u \mid S_{i-1})$
    Compute $u_i \leftarrow$ EM$(D, q^i, N_i, \varepsilon_0)$
    $S_i \leftarrow S_{i-1} \cup \{u_i\}$
end for
```

Theorem 5.1. *Suppose ϕ_D has sensitivity Δ. Given parameter $\varepsilon_0 > 0$,* DP-Greedy *is $k\varepsilon_0$-DP and (ε,δ)-DP for all $\delta > 0$ with $\varepsilon = \sqrt{2k\log(1/\delta)}\,\varepsilon_0 + k\varepsilon_0(e^{\varepsilon_0}-1)$. Moreover, it outputs a set S such that $\mathbb{E}[\phi_D(S)] \geq \frac{1}{2}\phi_D(\mathrm{OPT}) - O(\frac{k\Delta\log n}{\varepsilon_0})$ and makes $O(nk)$ oracle calls.*

In the special case where ϕ_D is decomposable (and hence so is ϕ'_D), DP-Greedy provides a stronger privacy guarantee compared to the general case, as stated next. The assumption that ϕ_D is 1-decomposable is without loss of generality, since we can apply our algorithms to the normalized function ϕ_D/Δ, and the guarantees for ϕ_D incur an additive error scaled by Δ.

Theorem 5.2. *Suppose ϕ'_D is 1-decomposable. Then given parameter $\varepsilon_0 \in (0,1]$,* DP-Greedy *(Algorithm 1) is (ε,δ)-DP for every $\delta > 0$ and $\varepsilon = (e^{\varepsilon_0/2}-1)(4+\log(1/\delta))$.*

Theorems 5.1 and 5.2 correspond to row *(i)* in Table 1. Informally, advanced composition implies that setting $\varepsilon_0 \approx \varepsilon/\sqrt{k}$ (omitting polylogarithmic factors in $1/\delta$) ensures (ε,δ)-DP. However, Theorem 5.2 implies that setting $\varepsilon_0 \approx \varepsilon$ suffices to ensure (ε,δ)-DP, avoiding the dependence on k that arises from composition in the general case. Substituting this into the utility guarantee of Theorem 5.1 yields the stated additive error in Table 1. Theorem 5.2 is proved by the same steps as in the privacy proof for the CPP problem in [33]. Since this result follows as a special case of our more general analysis in Section 6, it is omitted. In the remainder of this section, we prove Theorem 5.1.

Let OPT be an optimal solution, i.e., a subset of size at most k that maximizes ϕ. Since ϕ is monotone, we may assume that $|\mathrm{OPT}| = k$. Let S_i be the solution at the end of iteration i, and define $C_i = \mathrm{OPT} \setminus S_{i-1}$. The approximation guarantee of the greedy algorithm in [8] relies on a lemma which relates the overall distance between the sets S_{i-1} and C_i to the value $d(\mathrm{OPT})$. An adaptation of this lemma is stated next. The proof is deferred to Section B.

Lemma 5.3 (Adapted from [8]). *For all $i \in [k] \triangleq \{1,\dots,k\}$, $d(C_i, S_{i-1}) \geq \frac{(i-1)|C_i|}{k(k-1)}d(\mathrm{OPT})$.*

Proof. For the query complexity, observe that the algorithm performs k iterations, each requiring $O(n)$ oracle queries. The privacy guarantee follows directly from the composition theorems (Theorem 3.2), together with the ε_0-DP property of the exponential mechanism.

Thus, we proceed with the utility guarantee. Let $f'(S) = f(S)/2$, so that $\phi'(S) = f'(S) + d(S)$. Consider any iteration $i \in [k]$, and let us condition on the set S_{i-1} selected after the first $i-1$ iterations. We have:

$$\begin{aligned}\sum_{u\in C_i} f'(u \mid S_{i-1}) &\geq f'(C_i \cup S_{i-1}) - f'(S_{i-1})\\ &= f'(\mathrm{OPT}\cup S_{i-1}) - f'(S_{i-1}) \geq f'(\mathrm{OPT}) - f'(S_{i-1}),\end{aligned}$$

where the first inequality holds by the submodularity of f', and the second by monotonicity. Moreover, by Lemma 5.3:

$$\sum_{u\in C_i} d(u \mid S_{i-1}) = d(C_i, S_{i-1}) \geq \frac{(i-1)|C_i|}{k(k-1)}d(\mathrm{OPT}).$$

Letting $\alpha = 4\varepsilon_0^{-1}\Delta\log n$, we obtain:

$$\begin{aligned}\mathbb{E}[\phi'(u_i \mid S_{i-1})] &\geq \max_{u\in V\setminus S_{i-1}} \phi'(u \mid S_{i-1}) - \alpha\\ &\geq \frac{1}{|C_i|}\left(\sum_{u\in C_i}\phi'(u \mid S_{i-1})\right) - \alpha\\ &\geq \frac{f'(\mathrm{OPT}) - f'(S_{i-1})}{k} + \frac{i-1}{k(k-1)}d(\mathrm{OPT}) - \alpha.\end{aligned}$$

The first inequality follows from the guarantees of the exponential mechanism, along with the fact that the quality functions q^i_D have sensitivity at most 2Δ. The second inequality holds since $C_i \subseteq V \setminus S_{i-1}$.

Removing the conditioning on S_{i-1} and taking the expectation over all its possible realizations, the last inequality yields:

$$\mathbb{E}[\phi'(u_i)] \geq \frac{f'(\mathrm{OPT}) - \mathbb{E}[f'(S_k)]}{k} + \frac{i-1}{k(k-1)} d(\mathrm{OPT}) - \alpha,$$

where we have used the fact that $\mathbb{E}[f'(S_{i-1})] \leq \mathbb{E}[f'(S_k)]$ due to the monotonicity of f'. Summing over all $i \in [k]$, we get:

$$\mathbb{E}[\phi'(S_k)] \geq f'(\mathrm{OPT}) - \mathbb{E}[f'(S_k)] + \frac{1}{2} d(\mathrm{OPT}) - k\alpha.$$

Plugging in $\phi' = f/2 + d$ and $f' = f/2$, we obtain:

$$\frac{1}{2}\mathbb{E}[f(S_k)] + \mathbb{E}[d(S_k)] \geq \frac{f(\mathrm{OPT}) - \mathbb{E}[f(S_k)] + d(\mathrm{OPT})}{2} - k\alpha.$$

Rearranging terms gives $\mathbb{E}[\phi(S_k)] \geq \frac{1}{2}\phi(\mathrm{OPT}) - k\alpha$, which completes the proof. □

6 FASTER GREEDY ALGORITHM VIA SUBSET SAMPLING

In this section, we present our main contribution for the cardinality-constraint setting. The Sample Greedy algorithm for monotone submodular maximization, independently suggested by Buchbinder et al. [10] and Mirzasoleiman et al. [42], achieves a $(1 - 1/e - \gamma)$-approximation in expectation using only $O(n \log \gamma^{-1})$ oracle calls. Mitrovic et al. [43] proposed a variant for non-monotone submodular functions, guaranteeing a $(1 - 1/e)/e$-approximation, and adapted it to satisfy differential privacy by replacing greedy selections with randomized ones. However, these analyses rely on submodularity, which does not hold in our setting.

Nevertheless, we propose a Sample Greedy algorithm for DP MSD under cardinality constraints, stated as Algorithm 2. To the best of our knowledge, this is the fastest known algorithm for MSD with cardinality constraints, which is of interest even without privacy. Experimental results (Section 8) show that our method significantly improves running time over `DP-Greedy` while preserving utility. We further extend the analysis of Gupta et al. [33] to account for the subsampling step and obtain improved privacy guarantees for decomposable objectives.

Algorithm 2 performs greedy selections with respect to a function ϕ' that can be either the objective ϕ or an auxiliary function, and a utility function $g\colon [k] \to \mathbb{N}$, which determines the sample size in each iteration. We study two instantiations, each corresponding to a different choice of g and ϕ', yielding distinct trade-offs between running time and approximation guarantees.

6.1 Non-Oblivious Sample Greedy

The first instantiation we consider, referred to as `DP-NOSG`, selects the next element according to the auxiliary function $\phi'_D : 2^V \to \mathbb{R}$ given by $\phi'_D(S) = \frac{1}{2-\gamma} \cdot f_D(S) + d_D(S)$, and uses the utility function $g(i) = k - i + 1$ for all $i \in [k]$. We obtain the following result.

Theorem 6.1. *Suppose ϕ_D has sensitivity Δ. Then,* `DP-NOSG` *(Algorithm 2) with parameters $\varepsilon_0 > 0$ and $\gamma \in (0,1)$ is $k\varepsilon_0$-DP and (ε, δ)-DP for all $\delta > 0$, where $\varepsilon = \sqrt{2k \log(1/\delta)}\,\varepsilon_0 + k\varepsilon_0(e^{\varepsilon_0} - 1)$. Moreover, it outputs a set S such that $\mathbb{E}[\phi_D(S)] \geq (\frac{1}{2} - \gamma) \cdot \phi_D(\mathrm{OPT}) - O\Big(\frac{k\Delta \log n}{\varepsilon_0}\Big)$ and makes $O(n \log k \log \gamma^{-1})$ oracle calls.*

Algorithm 2 `DP-NOSG` (DP Non-Oblivious Sample Greedy) / `DP-OSG` (DP Oblivious Sample Greedy)

```
Input: Dataset $D$, ground set $V$, submodular $f_D$, pseudometric $d_D$,
    cardinality $k$, privacy parameter $\varepsilon_0$, utility parameter $\gamma$.
Output: Size $k$ subset of $V$
In DP-NOSG:
    Let $\phi'_D = \frac{1}{2-\gamma} f_D + d_D$ and $g(i) = k - i + 1$ for all $i \in [k]$
In DP-OSG:
    Let $\phi'_D = f_D + d_D$ and $g(i) = \min\{k, n - i + 1\}$ for all $i \in [k]$
Set $S_0 \leftarrow \emptyset$
for $i = 1, \ldots, k$ do
    Set $N_i \leftarrow V \setminus S_{i-1}$
    Sample a subset $V_i \subseteq N_i$ of size $\left\lceil |N_i| \cdot \min\left\{\frac{\log(1/\gamma)}{g(i)}, 1\right\}\right\rceil$
    uniformly at random.
    For all $u \in V_i$, define $q^i_D(u) = \phi'_D(u \mid S_{i-1})$
    Compute $u_i \leftarrow \mathrm{EM}(D, q^i_D, V_i, \varepsilon_0)$
    $S_i \leftarrow S_{i-1} \cup \{u_i\}$
end for
return $S_k$
```

By replacing the exponential mechanism with the true maximum, we get a faster, non-private algorithm for MSD, with the following guarantees.

Corollary 6.2. *Instantiated with* MAX *instead of* `EM`, `DP-NOSG` *outputs a set S such that $\mathbb{E}[\phi(S)] \geq (\frac{1}{2} - \gamma)\phi(\mathrm{OPT})$ and makes $O(n \log k \log \gamma^{-1})$ oracle calls.*

The next theorem states the improved privacy guarantees for decomposable objective functions. Note that under our definition of ϕ', if ϕ is 1-decomposable, then so is ϕ'.

Theorem 6.3. *Suppose $\phi_D : 2^V \to \mathbb{R}$ is 1-decomposable. Then given parameter $\varepsilon_0 \in (0,1]$, Algorithm 2 is (ε, δ)-DP for every $\delta > 0$ and $\varepsilon = (e^{\varepsilon_0/2} - 1)(4 + \log(1/\delta))$.*

Theorems 6.1 and 6.3 correspond to row *(ii)* in Table 1. As in the previous section, Theorem 5.2 implies that for decomposable objectives, setting $\varepsilon_0 \approx \varepsilon$ suffices to ensure (ε, δ)-DP. Substituting this into the utility guarantee of Theorem 6.1 yields the stated additive error in Table 1.

Before proceeding to the proof, we introduce additional notation. Let OPT be an optimal solution and S_i the set obtained at the end of iteration i. Recall that $N_i = V \setminus S_{i-1}$ contains all unselected elements after iteration $i - 1$, and $C_i = \mathrm{OPT} \setminus S_{i-1}$ denotes the subset of these elements belonging to OPT. Let $M_i \subseteq N_i$ be a set of size $g(i)$ maximizing $\sum_{v \in M_i} \phi'(v \mid S_{i-1})$. That is, M_i consists of the $g(i)$ elements with the largest marginal contributions during iteration i relative to the auxiliary function ϕ'.

Lemma 6.4. *For every iteration $i \in [k]$, conditioned on having selected a set S_{i-1} after the first $i - 1$ iterations, the following holds. $\mathbb{E}[\phi'(u_i \mid S_{i-1})] \geq \frac{1-\gamma}{|M_i|} \sum_{v \in M_i} \phi'(v \mid S_{i-1}) - \frac{4\Delta \log n}{\varepsilon_0}$*

Proof. Let $\alpha = 4\varepsilon_0^{-1}\Delta \log n$ and $i \in [k]$. Condition on a realization of S_{i-1} and V_i. By Theorem 3.5 and the 2Δ sensitivity bound for q_D^i, we have

$$\mathbb{E}\left[\phi'(u_i \mid S_{i-1})\right] \geq \max_{v \in V_i} \phi'(v \mid S_{i-1}) - \alpha \geq \max_{v \in V_i \cap M_i} \phi'(v \mid S_{i-1}) - \alpha$$
$$\geq \sum_{v \in V_i \cap M_i} \tfrac{\phi'(v|S_{i-1})}{|V_i \cap M_i|} - \alpha.$$

Note that second inequality holds even if $V_i \cap M_i = \emptyset$ because ϕ' is monotone and marginal gains are nonnegative. Removing the conditioning on V_i and taking the expectation over all its realizations gives

$$\begin{aligned}\mathbb{E}[\phi'(u_i \mid S_{i-1})] &\geq \mathbb{E}_{V_i}\Big[\sum_{v \in V_i \cap M_i} \tfrac{\phi'(v|S_{i-1})}{|V_i \cap M_i|}\Big] - \alpha \\ &\geq \tfrac{1-\gamma}{|M_i|} \sum_{v \in M_i} \phi'(v \mid S_{i-1}) - \alpha.\end{aligned} \tag{1}$$

To prove the last inequality, let G be the event $V_i \cap M_i \neq \emptyset$. If $\log(1/\gamma) \geq g(i)$, then $\mathbf{Pr}[G] = 1$ holds. Otherwise, the sampling in Line 6 ensures

$$\mathbf{Pr}[G] = \mathbf{Pr}[V_i \cap M_i \neq \emptyset] \geq 1 - \left(1 - \tfrac{g(i)}{|N_i|}\right)^{\frac{|N_i|\log(1/\gamma)}{g(i)}} \geq 1 - \gamma.$$

Let p_v be the probability that $v \in M_i$ appears in V_i conditioned on G. By the law of total expectation:

$$\begin{aligned}\mathbb{E}_{V_i}\Big[\sum_{v \in V_i \cap M_i} \tfrac{\phi'(v|S_{i-1})}{|V_i \cap M_i|}\Big] &= \mathbf{Pr}[G] \cdot \sum_{v \in M_i} p_v \phi'(v \mid S_{i-1}) \\ &\geq \tfrac{1-\gamma}{|M_i|} \sum_{v \in M_i} \phi'(v \mid S_{i-1})\end{aligned}$$

where the first equality uses the fact that the inner sum is zero conditioned on $\overline{G}$. Because V_i is chosen uniformly, p_v is identical for all $v \in M_i$, thus $p_v = 1/|M_i|$. This yields (1). □

We are now ready to complete the proof of Theorem 6.1.

Proof of Theorem 6.1. The complexity bound holds due to the fact that DP-NOSG evaluates $O(\frac{n}{k-i}\log\gamma^{-1})$ candidates in iteration i, the number of oracle calls across k iterations is $O(n \log k \log \gamma^{-1})$ using the standard Harmonic sum approximation. The full argument is deferred to Section C. Privacy follows from the composition theorems (Theorem 3.2) and the ε_0-DP guarantee of the exponential mechanism. We thus focus on the utility guarantee.

Let $f'(S) = \frac{1}{2-\gamma} f(S)$ and $\phi'(S) = f'(S) + d(S)$. Define $\alpha = 4\varepsilon_0^{-1}\Delta \log n$. Condition on having selected S_{i-1} prior to iteration $i \in [k]$. By Lemma 6.4, we have

$$\begin{aligned}\mathbb{E}[\phi'(u_i \mid S_{i-1})] &\geq \tfrac{1-\gamma}{|M_i|} \sum_{v \in M_i} \phi'(v \mid S_{i-1}) - \alpha \\ &\geq \tfrac{1-\gamma}{|C_i|} \sum_{v \in C_i} \phi'(v \mid S_{i-1}) - \alpha \\ &\geq (1-\gamma)\big(\tfrac{f'(\mathrm{OPT}) - f'(S_{i-1})}{k} + \tfrac{i-1}{k(k-1)} d(\mathrm{OPT})\big) - \alpha.\end{aligned}$$

The second inequality holds because $|M_i| = k - i + 1 \leq |C_i|$, and M_i contains the elements with the largest marginal gains. The third follows from submodularity of f' and Lemma 5.3 for d. Taking an expectation over S_{i-1} gives

$$\mathbb{E}[\phi'(u_i \mid S_{i-1})] \geq (1-\gamma)\left(\tfrac{f'(\mathrm{OPT}) - \mathbb{E}[f'(S_k)]}{k} + \tfrac{i-1}{k(k-1)} d(\mathrm{OPT})\right) - \alpha.$$

where we use the monotonicity of f' and the fact that $S_{i-1} \subseteq S_k$ to bound $\mathbb{E}[f'(S_k)] \geq \mathbb{E}[f'(S_{i-1})]$. Summing over $i \in [k]$ yields

$$\mathbb{E}[\phi'(S_k)] \geq (1-\gamma)(f'(\mathrm{OPT}) - \mathbb{E}[f'(S_k)] + \tfrac{1}{2} d(\mathrm{OPT})) - k\alpha.$$

Rearranging terms and substituting f' gives:

$$\mathbb{E}[\phi(S_k)] \geq (1-\gamma)\left(\tfrac{f(\mathrm{OPT})}{2-\gamma} + \tfrac{d(\mathrm{OPT})}{2}\right) - k\alpha \geq (\tfrac{1}{2} - \gamma)\phi(\mathrm{OPT}) - k\alpha.$$

□

Next, we address the stronger guarantee of Theorem 6.3, which applies when the function ϕ is decomposable. Intuitively, we show that for any (possibly adaptive) choice of the sets V_i from which the exponential mechanism selects the next element, the distributions over outputs under two neighboring datasets remain close. Hence, the distributions are also close on average, corresponding to the actual distribution over outputs induced by the algorithm. We provide a proof sketch, with the full proof given in Section C.

Proof of Theorem 6.3. Let D and D' be neighboring datasets such that $(D \setminus D') \cup (D' \setminus D) = \{x\}$. Suppose that instead of a set, Algorithm 2 outputs the sequence of selected elements in their order of selection. Let $U = (u_1, \ldots, u_k)$ be any such sequence, and $U_i = \{u_1, \ldots, u_i\}$ denote the prefix set consisting of the first i elements.

For any element $u \in V$, let $\beta_D^i(u) = \sum_{x \in D} \phi'_x(u \mid U_{i-1})$. By the definition of the exponential mechanism, the probability of selecting element u at iteration i, given dataset D and a sampled candidate set V_i, is:

$$\mathbf{Pr}[\mathsf{EM}(q_i, V_i, D, \varepsilon_0) = u] = \frac{\exp\left(\frac{\varepsilon_0}{2}\beta_D^i(u)\right)}{\sum_{u' \in V_i} \exp\left(\frac{\varepsilon_0}{2}\beta_D^i(u')\right)}.$$

where we have used that since ϕ'_D is 1-decomposable, it has a sensitivity $1/|D|$. The expression for D' follows analogously. Let $\mathcal{G}$ denote Algorithm 2. By the chain rule, the probability of $\mathcal{G}$ outputting sequence U is:

$$\mathbf{Pr}[\mathcal{G}(D) = (u_1, \ldots, u_k)] = \prod_{i=1}^{k} \mathbf{Pr}[\mathcal{G}(D)_i = u_i \mid S_{i-1} = U_{i-1}].$$

Let $\mathcal{V}_i$ denote the collection of all subsets of N_i of the size specified in Line 6. Since V_i is sampled uniformly at random from $\mathcal{V}_i$, for every $T \in \mathcal{V}_i$, we have $\mathbf{Pr}[V_i = T] = p_i$, where $p_i = 1/|\mathcal{V}_i|$ is independent of the dataset D. By the law of total probability, the i-th factor in the product above is:

$$\mathbf{Pr}[\mathcal{G}(D)_i = u_i \mid S_{i-1} = U_{i-1}] = \sum_{T \in \mathcal{V}_i : u_i \in T} \mathbf{Pr}[\mathsf{EM}(q_i, T, D, \varepsilon_0) = u_i] \cdot p_i.$$

Note that the selection probability is zero if $u_i \notin V_i$. Using the fact that $\frac{\sum a_j}{\sum b_j} \leq \max \frac{a_j}{b_j}$ for $a_j \geq 0$ and $b_j > 0$, we have:

$$\begin{aligned}\frac{\mathbf{Pr}[\mathcal{G}(D)_i = u_i \mid S_{i-1} = U_{i-1}]}{\mathbf{Pr}[\mathcal{G}(D')_i = u_i \mid S_{i-1} = U_{i-1}]} &= \frac{\sum_{T \in \mathcal{V}_i, u_i \in T} \mathbf{Pr}[\mathsf{EM}(q_i, T, D, \varepsilon_0) = u_i]}{\sum_{T \in \mathcal{V}_i, u_i \in T} \mathbf{Pr}[\mathsf{EM}(q_i, T, D', \varepsilon_0) = u_i]} \\ &\leq \max_{T \in \mathcal{V}_i : u_i \in T} \left[\frac{\mathbf{Pr}[\mathsf{EM}(q_i, T, D, \varepsilon_0) = u_i]}{\mathbf{Pr}[\mathsf{EM}(q_i, T, D', \varepsilon_0) = u_i]}\right].\end{aligned}$$

Let T_i be the candidate set attaining the maximum above. Taking the product over all iterations k:

$$\frac{\mathbf{Pr}[\mathcal{G}(D) = U]}{\mathbf{Pr}[\mathcal{G}(D') = U]} \leq \left(\prod_{i=1}^{k} \frac{\exp(\frac{\varepsilon_0}{2}\beta_D^i(u_i))}{\exp(\frac{\varepsilon_0}{2}\beta_{D'}^i(u_i))} \right) \left(\prod_{i=1}^{k} \frac{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2}\beta_{D'}^i(u))}{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2}\beta_D^i(u))} \right). \tag{2}$$

We consider two cases.

Case 1: $D = D' \cup \{x\}$**.** The first factor in (2) is bounded by

$$\exp\left(\frac{\varepsilon_0}{2} \sum_{i=1}^{k} \phi_x'(u_i \mid U_{i-1}) \right) \leq \exp(\tfrac{\varepsilon_0}{2})$$

due to 1-decomposability. The second factor is at most 1 since $\beta_D^i(u) \geq \beta_{D'}^i(u)$.

Case 2: $D' = D \cup \{x\}$**.** The first factor is at most 1, while the second factor becomes:

$$\prod_{i=1}^{k} \frac{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2}\phi_x'(u \mid U_{i-1})) \exp(\frac{\varepsilon_0}{2}\beta_D^i(u))}{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2}\beta_D^i(u))} \tag{3}$$

$$= \prod_{i=1}^{k} \mathbb{E}_{u \sim P_i} \left[\exp\left(\tfrac{\varepsilon_0}{2}\phi_x'(u \mid U_{i-1}) \right) \right], \tag{4}$$

where P_i is a distribution supported on T_i with weights proportional to $\exp(\frac{\varepsilon_0}{2}\beta_D^i(u))$. To bound this product of expectations, we apply the concentration bound from [16, 33]. By Lemma C.3, with probability at least $1 - \delta$:

$$\frac{\mathbf{Pr}[\mathcal{G}(D) = U]}{\mathbf{Pr}[\mathcal{G}(D') = U]} \leq \exp\left((e^{\varepsilon_0/2} - 1)(3 + \log(1/\delta)) \right).$$

Combining both cases, for any neighboring D, D' (under the change-one-element definition), with probability at least $1 - \delta$:

$$\frac{\mathbf{Pr}[\mathcal{G}(D) = U]}{\mathbf{Pr}[\mathcal{G}(D') = U]} \leq \exp\left(\frac{\varepsilon_0}{2} + (e^{\varepsilon_0/2} - 1)(3 + \log(1/\delta)) \right)$$
$$\leq \exp\left((e^{\varepsilon_0/2} - 1)(4 + \log(1/\delta)) \right).$$

Thus, Algorithm 2 is (ε, δ)-DP with $\varepsilon = (e^{\varepsilon_0/2} - 1)(4 + \log(1/\delta))$. □

So far, we have analyzed DP-NOSG, an instantiation of Algorithm 2 that makes $O_\gamma(n \log k)$ oracle calls. In the following section, we introduce a linear-time algorithm where the number of oracle calls is independent of k.

6.2 Oblivious Sample Greedy

For submodular objectives, the sample greedy method can eliminate the dependence on k while maintaining a constant-factor approximation. However, such results have not been established for the non-submodular MSD objective. We present DP-OSG, an efficient algorithm for DP MSD that makes only $O_\gamma(n)$ oracle calls, albeit with a weaker approximation factor. Our experiments in Section 8 demonstrate that DP-OSG achieves high utility under tight privacy constraints. Furthermore, it is significantly more scalable than DP-Greedy, making it better suited for interactive settings.

DP-OSG uses the utility function $g(i) = \min\{k, n - i + 1\}$ for all $i \in [k]$. A key component in the analysis of DP-NOSG is Lemma 5.3, which lower bounds the total distance between the current solution and the set of remaining unselected elements of OPT by a function of $d(\mathrm{OPT})$. While this technique yields tight guarantees for DP-Greedy and DP-NOSG, applying an analogous analysis with the current choice of g leads to a weaker approximation ratio of $1/6 - \gamma$.

Thus, we take a different approach by instantiating Algorithm 2 with $\phi' = \phi$. That is, in an *oblivious* manner. Moreover, instead of using Lemma 5.3, we bound the total distance between the current solution and the remaining unselected elements of OPT by a function of the marginal gain $d(\mathrm{OPT} \cup S_i) - d(S_i)$ (Lemma 6.7).

One can verify that the privacy guarantees stated in Theorems 6.1 and 6.3 hold for Algorithm 2 with the current choice of g and ϕ'.

Theorem 6.5. *Suppose ϕ has sensitivity Δ. Then,* DP-OSG *(Algorithm 2) outputs a set S such that $\mathbb{E}[\phi(S)] \geq (1 - (\frac{2}{e})^{(1-\gamma)(1-1/k)}) \cdot \phi(\mathrm{OPT}) - O(\frac{k\Delta \log n}{\varepsilon_0})$ and makes $O(n \log \gamma^{-1})$ oracle calls.*

Lemma C.4 shows that the approximation factor is greater than $1 - 2/e - \gamma - 1/k$, and we use this linearized form for simplicity. Theorem 6.5, together with the privacy guarantees in Theorems 6.1 and 6.3 correspond to row *(iii)* in Table 1.

By replacing the exponential mechanism with the true maximum, we get the first linear-time, non-private algorithm with a constant factor approximation.

Corollary 6.6. *Instantiated with* MAX *instead of* EM, DP-OSG *outputs a set S such that $\mathbb{E}[\phi(S)] \geq (1 - (\frac{2}{e})^{(1-\gamma)(1-1/k)}) \cdot \phi(\mathrm{OPT})$, and makes $O(n \log \gamma^{-1})$ oracle queries.*

Lemma 6.7. *For every iteration $i \geq 2$,*

$$d(\mathrm{OPT} \cup S_{i-1}) - d(S_{i-1}) \leq \left(1 + \frac{k-1}{i-1} \right) \cdot d(C_i, S_{i-1}).$$

Proof of Theorem 6.5. The proof of the query complexity is provided in Lemma C.1. We thus focus on the utility guarantee. Let $\alpha = 4\Delta \log(n)/\varepsilon_0$. Consider any iteration $2 \leq i \leq k$ and condition on the set S_{i-1} selected after iteration $i - 1$. By the submodularity and monotonicity of f, we have:

$$\sum_{u \in C_i} f(u \mid S_{i-1}) \geq f(\mathrm{OPT} \cup S_{i-1}) - f(S_{i-1}) \geq f(\mathrm{OPT}) - f(S_{i-1}).$$

Applying Lemma 6.7, we find that:

$$\sum_{v \in C_i} d(v \mid S_{i-1}) = d(C_i, S_{i-1}) \geq \frac{d(\mathrm{OPT} \cup S_{i-1}) - d(S_{i-1})}{1 + \frac{k-1}{i-1}}.$$

Let $M_i \subseteq N_i$ be a set of size $g(i) = \min\{k, |N_i|\}$ that maximizes $\sum_{v \in M_i} \phi(v \mid S_{i-1})$. Note that M_i comprises the k elements with the largest marginal contributions in iteration i, or is equal to N_i if $|N_i| \leq k$. We have:

$$\mathbb{E}[\phi(u_i \mid S_{i-1})] \geq \tfrac{1-\gamma}{g(i)} \sum_{v \in M_i} \phi(v \mid S_{i-1}) - \alpha \geq \tfrac{1-\gamma}{k} \sum_{v \in C_i} \phi(v \mid S_{i-1}) - \alpha$$
$$\geq (1 - \gamma) \left(\frac{\phi(\mathrm{OPT}) - \phi(S_{i-1})}{k(1 + \frac{k-1}{i-1})} \right) - \alpha.$$

The first inequality follows from the guarantees of the exponential mechanism and the sampling procedure, analogous to the proof of Lemma 6.4. For the second inequality, recall that $C_i \subseteq N_i$, and since $C_i \subseteq \mathrm{OPT}$, it holds that $|C_i| \leq k$. Thus, the inequality follows from the definition of M_i, together with the fact that $|C_i| \leq g(i) \leq k$.

Removing the conditioning on S_{i-1} and taking the expectation over all possible realizations yields:

$$\mathbb{E}[\phi(u_i)] \geq (1-\gamma)\left(\frac{\phi(\mathrm{OPT}) - \mathbb{E}[\phi(S_{i-1})]}{k(1+\frac{k-1}{i-1})}\right) - \alpha.$$

Rearranging terms, we obtain:

$$\phi(\mathrm{OPT}) - \mathbb{E}[\phi(S_i)] \leq \left(1 - \frac{1-\gamma}{k(1+\frac{k-1}{i-1})}\right)(\phi(\mathrm{OPT}) - \mathbb{E}[\phi(S_{i-1})]) + \alpha.$$

By recursively applying this inequality, we get:

$$\begin{aligned}\phi(\mathrm{OPT}) - \mathbb{E}[\phi(S_k)] &\leq \prod_{i=2}^{k}\left(1 - \frac{1-\gamma}{k(1+\frac{k-1}{i-1})}\right)\phi(\mathrm{OPT}) + k\alpha \\ &\leq \left(\frac{2}{e}\right)^{(1-\gamma)(1-1/k)} \phi(\mathrm{OPT}) + k\alpha,\end{aligned}$$

where the last inequality follows from Lemma C.4. The theorem follows by rearranging the final expression. □

7 LOCAL SEARCH ALGORITHM FOR MATROID CONSTRAINTS

In this section, we present our algorithm for matroid constraints. Borodin et al. [8] proposed a non-private local search algorithm achieving a $(1/2-\gamma)$-approximation using $O(n^2 + \gamma^{-1}nk^2 \log k)$ oracle calls.[3] Their algorithm begins with a base containing the highest-scoring independent set of size two and repeatedly performs improving swaps: replacing one element in the current solution with an outside element, provided independence is maintained and the objective strictly increases. The process terminates at a locally optimal solution. However, adapting this algorithm to satisfy differential privacy is not straightforward, as under the additive DP noise we cannot check for local optimality or ensure the selection of improving swaps.

Our algorithm, DP-SLS (Algorithm 3), uses the exponential mechanism to privately select an approximately-best available swap for a fixed number of iterations. To avoid forced suboptimal swaps, we include a dummy swap that allows the current solution to remain unchanged. We show that whenever the expected objective value in an iteration is low, the subsequent iteration increases it significantly. While the initial base S_0 is arbitrary, the first iteration ensures the expectation is immediately high enough to drive the solution toward a $(1/2-\gamma)$ approximation. Finally, the exponential mechanism is used one last time to output the best solution observed across all iterations.

Our analysis removes the $O(n^2)$ term in the complexity of [8]. We further reduce complexity by subsampling swaps in each iteration, obtaining $O(\gamma^{-1}nk\log k)$ oracle calls, a significant improvement over prior work. Our main result in this section is stated next.

Theorem 7.1. *Suppose $\phi_D : 2^V$ has sensitivity Δ, and let $\mathcal{M}$ be a matroid of rank k. For $T' = O(\gamma^{-1}k\log k)$,* DP-SLS *(Algorithm 3) with parameters $\varepsilon_0 > 0$ and $\gamma \in (0,1)$ is $T'\varepsilon_0$-DP, and (ε,δ)-DP for every $\delta > 0$, with $\varepsilon = \sqrt{2T'\log(1/\delta)}\varepsilon_0 + T'\varepsilon_0(e^{\varepsilon_0}-1)$. Moreover, it outputs a set S such that $\mathbb{E}[\phi_D(S)] \geq \left(\frac{1}{2}-\gamma\right)\cdot\phi_D(\mathrm{OPT}) - O\left(\frac{\Delta}{\varepsilon_0}\left(k\log n + \log\frac{1}{\gamma}\right)\right)$, and makes $O(\gamma^{-1}nk\log k)$ oracle calls.*

Theorem 7.1 corresponds to row *(iv)* in Table 1. The stated additive error in Table 1 is obtained using advanced composition (Theorem 3.2), which intuitively implies that setting $\varepsilon_0 \approx \varepsilon/\sqrt{\gamma^{-1}k\log k}$ (omitting polylogarithmic factors in $1/\delta$) ensures (ε,δ)-DP. Note that we now have $O(\gamma^{-1}k\log k)$ compositions. By replacing the exponential mechanism with the true maximum, we get a faster, non-private algorithm for the MSD problem with matroid constraints, for which the expected approximation guarantee is the same as that of the deterministic local search algorithm of [8].

Corollary 7.2. *Instantiated with* MAX *instead of* EM, DP-SLS *outputs a set S such that $\mathbb{E}[\phi(S)] \geq \left(\frac{1}{2}-\gamma\right)\cdot\phi(\mathrm{OPT})$ and makes $O(\gamma^{-1}nk\log k)$ oracle calls.*

Remark 7.3. In this section, we consider the broader class of Δ-sensitive functions. The technique of Gupta et al. [33] for decomposable functions relies crucially on the monotonic growth of the greedy solution sequence. Extending this approach to local-search algorithms, where elements are both added and removed, appears nontrivial and remains an interesting direction for future work.

Algorithm 3 DP Sample Local Search (DP-SLS)

```
Input: Dataset $D$, matroid $\mathcal{M}$ of rank $k$, privacy parameter $\varepsilon_0$,
    utility parameter $\gamma \in (0,1]$.
Let $S_0$ be an arbitrary base of $\mathcal{M}$.
Set $T \leftarrow \lceil \frac{2k\log(8k)}{\gamma(1-1/e)} \rceil + 1$
for $i = 1,\dots,T$ do
    Sample a subset $V_i \subseteq V$ of size $\lceil \frac{n}{k} \rceil$ uniformly at random
    Let $W_i \leftarrow \{(u,v) \in S_{i-1} \times (V_i \setminus S_{i-1}) \mid S_{i-1} - u + v \in \mathcal{I}\} \cup \{(w,w)\}$ for an arbitrary element $w \in S_{i-1}$.
    For all $(u,v) \in W_i$, define $q_D^i(u,v) = \phi_D(S_{i-1} - u + v)$.
    Compute $(u_i, v_i) \leftarrow \mathsf{EM}(D, q_D^i, W_i, \varepsilon_0)$
    Let $S_i \leftarrow S_{i-1} - u_i + v_i$
end for
For all $i = 1,\dots,T$, define $s_D(i) = \phi_D(S_i)$.
Compute $i^* \leftarrow \mathsf{EM}(D, s_D, [T], \varepsilon_0)$
return $S_{i^*}$
```

For the remainder of this section we prove Theorem 7.1. For simplicity, we assume throughout that the $\mathcal{M}$ has rank $k \geq 3$, and address the case $k = 2$ in Section E.

Given a set S and elements u, v, we use $S + u$, $S - u$ and $S - u + v$ as shorthands for $S \cup \{u\}$, $S \setminus \{u\}$ and $(S \setminus \{u\}) \cup \{v\}$ respectively. The following lemma is a known property of matroids.

Lemma 7.4 ([9]). *Let X and Y be two bases of a matroid $\mathcal{M} = (V, \mathcal{I})$. Then, there exists a bijection $h : X \to Y$ such that for every $u \in X$, $Y - h(u) + u \in \mathcal{I}$, and $h(u) = u$ for every $u \in X \cap Y$.*

Let OPT be an optimal solution. Since ϕ is monotone, we may assume, without loss of generality, that OPT is a base of $\mathcal{M}$. For $i \in [T]$, let S_{i-1} denote the base obtained after iteration $i-1$.

[3] The algorithm in [8] attains a 1/2-approximation but is not polynomial-time. The authors' proposal to perform only swaps that yield at least a γ-improvement at each iteration, rather than any improvement, leads to the stated approximation factor and complexity.

Lemma 7.5. *Suppose $\mathcal{M}$ has rank $k > 2$. Then,*

$$\sum_{u \in S_{i-1}} \phi(S_{i-1} - u + h(u)) \geq \phi(\text{OPT}) + (k-2)\phi(S_{i-1})$$

Using on this lemma, we establish the following result, characterizing the algorithm's progress.

Lemma 7.6. *For every iteration $i \in [T]$, we have*

$$\mathbb{E}\left[\phi(S_i) - \phi(S_{i-1})\right] \geq \tfrac{1-1/e}{k}\left(\phi(\text{OPT}) - 2\mathbb{E}\left[\phi(S_{i-1})\right]\right) - \tfrac{4\Delta \log(n)}{\varepsilon_0}$$

Intuitively, Lemma 7.5 implies that after obtaining a base S_{i-1}, a uniformly random swap from the set $M_i = \{(u, h(u)) : u \in S_{i-1}\}$ yields an expected gain of $\frac{1}{k}(\phi(\text{OPT}) - 2\phi(S_{i-1}))$. To prove Lemma 7.6, we show that the set of swaps considered at iteration i intersects M_i with probability at least $1 - 1/e$. Conditioned on this event, the swap selected by the exponential mechanism achieves, up to an additive error, the maximal gain available, which in turn is at least as large as the average gain of a swap in M_i. Combining these observations establishes the lemma. The full argument, which also handles potential negative gains, is presented in Section D.

Proof of Theorem 7.1. The privacy guarantee follows directly from the ε_0-DP of the exponential mechanism and the composition theorems (Theorem 3.2), noting that we now have $T' \triangleq T + 1 = O(\gamma^{-1} k \log k)$ compositions. For the query complexity, Algorithm 3 performs T iterations, and each iterations requires only $O(|S_{i-1}| \cdot |V_i|) = O(n)$ queries due to the sub-sampling step. Overall, the algorithm performs $O(\gamma^{-1} nk \log k)$ queries. We proceed to prove the utility guarantee. Let $\alpha = 4\Delta \log(n)/\varepsilon_0$. We may assume that $\frac{1-1/e}{k} \cdot \phi(\text{OPT}) - \alpha \geq \frac{\phi(\text{OPT})}{8k}$, since otherwise, it can be verified that $\phi(\text{OPT}) < 2k\alpha$, and the guarantees of Theorem 7.1 hold trivially. Let S_0 be the first base chosen by the algorithm. We have

$$\begin{aligned}\mathbb{E}\left[\phi(S_1)\right] &\geq \phi(S_0) + \frac{1-1/e}{k}\left(\phi(\text{OPT}) - 2\phi(S_0)\right) - \alpha \\ &\geq \frac{1-1/e}{k} \cdot \phi(\text{OPT}) - \alpha \geq \frac{\phi(\text{OPT})}{8k}\end{aligned}$$

where the first inequality holds by Lemma 7.6, and the second since $k \geq 2$. Now, fix an iteration $2 \leq i \leq T$, and let us condition on having selected some base S_{i-1} after the first $i-1$ iterations. If $\mathbb{E}[\phi(S_{i-1})] < \phi(\text{OPT})/(2+\gamma) - k\alpha$, then, by Lemma 7.6 and since $\alpha \geq 0$, we obtain

$$\mathbb{E}\left[\phi(S_i)\right] \geq \left(1 + \frac{\gamma(1-1/e)}{k}\right)\mathbb{E}\left[\phi(S_{i-1})\right] \tag{5}$$

We claim that given our choice of T, there must exist some $i \in [T]$ for which $\mathbb{E}[\phi(S_i)] \geq \phi(\text{OPT})/(2+\gamma) - k\alpha$. Assume for contradiction that this is not the case. Recursively applying the bound (5) yields

$$\begin{aligned}\mathbb{E}\left[\phi(S_T)\right] &\geq \left(1 + \frac{\gamma(1-1/e)}{k}\right)^{\frac{2k\log(8k)}{\gamma(1-1/e)}}\mathbb{E}\left[\phi(S_1)\right] \\ &\geq \frac{1}{8k}\left(1 + \frac{\gamma(1-1/e)}{k}\right)^{\frac{2k\log(8k)}{\gamma(1-1/e)}}\phi(\text{OPT}) \\ &> \phi(\text{OPT}).\end{aligned}$$

The last inequality from the bound $(1 + 1/x)^x \geq \exp(1 - 1/(2x))$ which holds for all $x > 0$. In particular, there exists some base S with $\phi(S) > \phi(\text{OPT})$, in contradiction. Thus, by the law of total expectation, we get

$$\begin{aligned}\mathbb{E}\left[\phi(S_{i^*})\right] &= \mathbb{E}\left[\mathbb{E}\left[\phi(S_{i^*})\right] \mid S_1, \ldots, S_T\right] \geq \mathbb{E}\left[\max_{i \in [T]} \phi(S_i)\right] - \tfrac{2\Delta \log T}{\varepsilon_0} \\ &\geq \max_{i \in [T]} \mathbb{E}\left[\phi(S_i)\right] - \tfrac{2\Delta \log T}{\varepsilon_0} \geq \tfrac{\phi(\text{OPT})}{2+\gamma} - k\alpha - \tfrac{2\Delta \log T}{\varepsilon_0} \\ &= \tfrac{\phi(\text{OPT})}{2+\gamma} - \tfrac{2\Delta}{\varepsilon_0} \cdot \left(2k \log n + \log\left(\tfrac{2k \log k}{\gamma(1-1/e)}\right)\right)\end{aligned}$$

The second inequality holds by the fact that $\mathbb{E}\left[\max_i X_i\right] \geq \max_i \mathbb{E}[X_i]$ for any jointly distributed random variables $X_1, \ldots, X_n$. As $1/(2+\gamma) \geq 1/2 - \gamma$, the theorem follows. □

8 EXPERIMENTS

In this section, we evaluate the quality and efficiency of our algorithm in two concrete data summarization applications: Amazon product summarization and Uber pick-up location summarization We examine the performance of out algorithms to study the tradeoffs between privacy, utility, and complexity.

Summary of Findings. For cardinality constraints, our results demonstrate that the proposed DP algorithms achieve competitive utility compared to the non-private baseline while drastically improving efficiency. DP-Greedy and DP-NOSG stay within a marginal 2.7% of the non-private baseline even at a strict $\varepsilon = 0.14$ budget. By fundamentally shifting the number of oracle calls from linear to logarithmic or no dependence on k, DP-NOSG and DP-OSG achieve speedups of 5.4× and 8.3×, respectively at $k = 100$. DP-OSG reduces the number of oracle calls by a factor of 40.6 compared to Greedy. For matroid constraints, DP-SLS maintains highly competitive utility, staying within 1.3% of the non-private baseline for $k \leq 12$ and $\varepsilon = 0.1$. While DP-SLS reduces the total number of oracle calls, the fixed iteration count leads to a manageable 60% execution time overhead compared to the non-private baseline.

8.1 Experimental Settings

We next present our settings for the experiments. All algorithms were implemented[4] in Python 3.9.19 using the Pandas and NumPy libraries. All experiments were run on an Intel Xeon CPU-based server with 24 cores and 96 GB of RAM.

Default parameters. Unless mentioned otherwise, the following parameters are used. We set $\varepsilon = 0.2$ and $\delta = |D|^{-1.5}$ where D is the sensitive dataset. For cardinality constraints, our per-iteration privacy parameter ε_0 can be set with either basic composition, advanced composition (e.g., Theorem 5.1) or the analysis for decomposable objective (e.g., Theorem 5.2). We follow Chaturvedi et al. [16], Mitrovic et al. [43] and pick the best initialization. For the partition matroid constraint, we pick the best out of basic and advanced composition. We fix $\gamma = 0.1$ for the utility parameter, and by default use $\lambda = 0.1$. Results are averaged over 10 runs.

Algorithms. As MSD has not been previously explored in a differentially private setting, we adopt the evaluation framework of [43]. We compare our proposed methods against several non-private baselines and a randomized baseline. Specifically, we evaluate the following:

[4] The implementation can be found at https://github.com/ronzadi/Differentially-Private-Max-Sum-Diversification.

- `Greedy`: The standard non-private greedy algorithm for cardinality constraints [8].
- `DP-Greedy`: The private adaptation of `Greedy` (Algorithm 1), which replaces the deterministic selection step with the Exponential Mechanism.
- `DP-NOSG` (DP Non-Oblivious Sample Greedy): The non-oblivious instantiation of Algorithm 2 (Section 6.1). It employs the Exponential Mechanism for selection and incorporates a subsampling step to reduce query complexity.
- `DP-OSG` (DP Oblivious Sample Greedy): The oblivious instantiation of Algorithm 2 (Section 6.2). It further reduces complexity by sampling a smaller portion of the ground set.
- `LS`: The non-private local search algorithm for general matroid constraints [8] discussed in Section 7.
- `DP-SLS` (DP Sample Local Search): Our private local search algorithm (Algorithm 3), which performs a predefined number of iterations, Identifying the best available swap via the exponential mechanism and utilizes subsampling to reduce query complexity.
- `Random`: Selects a random feasible subset. Since the output distribution of the exponential mechanism converges to uniform as $\varepsilon \longrightarrow 0$, this baseline serves as a lower bound on utility for private algorithms. Following prior work [16, 17, 43], we incorporate `Random` to contextualize the performance of our algorithms.

8.2 Datasets, Objective Functions, and Constraints

We evaluate our approach on two real-world data summarization tasks.

Amazon Product Selection. The first application follows the scenario in Example 1.1 and focuses on selecting a representative summary of highly-purchased Amazon products to maximize user reach.

Dataset. We utilize the "Health and Household"subset of the Amazon Reviews 2023 dataset [37], which contains product metadata (e.g., categories, price) and user review data. We treat the metadata as the public ground set and the review/purchase history as the private dataset. Specifically, we select the 1,000 most-reviewed products from the "Health Care" category as our ground set V and discretize their prices into four bins. The associated purchase dataset D consists of 1,375,389 purchases made by 1,198,080 users.
Objective function. Following Example 1.1, the goal is to select a subset of products $S \subseteq V$ that maximizes user reach. Let $P(u) \subseteq D$ be the set of users who purchased item u. We define the relevance function as: $f_D(S) = \frac{1}{|D|} \left|\bigcup_{u \in S} P(u)\right|$. This coverage objective, frequently employed in summarization tasks (e.g., [19]), is a monotone, decomposable, and submodular function. To ensure selection diversity, we incorporate a distance metric $d_J(u, v) = 1 - \frac{|C(u) \cap C(v)|}{|C(u) \cup C(v)|}$ based on the Jaccard distance of the category sets $C(u)$ and $C(v)$ for products u and v. That is, $d(S) = \sum_{u,v \in S} d_J(u, v)$
Constraints. We consider two types of constraints: (i) a *uniform* matroid of rank k (i.e., a cardinality constraint) which we refer to as Amazon-Cardinality, and (ii) the intersection of a uniform matroid of rank k with a partition matroid that limits selections to $\lceil k/4 \rceil$ products per price bin. For brevity, we refer to the latter setting as Amazon-Partition.

Uber Location Selection. The second application involves selecting a representative summary of Uber pickup locations in Manhattan. This setup adapts the location selection experiments from [16, 43]. The goal is to select a set of locations from a public candidate set, for instance, to serve as waiting spots for idle drivers.

Dataset. Following [16, 43], we use a sensitive dataset D of 20,000 Uber pickup locations in Manhattan from 2014 [28]. Each record consists of longitude and latitude coordinates. For the public candidate set, we follow the established methodology and consider a grid of 200 points over Manhattan. As noted by [16], the density of the points creates an easy instance where random selection performs near-optimally. To increase the difficulty random selection while keeping the instance essentially the same for the other algorithms, we follow [16] and augment the candidate set with 800 redundant copies of the grid's northern corner, yielding a ground set of size 1,000.
Objective function. We adopt the relevance function from [16, 43]. For location points $p = (p_x, p_y)$ and $p' = (p'_x, p'_y)$, let $d_1(p, p') = \frac{|p_x - p'_x| + |p_y - p'_y|}{M}$ be the normalized ℓ_1 (Manhattan) distance, where M is an upper bound on the ℓ_1 distance within the area such that $d_1(p, p') \in [0, 1]$. The relevance of a set S is defined as:

$$f_D(S) = \frac{1}{|D|} \sum_{p \in D} (1 - \min_{l \in S} d_1(l, p))$$

with $f_D(\emptyset) = 0$. This function is monotone, submodular, and 1-decomposable. However, as observed by [44] for a similar formulation, this objective does not inherently promote geographic diversity. For example, suppose congestion slows down traffic in and out of a certain area. If all selected waiting locations are concentrated there, drivers may struggle to reach passengers efficiently. To address this, we add a diversity term defined by the normalized sum of ℓ_1 distances between the selected points: $d(S) = \sum_{l,l' \in S} d_1(l, l')$
Constraints. We evaluate this task under a cardinality constraint k.

8.3 Utility Analysis

We provide a detailed experimental analysis of our algorithms, examining how the selected subset size k and the privacy parameter ε impact the MSD objective value $(1 - \lambda) f_D(S) + \frac{2\lambda}{k(k-1)} d(S)$. By default, we set $k = 6$ for the Uber dataset and for Amazon-Partition, and use $k = 60$ for Amazon-Cardinality.

Impact of ε. We evaluate the impact of the privacy parameter ε on the resulting objective value (Figure 3). As expected, a larger privacy budget increases the utility of all DP algorithms. For Amazon-Cardinality at $k = 60$, `DP-Greedy`, `DP-NOSG`, and `DP-OSG` achieve utility comparable to the non-private `Greedy` baseline even with a small budget of $\varepsilon = 0.14$, reaching within 2.26%, 2.7%, and 9.3% below the `Greedy` results, respectively. While `DP-OSG` is naturally marginally lower in utility than the non-oblivious variants at higher ε, it in fact performs slightly better in the low ε regime. This is because its objective function is not distorted, allowing the relevance signal to remain stronger relative to the noise. All proposed methods significantly outperform the `Random` baseline. Similar trends

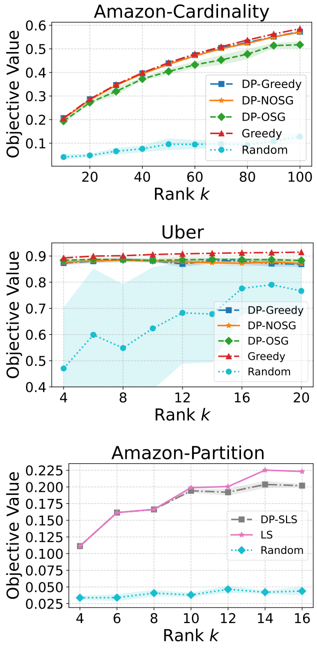

Figure 2: Objective value as the rank (maximal size of a feasible solution) k varies.

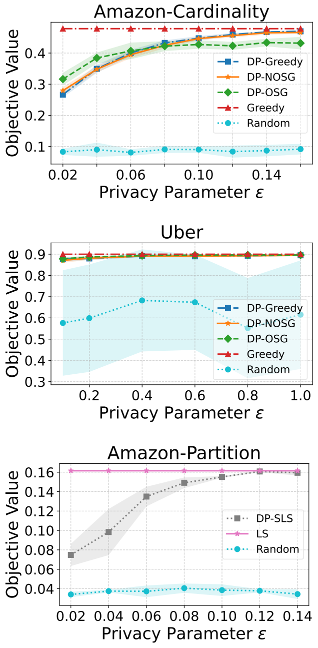

Figure 3: Objective value as the privacy parameter ε varies.

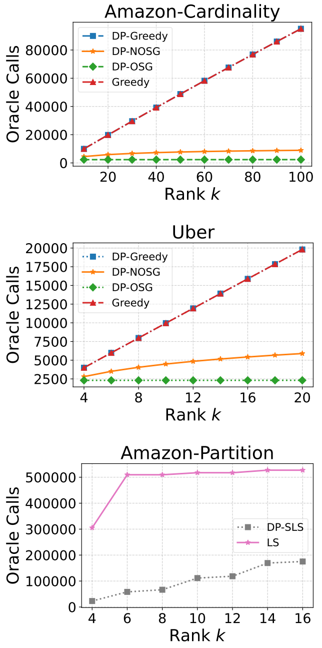

Figure 4: Number of oracle calls as the rank k varies.

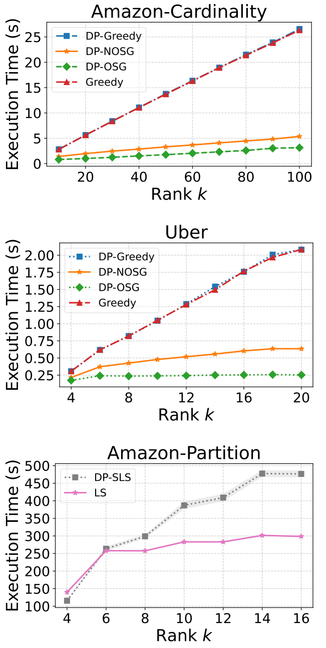

Figure 5: Execution time (seconds) as the rank k varies.

are observed for the Uber dataset. For Amazon-Partition at $k = 6$, we find that a small privacy parameter of $\varepsilon = 0.12$ is sufficient to reach a utility value within 1% of the non-private LS baseline.

Impact of k. We evaluate the impact of the rank k on the objective value. As shown in Figure 2, our algorithms perform competitively with the non-private Greedy baseline and significantly outperform Random. For the Uber dataset, DP-Greedy, DP-NOSG, and DP-OSG exhibit nearly identical performance, with an average utility gap relative to Greedy of at most 3.2%. This gap naturally widens as k increases due to the accumulation of DP noise. On the Amazon dataset, the utility of DP-OSG is slightly lower, averaging 8% below Greedy; however, this gap is notably smaller for lower values of k. Since our algorithms significantly outperform Random, its scale obscures the performance gap between the DP and non-private baseline. We provide a focused comparison excluding Random in Figure 7 to highlight these nuances.

For Amazon-Partition, the utility of DP-SLS remains comparable to the non-private LS baseline across all configurations. Specifically, DP-SLS performs close to LS for $k \leq 12$, with an average utility difference of only 1.3%. Beyond this threshold, while DP-SLS maintains a significant utility advantage over Random, the performance gap relative to LS widens, reaching 9.3% at $k = 16$.

Impact of λ. We examine the robustness of our results with respect to the parameter λ, which controls the trade-off between relevance and diversity. The results are presented in Figure 6. the high utility of our proposed algorithms relative to non-private baselines is consistently maintained across the various choices of λ between 0 and 0.8 .Overall, we find that varying λ has no significant impact on the relative quality of our algorithms. For the Uber dataset, the utility gap between our proposed algorithms and the Greedy baseline is upper-bounded by 2.8% on average. Similarly, for Amazon-Partition, the gap is upper-bounded by less than a marginal 1%. We observed similar trends for Amazon-Cardinality.

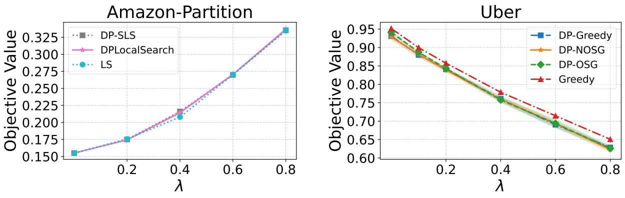

Figure 6: Objective value for different choices of λ.

8.4 Performance Analysis

We evaluate the computational efficiency of our algorithms by measuring both query complexity and execution time (Figures 4 and 5). In the submodular maximization literature, the number of oracle calls is considered the most robust metric for performance evaluation, as it is invariant to implementation and problem-specific details [40]. Nevertheless, we also report execution time to provide additional practical insights.

Number of Oracle Calls. Figure 4 shows the trend in the number of oracle calls as k varies. For cardinality constraints across both datasets, the results confirm that DP-Greedy requires the same

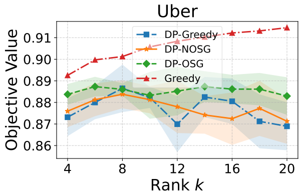

Figure 7: Objective value as ε varies for the Uber experiment, Random omitted.

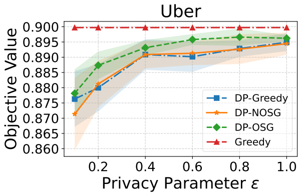

Figure 8: Objective value as k varies for the Uber experiment, Random omitted.

number of oracle calls as the non-private Greedy baseline, scaling linearly with the rank k. In contrast, DP-NOSG exhibits a clear logarithmic dependence on k, significantly outperforming both Greedy and DP-Greedy. For instance, on the Amazon dataset with $k = 100$, the query count for DP-NOSG is 10.6 times lower than that of Greedy. Notably, DP-OSG demonstrates even greater efficiency; aligning with its complexity analysis, its query count is invariant to k. For instance, for $k = 100$ on the Amazon dataset, DP-OSG reduces the query count by a factor of 40.6 compared to Greedy while maintaining comparable utility. For Amazon-Partition, the query count of the non-private LS baseline is dominated by the initialization phase, which requires an exhaustive search for the optimal feasible pair. This explains the plateau observed for $k \geq 6$, as increasing k does not expand the set of feasible initialization pairs. Our proposed DP-SLS avoids this exhaustive search, maintaining a lower query count. For instance, at $k = 10$, DP-SLS reduces the query count by a factor of 4.7 compared to LS.

Execution Time. Figure 4 shows the execution time trend as k varies. For cardinality constraints across both datasets, the execution time trends align closely with the query count results. DP-NOSG exhibits a clear logarithmic dependence on k, outperforming both Greedy and DP-Greedy; for the Amazon dataset at $k = 100$, DP-NOSG is 5.4 times faster than Greedy. As expected, DP-OSG demonstrates even greater efficiency, with an execution time that is essentially independent of k. Specifically, at $k = 100$ on the Amazon dataset, DP-OSG is 8.3 times faster than Greedy, while maintaining high utility. For Amazon-Partition, Figure 5 illustrates that despite its lower query complexity, DP-SLS is slower than the non-private LS baseline in the Amazon-Partition task. The discrepancy arises because the LS query bottleneck occurs during pair evaluation, which is computationally cheaper than the operations required to evaluate potential swaps in this task. Furthermore, the non-private LS baseline converged to a local optimum in fewer iterations than the fixed number of iterations required by the DP-SLS privacy analysis. Nevertheless, DP-SLS maintains a practical execution time of 500 seconds for $k = 16$ on the Amazon-Partition instance, a manageable 60% overhead compared to LS. While both algorithms can be relatively slow for real-time interactive settings, DP-SLS is currently the only algorithm that provides formal privacy guarantees.

9 CONCLUSION

In this work, we studied the max-sum diversification problem, a widely adopted formulation introduced by Borodin et al. [8], under differential privacy. We designed DP algorithms for both cardinality and matroid constraints. The proposed algorithms achieve utility guarantees comparable to their non-private counterparts while offering improved complexity, making them of independent interest beyond privacy considerations. We conducted experiments real-world datasets, showing that the proposed DP algorithms achieve high utility while substantially improving efficiency in practice for cardinality constraints.

Our work opens several interesting directions for future research. Immediate questions include whether stronger privacy guarantees can be achieved for decomposable objectives under matroid constraints, and whether the near-optimal $(1/2 - \gamma)$-approximation can be attained for cardinality constraints with $O_\gamma(n)$ queries. Another natural question is whether the DP algorithm for matroid constraints can be further accelerated in practice. Finally, addressing additional diversity models remains an important direction for future work.

REFERENCES

[1] Zeinab Abbassi, Vahab S Mirrokni, and Mayur Thakur. 2013. Diversity maximization under matroid constraints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 32–40.

[2] John M. Abowd. 2018. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2867. https://doi.org/10.1145/3219819.3226070

[3] Pankaj K Agarwal, Aryan Esmailpour, Xiao Hu, Stavros Sintos, and Jun Yang. 2024. Computing a well-representative summary of conjunctive query results. *Proceedings of the ACM on Management of Data* 2, 5 (2024), 1–27.

[4] Pankaj K Agarwal, Stavros Sintos, and Alex Steiger. 2020. Efficient indexes for diverse top-k range queries. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 213–227.

[5] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. 2011. Inapproximability of densest κ-subgraph from average case hardness. (2011).

[6] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. 2016. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 1046–1059.

[7] Lingfeng Bian, Weidong Yang, Jingyi Xu, and Zijing Tan. 2024. Discovering denial constraints based on deep reinforcement learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 120–129.

[8] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. 2017. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Transactions on Algorithms (TALG)* 13, 3 (2017), 1–25.

[9] Richard A Brualdi. 1969. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society* 1, 2 (1969), 161–167.

[10] Niv Buchbinder, Moran Feldman, and Roy Schwartz. 2017. Comparing apples and oranges: Query trade-off in submodular maximization. *Mathematics of Operations Research* 42, 2 (2017), 308–329.

[11] Pablo Castells, Neil Hurley, and Saul Vargas. 2021. Novelty and diversity in recommender systems. In *Recommender systems handbook*. Springer, 603–646.

[12] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. *Novelty and Diversity in Recommender Systems*. Springer US, Boston, MA, 881–918. https://doi.org/10.1007/978-1-4899-7637-6_26

[13] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. 2018. Fast coreset-based diversity maximization under matroid constraints. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 81–89.

[14] Matteo Ceccarello, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. 2017. MapReduce and streaming algorithms for diversity maximization in metric spaces of bounded doubling dimension. *Proceedings of the VLDB Endowment* 10, 5 (2017), 469–480.

[15] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. 2017. Local search for max-sum diversification. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 130–142.

[16] Anamay Chaturvedi, Huy Lê Nguyễn, and Lydia Zakynthinou. 2021. Differentially private decomposable submodular maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6984–6992.

[17] Anamay Chaturvedi, Huy Nguyen, and Thy Dinh Nguyen. 2023. Streaming submodular maximization with differential privacy. In *International Conference*

on Machine Learning. PMLR, 4116–4143.

[18] Erica Coppolillo, Giuseppe Manco, and Aristides Gionis. 2024. Relevance meets diversity: A user-centric framework for knowledge exploration through recommendations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 490–501.

[19] Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. 2013. Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1014–1022.

[20] Ting Deng and Wenfei Fan. 2014. On the complexity of query result diversification. *ACM Transactions on Database Systems (TODS)* 39, 2 (2014), 1–46.

[21] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3574–3583.

[22] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[23] Cynthia Dwork. 2019. Differential Privacy and the US Census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Amsterdam, Netherlands) *(PODS '19)*. Association for Computing Machinery, New York, NY, USA, 1. https://doi.org/10.1145/3294052.3322188

[24] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[25] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*. IEEE, 51–60.

[26] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348

[27] Sándor P. Fekete and Henk Meijer. 2003. Maximum Dispersion and Geometric Maximum Weight Cliques. *Algorithmica* 38, 3 (Dec. 2003), 501–511. https://doi.org/10.1007/s00453-003-1074-x

[28] FiveThirtyEight. 2014. Uber Pickups in New York City. https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

[29] Piero Fraternali, Davide Martinenghi, and Marco Tagliasacchi. 2012. Top-k bounded diversification. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. 421–432.

[30] Mehrdad Ghadiri and Mark Schmidt. 2019. Distributed Maximization of. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2077–2086.

[31] Sreenivas Gollapudi and Aneesh Sharma. 2009. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*. 381–390.

[32] Xintong Guo, Hong Gao, Yinan An, and Zhaonian Zou. 2020. Diversified top-k querying in knowledge graphs. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 319–336.

[33] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. 2010. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1106–1125.

[34] Ziyan Han, Wanjia Chen, Yunpeng Han, Rui Mao, and Jianbin Qin. 2026. Fast Diversified Top-k Rule Discovery via User-Guided Embeddings. *IEEE Transactions on Knowledge and Data Engineering* (2026).

[35] P Hansen and D Moon. 1988. *Dispersing facilities on a network*. Rutgers University. Rutgers Center for Operations Research [RUTCOR].

[36] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. 1997. Approximation algorithms for maximum dispersion. *Operations research letters* 21, 3 (1997), 133–137.

[37] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[38] Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S Mirrokni. 2014. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 100–108.

[39] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2020. *Mining of massive data sets*. Cambridge university press.

[40] Wenxin Li, Moran Feldman, Ehsan Kazemi, and Amin Karbasi. 2022. Submodular maximization in clean linear time. *Advances in neural information processing systems* 35 (2022), 17473–17487.

[41] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 94–103.

[42] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. 2015. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[43] Marko Mitrovic, Mark Bun, Andreas Krause, and Amin Karbasi. 2017. Differentially private submodular maximization: Data summarization in disguise. In *International Conference on Machine Learning*. PMLR, 2478–2487.

[44] Loay Mualem and Moran Feldman. 2022. Using partial monotonicity in submodular maximization. *Advances in Neural Information Processing Systems* 35 (2022), 2723–2736.

[45] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14 (1978), 265–294.

[46] Paolo Pellizzoni, Andrea Pietracaprina, and Geppino Pucci. 2025. Fully dynamic clustering and diversity maximization in doubling metrics. *ACM Transactions on Knowledge Discovery from Data* 19, 4 (2025), 1–45.

[47] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 15–22.

[48] Akbar Rafiey and Yuichi Yoshida. 2020. Fast and private submodular and k-submodular functions maximization with matroid constraints. In *International conference on machine learning*. PMLR, 7887–7897.

[49] Sekharipuram S Ravi, Daniel J Rosenkrantz, and Giri Kumar Tayi. 1994. Heuristic and special case algorithms for dispersion problems. *Operations research* 42, 2 (1994), 299–310.

[50] Omid Sadeghi and Maryam Fazel. 2021. Differentially private monotone submodular maximization under matroid and knapsack constraints. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2908–2916.

[51] Sebastian Perez Salazar and Rachel Cummings. 2021. Differentially private online submodular maximization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1279–1287.

[52] Alexander Schrijver et al. 2003. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer.

[53] Qi Song, Yinghui Wu, Peng Lin, Luna Xin Dong, and Hui Sun. 2018. Mining summaries for knowledge graph search. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1887–1900.

[54] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. 2017. Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12. *CoRR* abs/1709.02753 (2017). arXiv:1709.02753 http://arxiv.org/abs/1709.02753

[55] Yue Wang, Alexandra Meliou, and Gerome Miklau. 2018. Rc-index: Diversifying answers to range queries. *Proceedings of the VLDB Endowment* 11, 7 (2018).

[56] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and obfuscating user gender based on ratings. In *Proceedings of the sixth ACM conference on Recommender systems*. 195–202.

[57] Hassler Whitney. 1992. On the abstract properties of linear dependence. In *Hassler Whitney Collected Papers*. Springer, 147–171.

[58] Sepehr Zadeh, Mehrdad Ghadiri, Vahab Mirrokni, and Morteza Zadimoghaddam. 2017. Scalable feature selection via distributed diversity maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[59] Bo Zhang, Na Wang, and Hongxia Jin. 2014. Privacy concerns in online recommender systems: influences of control and user data input. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. 159–173.

[60] Chengyuan Zhang, Ying Zhang, Wenjie Zhang, Xuemin Lin, Muhammad Aamir Cheema, and Xiaoyang Wang. 2014. Diversified spatial keyword search on road networks. In *Advances in Database Technology-EDBT 2014: 17th International Conference on Extending Database Technology, Proceedings*.

A SUBMODULAR FUNCTIONS AND MATROIDS

Submodular set functions have numerous applications across areas such as machine learning, databases, economics, and network analysis. Their usefulness stems from the property of diminishing returns, which naturally arises in many practical situations, and informally states that the incremental benefit of adding an element to a smaller set is larger than adding it to a larger set. This property makes submodular functions well-suited to model the notion of relevance.

Definition A.1. [45] A set function $f : 2^V \to \mathbb{R}$ is submodular if $f(u \mid S) \geq f(u \mid T)$ for every two sets $S \subseteq T \subseteq V$ and element $u \in V \setminus T$.

A set function $f : 2^V \to \mathbb{R}$ is *non negative* if $f(S) \geq 0$ for every set $S \subseteq V$, and is *monotone* if $f(S) \leq f(T)$ for every two sets $S \subseteq T \subseteq V$. We consider only non-negative, monotone submodular functions, as is the case in the non-private setting [8].

The problem of maximizing a submodular function subject to a *matroid independence* constraint is one of earliest and most studied problems in submodular maximization [52]. In the context of max-sum diversification as well, matroids allow a substantial generalization of the types of constraints that can be modeled compared to simple cardinality [8]. For example, cardinality constraint is a special case called a *uniform matroid.* In a *partition matroid*, the ground set V is partitioned into disjoint subsets P_i, and the independent sets are $\{S \in 2^V : |S \cap P_i| \leq r_i\}$, i.e., sets that satisfy a separate cardinality bound for each block. As another example, a *transversal matroid* models representativeness. Let $A_1, \ldots, A_m$ a collection of categories, i.e, subsets of the ground set V (that may overlap). A subset $S \subseteq V$ is independent if there exists an injective mapping $h : S \to [m]$ such that $u \in A_{h(u)}$ for all $u \in S$, that is, no two items represent to the same category. Additional background an examples can be found in [52]. We now give the formal definition.

Definition A.2. [57] A matroid is a pair $(V, \mathcal{I})$, where $\mathcal{I} \subseteq 2^V$ is a collection of subsets called *independent sets*, with the following properties.

(*i*) if $A \subseteq B$ and $B \in \mathcal{I}$ then $A \in \mathcal{I}$ (Hereditary),
(*ii*) If $A, B \in \mathcal{I}$ and $|A| < |B|$ then $\exists u \in B \setminus A$ such that $A \cup \{u\} \in \mathcal{I}$ (Exchange).

A matroid is a combinatorial abstraction of the notion of independence from linear algebra. Following the terminology used for linear spaces, independent sets that are inclusion-wise maximal are called *bases*. An immediate corollary of the exchange property is that all bases have the same number of elements, and this number is referred to as the *rank* of the matroid. As is standard in the literature, we assume access to the set functions and the matroid via *oracles*. Specifically, the *value oracle* for f (respectively, d) returns the value $f(S)$ (respectively, $d(S)$) given a set S. The *independence oracle* associated with a matroid $(V, \mathcal{I})$ takes as input a set $S \subseteq V$ and answers whether $S \in \mathcal{I}$. The number of oracle calls performed by an algorithm is commonly used as a proxy for its time complexity, as the exact complexity may be application specific, or depend on implementation details such as the data structures used to maintain sets. In our algorithms, the number of queries to the value and independence oracles are the same up to constant factors. We therefore refer to the total number of queries to both as the number of *oracle calls.*

B OMITTED PROOFS FROM SECTION 5

In this section we provide the proof of Lemma 5.3, restated next for convenience.

Lemma 5.3 (Adapted from [8]). *For all $i \in [k] \triangleq \{1, \ldots, k\}$, $d(C_i, S_{i-1}) \geq \frac{(i-1)|C_i|}{k(k-1)} d(\mathrm{OPT})$.*

Let OPT be an optimal solution, i.e., a subset of size at most r that maximizes ϕ. Since ϕ is monotone, we may assume that $|\mathrm{OPT}| = r$. Let S_i be the solution at the end of step i. Define $A_i = S_{i-1} \cap \mathrm{OPT}$, $B_i = S_{i-1} \setminus A_i$ and $C_i = \mathrm{OPT} \setminus A_i$.

Lemma B.1 ([49]). *Suppose d is a pseudometric, and let X and Y be two disjoint subsets of V. Then, $|Y| \cdot d(X) \leq (|X| - 1) \cdot d(Y, X)$.*

Proof of Lemma B.1. If either set is empty, or if $|X| = 1$ then both sides equal zero and the inequality trivially holds. Thus, we assume that both sets are not empty and that $|X| \geq 2$. For distinct $x, x' \in X$ and for $y \in Y$, we have by the triangle inequality $d(x, x') \leq d(x, y) + d(x', y)$. Summing over all distinct pairs $x, x' \in X$, we obtain $d(X) \leq (|X| - 1) d(y, X)$, because each distance $d(y, x)$ appears in the sum exactly $|X| - 1$ times. Summing over all $y \in Y$, we get $|Y| d(X) \leq (|X| - 1) d(Y, X)$.

□

Proof of Lemma 5.3. We first prove the claim for the special case of $C_i = \{o\}$ for some $o \in V \setminus S_{i-1}$, which is not covered by the proof in [8]. Since $|S_{i-1}|$ increases by 1 in each round, it follows that $i = k$. Hence, it remains to prove that $d(o, S_{i-1}) \geq d(\mathrm{OPT})/k$. Note that our assumption implies that $\mathrm{OPT} = S_i \cup \{o\}$. By Lemma B.1, we have

$$d(S_{i-1}) \leq (|S_{i-1}| - 1) d(o, S_{i-1}) \leq (k - 1) d(o, S_{i-1}).$$

Therefore,

$$d(\mathrm{OPT}) = d(S_{i-1} \cup \{o\}) = d(S_{i-1}) + d(o, S_{i-1}) \leq k \cdot d(o, S_{i-1}).$$

For the case $|C_i| > 1$, we follow the proof of Borodin et al. [8]. By Lemma B.1, the following inequalities hold

$$(|C_i| - 1)d(B_i, C_i) \ge |B_i|\, d(C_i) \tag{6}$$

$$(|C_i| - 1)d(A_i, C_i) \ge |A_i|\, d(C_i) \tag{7}$$

$$(|A_i| - 1)d(A_i, C_i) \ge |C_i|\, d(A_i) \tag{8}$$

$$d(A_i, C_i) + d(A_i) + d(C_i) = d(\mathrm{OPT}) \tag{9}$$

where (9) holds since A_i and C_i are disjoint sets whose union equals OPT. Then, we multiply the equations (6),(7),(8),(9) by the following non-negative numbers respectively

$$\frac{1}{|C_i| - 1},\ \frac{|C_i| - |B_i|}{k(|C_i| - 1)},\ \frac{i-1}{k(k-1)},\ \frac{(i-1)\,|C_i|}{k(k-1)}.$$

Summing the multiplied equations, we get

$$d(A_i, C_i) + d(B_i, C_i) - \frac{(i-1)\,|C_i|\,(k - |C_i|)}{k(k-1)(|C_i| - 1)} d(C_i) \ge \frac{(i-1)\,|C_i|}{k(k-1)} d(\mathrm{OPT}).$$

Since $|C_i| \le r$ and $d(A_i, C_i) + d(B_i, C_i) = d(C_i, S_{i-1})$, we have

$$d(C_i, S_{i-1}) \ge \frac{(i-1)\,|C_i|}{k(k-1)} d(\mathrm{OPT}),$$

and the proof is complete. □

C OMITTED PROOFS FROM SECTION 6

The proof for the query complexity of Algorithm 2 is given by the following lemma.

Lemma C.1. *Algorithm 2 makes $O(n \log k \log(1/\gamma))$ queries when $g(i) = k - i + 1$ for all $i \in [k]$, and $O(n \log(1/\gamma))$ queries when $g(i) = \min\{k,\ n - i + 1\}$ for all $i \in [k]$.*

Proof. The number of queries performed by Algorithm 2 in iteration i is $|V_i| + 1$, and note that

$$|V_i| \le \frac{|N_i|\ \log(1/\gamma)}{g(i)} + 1.$$

Since a new element is added to S_i in each iteration, we have $|N_i| = |V \setminus S_{i-1}| = n - i + 1$. Hence, the total number of queries is bounded by

$$\sum_{i=1}^{k} |V_i| + 1 \le 2k + \log(1/\gamma) \sum_{i=1}^{r} \frac{n - i + 1}{g(i)}.$$

If $g(i) = r - i + 1$, the standard harmonic sum bound yields $\sum_{i=1}^{r} \frac{n-i+1}{g(i)} = O(n \log k)$, and the first part of the lemma follows. If $g(i) = \min\{k,\ n - i + 1\}$, we have

$$\sum_{i=1}^{r} \frac{n-i+1}{g(i)} \le \sum_{i=1}^{r} \frac{n-i+1}{r} + \sum_{i=1}^{r} \frac{n-i+1}{n-i+1} \le n + r = O(n),$$

which implies the second part. □

We now turn to complete the proof for the privacy guarantee stated in Theorem 6.3. To this end, we utilize the following concentration bound.

Lemma C.2 (Claim 3.8 in [16], based on [33]). *Consider an n-round probabilistic process. In each round $i \in [n]$, an adversary chooses a distribution $\mathcal{P}_i$ over $[0, 1]$ and a sample R_i is drawn from this distribution. Let $Z_1 = 1$ and $Z_{i+1} = Z_i - R_i Z_i$. We define the random variable $Y_j = \sum_{i=j}^{n} Z_i \mathbb{E}[R_i]$. Then for any $j \in [n]$, $\mathbf{Pr}[Y_j \ge q Z_j] \le \exp(3 - q)$.*

The above bound and the argument of [16, 33] yields the following lemma. We continue with the notation established in the proof of Theorem 6.3.

Lemma C.3. *The following inequality holds.*

$$\mathbf{Pr}\left[\prod_{i=1}^{k} \mathop{\mathbb{E}}_{u \sim \mathcal{P}_i}[\exp(\tfrac{\varepsilon_0}{2} \cdot \beta_t^i(u))] \ge (e^{\varepsilon_0/2} - 1)(3 + \log(1/\delta))\right] \le \delta$$

Proof. For all $\varepsilon_0 \in (0,1]$, we have

$$\prod_{i=1}^{k} \mathop{\mathbb{E}}_{u\sim\mathcal{P}_i}[\exp(\tfrac{\varepsilon_0}{2}\cdot\beta_t^i(u))] \le \prod_{i=1}^{k} \mathop{\mathbb{E}}_{u\sim\mathcal{P}_i}[1+(e^{\varepsilon_0/2}-1)\cdot\beta_t^i(u)]$$

$$\le \exp\left((e^{\varepsilon_0/2}-1)\sum_{i=1}^{k} \mathop{\mathbb{E}}_{u\sim\mathcal{P}_i}[\beta_t^i(u)]\right).$$

where the first inequality holds since $e^x \le 1+\frac{e^{\varepsilon_0/2}-1}{\varepsilon_0/2}x$ for all $x \in [0,\varepsilon_0/2]$, and the second uses $1+x \le e^x$ for all x. It remains to show that the sum of expectations is bounded with high probability. Consider the following probabilistic process. In each round $i \in [k]$, u_i is drawn from $\mathcal{P}_i$. Let $Z_i = 1-\phi_t'(U_{i-1})$ be the total remaining marginal utility at the beginning of iteration i. Define the random variable $R_{i-1}(u) = \beta_t^i(u)/Z_{i-1}$ which is determined by the sample of $u \sim \mathcal{P}_i$, and denotes the percentage increase in utility in iteration i (If $Z_i = 0$, we set $R_i = 0$). Since ϕ_t' is monotone and has range in $[0,1]$, we have $R_{i-1}(u) \in [0,1]$. Note that using these definitions, we have $Z_i = Z_{i-1} - R_{i-1}Z_{i-1}$. Define, $Y_j = \sum_{i=j}^{k} Z_i \cdot \mathbb{E}_{u\sim\mathcal{P}_i}[R_i(u)]$, and observe that the sum we want to bound is exactly Y_1. Letting $q = 3+\log(1/\delta)$ and using Lemma C.2, we conclude that

$$\mathbf{Pr}\left[\sum_{i=1}^{k} \mathop{\mathbb{E}}_{u\sim\mathcal{P}_i}[\beta_t^i(u)] \ge q\right] = \mathbf{Pr}[Y_1 \ge q] \le \mathbf{Pr}[Y_1 \ge qZ_1] \le \exp(3-q) = \delta$$

where the second inequality holds since $Z_1 \in [0,1]$. □

The remainder of this section is devoted to the proof of Theorem 6.5, restated next.

Theorem 6.5. *Suppose ϕ has sensitivity Δ. Then,* `DP-OSG` *(Algorithm 2) outputs a set S such that $\mathbb{E}[\phi(S)] \ge (1-(\frac{2}{e})^{(1-\gamma)(1-1/k)})\cdot\phi(\mathrm{OPT}) - O(\frac{k\Delta\log n}{\varepsilon_0})$ and makes $O(n\log\gamma^{-1})$ oracle calls.*

Lemma 6.7. *For every iteration $i \ge 2$,*

$$d(\mathrm{OPT}\cup S_{i-1}) - d(S_{i-1}) \le \left(1+\frac{k-1}{i-1}\right)\cdot d(C_i, S_{i-1}).$$

Proof of Lemma 6.7. By Lemma B.1, we have $|S_{i-1}|\cdot d(C_i) \le (|C_i|-1)\cdot d(C_i,S_{i-1})$. Moreover, since $|S_{i-1}| = i-1$ and $|C_i| \le k$, we have $d(C_i) \le \frac{k-1}{i-1}\cdot d(C_i,S_{i-1})$. Therefore,

$$d(\mathrm{OPT}\cup S_{i-1}) - d(S_{i-1}) = d(C_i\cup S_{i-1}) - d(S_{i-1}) = d(C_i) + d(C_i,S_{i-1}) \le \left(1+\frac{k-1}{i-1}\right)d(C_i,S_{i-1}).$$

where the second equality holds because S_{i-1} and C_i are disjoint. □

Lemma C.4. *The following inequalities hold:*

$$\prod_{i=1}^{k-1}\left(1-\frac{1-\gamma}{k(1+\frac{k-1}{i})}\right) \le \left(\frac{2}{e}\right)^{(1-\gamma)(1-1/k)} \le \frac{2}{e}+\gamma+\frac{1}{k}.$$

Proof. We first evaluate the sum within the exponent. We have:

$$\frac{1}{k}\sum_{i=1}^{k-1}\frac{i}{i+k-1} = \frac{1}{k}\sum_{i=1}^{k-1}\left(1-\frac{k-1}{i+k-1}\right) = \frac{k-1}{k} - \frac{k-1}{k}\cdot\sum_{i=1}^{k-1}\frac{1}{i+k-1}$$

$$= \frac{k-1}{k} - \frac{k-1}{k}\cdot\sum_{j=k}^{2k-2}\frac{1}{j} \ge \frac{k-1}{k} - \frac{k-1}{k}\cdot\log(2)$$

$$= \left(1-\frac{1}{k}\right)(1-\log 2),$$

where in the inequality we have used the integral bound:

$$\sum_{j=k}^{2k-2}\frac{1}{j} \le \int_{k-1}^{2k-2}\frac{1}{x}\,dx = \log(2k-2)-\log(k-1) = \log(2).$$

We next observe that the bound $\log(1-x) \le -x$, which holds for all $x \in (0,1)$, together with the preceding derivation, implies that:

$$\sum_{i=1}^{k-1}\log\left(1-\frac{1-\gamma}{k(1+\frac{k-1}{i})}\right) \le -(1-\gamma)\sum_{i=1}^{k-1}\frac{1}{k(1+\frac{k-1}{i})} \le -(1-\gamma)\left(1-\frac{1}{k}\right)(1-\log 2).$$

The first inequality in the lemma follows by exponentiating both sides of the result above. For the second inequality, observe that:

$$\begin{aligned}(2/e)^{(1-\gamma)(1-1/k)} \le (2/e)^{(1-\gamma-1/k)} &= e^{(1-\gamma-1/k)\log(2/e)} = \frac{2}{e} \cdot e^{\log(e/2)(\gamma+1/k)} \\ &\le \frac{2}{e} \cdot (1 + (e-1)\log(e/2)(\gamma + 1/k)) \le \frac{2}{e} + \gamma + \frac{1}{k},\end{aligned}$$

where the second inequality follows from the fact that $e^x \le 1 + (e-1)x$ for all $x \in [0,1]$. □

D OMITTED PROOFS FROM SECTION 7

We use the following notation, some of which has been introduced in Section 7. Let S_i denote the base selected at iteration i, $C_i = \mathrm{OPT} \setminus S_{i-1}$, $B_i = S_{i-1} \setminus \mathrm{OPT}$, and let $h : S_{i-1} \longrightarrow \mathrm{OPT}$ be a bijection satisfying the condition in Lemma 7.4. Denote $B_i = \{b_1, \dots, b_t\}$, and let $c_j = h(b_j)$ for each $j \in [t]$. The following result was proved in [8].

Lemma D.1 (Lemmas 5 and 7, [8]). *Suppose $\mathcal{M}$ has rank $r > 2$ and $t \ge 2$. Then,*

(i) $\sum_{j=1}^{t}[f(S_{i-1} - b_j + c_j) - f(S_{i-1})] \ge f(\mathrm{OPT}) - 2f(S_{i-1})$,

(ii) $\sum_{j=1}^{t}[d(S_{i-1} - b_j + c_j) - d(S_{i-1})] \ge d(\mathrm{OPT}) - 2d(S_{i-1})$.

Proof of Lemma 7.5. Recall that $h(u) = u$ for every $u \in S_{i-1} \cap \mathrm{OPT}$. Thus, when $t = 0$, we have $S_{i-1} = \mathrm{OPT}$, so the inequality becomes $r\phi(\mathrm{OPT}) \ge (r-1)\phi(\mathrm{OPT})$. When $t = 1$, the left-hand side becomes $\phi(\mathrm{OPT}) + (r-1)\phi(S_{i-1})$. Since the claim holds in both cases, we assume $t \ge 2$. We get

$$\begin{aligned}\sum_{u \in S_{i-1}} \phi(S_{i-1} - u + h(u)) &= \sum_{i=1}^{t} \phi(S_{i-1} - b_i + c_i) + (r-t)\phi(S_{i-1}) \\ &\ge \phi(\mathrm{OPT}) + (t-2)\phi(S_{i-1}) + (r-t)\phi(S_{i-1}) \\ &= \phi(\mathrm{OPT}) + (r-2)\phi(S_{i-1}).\end{aligned}$$

where the inequality follows by summing the inequalities in Lemma D.1 and rearranging. □

Lemma D.2. *Let $V_i \subseteq V$ be a uniformly random subset of size $\lceil \frac{n}{r} \rceil$. Then,*

$$\mathbf{Pr}[(S_{i-1} \times V_i) \cap M_i \neq \emptyset] \ge 1 - 1/e.$$

Proof of Lemma D.2. Consider the probability that V_i does not contain any element of $\{h(u) : u \in S_{i-1}\}$. If $|V_i| > n - r$, this probability is zero. Otherwise, it equals

$$\frac{\binom{n-k}{|V_i|}}{\binom{n}{|V_i|}} = \prod_{i=0}^{|V_i|-1} \frac{n-k-i}{n-i} = \prod_{i=1}^{|V_i|} \left(1 - \frac{r}{n-i+1}\right) \le \left(1 - \frac{r}{n}\right)^{|V_i|} \le e^{-\frac{r|V_i|}{n}} \le e^{-1}.$$

where the last inequality holds since $r \cdot |V_i| = r \cdot \lceil \frac{n}{r} \rceil \ge n$. The claim follows. □

Proof of Lemma 7.6. Denote $\alpha = 4\Delta \log(n)/\varepsilon_0$. Fix an iteration $i \in [T]$ and let us condition on having selected a base S_{i-1} after iteration $i-1$, and on some realization of the set V_i at iteration i. Let $M_i = \{(u, h(u)) : u \in S_{i-1}\}$. By Theorem 3.5, the exponential mechanism guarantees that

$$\begin{aligned}\mathbb{E}\left[\phi(S_{i-1} - u + v) - \phi(S_{i-1})\right] &\ge \max_{(u,v) \in W_i} [\phi(S_{i-1} - u + v) - \phi(S_{i-1})] - \alpha \\ &= \max\{0, \max_{\substack{(u,v) \in S_{i-1} \times (V_i \setminus S_{i-1}) \\ S_{i-1} - u + v \in \mathcal{I}}} [\phi(S_{i-1} - u + v) - \phi(S_{i-1})]\} - \alpha \\ &\ge \max\{0, \max_{(u,v) \in (S_{i-1} \times V_i) \cap M_i} [\phi(S_{i-1} - u + v) - \phi(S_{i-1})]\} - \alpha \\ &\ge \max\left\{0, \sum_{(u,v) \in (S_{i-1} \times V_i) \cap M_i} \frac{\phi(S_{i-1} - u + v) - \phi(S_{i-1})}{|(S_{i-1} \times V_i) \cap M_i|}\right\} - \alpha\end{aligned}$$

where the equality in the second line holds since W_i always contains a dummy swap that does not change the current solution. Unfixing the implicit conditioning on V_i in iteration i and taking expectation over all their possible realizations, we get

$$\begin{aligned}\mathbb{E}\left[\phi(S_{i-1}-u+v)-\phi(S_{i-1})\right] &\geq \mathop{\mathbb{E}}_{V_i}\left[\max\left\{0, \sum_{(u,v)\in(S_{i-1}\times V_i)\cap M_i}\frac{\phi(S_{i-1}-u+v)-\phi(S_{i-1})}{|(S_{i-1}\times V_i)\cap M_i|}\right\}-\alpha\right]\\ &\geq \max\left\{0, \mathop{\mathbb{E}}_{V_i}\left[\sum_{(u,v)\in(S_{i-1}\times V_i)\cap M_i}\frac{\phi(S_{i-1}-u+v)-\phi(S_{i-1})}{|(S_{i-1}\times V_i)\cap M_i|}\right]\right\}-\alpha\\ &\geq \frac{1-1/e}{r}\sum_{(u,v)\in M_i}\left[\phi(S_{i-1}-u+v)-\phi(S_{i-1})\right]-\alpha. \qquad (10)\end{aligned}$$

The second inequality follows from Jensen's inequality, as the function $f(x)=\max\{0,x\}$ is convex. We now focus on proving the last inequality. First, observe that if the sum in (10) is negative, then the final inequality holds trivially. We therefore assume that the sum is non-negative. Consider the following probabilistic experiment. We sample V_i as in Algorithm 3. If $(S_{i-1}\times V_i)\cap M_i\neq\emptyset$, we select a uniformly random $(u,v)\in(S_{i-1}\times V_i)\cap M_i$. Otherwise, output $\perp$. Let $p_{u,v}$ denote the probability that $(u,v)\in M_i$ is selected conditioned on the event $(S_{i-1}\times V_i)\cap M_i\neq\emptyset$, which we denote by G. We have

$$\begin{aligned}\mathop{\mathbb{E}}_{V_i}\left[\sum_{(u,v)\in(S_{i-1}\times V_i)\cap M_i}\frac{\phi(S_{i-1}-u+v)-\phi(S_{i-1})}{|(S_{i-1}\times V_i)\cap M_i|}\right] &= \mathbf{Pr}[G]\cdot\sum_{(u,v)\in M_i}p_{u,v}\cdot\left[\phi(S_{i-1}-u+v)-\phi(S_{i-1})\right]\\ &\geq \frac{1-1/e}{r}\sum_{(u,v)\in M_i}\left[\phi(S_{i-1}-u+v)-\phi(S_{i-1})\right]\end{aligned}$$

where in the first equality we use the law of total expectation, observing that the inner sum is empty and thus equals 0 conditioned on $\overline{G}$. In the second inequality, we observe that since V_i is chosen uniformly at random, the probabilities $p_{u,v}$ must be the same for all $(u,v)\in M_i$, and therefore equal $1/|M_i|=1/r$. By Lemma D.2 we have $\mathbf{Pr}[G]\geq 1-1/e$, and the inequality (10) follows. Using Lemma 7.5, we get

$$\mathbb{E}\left[\phi(S_{i-1}-u+v)\right]\geq\phi(S_{i-1})+\frac{1-1/e}{r}\left(\phi(\mathrm{OPT})-2\phi(S_{i-1})\right)-\alpha.$$

Finally, the lemma follows by unfixing S_{i-1} and taking an expectation over its possible realizations.

□

E LOCAL SEARCH FOR RANK 2 MATROID

We follow the notation of Section 7. The difficulty arises because the second inequality in Lemma D.1 may fail when $k=2$. Nevertheless, as its proof shows, Lemma 7.5 holds for $t=0$ or $t=1$ even with $k=2$, so Lemma 7.6 remains valid in these cases. Therefore, to complete the proof of Theorem 7.1, it suffices to verify that the inequality in Lemma 7.6 continues to hold in iterations with $k=t=2$, namely when $S_{i-1}=\{b_1,b_2\}$ and $C_i=\{c_1,c_2\}$.

Lemma E.1. *Suppose that $\mathcal{M}$ has rank $r=2$. Conditioned on having selected a base $S_{i-1}=\{b_1,b_2\}$ after iteration $i-1$, the following inequality holds.*

$$\mathbb{E}\left[\phi(S_i)\right]\geq\phi(S_{i-1})+\frac{1-1/e}{r}\left(\phi(\mathrm{OPT})-2\phi(S_{i-1})\right)-\frac{4\Delta\log(n)}{\varepsilon_0}$$

Proof. Since $\mathcal{M}$ has rank 2, our assumption implies that $\mathrm{OPT}=\{c_1,c_2\}$. Without loss of generality, assume that $f(b_1)\geq f(b_2)$. By the triangle inequality,

$$d(S_{i-1}-b_2+c_1)+d(S_{i-1}-b_2+c_2)=d(b_1,c_1)+d(b_1,c_2)\geq d(c_1,c_2)=d(\mathrm{OPT}).$$

Moreover,

$$\begin{aligned}f(\mathrm{OPT})-f(S_{i-1}) &\leq f(\{b_1,b_2,c_1,c_2\})-f(\{b_1,b_2\})\\ &\leq f(\{b_1,b_2,c_1\})+f(\{b_1,b_2,c_2\})-2\cdot f(\{b_1,b_2\})\\ &\leq f(\{b_1,c_1\})+f(\{b_1,c_2\})-2\cdot f(\{b_1\})\\ &= f(S_{i-1}-b_2+c_1)+f(S_{i-1}-b_2+c_2)-2\cdot f(\{b_1\})\end{aligned}$$

where the first inequality follows from monotonicity, and the second and third follow from submodularity. Rearranging and using the fact that $f(S_{i-1})\leq f(\{b_1\})+f(\{b_2\})\leq 2f(\{b_1\})$ yields

$$f(S_{i-1}-b_2+c_1)+f(S_{i-1}-b_2+c_2)\geq f(\mathrm{OPT}).$$

Now, let $M_i = \{(b_2, c_1), (b_2, c_2)\}$. By summing the previous inequalities, we obtain

$$\sum_{(u,v)\in M_i} [\phi(S_{i-1} - u + v) - \phi(S_{i-1})] \geq \phi(\mathrm{OPT}) - 2\phi(S_{i-1}).$$

The remainder of the proof is analogous to that of Lemma 7.6, but uses the current definition of M_i. A similar derivation yields

$$\begin{aligned} \mathbb{E}\left[\phi(S_{i-1} - u + v) - \phi(S_{i-1})\right] &\geq \frac{1 - 1/e}{r} \sum_{(u,v)\in M_i} [\phi(S_{i-1} - u + v) - \phi(S_{i-1})] - \alpha \\ &\geq \frac{1 - 1/e}{r} \left(\phi(\mathrm{OPT}) - 2\phi(S_{i-1})\right) - \alpha \end{aligned}$$

using that $\mathbf{Pr}[\{c_1, c_2\} \cap V_i \neq \emptyset] \geq 1 - e^{-1}$, by an argument similar to that used in the proof of Lemma D.2. □