# A Lower Bound for Parallel Submodular Minimization

Eric Balkanski
Harvard University
ericbalkanski@g.harvard.edu

Yaron Singer
Harvard University
yaron@seas.harvard.edu

**Abstract**

In this paper, we study the problem of submodular function minimization in the adaptive complexity model. Seminal work by Grötschel, Lovász, and Schrijver [GLS81, GLS88] shows that with oracle access to a function $f$, the problem of submodular minimization can be solved exactly with poly$(n)$ many queries to $f$. A long line of work has since then been dedicated to the acceleration of submodular minimization. In particular, recent work obtains a (strongly) polynomial time algorithm with $\tilde{\mathcal{O}}(n^3)$ query complexity [LSW15]. A natural way to accelerate computation is via parallelization, though very little is known about the extent to which submodular minimization can be parallelized.

A natural measure for the parallel runtime of a black-box optimization algorithm is its *adaptivity*, as recently introduced in the context of submodular maximization [BS18a]. Informally, the adaptivity of an algorithm is the number of sequential rounds it makes when each round can execute polynomially-many function evaluations in parallel. In the past two years there have been breakthroughs in the study of adaptivity for both submodular maximization and convex minimization, in particular an exponential improvement in the parallel running time of submodular maximization was obtained with a $\mathcal{O}(\log n)$-adaptive algorithm [BS18a]. Whether submodular minimization can enjoy, thanks to parallelization, the same dramatic speedups as submodular maximization is unknown. To date, we do not know of any polynomial time algorithm for solving submodular minimization whose adaptivity is subquadratic in $n$.

We initiate the study of the adaptivity of submodular function minimization by giving the first non-trivial lower bound for the parallel runtime of submodular minimization. We show that there is no $o(\frac{\log n}{\log \log n})$-adaptive algorithm with poly$(n)$ queries which solves the problem of submodular minimization. This is the first adaptivity lower bound for unconstrained submodular optimization (whether for maximization or minimization) and the analysis relies on a novel and intricate construction of submodular functions.

# 1 Introduction

In this paper we study the adaptive complexity of minimizing a submodular function. Submodular minimization is a fundamental optimization problem in theoretical computer science, operations research, and machine learning. It generalizes classical problems such as minimum cut and clustering, and has a broad range of applications in domains such as computer vision [BVZ01, KKT08, KT10, JB11, JBS13] and speech analysis [LB10, LB11].

The problem of minimizing a submodular function $f : 2^N \to \mathbb{R}$ over $n$ elements with black-box access to $f$ can be solved exactly using $\tilde{\mathcal{O}}(n^5)$ queries (function evaluations), using the ellipsoid algorithm, as first shown by Grötschel, Lovász, and Schrijver [GLS81, GLS88]. Since the ellipsoid algorithm is slow in practice and $\tilde{\mathcal{O}}(n^5)$ queries becomes infeasible even for moderate values of $n$, there has been a long line of work devoted to the acceleration of submodular minimization (e.g. [Cun85, Sch00, IFF01, Iwa02, CJK14, LJJ15, LSW15, CLSW17, HRRS19, ALS19]). Recent breakthroughs have led to the algorithms with the best known query complexity for submodular minimization under different regimes: $\tilde{\mathcal{O}}(n^3)$ for (strongly) polynomial time algorithms [LSW15], $\tilde{\mathcal{O}}(n^2 \log M)$ for weakly-polynomial time algorithms (where $|f(S)| \leq M$ for all $S \subseteq N$) [LSW15], $\tilde{\mathcal{O}}(nM^2)$ for pseudo-polynomial time algorithms [CLSW17, ALS19], and $\tilde{\mathcal{O}}(n/\epsilon^2)$ for $\epsilon$-approximate algorithms [ALS19].

An effective way to accelerate computation is via *parallelization*. A convenient metric for parallelism in black-box optimization models is *adaptivity*, as recently introduced in the context of submodular optimization to quantify the information theoretic complexity of black-box optimization in a parallel computation model [BS18a]. Informally, the adaptivity of an algorithm is the number of sequential rounds it makes when each round can execute polynomially-many queries in parallel. The concept of adaptivity is heavily studied in computer science and optimization as it provides a measure of efficiency of parallel computation. In the past two years there have been breakthroughs in the study of adaptivity in two related domains: submodular maximization and convex minimization.

## 1.1 Distant relatives: submodular maximization and convex minimization

Submodular minimization has two distant relatives: submodular maximization and convex minimization, both have a well established set of results on adaptivity. Due to the structure of submodular functions, submodular maximization is a natural distant relative of submodular minimization. For convex minimization, due to the seminal result by Lovász [Lov83] we know that an optimal solution to submodular minimization can be obtained by minimizing a convex relaxation of the submodular function. Therefore, upper bounds on submodular minimization are intimately related to those for convex minimization.

For submodular maximization, a recent line of work initiated by this set of authors in [BS18a] studies the adaptive complexity of submodular maximization. The first main result in [BS18a] is that for the canonical problem of maximizing a submodular function under a cardinality constraint one can obtain a constant factor approximation in $\mathcal{O}(\log n)$ rounds. This is an exponential speedup over any previously known constant factor approximation algorithm for this problem. Moreover, the second main result in [BS18a] is that no algorithm can obtain a $\frac{1}{\log n}$-approximation in $o(\frac{\log n}{\log \log n})$ rounds. Since these two results, there has been a long line of work showing how to obtain tight approximation guarantees in logarithmically many rounds for some of the most general problems of submodular maximization [BRS19a, EN19, FMZ19, CQ19b, BBS18, CFK19, ENV19, CQ19a, BRS19b]. The most relevant is the result by Chen, Feldman, and Karbasi [CFK19] showing that for the problem of unconstrained submodular maximization, whose optimal approximation is $1/2$, one can obtain a $1/2 - \epsilon$ approximation in $\tilde{\mathcal{O}}(1/\epsilon)$ adaptive rounds.

For convex minimization, despite a great deal of interest in parallelizing convex optimization, dramatic improvements in parallel runtime as those for submodular maximization cannot be obtained. Over 25 years ago, Nemirovski showed a lower bound of $\tilde{\Omega}(n^{1/3})$ rounds of adaptivity for minimizing an $n$-dimensional convex functions up to accuracy $\epsilon$ over an $\ell_\infty$ ball [Nem94]. This lower bound has only recently been improved to $\tilde{\Omega}(n^{1/2})$ in breakthrough work [BJL+19]. A similar construction to that of Nemirovski can be applied to show $\tilde{\Omega}(n^{1/3})$ lower bounds over $\ell_2$ spaces [BS18b] and more intricate constructions can be used to obtain lower bounds for minimization over general $\ell_p$ spaces [DG18].

### 1.2 Main result: an adaptivity lower bound for submodular minimization

While the adaptive complexity of submodular maximization and convex minimization is quite well understood, we have a very limited understanding of the adaptive complexity of submodular minimization. The polynomial time algorithm in [LSW15] with $\tilde{\mathcal{O}}(n^3)$ query complexity requires $\tilde{\mathcal{O}}(n^2)$ iterations, and thus has adaptivity that is at least $\tilde{\mathcal{O}}(n^2)$.

The question of whether submodular minimization can enjoy the same dramatic speedups as submodular maximization remains a mystery. To date, we do not know of any polynomial time algorithm for solving submodular minimization whose adaptivity is subquadratic in $n$. In terms of lower bounds, since the convex relaxation of submodular functions takes a particular form, none of the lower bounds for convex minimization apply. In this paper we initiate the study of the adaptivity of submodular function minimization by giving the first non-trivial lower bound for the parallel runtime of submodular minimization.

**Theorem 1.** *There is no $\frac{\log(n/4096)}{4 \log \log n}$-adaptive algorithm with $\mathrm{poly}(n)$ query complexity which, for any submodular function $f$, finds an optimal solution to $\min_S f(S)$ with probability $e^{-o\left(\sqrt{n} \log^{-3} n\right)}$.*

This lower bound excludes not only any deterministic $o(\frac{\log n}{\log \log n})$-adaptive algorithm, but also any randomized algorithm which finds an optimal solution with constant probability. The bound we obtain is similar to the bound obtained in [BS18a] for submodular maximization. However, the constructions are fundamentally different. Of course, one is for minimization and the other is for maximization, but another important difference is that the lower bound for submodular maximization is for constrained optimization and employs monotone submodular functions. We note that there are no known adaptivity lower bounds for the problem of unconstrained submodular maximization and that we obtain the first such lower bound for unconstrained submodular optimization.

### 1.3 Organization of the paper

In the remainder of the introduction, we discuss additional related work in Section 1.4, cover basic preliminaries in Section 1.5, and present a technical overview for the lower bound construction in Section 1.6. In Section 2, we show that there is no non-adaptive, i.e. 1-adaptive, algorithm for submodular minimization with a construction that serves as a warm-up to the construction for the main result, which we present in Section 3.

### 1.4 Further related work

As previously discussed, a long line of work has focused on accelerating submodular minimization (e.g. [Cun85, GLS88, Sch00, IFF01, Iwa02, CJK14, LJJ15, LSW15, CLSW17, HRRS19, ALS19]). Parallel algorithms with low adaptivity have recently been studied for submodular maximization [BS18a, BRS19a, EN19, FMZ19, CQ19b, BBS18, CFK19, ENV19, CQ19a, BRS19b] and convex minimization (see [Nem94, BJL+19, BS18b, DG18] for lower bounds and [DBW12, SBB+18] for upper bounds). However, very little is known about parallel submodular minimization. A result in the optimization from samples model [BS17] implies that there is no 1-adaptive algorithm for submodular minimization. Recent work on graph algorithms in the streaming model introduces a new powerful communication problem called the hidden-pointer chasing problem to show general lower bounds [ACK19]. The authors show via a reduction from this problem that there is no $r$-adaptive algorithms with query complexity $\tilde{o}(n^2/r^5)$ for submodular minimization. In conclusion, for the problem of submodular minimization, when $\text{poly}(n)$ queries are allowed per round, we only know that the number of rounds needed is between 1 and $\Omega(n^2)$.

### 1.5 Preliminaries

**Submodularity.** A function $f : 2^N \to \mathbb{R}$ over ground set $N$ of size $n$ is *submodular* if the marginal contributions $f_S(a) := f(S \cup a) - f(S)$ of an element $a \in N \setminus S$ to a set $S \subseteq N$ are diminishing, meaning $f_S(a) \geq f_T(a)$ for all $S \subseteq T \subseteq N$ and $a \in N \setminus T$. Throughout the paper, we abuse notation by writing $S \cup a$ instead of $S \cup \{a\}$. An optimal solution to the problem of minimizing a submodular function $f$ is a set $O$ such that $f(O) = \min_{S \subseteq N} f(S)$.

**Adaptivity.** Informally, the *adaptivity* of an algorithm is the number of sequential rounds it makes when polynomially-many queries can be executed in parallel in each round.

**Definition** (Adaptivity [BS18a])**.** *Given a value oracle for $f$, an algorithm is $r$-**adaptive** if every query $f(S)$ for the value of a set $S$ occurs at a round $i \in [r]$ s.t. $S$ is independent of the values $f(S')$ of all other queries at round $i$, with* $\text{poly}(n)$ *queries at every round.*

### 1.6 Technical overview

We start by reviewing the high level approach to adaptivity lower bounds for submodular maximization and convex minimization. These lower bounds construct functions $f^P$ that depend on a partition $P$ of either the ground set of elements $N$ for submodular functions or the indices $[n]$ for $n$-dimensional convex functions, into sets $P_1, \ldots, P_R$. The central part of the analysis is an information theoretic argument that shows that, with high probability, an algorithm cannot learn the set $P_i$ using less than $i$ rounds of polynomially many queries. However, the algorithm needs to learn set $P_R$, which is the optimal solution, to optimize $f^P$.

The main technical difficulty is in the construction of such functions, which requires new technical ideas for the problem of submodular minimization. A main difference with the lower bound for submodular maximization in [BS18a] is that our lower bound is for unconstrained optimization, instead of for constrained optimization. In particular, we construct *non-monotone* functions, while the functions in [BS18a] are *monotone*, and obtain the first adaptivity lower bound for unconstrained submodular optimization (whether for maximization or minimization). Regarding convex minimization, the functions constructed for the adaptivity lower bounds in [Nem94, BJL+19, BS18b, DG18] are highly non-submodular.

In Section 2, as a warm-up, we give a construction of submodular functions that cannot be minimized in a single round of queries. These functions $f^P$ depend a partition $P = (P_1, P_2, P_3)$ of the ground set $N = P_1 \cup P_2 \cup P_3$ such that elements in $P_2$ and $P_3$ are indistinguishable after one round of queries (formal definition of indistinguishability in Definition 1). For a partition $P' = (P_1, P'_2, P'_3)$ with the same set $P_1$ as partition $P$, we show that $f^P(S) = f^{P'}(S)$ for all queries $S$. Thus, the algorithm cannot learn part $P_3$ of partition $P = (P_1, P_2, P_3)$ in one round of queries, and the construction is such that $P_3$ is the unique optimal solution. In particular, $f^P(P_2) > f^P(P_3)$ even though $f^P(S) = f^{P'}(S)$ for all queries $S$. In our construction, we use elements in $P_1$ to diminish the marginal contribution of elements in $P_2$ and make them equal to the contribution of elements in $P_3$, thus making $P_2$ and $P_3$ indistinguishable. In other words, if $|S \cap P_1| = 0$, we have $f_S(a_2) > f_S(a_3)$ for $a_2 \in P_2$ and $a_3 \in P_3$, but if $|S \cap P_1|$ is large enough, then we have $f_S(a_2) = f_S(a_3)$. Since elements in $P_1$ cause elements in $P_2$ and $P_3$ to be indistinguishable, we say that elements in $P_1$ mask the partition of $P_2 \cup P_3$ (formal definition of masking in Definition 2).

The main technical challenge is in generalizing the construction from Section 2 to obtain the lower bound for $\Omega(\frac{\log n}{\log \log n})$ rounds, which we do in Section 3. Unlike the previous lower bound constructions for submodular and convex optimization, a main non-trivial part of our analysis is to show that the functions constructed are submodular. We construct functions $f^P$ that depend on a partition $P = (P_1, \ldots, P_R)$ such that at each round $r$, for all queries $f^P(S)$ of an algorithm at that round, elements in $P_r$ mask elements in $P_{r+1}, \ldots, P_R$. Thus, at each round $r$, an algorithm can learn at most part $P_r$ of the partition, but it will not learn any information about the partition of elements in $P_{r+1}, \ldots, P_R$, which remain indistinguishable to the algorithm. One challenge with multiple rounds is to mask elements which also need to mask other elements while maintaining submodularity, i.e., we need elements in $P_1$ to mask elements in $P_2 \cup P_3 \cup P_4$ in the first round of queries, but we also need elements in $P_2$ to mask elements in $P_3 \cup P_4$ in the second round of queries.

# 2 Warm-up: Non-Adaptive Algorithms

In this section, we show that there is no non-adaptive, i.e. 1-adaptive, algorithm for submodular minimization. The construction and the analysis for this hardness result are a warm-up towards the $\Omega(\frac{\log n}{\log \log n})$ rounds lower bound in Section 3. We first construct the functions that are hard to optimize in Section 2.1. After giving the formal definition, we discuss and explain this construction in detail as it will be generalized in the next section. In Section 2.2, we define two key technical tools for the analysis which capture when sets of elements are indistinguishable to the algorithm. Finally, in Section 2.3, we give the analysis of the construction from Section 2.1 using the tools from Section 2.2.

## 2.1 The construction

We construct a family of functions $f^P$ which depends on a partition $P$ of the ground set $N$ into 3 parts $P_1, P_2, P_3$ each of size $n/3$ and such that $P_1 \cup P_2 \cup P_3 = N$. We denote all such partitions by $\mathcal{P}_3$. The main idea of the construction is that after one round of queries, elements in $P_2$ and $P_3$ are indistinguishable to the algorithm. However, $P_3$ is the optimal solution and the algorithm needs to learn to distinguish elements in $P_3$ from elements in $P_2$ to optimize $f^P$.

**Definition of the function** $f^P$. For a partition $P = (P_1, P_2, P_3) \in \mathcal{P}_3$, the function $f^P$ is

$$f^P(S) = (1 - 2m(S)) \cdot \min\left(|S \cap P_2|, \frac{1}{4}|P_2|\right) - m(S) \cdot \left(|S \cap P_2| - \frac{1}{4}|P_2|\right)_+ + (1 - 2m(S)) \cdot \min\left(|S \cap P_3|, \frac{1}{4}|P_3|\right) - \left(|S \cap P_3| - \frac{1}{4}|P_3|\right)_+ + 2m(S) \cdot |P_2 \cup P_3|$$

where $x_+$ is defined as $x_+ = \max(0, x)$ and where $m(S) \in [0, 1]$ is a monotonically increasing function such that $m(\emptyset) = 0$ and $m(S) = 1$ if $|S \cap P_1| \geq \sqrt{n}$:

$$m(S) = \min\left(1, n^{-1/2}|S \cap P_1|\right).$$

The family of functions $\mathcal{F}_3$ consists of all functions $f^P$ such that $P \in \mathcal{P}_3$, i.e., $\mathcal{F}_3 = \{f^P : P \in \mathcal{P}_3\}$. The analysis will show that there is no non-adaptive algorithm that minimizes $\mathcal{F}_3$.

**Explanation for the function** $f^P$. It is easier to understand the function in terms of the marginal contributions $f_S(a) = f(S \cup a) - f(S)$ of the elements. For the remainder of this section, we abuse notation and denote the function $f^P$ defined in terms of the partition $P = (P_1, P_2, P_3)$ by $f$ when it is clear from context.

The marginal contribution $f_S(a)$ of an element $a \in P_2 \cup P_3$ to a set $S$ such that $|S \cap P_1| = 0$ is

$$f_S(a) = \begin{cases} 1 & \text{if } (a \in P_2 \text{ and } |S \cap P_2| < \frac{1}{4}|P_2|) \text{ or } (a \in P_3 \text{ and } |S \cap P_3| < \frac{1}{4}|P_3|) \\ 0 & \text{if } a \in P_2 \text{ and } |S \cap P_2| \geq \frac{1}{4}|P_2| \\ -1 & \text{if } a \in P_3 \text{ and } |S \cap P_3| \geq \frac{1}{4}|P_3| \end{cases} \tag{1}$$

From these marginal contributions, it is easy to see that $f(P_2) > f(\emptyset) > f(P_3)$. We will later show that $P_3$ is actually the unique optimal solution. We also observe that if $|S \cap P_1| = 0$ and $|S| < \frac{1}{4}|P_2| = \frac{1}{4}|P_3|$, then $f(S) = |S \cap (P_2 \cup P_3)|$.

The marginal contribution of an element $a \in P_2 \cup P_3$ to a set $S$ such that $|S \cap P_1| \geq \sqrt{n}$ is

$$f_S(a) = -1. \tag{2}$$

Note that for sets $S$ such that $|S \cap P_2|, |S \cap P_3| \geq \frac{1}{4}|P_2|$ and $|S \cap P_1| = 0$, we have $f_S(a_2) = 0 > -1 = f_S(a_3)$ for $a_2 \in P_2$ and $a_3 \in P_3$. However, if $|S \cap P_1| \geq \sqrt{n}$, we have $f_S(a_2) = -1 = f_S(a_3)$.

More generally, the marginal contribution $f_S(a)$ of an element $a \in P_2 \cup P_3$ to a set $S$ is

$$f_S(a) = \begin{cases} 1 - 2m(S) & \text{if } (a \in P_2 \text{ and } |S \cap P_2| < \frac{1}{4}|P_2|) \text{ or } (a \in P_3 \text{ and } |S \cap P_3| < \frac{1}{4}|P_3|) \\ -m(S) & \text{if } a \in P_2 \text{ and } |S \cap P_2| \geq \frac{1}{4}|P_2| \\ -1 & \text{if } a \in P_3 \text{ and } |S \cap P_3| \geq \frac{1}{4}|P_3| \end{cases} \tag{3}$$

Recall that $m(S) \in [0, 1]$ is a monotonically increasing function such that $m(\emptyset) = 0$ and $m(S) = 1$ if $|S \cap P_1| \geq \sqrt{n}$. The function $m(S)$, which we call the masking function, smoothly interpolates the marginal contributions from equation 1, where $|S \cap P_1| = 0$ and where elements in $P_2$ and $P_3$

have different contributions, to the marginal contributions from equation 2, where $|S \cap P_1| \geq \sqrt{n}$ and where elements in $P_2$ and $P_3$ have equal contributions.

Finally, the marginal contribution $f_S(a)$ of an element $a \in P_1$ to any set $S$ is such that

$$f_S(a) \geq 0.$$

Since these contributions are non-negative, elements in $P_1$ are not in the optimal solution.

### 2.2 Partition independence and masking a partition

As previously mentioned, the main idea of the construction of $\mathcal{F}_3$ is that elements in $P_2$ and $P_3$ are indistinguishable to the algorithm. In this section, we formalize what it means for sets of elements to be indistinguishable, which is central to the analysis. Since the next two definitions are also used in the next section, we consider an arbitrary family $\mathcal{P}_R$ of partitions $P = (P_0, \dots, P_R)$. Given a partition $(P_0, \dots, P_R)$, we denote by $P_{r:}$ the union of parts $P_i$ such that $i \geq r$, i.e. $P_{r:} := \cup_{i=r}^{R} P_i$. Informally, we say that queries $f^P(S)$ are *independent of the partition of* $P_{r:}$ if the values $f^P(S)$ of these queries do not contain information about which elements are in $P_r$, or $P_{r+1}, \dots,$ or $P_R$.

**Definition 1.** *Given a family of partitions $\mathcal{P}_R$ and a partition $P = (P_0, \dots, P_R) \in \mathcal{P}_R$, let $P'$ be a partition chosen uniformly at random from $\left\{(P'_0, \dots, P'_R) \in \mathcal{P}_R : P'_0 = P_0, \dots, P'_{r-1} = P_{r-1}\right\}$ for some $r < R$. A query $f^P(S)$ is* independent of the partition of $P_{r:} = \cup_{j=r}^{R} P_j$ *if $f^P(S) = f^{P'}(S)$ with high probability over $P'$.*

For example, in the construction from Section 2.1, for a set $S$ and partition $P = (P_1, P_2, P_3)$ such that $|S \cap P_1| \geq \sqrt{n}$, query $f^P(S)$ is independent of the partition of $P_2 \cup P_3$ since

$$\begin{aligned} f^P(S) &= -|S \cap (P_2 \cup P_3)| + 2|P_2 \cup P_3| \\ &= -|S \cap (N \setminus P_1)| + \frac{4n}{3} \\ &= -|S \cap (P'_2 \cup P'_3)| + 2|P'_2 \cup P'_3| = f^{P'}(S) \end{aligned}$$

for any $P' = (P_1, P'_2, P'_3)$. Next, we say that a set of elements $S$ masks the partition of $P_{r:}$ if queries $T$ such that $T \supseteq S$ are independent of the partition of $P_{r:}$.

**Definition 2.** *Given a partition $P = (P_0, P_1 \dots, P_R)$, a set $S$ of elements* masks the partition of *$P_{r:}$ if for all sets $T \supseteq S$, $f^P(T)$ is independent of the partition of $P_{r:}$.*

The function $m(S)$ from our construction is called the masking function because when $m(S) = 1$, the set $S$ masks the partition of $P_2 \cup P_3$. In other words, $S$ causes elements in $P_2$ and $P_3$ to be indistinguishable to the algorithm.

### 2.3 The analysis of the construction

We consider a non-adaptive algorithm $\mathcal{A}$ and a uniformly random partition $P = (P_1, P_2, P_3)$ in $\mathcal{P}_3$. We argue that with high probability over both the randomization of $P$ and the decisions of $\mathcal{A}$, the solution returned by $\mathcal{A}$ is not optimal. The analysis consists of three main parts, which are similar to the main parts for the analysis of the main result in the next section.

**Elements in $P_2$ and $P_3$ are indistinguishable after one round of queries.** The first part argues that, with high probability, for all queries $S$ by a non-adaptive algorithm, $f^P(S)$ is independent of the partition of $P_2 \cup P_3$.

**Lemma 1.** *Let $P$ be a uniformly random partition in $\mathcal{P}_3$. For any collection $\mathcal{S}$ of* poly$(n)$ *non-adaptive queries, with probability $1 - e^{-\Omega(n)}$, for all $S \in \mathcal{S}$, $f^P(S)$ is independent of the partition of $P_2 \cup P_3$.*

*Proof.* Consider any set $S$ which is independent of the randomization of $P$. There are two cases depending on the size of $S$. First, if $|S| \leq \frac{1}{12}n$, then $|S \cap P_2| \leq \frac{1}{4}|P_2|$ and $|S \cap P_3| \leq \frac{1}{4}|P_3|$. Thus,

$$\begin{aligned} f^P(S) &= (1 - 2m(S)) \cdot |S \cap (P_2 \cup P_3)| + 2m(S) \cdot |P_2 \cup P_3| \\ &= \left(1 - \min\left(1, n^{-1/2}|S \cap P_1|\right)\right) \cdot |S \cap (P_2 \cup P_3)| + 2\min\left(1, n^{-1/2}|S \cap P_1|\right) \cdot |P_2 \cup P_3| \end{aligned}$$

and we get $f^P(S) = f^{P'}(S)$ for any partition $P' = (P_1, P'_2, P'_3)$ since $P_2 \cup P_3 = N \setminus P_1 = P'_2 \cup P'_3$.

In the second case, we have $|S| \geq \frac{1}{12}n$. Since $P_1$ is a uniformly random set of size $n/3$, by the Chernoff bound, $|S \cap P_1| \geq \sqrt{n}$ with probability $1 - e^{-\Omega(n)}$. If $|S \cap P_1| \geq \sqrt{n}$, then $m(S) = 1$. Thus,

$$f^P(S) = -|S \cap (P_2 \cup P_3)| + 2|P_2 \cup P_3|.$$

and we get $f^P(S) = f^{P'}(S)$ for any partition $P' = (P_1, P'_2, P'_3)$. Thus, with probability $1 - e^{-\Omega(n)}$, $f^P(S)$ is independent of the partition of $P_2 \cup P_3$, and by a union bound, this holds for any collection of poly$(n)$ sets $S$. □

**If elements in $P_2$ and $P_3$ are indistinguishable, then the solution returned is not optimal.** The second part of the analysis argues that if all queries $f^P(S)$ of an algorithms are independent of the partition of $P_2 \cup P_3$, then, with high probability, the solution returned by this algorithm is not optimal.

**Lemma 2.** *Let $P$ be a uniformly random partition in $\mathcal{P}_3$. Consider an algorithm $\mathcal{A}$ such that all the queries $f^P(S)$ made by $\mathcal{A}$ are independent of the partition of $P_2 \cup P_3$. Then, the (possibly randomized) solution $S$ returned by $\mathcal{A}$ is, with probability $1 - e^{-\Omega(n)}$, not a minimizer of $f^P$.*

*Proof.* First, we argue that $P_3$ is the unique optimal solution. Because of the $2m(S) \cdot |P_2 \cup P_3|$ term in the definition of $f(S)$, $f_S(a) > 0$ for all $a \in P_1$ and for all $S$. Thus, an optimal solution does not contain any element from $P_1$. If we assume that $|S \cap P_1| = 0$, then for $a \in P_2$, $f_S(a) = 1$ if $|S \cap P_2| < \frac{1}{4}P_2$ and $f_S(a) = 0$ otherwise. This implies that an optimal solution does not contain any element from $P_2$ either. Thus, an optimal solution $S$ is such that $S \subseteq P_3$ and it is then to easy to see that $P_3$ is the unique optimal solution.

Consider an algorithm $\mathcal{A}$ such that all queries $f^P(S)$ of $\mathcal{A}$ are independent of the partition of $P_2 \cup P_3$. Thus, the solution $S$ returned by $\mathcal{A}$ is conditionally independent of the randomization of the partition $P$ given $P_1$. If $|S \cap (P_2 \cup P_3)| \neq |P_3|$ then $S$ is not optimal since $P_3$ is the unique optimal solution. If $|S \cap (P_2 \cup P_3)| = |P_3|$, then, by the Chernoff bound,

$$\Pr\left(|S \cap P_2| > 0\right) = 1 - e^{-\Omega(n)}$$

since $S$ is conditionally independent of the randomization of the partition $P$ given $P_1$. If $|S \cap P_2| > 0$, then $S$ is not optimal. □

**The function $f^P$ is submodular.** The last part of the analysis shows that the functions constructed are submodular.

**Lemma 3.** *$\mathcal{F}_3$ is a family of submodular functions.*

*Proof.* Since $m(S)$ is a monotonically increasing function such that $m(S) \in [0,1]$ for all $S$, we get that $f_S(a) \geq f_T(a)$ for all $S \subseteq T$ and $a \in (P_2 \cup P_3) \setminus T$ by Equation 3.

For an element $a \in P_1$, if $|S \cap P_1| < n^{1/2}$, then we have

$$f_S(a) = \frac{1}{n^{1/2}}\Bigg(2|P_2 \cup P_3| - 2\min\left(|S \cap P_2|, \frac{1}{4}|P_2|\right) - \left(|S \cap P_2| - \frac{1}{4}|P_2|\right)_+ - 2\min\left(|S \cap P_3|, \frac{1}{4}|P_3|\right)\Bigg) \geq 0,$$

which is monotonically decreasing as $S$ grows larger. If $|S \cap P_1| \geq n^{1/2}$, then $f_S(a) = 0$. Thus $f_S(a) \geq f_T(a)$ for all $S \subseteq T$ and $a \in P_1 \setminus T$.

Since $f_S(a) \geq f_T(a)$ for all $S \subseteq T$ and $a \in N \setminus T$, the function $f$ is submodular. □

By combining Lemma 1, Lemma 2, and Lemma 3, we get the hardness result for non-adaptive algorithms.

**Theorem 2.** *There is no* 1*-adaptive algorithm with* $\text{poly}(n)$ *query complexity which, for any submodular function $f$, finds an optimal solution to $\min_S f(S)$ with probability $e^{-o(n)}$.*

*Proof.* Consider a uniformly random partition $P \in \mathcal{P}_3$ and an algorithm $\mathcal{A}$ which queries $f^P$. By Lemma 1, after one round of queries, with probability $1 - e^{-\Omega(n)}$ over both the randomization of $P$ and of the algorithm, all the queries $f^P(S)$ made by $\mathcal{A}$ are independent of the partition of $P_2 \cup P_3$. By Lemma 2, this implies that, with probability $1 - e^{-\Omega(n)}$, the solution $S$ returned by $\mathcal{A}$ is not a minimizer for $f^P$. By the probabilistic method, this implies that there exists a partition $P \in \mathcal{P}_3$ for which, with probability $1 - e^{-\Omega(n)}$, $\mathcal{A}$ does not return a minimizer of $f^P \in \mathcal{F}_3$ after one round of queries. Finally, $\mathcal{F}_3$ is a family of submodular functions by Lemma 3 □

# 3 Main Result

In this section, we show that there is no $o(\frac{\log n}{\log \log n})$ adaptive algorithm for submodular minimization, which is our main result. The main technical challenge to obtain this result is to generalize the construction from the 1-round lower bound in the previous section to a construction that holds for $\Omega(\frac{\log n}{\log \log n})$ rounds, while maintaining submodularity.

We describe and discuss the construction in Section 3.1. After Section 3.1, the remainder of the section is devoted to the analysis of the construction, which follows a similar structure as the analysis for non-adaptive algorithms in the previous section. We first argue that elements in $P_{r+1:} = \cup_{i=r+1}^{R} P_i$ are indistinguishable to the algorithm after queries at round $r$ in Section 3.2. In Section 3.3, we show that if $P_{R-1}$ and $P_R$ are indistinguishable to the algorithm, then the solution returned by the algorithm is not optimal. We show that the functions constructed are submodular in Section 3.4. We then conclude with the main result in Section 3.5.

### 3.1 The construction

We consider partitions of the ground set $N$ into $R+1$ parts $P_0, P_1, \ldots, P_R$ of decreasing size:

$$\mathcal{P}_R = \left\{ (P_0, P_1, \ldots, P_R) : |P_i| = \frac{n}{\log^{2i} n} \text{ for } i \geq 1, |P_0| = n - \sum_{i=1}^{R} |P_i|, \bigcup_{i=0}^{R} P_i = N \right\}.$$

The main idea of the construction is that after $r$ rounds of queries, elements in $P_{r+1}, \ldots, P_R$ are indistinguishable to the algorithm. More precisely, at any round $r$, the queries $f^P(S)$ of the algorithm at that round are independent of the partition of $P_{r+1:}$ (recall Definition 1 of partition independence). However, $P_R$ is the optimal solution and the algorithm needs to learn to distinguish elements in $P_R$ from elements in $P_{R-1}$ to optimize $f^P$. Thus, the algorithm cannot find the optimal solution $P_R$ in less than $R-1$ rounds.

**Definition of the function $f^P$.** For a partition $P = (P_0, \ldots, P_R) \in \mathcal{P}_R$, we define

$$f^P(S) = \sum_{i=1}^{R-1} \Big( \big(1 - 2m^i(S)\big) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot \left(|S \cap P_i| - x_i|P_i|\right)_+ + m^i(S) \cdot 2|P_i| \Big)$$
$$+ \big(1 - 2m^R(S)\big) \cdot \min\left(|S \cap P_R|, x_R|P_R|\right) - \left(|S \cap P_R| - x_R|P_R|\right)_+ + m^R(S) \cdot 2|P_R|$$

where $m^i(S)$ is is a monotonically increasing function, such that $m^i(S) \in [0,1]$, that is called the masking function for part $P_i$ and is defined as

$$m^i(S) = 1 - \prod_{j=1}^{i-1} \left( 1 - \frac{16R}{|P_j|} \cdot \left(|S \cap P_j| - x_j|P_j|\right)_+ \right)_+ .$$

Finally, we set $x_i = \frac{i}{4R}$, $|P_i| = \frac{n}{\log^{2i} n}$, and $R = \frac{\log\left(\frac{n}{64^2}\right)}{4 \log \log n} + 2$. In particular, this implies that $|P_i| = \Omega(\sqrt{n})$ for all $i$. We analyze the family of functions $\mathcal{F}_R$ defined as

$$\mathcal{F}_R = \{f^P : P \in \mathcal{P}_R\}.$$

**Explanation for the function $f^P$.** The functions $f^P$ are easier to understand in terms of the marginal contributions of elements. The marginal contribution of an element $a \in P_i$, $i < R$ is

$$f_S(a) = \begin{cases} 1 - 2m^i(S) & \text{if } |S \cap P_i| \leq x_i|P_i| \\ -m^i(S) + \sum_{j=i+1}^{R} m^j_S(a) \cdot M_j & \text{otherwise} \end{cases}$$

for some $M_{i+1}, \ldots, M_R \geq 0$. The term $\sum_{j=i+1}^{R} m^j_S(a) \cdot M_j$ is contribution of $a$ to masking the partition of $P_{j:}$, for $j > i$. In contrast, the marginal contribution of an element $a \in P_R$ is

$$f_S(a) = \begin{cases} 1 - 2m^R(S) & \text{if } |S \cap P_R| \leq \frac{1}{4}|P_R| \\ -1 & \text{otherwise} \end{cases}$$

We make the following additional observations about the function $f^P$.

- For any $S$, if there exists $r$ such that $|S \cap P_i| \leq x_i|P_i|$ for all $i > r$, then for any $a_i \in P_i$ with $i > r$, we have

$$f_S(a_i) = 1 - 2m^i(S) = 1 - 2\left(1 - \prod_{j=1}^{r-1}\left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+\right)$$

  since $(|S \cap P_j| - x_j|P_j|)_+ = 0$ for all $j > r$. Thus, elements in $P_{r+1}, \ldots, P_R$ all have equal contributions and are indistinguishable from such queries $f(S)$.

- If $S$ is such that $|S \cap P_r| \geq \left(x_r + \frac{1}{16R}\right)|P_r|$, then

$$\frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)_+ \geq 1,$$

  which implies that $m^i(S) = 1$ for all $i > r$. If $m^i(S) = 1$, then $m^i_S(a) = 0$ since $m^i(S)$ is a monotonically increasing function upper bounded by 1, and we get

$$f_S(a) = -1$$

  for all $a \in P_{r+1:}$. Thus, elements in $P_{r+1}, \ldots, P_R$ all have equal contributions and are indistinguishable from such queries $f(S)$.

- This next observation is about the contributions of elements to the masking functions $m^j(S)$. For all $i$, elements in $P_i$ do not contribute to the masking functions if $|S \cap P_i| < x_i|P_i|$, i.e. $m^j_S(a) = 0$ for all $j$. This design choice is due to the fact that if $m^j_S(a) > 0$ for $a \in P_i$, then elements in $P_i$ are *not* indistinguishable from elements in $P_R$. We set the value of $x_1, \ldots, x_R$ to be monotonically increasing for the following reason. At round $r$ of an algorithm, if queries from previous rounds are all independent of the partition of $P_{r:}$, then for query $S$ at round $r$, for all $a \in P_i$ with $i > r$, there are two cases. In the first case, $|S \cap P_i| < x_i|P_i|$ and $m^j_S(a) = 0$ for all $j$. In the second case, $|S \cap P_i| \geq x_i|P_i|$. Since $x_r < x_i$, with high probability, we have $|S \cap P_r| \geq (x_r + \frac{1}{16R})|P_r|$, which implies that $m^j(S) = 1$ and $m^j_S(a) = 0$ for $j > r$. Thus, for queries at round $r$, elements in $P_{r+1:}$ do not contribute to the masking functions, which is important for them to be indistinguishable.

- Finally, consider $S \subseteq T$ such that, for $a \in P_i$ for some $i$, $|S \cap P_i| < x_i|P_i|$ and $|T \cap P_i| \geq x_i|P_i|$. Since $f^P$ is submodular, we must have

$$f_S(a) = 1 - 2m^i(S) \geq -m^i(T) + \sum_{j=i+1}^{R} m^j_T(a) = f_T(a)$$

  for all such $S, T$, which is non-trivial and requires a careful design of the masking functions $m^j(\cdot)$ as well as other aspects of the function $f^P$. In fact, showing that the function $f^P$ is submodular is a main non-trivial part of the analysis (Lemma 8).

### 3.2 Indistinguishability

We consider a uniformly random function $f^P \in \mathcal{F}_R$ defined by partition $P = (P_0, P_1, \ldots, P_R)$. We show that, with high probability, for all queries $S$ in the first $r$ rounds of an algorithm, $f^P(S)$ is independent of the partition of $P_{r+1:}$. In other words, elements in $P_{r+1}, \ldots, P_R$ are all indistinguishable to the algorithm after round $r$ of queries.

**Lemma 4.** *Let $P$ be a uniformly random partition in $\mathcal{P}_R$. Consider an algorithm $\mathcal{A}$ at round $r$ such that all the queries $f^P(S)$ from previous rounds are independent of the partition of $P_{r:}$. Then, for any collection $\mathcal{S}$ of* poly$(n)$ *non-adaptive queries at round $r$, we have that, with probability $1 - e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}$, for all $S \in \mathcal{S}$, $f^P(S)$ is independent of the partition of $P_{r+1:}$.*

The remainder of Section 3.2 is devoted to the proof of Lemma 4. We first note that

$$\sum_{i=1}^{r} \left( \left(1 - 2m^i(S)\right) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot \left(|S \cap P_i| - x_i|P_i|\right)_+ + m^i(S) \cdot 2|P_i| \right)$$

is independent of the partition of $P_{r+1:}$. Thus, we need to show that

$$\sum_{i=r+1}^{R-1} \left( \left(1 - 2m^i(S)\right) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot \left(|S \cap P_i| - x_i|P_i|\right)_+ + m^i(S) \cdot 2|P_i| \right)$$
$$+ \left(1 - 2m^R(S)\right) \cdot \min\left(|S \cap P_R|, x_R|P_R|\right) - \left(|S \cap P_R| - x_R|P_R|\right)_+ + m^R(S) \cdot 2|P_R|$$

is independent of the partition of $P_{r+1:}$. There are two cases depending on the size of $S \cap P_{r:}$, which we address in Lemma 5 and Lemma 6.

**Lemma 5.** *Let $P$ be a uniformly random partition in $\mathcal{P}_R$. Consider an algorithm $\mathcal{A}$ at round $r$ such that all the queries $f^P(S)$ from previous rounds are independent of the partition of $P_{r:}$. Then, for any query $f^P(S)$ by $\mathcal{A}$ at round $r$ such that $|S \cap P_{r:}| \leq \left(x_r + \frac{1}{8R}\right)|P_{r:}|$, we have that, with probability $1 - e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}$, query $f^P(S)$ is independent of the partition of $P_{r+1:}$.*

*Proof.* Consider a query $f^P(S)$ at round $r$ such that the queries from previous rounds are independent of the partition of $P_{r:}$. This implies that $S$ is conditionally independent of the randomization of $P = (P_0, \ldots, P_R)$ given $P_0, \ldots P_{r-1}$. We also assumed that

$$|S \cap P_{r:}| \leq \left(x_r + \frac{1}{8R}\right)|P_{r:}| = \left(x_{r+1} - \frac{1}{8R}\right)|P_{r:}|.$$

By the Chernoff bound, for any $i \geq r$, with $\mu \leq \left(x_{r+1} - \frac{1}{8R}\right)|P_i| = \Omega\left(\frac{\sqrt{n}}{\log n}\right), \delta = O\left(\frac{1}{\log n}\right)$,

$$\Pr\left(|S \cap P_i| \geq x_{r+1}|P_i|\right) = e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}.$$

If $|S \cap P_i| \leq x_{r+1}|P_i|$, then, since $x_i$ is monotonically increasing, we have that for any $i \geq r+1$,

$$|S \cap P_i| \leq x_i|P_i|.$$

This implies that $\min\left(|S \cap P_i|, x_i|P_i|\right) = |S \cap P_i|$ and $\left(|S \cap P_i| - x_i|P_i|\right)_+ = 0$ for all $i \geq r+1$. So,

$$\sum_{i=r+1}^{R-1} \left( \left(1 - 2m^i(S)\right) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot \left(|S \cap P_i| - x_i|P_i|\right)_+ + m^i(S) \cdot 2|P_i| \right)$$
$$+ \left(1 - 2m^R(S)\right) \cdot \min\left(|S \cap P_R|, x_R|P_R|\right) - \left(|S \cap P_R| - x_R|P_R|\right)_+ + m^R(S) \cdot 2|P_R|$$
$$= \sum_{i=r+1}^{R-1} \left( \left(1 - 2m^i(S)\right) \cdot |S \cap P_i| + m^i(S) \cdot 2|P_i| \right) + \left(1 - 2m^R(S)\right) \cdot |S \cap P_R| + m^R(S) \cdot 2|P_R|$$
$$= |S \cap P_{r+1:}| \cdot \sum_{i=r+1}^{R} \left(1 - 2m^i(S)\right) + \cdot \sum_{i=r+1}^{R} m^i(S) \cdot 2|P_i|$$

Note that $|S \cap P_{r+1:}|$ is independent of the partition of $P_{r+1:}$. It remains to show that $m^i(S)$ is also independent of the partition of $P_{r+1:}$ for all $i > r$. Note that

$$\begin{aligned} m^i(S) &= 1 - \prod_{j=1}^{r} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+ \cdot \prod_{j=r+1}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+ \\ &= 1 - \prod_{j=1}^{r} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+ \end{aligned}$$

since $(|S \cap P_j| - x_j|P_j|)_+ = 0$ for all $j \geq r+1$. Thus $m^i(S)$ is independent of the partition of $P_{r+1:}$ for all $i$. As previously noted,

$$\sum_{i=1}^{r} \left(\left(1 - 2m^i(S)\right) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot (|S \cap P_i| - x_i|P_i|)_+ + m^i(S) \cdot 2|P_i|\right)$$

is also independent of the partition of $P_{r+1:}$, and we get that $f^P(S)$ is independent of this partition as well. □

**Lemma 6.** *Let $P$ be a uniformly random partition in $\mathcal{P}_R$. Consider an algorithm $\mathcal{A}$ at round $r$ such that all the queries $f^P(S)$ from previous rounds are independent of the partition of $P_{r:}$. Then, for any query $f^P(S)$ by $\mathcal{A}$ at round $r$ such that $|S \cap P_{r:}| \geq \left(x_r + \frac{1}{8R}\right)|P_{r:}|$, we have that, with probability $1 - e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}$, query $f^P(S)$ is independent of the partition of $P_{r+1:}$.*

*Proof.* Consider a query $f^P(S)$ at round $r$ such that the queries from previous rounds are independent of the partition of $P_{r:}$. As for Lemma 6, this implies that $S$ is conditionally independent of the randomization of $P = (P_0, \ldots, P_R)$ given $P_0, \ldots P_{r-1}$. We also assumed that $|S \cap P_{r:}| \geq \left(x_r + \frac{1}{8R}\right)|P_{r:}|$.

By the Chernoff bound, with $\mu \geq \left(x_r + \frac{1}{8R}\right)|P_r| = \Omega\left(\frac{\sqrt{n}}{\log n}\right)$ and $\delta = O\left(\frac{1}{\log n}\right)$, we have

$$\Pr\left(|S \cap P_r| \leq \left(x_r + \frac{1}{16R}\right)|P_r|\right) = e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}.$$

If $|S \cap P_r| \geq \left(x_r + \frac{1}{16R}\right)|P_r|$, then $\frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)_+ \geq 1$ and $m^i(S) = 1$ for all $i \geq r+1$. With $m^i(S) = 1$ for all $i \geq r+1$, we get

$$\begin{aligned} &\sum_{i=r+1}^{R-1} \left(\left(1 - 2m^i(S)\right) \cdot \min\left(|S \cap P_i|, x_i|P_i|\right) - m^i(S) \cdot (|S \cap P_i| - x_i|P_i|)_+ + m^i(S) \cdot 2|P_i|\right) \\ &\qquad + \left(1 - 2m^R(S)\right) \cdot \min\left(|S \cap P_R|, x_R|P_R|\right) - (|S \cap P_R| - x_R|P_R|)_+ + m^R(S) \cdot 2|P_R| \\ &= \sum_{i=r+1}^{R} \left(-\min\left(|S \cap P_i|, x_i|P_i|\right) - (|S \cap P_i| - x_i|P_i|)_+ + 2|P_i|\right) \\ &= \sum_{i=r+1}^{R} \left(-|S \cap P_i| + 2|P_i|\right) \\ &= -\,|S \cap P_{r+1:}| + \sum_{i=r+1}^{R} 2|P_i| \end{aligned}$$

which is independent of the partition of $P_{r+1:}$. As previously noted,

$$\sum_{i=1}^{r} \left(\left(1-2m^i(S)\right)\cdot \min\left(|S\cap P_i|, x_i|P_i|\right) - m^i(S)\cdot \left(|S\cap P_i| - x_i|P_i|\right)_+ + m^i(S)\cdot 2|P_i|\right)$$

is also independent of the partition of $P_{r+1:}$, and we get that $f^P(S)$ is independent of this partition as well. □

We are now ready to prove Lemma 4

*Proof of Lemma 4.* Consider a query $f^P(S)$ by an algorithm at round $r$ such that all the queries from previous rounds are independent of the partition of $P_{r:}$. By Lemma 5 and Lemma 6, with probability $1-e^{-\Omega\left(\sqrt{n}\log^{-3} n\right)}$, $f^P(S)$ is independent of the partition of $P_{r+1:}$. By a union bound, this holds for $\text{poly}(n)$ queries at round $r$. □

### 3.3 Indistinguishability implies non-optimal solution

The next part of the analysis argues that if all queries $f^P(S)$ of an algorithm are independent of the partition of $P_{R-1}\cup P_R$, then, with high probability, the solution returned by this algorithm is not optimal.

**Lemma 7.** *Let $P$ be a uniformly random partition in $\mathcal{P}_R$. Consider an algorithm $\mathcal{A}$ such that all the queries $f^P(S)$ made by the algorithm $\mathcal{A}$ are independent of the partition of $P_{R-1}\cup P_R$. Then, the (possibly randomized) solution $S$ returned by $\mathcal{A}$ is not a minimizer of $f^P$ w.p. $1-e^{-\Omega(\sqrt{n})}$.*

*Proof.* First, we argue that $P_R$ is the unique optimal solution for minimizing $f^P$. We analyze the different terms in the definition of $f^P$. Note that for all $i<R$,

$$\left(1-2m^i(S)\right)\cdot \min\left(|S\cap P_i|, x_i|P_i|\right) - m^i(S)\cdot \left(|S\cap P_i| - x_i|P_i|\right)_+ + m^i(S)\cdot 2|P_i|$$

is non-negative and is equal to zero if $S=P_R$, so $P_R$ is a minimizer for each of these terms. Regarding

$$\left(1-2m^R(S)\right)\cdot \min\left(|S\cap P_R|, x_R|P_R|\right) - \left(|S\cap P_R| - x_R|P_R|\right)_+ + m^R(S)\cdot 2|P_R|,$$

it is monotonically increasing as $m^R(S)$ increases. Since $m^R(P_R)=0$, it is then easy to see that $S=P_R$ is a unique minimizer for this term. Thus, $S=P_R$ is the unique optimal solution for minimizing $f^P$.

Next, consider an algorithm $\mathcal{A}$ such that all queries $f^P(S)$ of $\mathcal{A}$ are independent of the partition of $P_{R-1}\cup P_R$. Thus, the solution $S$ returned by $\mathcal{A}$ is conditionally independent of the randomization of the partition $P$ given $P_1,\ldots,P_{R-2}$. If $|S\cap(P_{R-1}\cup P_R)|\neq|P_R|$ then $S$ is not optimal since $P_R$ is the unique optimal solution. If $|S\cap(P_{R-1}\cup P_R)|=|P_R|=\Omega(\sqrt{n})$, then, by the Chernoff bound,

$$\Pr\left(|S\cap P_{R-1}|>0\right) = 1-e^{-\Omega\left(\sqrt{n}\right)}$$

since $S$ is conditionally independent of the randomization of the partition $P$ given $P_1,\ldots,P_{R-2}$. If $|S\cap P_{R-1}|>0$, then $S$ is not optimal. □

### 3.4 Submodularity

We show that the family of functions $\mathcal{F}_R$ that we constructed is a family of submodular functions, which is a main non-trivial part of the analysis.

**Lemma 8.** *For any $f^P \in \mathcal{F}_R$, $f^P$ is a submodular function.*

*Proof.* We consider a function $f^P \in \mathcal{F}^R$. For the remainder of this proof, we abuse notation and denote $f^P$ by $f$. We wish to show that the marginal contributions are diminishing, i.e., $f_S(a) \geq f_T(a)$ for all $S \subseteq T \subseteq N$ and $a \in N \setminus T$. We consider $S \subseteq T \subseteq N$ and $a \in N \setminus T$ where $a \in P_r$ for some $r \in [R]$. There are three cases. In case 1, $|S \cap P_r|, |T \cap P_r| < x_r|P_r|$. Since $m^r(S) \leq m^r(T)$ for all $S \subseteq T$, we get

$$f_S(a) = 1 - 2m^r(S) \geq 1 - 2m^r(T) = f_T(a).$$

In case 2, we assume that $|S \cap P_r|, |T \cap P_r| \geq x_r|P_r|$. If $r = R$, then $f_S(a) = f_T(a) = -1$. For the remainder of case 2, we assume that $r < R$. Note that for $i \leq r$, $m^i_S(a) = 0$. For $i > r$, we have

$$\begin{aligned}
&m^i(S \cup a) - m^i(S) \\
=&1 - \left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| + 1 - x_r|P_r|)_+\right)_+ \cdot \prod_{j=1, j\neq r}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+ \\
&\quad - \left(1 - \left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)_+\right)_+ \cdot \prod_{j=1, j\neq r}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+\right) \\
=&\left(\left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)\right)_+ - \left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| + 1 - x_r|P_r|)\right)_+\right) \\
&\quad \cdot \prod_{j=1, j\neq i}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+ \\
=&\left(\frac{16R}{|P_r|} - \left(\frac{16R}{|P_r|} \cdot (|S \cap P_r| + 1 - x_r|P_r|) - 1\right)_+\right)_+ \cdot \prod_{j=1, j\neq r}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|S \cap P_j| - x_j|P_j|)_+\right)_+,
\end{aligned}$$

and similarly,

$$\begin{aligned}
&m^i(T \cup a) - m^i(T) \\
=&\left(\frac{16R}{|P_r|} - \left(\frac{16R}{|P_r|} \cdot (|T \cap P_r| + 1 - x_r|P_r|) - 1\right)_+\right)_+ \cdot \prod_{j=1, j\neq r}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|T \cap P_j| - x_j|P_j|)_+\right)_+.
\end{aligned}$$

Since $S \subseteq T$, we get $m^i_S(a) \geq m^i_T(a)$ for $i > r$. We also have

$$\begin{aligned}
f_S(a) &= -m^r(S) \cdot \big((|S \cap P_r| + 1 - x_r|P_r|)_+ - (|S \cap P_r| - x_r|P_r|)_+\big) \\
&\quad + \sum_{i=r+1}^{R-1} \big(-2m^i_S(a) \cdot \min(|S \cap P_i|, x_i|P_i|) - m^i_S(a) \cdot (|S \cap P_i| - x_i|P_i|)_+ + m^i_S(a) \cdot 2|P_i|\big) \\
&\quad - 2m^R_S(a) \cdot \min(|S \cap P_R|, x_R|P_R|) + m^R_S(a) \cdot 2|P_R| \\
&= -m^r(S) + \sum_{i=r+1}^{R-1} m^i_S(a) \big(2|P_i| - 2\min(|S \cap P_i|, x_i|P_i|) - (|S \cap P_i| - x_i|P_i|)_+\big) \\
&\quad + m^R_S(a) \cdot (2|P_R| - 2\min(|S \cap P_R|, x_R|P_R|))
\end{aligned}$$

and similarly,

$$\begin{aligned}
f_T(a) = -m^r(T) &+ \sum_{i=r+1}^{R-1} m^i_T(a) \big(2|P_i| - 2\min(|T \cap P_i|, x_i|P_i|) - (|T \cap P_i| - x_i|P_i|)_+\big) \\
&+ m^R_T(a)(2|P_R| - 2\min(|T \cap P_R|, x_R|P_R|))
\end{aligned}$$

Since $m^r(S) \leq m^r(T)$ for all $S \subseteq T$ and $m^i_S(a) \geq m^i_T(a)$ for $i > r$, we get $f_S(a) \geq f_T(a)$.

The third case is if $|S \cap P_r| < x_r|P_r|$ and $|T \cap P_r| \geq x_r|P_r|$. Then, similarly as in case 1,

$$f_S(a) = 1 - 2m^r(S)$$

If $r = R$, then $f_T(a) = -1 \leq f_S(a)$. For the remainder of case 3, we assume that $r < R$. Similarly as for case 2, we have

$$\begin{aligned}
f_T(a) = -m^r(T) &+ \sum_{i=r+1}^{R-1} m^i_T(a) \big(2|P_i| - 2\min(|T \cap P_i|, x_i|P_i|) - (|T \cap P_i| - x_i|P_i|)_+\big) \\
&+ m^R_T(a)(2|P_R| - 2\min(|T \cap P_R|, x_R|P_R|)) \\
\leq -m^r(T) &+ \sum_{i=r+1}^{R} 2m^i_T(a)|P_i|.
\end{aligned}$$

Next, note that

$$1 - m^r(S) = \prod_{j=1}^{r-1} \left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)_+\right)_+.$$

We also recall from case 2 that for $i \geq r$,

$$\begin{aligned}
&m^i_T(a) \\
&= \left(\frac{16R}{|P_r|} - \left(\frac{16R}{|P_r|} \cdot (|T \cap P_r| + 1 - x_r|P_r|) - 1\right)_+\right)_+ \cdot \prod_{j=1, j \neq r}^{i-1} \left(1 - \frac{16R}{|P_j|} \cdot (|T \cap P_j| - x_j|P_j|)_+\right)_+ \\
&\leq \frac{16R}{|P_r|} \cdot \prod_{j=1}^{r-1} \left(1 - \frac{16R}{|P_j|} \cdot (|T \cap P_j| - x_j|P_j|)_+\right)_+
\end{aligned}$$

Together with $\frac{|P_r|}{16R} \geq \sum_{i=r+1}^{R} 2|P_i|$, we get that

$$\sum_{i=r+1}^{R} 2m_T^i(a)|P_i| \leq \prod_{j=1}^{r-1} \left(1 - \frac{16R}{|P_j|} \cdot (|T \cap P_j| - x_j|P_j|)_+\right)_+.$$

We conclude that

$$\begin{aligned} f_S(a) &= 1 - 2m^r(S) \\ &= -m^r(S) + \prod_{j=1}^{r-1} \left(1 - \frac{16R}{|P_r|} \cdot (|S \cap P_r| - x_r|P_r|)_+\right)_+ \\ &\geq -m^r(T) + \sum_{i=r+1}^{R} 2m_T^i(a)|P_i| \\ &\geq f_T(a). \end{aligned}$$

□

### 3.5 Proof of the main result

We combine Lemma 4, Lemma 7, and Lemma 8 to obtain the main result.

**Theorem 1.** *There is no $\frac{\log(n/4096)}{4 \log \log n}$-adaptive algorithm with* $\operatorname{poly}(n)$ *query complexity which, for any submodular function $f$, finds an optimal solution to* $\min_S f(S)$ *with probability $e^{-o\left(\sqrt{n} \log^{-3} n\right)}$.*

*Proof.* Consider a uniformly random partition $P \in \mathcal{P}_R$ with $R = \frac{\log\left(\frac{n}{64^2}\right)}{4 \log \log n} + 2$ and an algorithm $\mathcal{A}$ which queries $f^P$. By Lemma 4, after $R-2$ round of queries, with probability $1-e^{-\Omega\left(\sqrt{n} \log^{-3} n\right)}$ over both the randomization of $P$ and the algorithm, all the queries $f^P(S)$ made by $\mathcal{A}$ are independent of the partition of $P_{R-1} \cup P_R$. By Lemma 7, this implies that, with probability $1 - e^{-\Omega\left(\sqrt{n} \log^{-3} n\right)}$, the solution $S$ returned by $\mathcal{A}$ is not a minimizer for $f^P$. By the probabilistic method, this implies that there exists a partition $P \in \mathcal{P}_R$ for which, with probability $1 - e^{-\Omega\left(\sqrt{n} \log^{-3} n\right)}$, $\mathcal{A}$ does not return a minimizer of $f^P \in \mathcal{F}_R$ after $R - 2 = \frac{\log\left(\frac{n}{64^2}\right)}{4 \log \log n}$ rounds of queries. Finally, $\mathcal{F}_R$ is a family of submodular functions by Lemma 8 □

# References


[ACK19] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial pass lower bounds for graph streaming algorithms. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 265–276. ACM, 2019.

[ALS19] Brian Axelrod, Yang P Liu, and Aaron Sidford. Near-optimal approximate discrete and continuous submodular function minimization. *arXiv preprint arXiv:1909.00171*, 2019.

[BBS18] Eric Balkanski, Adam Breuer, and Yaron Singer. Non-monotone submodular maximization in exponentially fewer iterations. In *Advances in Neural Information Processing Systems*, pages 2353–2364, 2018.

[BJL$^+$19] Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Complexity of highly parallel non-smooth convex optimization. *arXiv preprint arXiv:1906.10655*, 2019.

[BRS19a] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 283–302. Society for Industrial and Applied Mathematics, 2019.

[BRS19b] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An optimal approximation for submodular maximization under a matroid constraint in the adaptive complexity model. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 66–77. ACM, 2019.

[BS17] Eric Balkanski and Yaron Singer. Minimizing a submodular function from samples. In *Advances in Neural Information Processing Systems*, pages 814–822, 2017.

[BS18a] Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1138–1151. ACM, 2018.

[BS18b] Eric Balkanski and Yaron Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization. *arXiv preprint arXiv:1808.03880*, 2018.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[CFK19] Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–113. ACM, 2019.

[CJK14] Deeparnab Chakrabarty, Prateek Jain, and Pravesh Kothari. Provable submodular minimization using wolfe's algorithm. In *Advances in Neural Information Processing Systems*, pages 802–809, 2014.

[CLSW17] Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic submodular function minimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1220–1231. ACM, 2017.

[CQ19a] Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 78–89. ACM, 2019.

[CQ19b] Chandra Chekuri and Kent Quanrud. Submodular function maximization in parallel via the multilinear relaxation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 303–322. SIAM, 2019.

[Cun85] William H Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985.

[DBW12] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[DG18] Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. *arXiv preprint arXiv:1811.01903*, 2018.

[EN19] Alina Ene and Huy L Nguy?n. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 274–282. SIAM, 2019.

[ENV19] Alina Ene, Huy L Nguy?n, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 90–101. ACM, 2019.

[FMZ19] Matthew Fahrbach, Vahab Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with optimal approximation, adaptivity and query complexity. *SODA*, 2019.

[GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. Geometric algorithms and combinatorial optimization. 1988.

[HRRS19] Yassine Hamoudi, Patrick Rebentrost, Ansis Rosmanis, and Miklos Santha. Quantum and classical algorithms for approximate submodular function minimization. *arXiv preprint arXiv:1907.05378*, 2019.

[IFF01] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[Iwa02] Satoru Iwata. A fully combinatorial algorithm for submodular function minimization. *Journal of Combinatorial Theory, Series B*, 84(2):203–212, 2002.

[Iwa03] Satoru Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.

[JB11] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR 2011*, pages 1897–1904. IEEE, 2011.

[JBS13] Stefanie Jegelka, Francis Bach, and Suvrit Sra. Reflection methods for user-friendly submodular optimization. In *Advances in Neural Information Processing Systems*, pages 1313–1321, 2013.

[KKT08] Pushmeet Kohli, M Pawan Kumar, and Philip HS Torr. $P^3$ & beyond: Move making algorithms for solving higher order functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2008.

[KT10] Pushmeet Kohli and Philip HS Torr. Dynamic graph cuts and their applications in computer vision. In *Computer Vision*, pages 51–108. Springer, 2010.

[LB10] Hui Lin and Jeff Bilmes. An application of the submodular principal partition to training data subset selection. In *NIPS workshop on Discrete Optimization in Machine Learning*, 2010.

[LB11] Hui Lin and Jeff Bilmes. Optimal selection of limited vocabulary speech corpora. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[LJJ15] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[Lov83] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065. IEEE, 2015.

[Nem94] Arkadi Nemirovski. On parallel complexity of nonsmooth convex optimization. *Journal of Complexity*, 10(4):451–463, 1994.

[SBB+18] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

[Sch00] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.