
# 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, 19(11): 3146 - 3159, 2026.
doi:10.14778/3836663.3836679

**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 range of tasks, including query result diversification [3, 8, 19, 28, 54], graph summarization [31, 52], rule mining [7, 33], spatial keyword search [61], as well as recommender systems [10, 17, 47] and feature selection [29, 57].

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 [30], 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 [55, 60], 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 [18]. However, optimizing coverage alone may still yield a redundant set of items: Figure 1a shows results on the Amazon Reviews dataset [36], 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 [11].

To encourage diversity, each product is associated with a set of subcategories, and pairwise distances are defined using the Jaccard distance [38] 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.


Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836679

| 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.**

Critically, the selection procedure must also preserve users' privacy. Since the objective is derived from individual transaction 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 [21, 23] 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, 22] and governmental organizations [20, 25, 53]. 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 [15, 32, 42, 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. [32] provides improved privacy guarantees. The precise results, corresponding to row *(i)* of Table 1, are discussed in Section 5.

However, this algorithm makes $\Theta(nk)$ 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. Such scaling becomes a significant bottleneck in high-concurrency environments or interactive systems where each execution of the algorithm corresponds to a single user query and many such queries must be processed simultaneously. Even in well-resourced systems, the cumulative computational overhead of $\Theta(nk)$ evaluations per instance limits total system throughput and increases latency, underscoring the need for more efficient algorithms that better suit high-volume workloads.

**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 [9, 41, 42]. While these analyses rely on submodularity, which does not hold in our setting, we nonetheless 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. [32] to incorporate the subsampling step and prove stronger guarantees for decomposable objectives. Following established literature in both private [42] and non-private [9, 41] submodular maximization, our utility guarantees are provided in expectation, as the randomness inherent in the subsampling process typically precludes the strong concentration required for meaningful high probability bounds. The precise results, corresponding to rows *(ii)* and *(iii)* of Table 1, are detailed in Section 6.

[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 $(\varepsilon, \delta)$-DP algorithms and their query complexity. Here $\Delta$ is the sensitivity (equivalent to the decomposability parameter for decomposable objectives), $n$ is the ground set size, $k$ is the maximal feasible solution size, and $\gamma$ controls the utility–complexity trade-off. For general $\Delta$-sensitive functions with cardinality constraints, additive errors incur an additional $\sqrt{k/\log(1/\delta)}$ factor. Setting $\varepsilon = \infty$ yields non-private algorithms with the same multiplicative approximation ratios. The $1/2$-approximation ratio is tight under standard complexity-theoretic assumptions (see Section 2).**

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

**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 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 [26, 34, 35, 49]. Gollapudi and Sharma [30] 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. [37] introduced composable coresets for streaming and distributed settings. Ceccarello et al. [13] later generalized these constructions to metrics with bounded doubling dimension, and Ceccarello et al. [12] 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. [14] 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.

**DP Submodular Maximization.** While MSD has not been studied in the context of differential privacy, a closely related line of work focuses on DP submodular maximization. The work of Gupta et al. [32] was the first to address the problem, focusing on decomposable objective functions under cardinality constraints. Mitrovic et al. [42] considered the more general case of submodular objectives with bounded sensitivity, proposing algorithms for both monotone and non-monotone functions under matroid and $p$-extendable systems constraints. Chaturvedi et al. [15] revisited decomposable objectives, extending the results of Gupta et al. [32] from cardinality to matroid constraints and from monotone to non-monotone functions. Rafiey and Yoshida [48] studied private submodular and $k$-submodular maximization under matroid constraints. Other notable advances include Salazar and Cummings [51], who studied the problem in the online setting under cardinality constraints; Sadeghi and Fazel [50], who considered bounded-curvature objectives; Zadicario and Milo [58], who addressed knapsack constraints for both monotone and non-monotone objectives; and Chaturvedi et al. [16], who proposed a streaming algorithm for cardinality constraints. Yet, all of these approaches apply only to submodular objectives and do not directly extend to MSD.

**Hardness and Lower Bounds.** Cardinality-constrained max-sum dispersion is known to be NP-hard even with metric distances [34]; hence, the more general MSD problem is also NP-hard. Moreover, Borodin et al. [8] showed that achieving an approximation factor better than $1/2$ is hard under the planted-clique hardness assumption [5]. As summarized in Table 1, the guarantees in rows (i), (ii), and (iv) match or nearly match this tight approximation factor. In contrast, row (iii) achieves improved complexity while providing a slightly weaker approximation guarantee.

Additionally, the DP MSD problem in Section 4 generalizes DP monotone submodular maximization, and hence known lower bounds for that problem extend to our setting. In particular, Gupta et al. [32] proved that any $\varepsilon$-DP algorithm for maximizing a submodular function under a cardinality constraint $k$ over a ground set of size $n$ must incur expected additive error $\Omega(\Delta k\varepsilon^{-1}\log(n/k))$. Chaturvedi et al. [16] extended this to the $(\varepsilon,\delta)$-DP setting, showing that any such algorithm achieving multiplicative factor $c$ must incur additive error $\Omega(\Delta k\varepsilon^{-1}c\log(\varepsilon/\delta))$, assuming $n \geq \delta^{-1}k(e^{\varepsilon}-1)$ and $c \geq 4\delta/(e^{\varepsilon}-1)$. In both cases, the hard instances are 1-decomposable, implying that our results in rows (i)–(iii) of Table 1 match the corresponding lower bounds up to logarithmic factors. These results recover the best-known additive error bounds for submodular maximization, and note that even in this restricted setting the tightness of logarithmic factors remains open [32]. For general bounded-sensitivity functions (row (iv), and rows (i)–(iii) with an additional $\sqrt{k/\log(1/\delta)}$ factor), our guarantees incur an additional $\sqrt{k}$-factor compared to the known lower bounds, which are not known to be tight in this more general setting. A similar $\tilde{O}(\sqrt{k})$ gap also appears in the state-of-the-art results for bounded-sensitivity submodular maximization under cardinality constraints [42].

## 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* [56] 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. See Appendix A in [59] for additional background and examples of matroids.

**Complexity Model.** We adopt the oracle complexity model, the standard abstraction in set function optimization, as it captures algorithmic complexity independently of application-specific details such as the cost of evaluating the objective function. Thus, 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 $S \in \mathcal{I}$. Complexity is measured by the number of oracle calls. While each marginal gain $d(u \mid S)$ may require up to $k$ pairwise evaluations $d(u,v)$, we treat $d$ as a set function and count each evaluation of $d(S)$ as a single oracle call for consistency with the submodular objective. Although the number of oracle calls is generally a reliable proxy for runtime, discrepancies may arise in practice, as evaluating the oracle on larger sets can be slower in some applications.

**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 [21, 23]).* A randomized algorithm $\mathcal{A}$ is $(\varepsilon,\delta)$-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 $\varepsilon$-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 [23].

Theorem 3.2 (Composition [23, 24]). *Let $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be $\varepsilon$-DP algorithms. Their $k$-fold adaptive composition $\mathcal{A}_{[k]}$, which outputs $y_i = \mathcal{A}_i(D, y_1, \ldots, y_{i-1})$ for $i = 1, \ldots, k$, 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 $(\varepsilon,\delta)$-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 are typically calibrated to the sensitivity of the underlying function with respect to single-record modifications of the dataset, defined formally as follows.

*Definition 3.3 (Sensitivity [23]).* The *sensitivity* of a function $q_D : \mathcal{R} \to \mathbb{R}$ that depends on a dataset $D$ is defined as $\Delta_q = \sup_{r\in\mathcal{R}} \sup_{D\sim D'} |q_D(r) - q_{D'}(r)|$. In particular, the sensitivity of a set function $q_D : 2^V \to \mathbb{R}$ is $\sup_{S\subseteq V} \sup_{D\sim D'} |q_D(S) - q_{D'}(S)|$.

The exponential mechanism [40] 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 [40]).* 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 $\varepsilon$, the exponential mechanism $\mathtt{EM}(D, q, \mathcal{R}, \varepsilon)$ 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 $\Delta_q$ is provided alongside $q$ and do not consider it as a separate input.

Theorem 3.5 ([6, 40]). *The exponential mechanism is $\varepsilon$-DP. Moreover,*

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

The utility guarantee in expectation, which we use in this work, is due to [6]. The privacy guarantee is due to [40], as is the 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. The subscript $D$ is omitted when it is clear from context.

*Remark* 4.2. The relevance function $f$ is submodular and the diversity term $d(S)$, being a sum of pairwise distances, is *supermodular*. Hence, the objective $\phi$ is not necessarily submodular.

*Example* 4.3. 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 $v \in V$, and the diversity is 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 $\phi_D$ is called $\Delta$-*decomposable* [32, 42] if it can be written as $\phi_D(S) = \sum_{x \in D} \phi_x(S)$, where each $\phi_x : 2^V \to [0, \Delta]$ depends only on the individual record $x$. A $\Delta$-decomposable function has sensitivity $\Delta$. Prior work has studied DP maximization of decomposable submodular functions [15, 16, 32], as they allow stronger privacy guarantees compared to the general case. Our algorithms for cardinality constraints similarly leverage this structure to provide privacy guarantees that nearly match known lower bounds for decomposable objectives.

*Example* 4.4. 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

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

where $m = |D|$. The normalization ensures that $\phi_x(S) \in [0, 1/m]$. The total objective is the average of these individual contributions:

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

Thus, $\phi$ is $1/m$-decomposable. We now derive the sensitivity bound for $\phi$. If $D = (D' \setminus \{x'\}) \cup \{x\}$, then for any $S \subseteq V$, $|\phi_D(S) - \phi_{D'}(S)| = |\phi_x(S) - \phi_{x'}(S)| \leq 1/m$, and thus $\phi$ has sensitivity bounded by $1/m$.

## 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 shown empirically in Section 8, its complexity scales linearly in $k$, which can become prohibitive when $k$ is large. *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 $\phi'_D(S) = \frac{1}{2} f_D(S) + \lambda d_D(S)$ rather than the objective $\phi_D$.

**Algorithm 1** DP-Greedy

**Input:** Dataset $D$, ground set $V$, monotone submodular $f_D$, pseudometric $d_D$, cardinality $k$, privacy parameter $\varepsilon_0$, diversity parameter $\lambda$.

```
Define φ'_D(·) = (1/2) f_D(·) + λ d_D(·)
Set S_0 ← ∅
for i = 1, ..., k do
    Let N_i ← V \ S_{i−1}
    For all u ∈ N_i, define q^i_D(u) = φ'_D(u | S_{i−1})
    Compute u_i ← EM(D, q^i, N_i, ε_0)
    S_i ← S_{i−1} ∪ {u_i}
end for
return S_k
```

Theorem 5.1. *Suppose $\phi_D$ has sensitivity $\Delta$. Given parameter $\varepsilon_0 > 0$,* DP-Greedy *is $k\varepsilon_0$-DP and $(\varepsilon, \delta)$-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 feasible 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 $\phi_D$ is decomposable (and hence so is $\phi'_D$), DP-Greedy provides a stronger privacy guarantee compared to the general case, as stated next. The assumption that $\phi_D$ is 1-decomposable is without loss of generality, since we can apply our algorithms to the normalized function $\phi_D/\Delta$, and the guarantees for $\phi_D$ incur an additive error scaled by $\Delta$.

Theorem 5.2. *Suppose $\phi'_D$ is 1-decomposable. Then given parameter $\varepsilon_0 \in (0, 1]$ ,* DP-Greedy *(Algorithm 1) is $(\varepsilon, \delta)$-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 $\varepsilon_0 = O(\varepsilon/\sqrt{\log(1/\delta)k})$ ensures $(\varepsilon, \delta)$-DP. However, Theorem 5.2 implies that setting $\varepsilon_0 = O(\varepsilon/\log(1/\delta))$ suffices to ensure $(\varepsilon, \delta)$-DP, avoiding the dependence on $k$ that arises from $k$ compositions 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 [32]. 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 $\phi$. Since $\phi$ is monotone, we may assume that $|\text{OPT}| = k$. Let $S_i$ be the solution at the end of iteration $i$, and define $C_i = \text{OPT} \setminus S_{i-1}$. 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 $d(\text{OPT})$. An adaptation of this lemma is stated next. The proof, as well as additional omitted proofs, is given in the full version [59].

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

Proof Sketch of Theorem 5.1. For query complexity, note that the algorithm performs $k$ iterations of $O(n)$ oracle calls. The privacy follows from the $\varepsilon_0$-DP of the exponential mechanism and composition (Theorem 3.2). For utility, let $f'(S) = f(S)/2$. Consider iteration $i \in [k]$ and condition on $S_{i-1}$. The submodularity and monotonicity of $f'$ imply that $\sum_{u \in C_i} f'(u \mid S_{i-1}) \geq f'(\text{OPT}) - f'(S_{i-1})$. Furthermore, by Lemma 5.3, we have $\sum_{u \in C_i} d(u \mid S_{i-1}) \geq \frac{(i-1)|C_i|}{k(k-1)} d(\text{OPT})$. Thus, the exponential mechanism, executed with the marginal gain quality functions $q_D^i$ of sensitivity $2\Delta$, guarantees that

$$\mathbb{E}[\phi'(u_i \mid S_{i-1})] \geq \max_{u \in V \setminus S_{i-1}} \phi'(u \mid S_{i-1}) - \alpha \geq \frac{\sum_{u \in C_i} \phi'(u|S_{i-1})}{|C_i|} - \alpha \geq \frac{f'(\text{OPT}) - f'(S_{i-1})}{k} + \frac{\lambda(i-1)}{k(k-1)} d(\text{OPT}) - \alpha$$

for $\alpha = O(\varepsilon_0^{-1} \Delta \log n)$. Monotonicity, together with first removing the conditioning on $S_{i-1}$ and then taking expectation over all its realizations, yields

$$\mathbb{E}[\phi'(u_i|S_{i-1})] \geq \frac{f'(\text{OPT}) - \mathbb{E}[f'(S_k)]}{k} + \frac{\lambda(i-1)}{k(k-1)} d(\text{OPT}) - \alpha.$$

Summing over $i \in [k]$ gives

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

Plugging the definitions for $f', \phi'$ and rearranging 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 Buchbinder et al. [9], Mirzasoleiman et al. [41] achieves a $(1 - 1/e - \gamma)$-approximation in expectation using only $O(n \log \gamma^{-1})$ oracle calls. Mitrovic et al. [42] proposed a variant for non-monotone submodular functions, guaranteeing a $(1 - 1/e)/e$-approximation, and adapted it to satisfy differential privacy. 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. It achieves strong utility guarantees while using substantially fewer oracle calls than DP-Greedy. 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. [32] to account for the subsampling step and obtain improved privacy guarantees for decomposable objectives.

**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 $\gamma$, diversity parameter $\lambda$.

1: In DP-NOSG:
   $\phi'_D(\cdot) = \frac{1}{2-\gamma} f_D(\cdot) + \lambda d_D(\cdot)$, $g(i) = k - i + 1$ for all $i \in [k]$
2: In DP-OSG:
   $\phi'_D(\cdot) = f_D(\cdot) + \lambda d_D(\cdot)$, $g(i) = \min\{k, n - i + 1\}$ for all $i \in [k]$
3: Set $S_0 \leftarrow \emptyset$
4: **for** $i = 1, \ldots, k$ **do**
5: Set $N_i \leftarrow V \setminus S_{i-1}$
6: 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.
7: For all $u \in V_i$, define $q_D^i(u) = \phi'_D(u \mid S_{i-1})$
8: Compute $u_i \leftarrow \mathsf{EM}(D, q_D^i, V_i, \varepsilon_0)$
9: $S_i \leftarrow S_{i-1} \cup \{u_i\}$
10: **end for**
11: **return** $S_k$

Algorithm 2 performs greedy selections with respect to a function $\phi'$ that can be either the objective $\phi$ or an auxiliary function, and a utility function $g\colon [k] \to \mathbb{N}$, where $g(i)$ determines the size of the sampled candidate set in iteration $i$. We study two instantiations corresponding to different choices of $g$ and $\phi'$, yielding different 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 $\phi'_D(S) = \frac{1}{2-\gamma} \cdot f_D(S) + \lambda d_D(S)$, and uses the utility function $g(i) = k - i + 1$ for all $i \in [k]$. We obtain the following result.

Theorem 6.1. *Suppose $\phi_D$ has sensitivity $\Delta$. Then,* DP-NOSG *(Algorithm 2) with parameters $\varepsilon_0 > 0$ and $\gamma \in (0, 1)$ is $k\varepsilon_0$-DP and $(\varepsilon, \delta)$-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 feasible set $S$ such that $\mathbb{E}[\phi_D(S)] \geq (\frac{1}{2} - \gamma) \cdot \phi_D(\text{OPT}) - O\left(\frac{k\Delta \log n}{\varepsilon_0}\right)$ and makes $O(n \log k \log \gamma^{-1})$ oracle calls.*

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 feasible set $S$ such that $\mathbb{E}[\phi(S)] \geq (\frac{1}{2} - \gamma)\phi(\text{OPT})$ 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 $\phi'$, if $\phi$ is 1-decomposable, then so is $\phi'$.

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 $(\varepsilon, \delta)$-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 $\varepsilon_0 = O(\varepsilon/\log(1/\delta))$ suffices to ensure $(\varepsilon, \delta)$-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 $\phi'$. We prove the following.

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.* $\mathbb{E}[\phi'(u_i \mid S_{i-1})] \geq \frac{1-\gamma}{|M_i|} \sum_{v \in M_i} \phi'(v \mid S_{i-1}) - \frac{4\Delta \log n}{\varepsilon_0}$

Intuitively, the proof leverages the guarantee of the EM to show that the expected marginal gain of the selected $u_i$ is at least the average gain across elements in $V_i \cap M_i$ up to a bounded additive error. Since this intersection is non-empty with probability at least $1-\gamma$, and each element in $M_i$ has an equal probability of being sampled into $V_i$, the law of total expectation yields the desired bound. This follows from the definition of $M_i$ as the set of highest contributing elements. The proof is provided in [59].

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 [59]. Privacy follows from the composition theorems (Theorem 3.2) and the $\varepsilon_0$-DP guarantee of the exponential mechanism. We thus focus on the utility guarantee.

Let $f' = \frac{1}{2-\gamma} f$ and $\phi' = f' + \lambda d$. 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

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

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

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

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

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

Rearranging terms and substituting $f'$ gives:

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

□

Next, we address the stronger guarantee of Theorem 6.3, which applies when the function $\phi$ is decomposable. We build on the technique of [32], which bounds the privacy loss of the greedy algorithm for submodular maximization by the sum of expected marginal gains of a single individual's function $f_x$ across all rounds, where the expectation is taken over a distribution induced by the functions of the other individuals. This quantity is bounded with high probability by the sum of realized marginal gains $\sum_{i=1}^{k} [f_x(S_i) - f_x(S_{i-1})] = f_x(S_k) - f_x(S_0) \leq \Delta$, which telescopes to a constant independent of $k$. We extend this technique to Algorithm 2 by providing a unified analysis for the subsampling steps and the EM. Specifically, we prove that for any deterministic and adaptive choice of $V_i$, the privacy loss can be bounded by a sum of expected marginal gains, which is bounded by a telescoping sum with high probability, analogous to [32]. This extension enables our utility bound to outperform a naive composition-based analysis of repeated EM, and approach existing lower bounds.

Proof Sketch of Theorem 6.3. Let $D$ and $D'$ be two datasets such that $(D \setminus D') \cup (D' \setminus D) = \{x\}$. We first bound the output probability ratio for this add/remove pair, and then apply a reduction to derive the bound for neighboring datasets under the substitution neighboring relation. Suppose that instead of a set, Algorithm 2 outputs the sequence of picked elements in their picking order. Let $U = (u_1, \ldots, u_k)$ be such any sequence. By the definition of the exponential mechanism, the probability of selecting element $u$ at iteration $i$ given dataset $D$ and a sampled set of candidates $V_i$ is:

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

where we define $\beta_D^i(u) = \sum_{x \in D} \phi_x'(u \mid \{u_1, \ldots, u_{i-1}\})$. For brevity, denote Algorithm 2 by $\mathcal{G}$, and drop irrelevant parameters. By the chain rule and the law of total probability over the uniform sampling of $V_i$, the probability ratio $\frac{\mathbf{Pr}[\mathcal{G}(D)=U]}{\mathbf{Pr}[\mathcal{G}(D')=U]}$ is bounded by:

$$\left(\prod_{i=1}^{k} \frac{\exp(\frac{\varepsilon_0}{2} \cdot \beta_D^i(u_i))}{\exp(\frac{\varepsilon_0}{2} \cdot \beta_{D'}^i(u_i))}\right)\left(\prod_{i=1}^{k} \frac{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2} \cdot \beta_{D'}^i(u))}{\sum_{u \in T_i} \exp(\frac{\varepsilon_0}{2} \cdot \beta_D^i(u))}\right). \tag{1}$$

where $T_i$ is a worst-case realization of $V_i$ containing $u_i$. We consider two cases.

**Case 1:** $D = D' \cup \{x\}$**.** By the 1-decomposability of $\phi'$, the first factor in (1) is bounded by $\exp(\frac{\varepsilon_0}{2} \sum_i (\beta_D^i(u_i) - \beta_{D'}^i(u_i))) \leq \exp(\varepsilon_0/2)$. The second factor is at most 1 since $\beta_x^i \geq 0$.

**Case 2:** $D' = D \cup \{x\}$**.** The first factor in (1) is at most 1 . The second factor is a product of expectations $\prod_{i=1}^{k} \mathop{\mathbb{E}}_{u \sim P_i} [\exp(\frac{\varepsilon_0}{2} \beta_x^i(u))]$, where $P_i(v) \propto \exp(\varepsilon_0 \beta_D^i(v))$ for $v \in T_i$. Using the concentration bounds from [15, 32], the product of expectations is bounded by $\exp((e^{\varepsilon_0/2} - 1)(3 + \log(1/\delta)))$ with probability $1 - \delta$.

Combining these add/remove cases and applying a standard reduction establishes an $((e^{\varepsilon_0/2} - 1)(4 + \log(1/\delta)), \delta)$-DP guarantee for neighboring datasets under the substitution relation. □

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

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 remaining unselected elements of OPT as a function of $d(\text{OPT})$. While this yields tight guarantees for DP-Greedy and DP-NOSG, applying the same technique with the current choice of $g$ leads only to a $1/6-\gamma$ approximation. We therefore take a different approach and instantiate Algorithm 2 with $\phi' = \phi$, i.e., in an oblivious manner. Instead of relying on Lemma 5.3, we bound the total distance to the remaining elements of OPT using the marginal gain $d(\text{OPT} \cup S_i) - d(S_i)$ (Lemma 6.7). This yields a new progress inequality for the oblivious variant, leading to a tighter approximation factor.

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 $\phi'$.

Theorem 6.5. *Suppose $\phi$ has sensitivity $\Delta$. Then,* DP-OSG *(Algorithm 2) outputs a feasible set $S$ such that*

$$\mathbb{E}[\phi(S)] \geq (1-(\tfrac{2}{e})^{(1-\gamma)(1-1/k)}) \cdot \phi(\text{OPT}) - O(\frac{k\Delta \log n}{\varepsilon_0})$$

*and makes $O(n \log \gamma^{-1})$ oracle calls.*

Lemma C.4 in [59] 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 feasible set $S$ such that $\mathbb{E}[\phi(S)] \geq (1-(\tfrac{2}{e})^{(1-\gamma)(1-1/k)}) \cdot \phi(\text{OPT})$, and makes $O(n \log \gamma^{-1})$ oracle queries.*

The proof of Theorem 6.5 relies on the following structural progress lemma, whose proof is deferred to the full version [59].

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

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

Proof Sketch of Theorem 6.5. Intuitively, since DP-OSG evaluates $O(\frac{n}{k}\log\gamma^{-1})$ candidates in each iteration, the number of oracle calls across $k$ iterations is $O(n\log\gamma^{-1})$. We focus here on the utility guarantee.

Let $\alpha = 4\varepsilon_0^{-1}\Delta \log n$. Consider any iteration $2 \leq i \leq k$ and condition on having selected set $S_{i-1}$ after iteration $i-1$. By submodularity and monotonicity, $\sum_{u\in C_i} f(u \mid S_{i-1}) \geq f(\text{OPT}\cup S_{i-1}) - f(S_{i-1})$. By Lemma 6.7,

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

Let $M_i \subseteq N_i$ be a set of size $g(i) = \min\{k, |N_i|\}$ that maximizes $\sum_{v\in M_i}\phi(v \mid S_{i-1})$. By definition of $M_i$, we have

$$\mathbb{E}[\phi(u_i \mid S_{i-1})] \geq \tfrac{1-\gamma}{g(i)} \sum_{v\in M_i} \phi(v \mid S_{i-1}) - \alpha$$

$$\geq \tfrac{1-\gamma}{k} \sum_{v\in C_i} \phi(v \mid S_{i-1}) - \alpha \geq (1-\gamma)\left(\frac{\phi(\text{OPT})-\phi(S_{i-1})}{k\left(1+\frac{k-1}{i-1}\right)}\right) - \alpha$$

The first inequality follows by an argument analogous to that in the proof of Lemma 6.4. The second inequality follows from the definition of $M_i$, together with the fact that $|C_i| \leq g(i) = |M_i| \leq k$. Removing the conditioning on $S_{i-1}$ and taking an expectation over all its possible realizations yields

$$\mathbb{E}[\phi(u_i \mid S_{i-1})] \geq (1-\gamma)\left(\frac{\phi(\text{OPT})-\phi(S_{i-1})}{k\left(1+\frac{k-1}{i-1}\right)}\right) - \alpha \tag{2}$$

Rearranging gives

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

By recursively applying the last inequality after removing the conditioning on $S_{i-i}$, taking the expectation over all its realizations, we get

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

where the second inequality is by Lemma C.4 in [59]. The theorem now follows by rearranging. □

# 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 achieving a $(1/2-\gamma)$-approximation in $O(n^2 + \gamma^{-1}nk^2\log k)$ oracle calls[3] starting from an initial base containing the highest-scoring feasible pair of elements. However, privatizing this approach is non-trivial: additive DP noise prevents both verifying local optimality and guaranteed selection of strictly improving swaps.

Our algorithm, DP-SLS (Algorithm 3), executes a fixed number of local search iterations, using the exponential mechanism to privately select an approximately best available swap from a subsampled candidate set. To avoid forced suboptimal swaps when no improving swap exists, we introduce a dummy swap that allows the current solution to remain unchanged. Finally, we apply the exponential mechanism to privately select and output the highest-scoring solution observed throughout the execution.

Our analysis exploits the structure of the underlying metric space to establish sufficient expected progress and achieve a near-optimal approximation ratio, while avoiding higher-order error terms introduced by continuous relaxations [48]. We show that whenever the expected objective value in a given iteration is low, the subsequent iteration increases it significantly. In particular,

[3]The algorithm in [8] attains a 1/2-approximation but is not polynomial-time. The authors' proposal to perform only swaps yielding at least a $\gamma$-improvement at each iteration, rather than any improvement, leads to the stated guarantees.

the first iteration already guarantees a sufficiently large expected objective value, enabling a multiplicative progress argument that yields a $(1/2-\gamma)$-approximation within $O(\gamma^{-1}k\log k)$ iterations.

By basing our approach on local search, we depart from predominantly greedy methods used in DP submodular maximization. This approach yields a combinatorial proof with a tighter additive error term, while simultaneously eliminating the $O(n^2)$ factor present in the complexity of [8]. We further reduce complexity by subsampling candidate swaps in each iteration, ultimately requiring $O(\gamma^{-1}nk\log k)$ oracle calls. This provides a substantial improvement over prior work while maintaining the same near-optimal expected approximation factor.

Theorem 7.1. *Suppose $\phi_D : 2^V \to \mathbb{R}$ has sensitivity $\Delta$, 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 $(\varepsilon,\delta)$-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 feasible set $S$ such that*

$$\mathbb{E}[\phi_D(S)] \geq \left(\tfrac{1}{2}-\gamma\right)\cdot\phi_D(\mathrm{OPT}) - O\left(\tfrac{\Delta}{\varepsilon_0}\left(k\log n + \log\tfrac{1}{\gamma}\right)\right),$$

*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 $\varepsilon_0 = O(\varepsilon/\sqrt{\gamma^{-1}k\log k\log\delta^{-1}})$ ensures $(\varepsilon,\delta)$-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 feasible 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 $\Delta$-sensitive functions. The technique of Gupta et al. [32] 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.

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 [59].

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. Let OPT be an optimal solution. Since $\phi$ 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 $i-1$.

Lemma 7.4. *Suppose $\mathcal{M}$ has rank $k > 2$. There exists a bijection $h : S_{i-1} \to \mathrm{OPT}$ such that $h(u) = u$ for every $u \in S_{i-1}\cap\mathrm{OPT}$, and $\mathrm{OPT} - h(u) + u \in \mathcal{I}$ for every $u \in S_{i-1}$. Moreover,*

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

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

**Algorithm 3** DP Sample Local Search (DP-SLS)

**Input:** Dataset $D$, ground set $V$, submodular $f_D$, pseudometric $d_D$, matroid $\mathcal{M}$ of rank $k$, privacy parameter $\varepsilon_0$, utility parameter $\gamma \in (0,1]$, diversity parameter $\lambda$.

1: Let $\phi_D(\cdot) = f_D(\cdot) + \lambda d_D(\cdot)$
2: Let $S_0$ be an arbitrary base of $\mathcal{M}$.
3: Set $T \leftarrow \lceil \frac{2k\log(8k)}{\gamma(1-1/e)} \rceil + 1$
4: **for** $i = 1, \ldots, T$ **do**
5: Sample a subset $V_i \subseteq V$ of size $\lceil \frac{n}{k} \rceil$ uniformly at random
6: Let $W_i \leftarrow \{(u,v) : u \in S_{i-1}, v \in V_i \setminus S_{i-1}, S_{i-1} - u + v \in \mathcal{I}\} \cup \{(w,w)\}$, where $w \in S_{i-1}$ is arbitrary.
7: For all $(u,v) \in W_i$, define $q_D^i(u,v) = \phi_D(S_{i-1}-u+v)$.
8: Compute $(u_i, v_i) \leftarrow \mathtt{EM}(D, q_D^i, W_i, \varepsilon_0)$
9: Let $S_i \leftarrow S_{i-1} - u_i + v_i$
10: **end for**
11: For all $i = 1, \ldots, T$, define $s_D(i) = \phi_D(S_i)$.
12: Compute $i^* \leftarrow \mathtt{EM}(D, s_D, [T], \varepsilon_0)$
13: **return** $S_{i^*}$

Lemma 7.5. *For every iteration $i \in [T]$, we have*

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

Intuitively, Lemma 7.4 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 $\frac{1}{k}(\phi(\mathrm{OPT}) - 2\phi(S_{i-1}))$. To prove Lemma 7.5, 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 [59].

Proof Sketch of Theorem 7.1. The privacy guarantee follows from the $\varepsilon_0$-DP of the exponential mechanism and composition theorems (Theorem 3.2) over $O(\gamma^{-1}k\log k)$ iterations. For complexity, each iteration requires $O(|S_{i-1}|\cdot|V_i|) = O(n)$ oracle calls due to the subsampling step. No additional calls are required in Line 12 as all values are computed during the preceding step. In total, the algorithm makes $O(\gamma^{-1}nk\log k)$ oracle calls. For utility, we first obtain $\mathbb{E}[\phi(S_1)] = \Omega(\frac{\phi(\mathrm{OPT})}{k})$ using Lemma 7.5, assuming $\phi(\mathrm{OPT})$ is large enough that the bound in the theorem is non-trivial. Conditioning on the selection of $S_{i-1}$, Lemma 7.5 implies:

$$\mathbb{E}\left[\phi(S_i)\right] \geq \left(1 + \frac{\gamma(1-1/e)}{k}\right)\mathbb{E}\left[\phi(S_{i-1})\right] \tag{3}$$

for every $i$ for which the bound $\mathbb{E}[\phi(S_{i-1})] < \phi(\mathrm{OPT})/(2+\gamma) - k\alpha$ holds. Thus, there must exist some $i$ for which the bound fails; otherwise, recursive application of (3) yields $\mathbb{E}[\phi(S_T)] > \phi(\mathrm{OPT})$ in contradiction. The proof is completed by applying the law of total expectation over the sampling of $S_1, \ldots, S_T$ and concluding that the selected $S_{i^*}$ has expected quality at least $\max_{i\in[T]}\mathbb{E}[\phi(S_i)]$ up to a bounded additive error. This maximum is at least $\phi(\mathrm{OPT})/2$ up to the stated additive error. □

# 8 EXPERIMENTS

In this section, we evaluate the utility and efficiency of our proposed algorithms across two data summarization applications: Uber pickup location summarization and Amazon product summarization. Based on these applications, we characterize the trade-offs between privacy, utility, and complexity.

**Summary of Findings.** For cardinality constraints, 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 1% of the non-private baseline even at a strict $\varepsilon = 0.1$ budget. By shifting the query complexity to logarithmic or zero dependence on $k$, `DP-NOSG` and `DP-OSG` achieve speedups of 14× and 50× at $k = 100$. For matroid constraints, `DP-SLS` maintains competitive utility, staying within 1% of the non-private baseline at $\varepsilon = 0.1$. While `DP-SLS` reduces the total number of oracle calls, its fixed iteration count leads to an execution time overhead of up to 2.2× compared to the non-private baseline; however, despite scaling more slowly with $k$, it exhibits the improved scaling with $n$ predicted by our analysis.

## 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 $\varepsilon = 0.1$ and $\delta = |D|^{-1.5}$ where $D$ is the sensitive dataset. For cardinality constraints, our per-iteration privacy parameter $\varepsilon_0$ 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 [15, 42] and pick the initialization that achieves $(\varepsilon, \delta)$-DP with the smallest noise scale. 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 $\lambda = 0.5$. Results are averaged over 10 runs.

**Baselines.** As MSD has not been previously explored in a DP setting, we compare our proposed methods against non-private, random, and DP submodular maximization baselines.

- **`Greedy`** (Cardinality): The standard non-private greedy algorithm for cardinality constraints [8].
- **`LS`** (Matroid): The non-private local search algorithm for general matroid constraints [8] discussed in Section 7.
- **`Random`** (Cardinality/Matroid): Selects a random feasible subset. Since the output distribution of the exponential mechanism converges to uniform as $\varepsilon \to 0$, this baseline serves as a lower bound on utility for private algorithms. Following prior work [15, 16, 42], we incorporate `Random` to contextualize the performance of our algorithms.
- **`DP-RelG`** (Cardinality / Matroid): The DP greedy algorithm of [42] for DP submodular maximization. It greedily selects feasible elements using the `EM`, applied only to the submodular component of our objective (i.e., $\lambda = 0$). This baseline isolates the effect of removing the diversity term by optimizing relevance alone.
- **`DP-Mix`** (Cardinality): A hybrid baseline combining the random baseline with `DP-RelG`: $k/2$ items are selected uniformly at random, and the remaining $k/2$ are selected using `DP-RelG`.

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

## 8.2 Datasets, Objective Functions, and Constraints

We evaluate our approach on two real-world data summarization tasks: Uber location selection and Amazon product selection.

**Uber Location Selection.** The first application involves selecting a representative summary of Uber pickup locations in Manhattan. This setup adapts the location selection experiments from [15, 42, 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.** We consider a sensitive dataset $D$ of $m = 600,000$ Uber pickup locations in Manhattan from 2014 [27]. Each record consists of longitude and latitude coordinates. For the public ground set, we follow the established methodology and consider a grid of $n$ points over Manhattan, where by default $n = 1000$.
**Objective function.** For relevance, we adopt the Uber "convenience" score from Mitrovic et al. [43], defined as follows. For a passenger location $a = (x_a, y_a)$ and a grid point $b = (x_b, y_b)$ define $c(a, b) = 2 - \frac{2}{1+e^{-200\|a-b\|_1}}$, where $\|a - b\|_1 = |x_a - x_b| + |y_a - y_b|$ is the Manhattan distance between the two points. The relevance of a set $S$ is defined as $f_D(S) = \frac{1}{m}\sum_{a\in D} \min_{b\in S} c(a, b)$ with $f_D(\emptyset) = 0$. This function is monotone, submodular, and $1/m$-decomposable (See Appendix F in [59] ). However, as observed in [44], it 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 distances between the selected points: $d(S) = \sum_{b,b'\in S} \frac{\|b-b'\|_1}{M}$, where $M$ is an upper bound on the distance within the area such that $\|b - b'\|_1 \in [0, 1]$
**Constraints.** We evaluate this task under a cardinality constraint $k$.

**Amazon Product Selection.** The second 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 [36], 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 3000 products from the *Health Care* category as our ground set $V$ and discretize their prices into four bins. The dataset $D$ contains the purchase records of 1,317,193 users.
**Objective function.** We consider the objective function from our running example (Example 4.4), where the goal is to select a subset $S \subseteq V$ maximizing user reach, with diversity given by the Jaccard distance over category sets.
**Constraints.** We consider 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. We refer to the latter setting as Amazon-Partition. We also conducted experiments under a cardinality constraint. The scalability trends were similar to those observed for Uber, which is expected since the two settings differ primarily in the cost of evaluating the objective function rather than in the algorithmic behavior. We likewise observed similar utility trends, and therefore defer these results to Appendix G in [59]

## 8.3 Utility Analysis

We examine how the selected subset size $k$ and the privacy parameter $\varepsilon$ impact the MSD objective value $(1-\lambda)f_D(S) + \frac{2\lambda}{k(k-1)}d(S)$. By default, we set $k = 20$ for the Uber experiment, $k = 6$ for the Amazon-Partition experiment.

**Impact of $k$.** We evaluate the impact of the rank $k$ on the objective value (Figure 7). Overall, our algorithms perform competitively with the non-private baselines and consistently outperform the other DP baselines. For the Uber dataset, our three algorithms remain within 3.4% of the non-private `Greedy` baseline on average, with the gap increasing in $k$ due to accumulated DP noise. In contrast, `DP-Mix` is on average 17% below `Greedy`, `DP-RelG` is 20% below, and `Random` is 26% below. For Amazon-Partition, the utility of `DP-SLS` remains nearly identical to the non-private `LS` baseline across the examined range, and is sometimes even slightly better (within 0.1% on average). In contrast, the baselines `DP-RelG` and `Random` perform substantially worse, remaining approximately 25% and 30% below `LS`, respectively. We also note that the objective value need not increase monotonically with $k$. This is because the normalization factor of the diversity increases with $k$, and stricter partition constraints for smaller $k$ may restrict the algorithm to naturally more diverse solutions.

**Impact of $\varepsilon$.** We evaluate the impact of the privacy parameter $\varepsilon$ on the resulting objective value (Figure 6). As expected, a larger privacy budget improves the utility of all DP algorithms. For the Uber dataset, all our algorithms significantly outperform the DP baselines and achieve utility comparable to the non-private `Greedy` baseline, reaching within 1% even at $\varepsilon = 0.1$. In contrast, `DP-Mix`, `DP-RelG`, and `Random` remain substantially lower, at 19%, 27%, and 34% below, respectively. For Amazon-Partition, we similarly find that at $\varepsilon = 0.1$, `DP-SLS` reaches utility within 1% of the non-private `LS` baseline and significantly outperforms `DP-RelG` and `Random`, which remain 38% and 48% below `LS`, respectively. `DP-RelG` slightly benefits from lower privacy budgets, as the increased randomness induces more diverse selections, resulting in performance closer to `Random` in this regime.

**Impact of $\lambda$.** We evaluate the robustness of our algorithms with respect to $\lambda$, which controls the trade-off between relevance and diversity. The results, shown in Figure 8, demonstrate that the utility of our algorithms relative to the corresponding non-private baselines remains consistently high across all examined values $\lambda \in [0, 0.8]$, indicating that the observed performance is largely insensitive to the choice of $\lambda$. For the Uber dataset, the utility gap from the non-private `Greedy` baseline remains below 1% on average for `DP-Greedy` and `DP-NOSG`, and below 3% for `DP-OSG`. In contrast, the other DP baselines perform substantially worse, with `Random` reaching up to 38% lower utility on average. For Amazon-Partition, the utility gap of `DP-SLS` from the non-private `LS` baseline remains below a marginal 0.1% across all examined values of $\lambda$.

As expected, the performance of relevance-only baselines deteriorates as $\lambda$ increases as diversity becomes more dominant in the objective. When $\lambda = 0$, corresponding to relevance-only optimization, our algorithms achieve essentially the same utility as `DP-RelG`. This shows that our approach recovers the performance of prior work in this special case while naturally extending to more general relevance-diversity objectives.

## 8.4 Performance Analysis

We evaluate the computational efficiency of our algorithms by measuring both query complexity and execution time (Figures 2 and 4). 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 [39]. Nevertheless, we also report execution time to provide additional practical insights. Note that we do not attempt to optimize the oracle evaluation time.

**Number of Oracle Calls.** Figure 2 reports the number of oracle calls as $k$ varies. For cardinality constraints, `DP-Greedy` matches the non-private `Greedy` baseline, scaling linearly in $k$, while `DP-NOSG` exhibits a logarithmic dependence and significantly reduces query complexity (e.g., 15× fewer calls than `Greedy` at $k = 100$ on Uber). `DP-OSG` is even more efficient, with a query count essentially independent of $k$, achieving up to a 57× reduction at $k = 100$ while maintaining comparable utility. For Amazon-Partition, the query count of the non-private `LS` baseline is dominated by its initialization phase, which requires an exhaustive $\Theta(n^2)$ search over feasible pairs and is independent of $k$; in practice, the subsequent local search terminates after fewer iterations than the worst-case bound. In contrast, `DP-SLS` avoids this exhaustive initialization and maintains a lower query count.

Figure 3 shows the dependence on $n$. On the Uber dataset, all algorithms exhibit linear scaling in $n$, while our `DP-NOSG` and `DP-OSG` achieve a smaller slope due to their improved dependence on $k$. For Amazon-Partition, `LS` exhibits quadratic scaling in $n$ due to its initialization phase, which involves a brute-force search over pairs of elements, whereas our `DP-SLS` scales linearly.

**Execution Time.** Figure 4 illustrates execution time as $k$ varies. For cardinality constraints, the trends align with our query complexity analysis. `DP-NOSG` exhibits a logarithmic dependence on $k$, outperforming both `Greedy` and all DP baselines; on Uber ($k = 100$), it is 14× faster than `Greedy`. `DP-OSG` is even more efficient, achieving a 50× speedup while preserving comparable utility.

For Amazon-Partition, `DP-SLS` is slower than the non-private `LS` baseline despite its fewer oracle calls. This discrepancy arises because the `LS` bottleneck is dominated by pair evaluations, which are computationally cheaper in this application than evaluating the larger sets required for potential swaps. Moreover, the non-private baseline often converges to a local optimum in substantially fewer iterations than the fixed number required by the `DP-SLS` theoretical analysis. Overall, `DP-SLS` is approximately 2× slower than `LS` on average. While both algorithms may be slow for interactive settings, `DP-SLS` uniquely provides formal privacy and utility guarantees. In

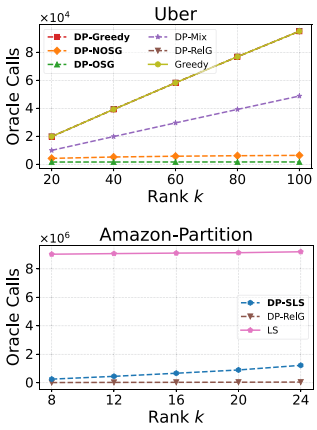


**Figure 2: Number of oracle calls as the rank $k$ varies.**

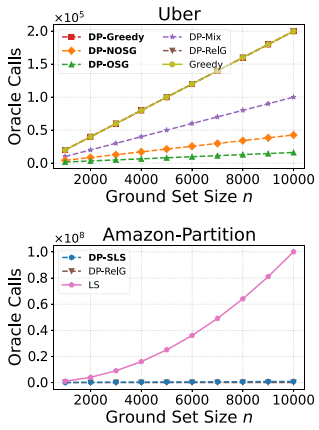


**Figure 3: Number of oracle calls as the ground set size $n$ varies.**

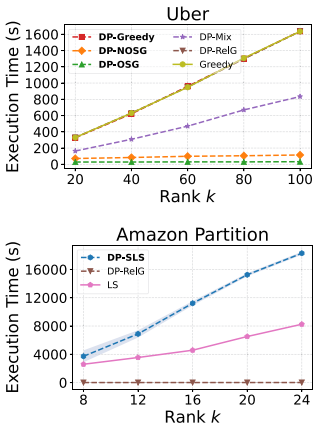


**Figure 4: Execution time (seconds) as the rank $k$ varies.**

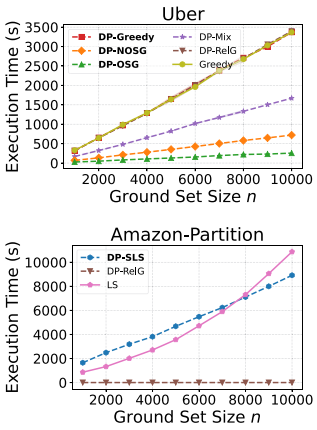


**Figure 5: Execution time (seconds) as the ground set size $n$ varies.**

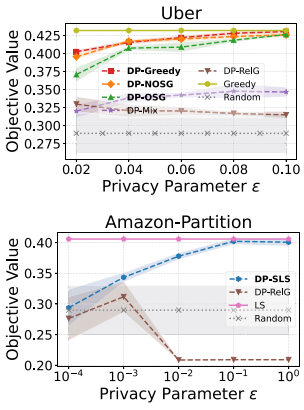


**Figure 6: Objective value as the privacy parameter $\varepsilon$ varies.**

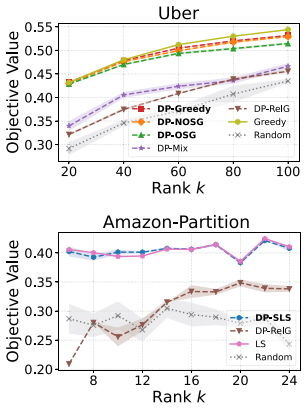


**Figure 7: Objective value as the rank (maximal size of a feasible solution) $k$ varies.**

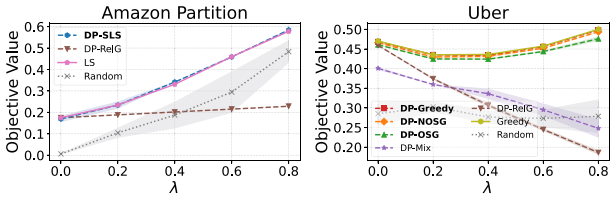


**Figure 8: Objective value as the parameter $\lambda$ varies.**

contrast, DP-RelG is significantly faster, but achieves substantially lower utility and lacks a constant-factor approximation guarantee.

Figure 5 depicts scalability as $n$ varies. For cardinality constraints, DP-OSG and DP-NOSG are the most scalable DP methods, and their execution-time trends match the query-complexity trends. For Amazon-Partition, DP-SLS exhibits the improved scalability in $n$ predicted by our bottleneck analysis; for sufficiently large ground sets ($n \geq 8000$), it becomes faster than the LS baseline.

## 9 CONCLUSION AND LIMITATIONS

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 general matroid constraints. These algorithms achieve utility guarantees comparable to those of their non-private counterparts while also offering improved complexity, making them attractive even when privacy is not required. Experiments on real-world datasets demonstrate that our approach attains high utility and substantially improves practical efficiency for cardinality constraints.

Our work has several limitations that suggest interesting directions for future research. First, as discussed in Remark 7.3, our current guarantees for matroid constraints apply to arbitrary $\Delta$-sensitive functions and do not exploit decomposability to obtain stronger utility bounds. Achieving such improvements appears to be non-trivial. Second, our fastest $O_\gamma(n)$ oracle-call algorithm provides a constant-factor approximation but does not attain the tight $1/2$ ratio. It remains open whether a near-optimal $(1/2-\gamma)$-approximation can be achieved for cardinality constraints with $O_\gamma(n)$ calls. Third, while Algorithm 3 has strictly better worst-case query complexity than the non-private baseline of Borodin et al. [8], it can be slower in practice in some regimes due to realizing its worst-case complexity in every execution. Developing practical accelerations while preserving its high utility is an interesting direction. Finally, extending our framework to additional diversity models remains an important avenue for future work.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback. This research was supported by the Israel Science Foundation (ISF) under grant 2707/22 of the Breakthrough Research Grant (BRG) Program.

# 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 $\kappa$-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] 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.

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

[11] 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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Anamay Chaturvedi, Huy Nguyen, and Thy Dinh Nguyen. 2023. Streaming submodular maximization with differential privacy. In *International Conference on Machine Learning*. PMLR, 4116–4143.

[17] 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.

[18] 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.

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

[20] 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.

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

[22] 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

[23] 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.

[24] 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.

[25] 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

[26] 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

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

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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).

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

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

[36] 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).

[37] 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.

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

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Marko Mitrovic, Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. 2018. Data summarization at scale: A two-stage submodular approach. In *International Conference on Machine Learning*. PMLR, 3596–3605.

[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] 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.

[53] 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

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

[55] 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.

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

[57] 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.

[58] Ron Zadicario and Tova Milo. 2026. Differentially Private Submodular Maximization with a Knapsack Constraint. *arXiv preprint arXiv:2606.14951* (2026). https://arxiv.org/abs/2606.14951

[59] Ron Zadicario and Tova Milo. 2026. Fast and Private Max-Sum Diversification. *arXiv preprint arXiv:2607.17196* (2026). https://arxiv.org/abs/2607.17196

[60] 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.

[61] 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*.