A polynomial lower bound on adaptive complexity of submodular maximization

Wenzheng Li
Stanford University
wzli@stanford.edu

Paul Liu
Stanford University
paul.liu@stanford.edu

Jan Vondrák
Stanford University
jvondrak@stanford.edu

In large-data applications, it is desirable to design algorithms with a high degree of parallelization. In the context of submodular optimization, adaptive complexity has become a widely-used measure of an algorithm's "sequentiality". Algorithms in the adaptive model proceed in rounds, and can issue polynomially many queries to a function f in each round. The queries in each round must be independent, produced by a computation that depends only on query results obtained in previous rounds.

In this work, we examine two fundamental variants of submodular maximization in the adaptive complexity model: cardinality-constrained monotone maximization, and unconstrained non-monotone maximization. Our main result is that an r-round algorithm for cardinality-constrained monotone maximization cannot achieve a factor better than $1 - 1/e - \Omega(\min\{\frac{1}{r}, \frac{\log^2 n}{r^3}\})$, for any $r < n^c$ (where $c > 0$ is some constant). This is the first result showing that the number of rounds must blow up polynomially large as we approach the optimal factor of $1 - 1/e$.

For the unconstrained non-monotone maximization problem, we show a positive result: For every instance, and every $\delta > 0$, either we obtain a $(1/2 - \delta)$-approximation in 1 round, or a $(1/2 + \Omega(\delta^2))$-approximation in $O(1/\delta^2)$ rounds. In particular (in contrast to the cardinality-constrained case), there cannot be an instance where (i) it is impossible to achieve a factor better than $1/2$ regardless of the number of rounds, and (ii) it takes r rounds to achieve a factor of $1/2 - O(1/r)$.

1 INTRODUCTION

Let E be a set of size n, and $f : 2^E \to \mathbb{R}_+$ a function satisfying $f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$ for all $S \subseteq T \subseteq E \setminus \{e\}$. Such a function is called *submodular*. When $f(S) \leq f(T)$ for all $S \subseteq T$, f is called *monotone*. Submodular functions capture a notion of *diminishing returns*, where the gain $f(S \cup \{e\}) - f(S)$ gets smaller as S gets larger. This notion arises naturally in combinatorial optimization, algorithmic game theory, and machine learning, among other domains (see [25] and the references contained therein). As such, there has been a wealth of research on submodularity over the past few decades.

As datasets grow larger in size however, there has been renewed attention examining submodular optimization in several computing models for large-scale data. These models typically assume oracle access to a submodular function f and restrict the computation in certain ways. Such models include streaming [1, 24, 29], MapReduce [13, 14, 27, 28, 30, 31], and more recently the adaptive complexity model [4–7, 9, 10, 15, 17, 19, 24]. Algorithms in the streaming model examine elements of E one at a time, with the only constraint being limited memory. Algorithms in the MapReduce model partition the dataset among many machines with smaller memory, and run local computations on each machine independently. In both models, algorithms are typically memory-efficient, with algorithms running sequentially once the data to be processed is small enough. In many applications however, oracle queries to f are the dominant computational bottleneck. Thus, long chains of sequential queries drastically slow down an algorithm. For these applications, the adaptive model of Balkanski and Singer [5] offers more relevant constraints. Algorithms in the adaptive model proceed in *rounds*, and can issue polynomially many queries to f in each round. The queries in each round must be independent, and can be generated by the algorithm based on query results obtained in previous rounds. Informally, the adaptive model measures complexity by the longest chain of sequentially dependent calls to f in the algorithm. Consequently, standard greedy algorithms which examine one element at a time have essentially worst possible adaptive complexity.

Over the past two years, there has been a burst of work in adaptivity-efficient algorithms for maximizing submodular functions [4–7, 9, 10, 15, 17, 19, 24]. For maximizing a monotone submodular function under a cardinality constraint, several independently developed algorithms are known [3, 10, 15, 19] which surprisingly all achieve a close-to-optimal $1 - 1/e - \varepsilon$ approximation ratio using $O(\frac{1}{\epsilon^2} \log n)$ adaptive rounds. Moreover, Breuer et al. [7] have developed an $O(\frac{1}{\varepsilon^2} \log n \log^2 \log k)$-adaptive algorithm that outperforms all current theoretically state-of-the-art algorithms in practice. In the case of a matroid constraint, the best theoretical results achieve a $(1 - 1/e - \varepsilon)$-approximation with $O(\frac{1}{\epsilon^3} \log n \log k)$ adaptivity [4, 9], where k is the rank of the underlying matroid. The known results are somewhat weaker in the cardinality-constrained non-monotone setting. Mirrokni et al. [18] developed a $(0.039 - \varepsilon)$-approximation in $O(\frac{1}{\varepsilon} \log n)$ adaptive rounds, Balkanski et al. [2] designed a $1/2e$-approximation in $O(\log^2 n)$ rounds, and Ene et al. [17] achieved a $(1/e - \varepsilon)$-approximation in $O(\frac{1}{\varepsilon^2} \log^2 n)$ rounds.

In the unconstrained non-monotone case, two independent works developed a $(1/2 - \varepsilon)$-approximation in $\tilde{O}(1/\varepsilon)$ rounds [11, 16]. This is achieved through a low-adaptivity version of the *double greedy* algorithm of Buchbinder et al [8]. Interestingly, the unconstrained non-monotone case seems unique in that the number of rounds is *independent of n*. From hardness results in the sequential model, it is known that $1/2$ is the optimal factor for unconstrained submodular maximization [21], and $1 - 1/e$ is optimal for cardinality constrained monotone submodular maximization [20].

Considering that so many different algorithms are exhibiting a similar behavior — adaptive complexity blowing up polynomially as we approach the optimal approximation factor, a natural question is whether this is necessary. The only non-trivial lower bound that we are aware of appears in the initial work by Balkanski and Singer [5]: $\Omega\left(\frac{\log n}{\log \log n}\right)$ rounds are necessary to achieve a $\frac{1}{\log n}$-approximation for the cardinality-constrained monotone maximization problem. No stronger lower bounds were known for achieving constant factors, even close to $1 - 1/e$.

1.1 Our results

In this work, we prove two main results.

Monotone submodular maximization. Our first result is a *polynomial* lower bound on the adaptive complexity of algorithms for cardinality-constrained monotone submodular maximization.

Theorem 2.1. *For any $r, n \in \mathbb{N}$, $r < n^c$, where $c > 0$ is some absolute constant, there is no algorithm using r rounds of queries and achieving better than a $(1 - 1/e - \Omega(\min\{\frac{1}{r}, \frac{\log^2 n}{r^3}\}))$-approximation for monotone submodular maximization subject to a cardinality constraint (on a ground set of size n).*

This is the first result showing that if we want to approach the optimal factor of $1 - 1/e$, the adaptive complexity must blow up to polynomially large factors. (The hard instances of [5] are unrelated to the factor of $1 - 1/e$; they allow one to compute the optimal solution in $O\left(\frac{\log n}{\log \log n}\right)$ rounds.)

As the statement of the result suggests, we consider two regimes of $\varepsilon := \Omega(\min\{\frac{1}{r}, \frac{\log^2 n}{r^3}\})$: For $\varepsilon > c'/\log n$, our lower bound on the number of rounds to achieve a $(1-1/e-\varepsilon)$-approximation is $\Omega(1/\varepsilon)$. For $\varepsilon < c'/\log n$, our lower bound is $\Omega\left(\frac{\log^{2/3} n}{\varepsilon^{1/3}}\right)$. Apart from this quantitative difference, the first regime has another feature: We provide a single (randomized) instance for a given n such that achieving a $(1 - 1/e - \varepsilon)$-approximation requires $\Omega(1/\varepsilon)$ rounds for every $\varepsilon > c'/\log n$. The instances in the second regime are different depending on the value of ε.

As a building block for our result, we design a simple hard instance for (sequential) monotone submodular maximization. It implies the following result, which could be of independent interest.

Theorem B.2. *For the monotone submodular maximization problem subject to a cardinality constraint, $\max\{f(S) : |S| \leq k\}$, any $(1 - 1/e + \Omega(n^{-1/4}))$-approximation algorithm on instances with n elements would require exponentially many value queries.*

As far as we know, the lower bounds known so far [20, 34, 35] only showed the hardness of $(1 - 1/e + \Omega(1/\log n))$-approximations.

Unconstrained submodular maximization. Following our hardness result, it is natural to ask whether a similar lower bound holds for unconstrained submodular maximization. Here, the optimal approximation factor is $1/2$ and it is known that a $(1/2-\varepsilon)$-approximation can be achieved in $\tilde{O}(1/\varepsilon)$ adaptive rounds [11, 16]. Hence, a lower bound analogous to our first result, where it takes $\Omega(1/\epsilon)$ rounds to approach the optimal factor within ε, would be *optimal* here. Nevertheless, we show that the situation here is substantially different.

Theorem 4.1. *Let $f : 2^E \to \mathbb{R}_+$ be a non-monotone submodular function with maximum value OPT, and let R denote a uniformly random subset of E. If $\mathbf{E}[f(R)] \leq (1/2 - \delta)OPT$, then there is an algorithm using $O(1/\delta^2)$ adaptive rounds that achieves value $(1/2 + \Omega(\delta^2))OPT$.*

Phrased differently, a $1/2 - \varepsilon$ approximation to unconstrained non-monotone submodular optimization takes at most $O\left(\min\{\frac{1}{\delta^2}, \frac{1}{\varepsilon}\}\right)$ rounds, where $\mathbf{E}[f(R)] = (\frac{1}{2} - \delta)OPT$. Unlike the monotone case, there is no blowup when $\varepsilon \to 0$, as long as $\mathbf{E}[f(R)]$ is bounded away from $\frac{1}{2}OPT$.

Also, this means that the hardest instances for many rounds are the easiest ones for 1 round. There are no instances of unconstrained submodular maximization such that (i) it is impossible to achieve a factor better than $1/2$ regardless of the number of rounds, (ii) it takes $\Omega(1/\varepsilon)$ rounds to achieve a factor of $1/2 - \varepsilon$. Either it takes a constant number of rounds to achieve a factor better than $1/2$, or a random set is already very close to $\frac{1}{2}OPT$.

1.2 Our techniques

Lower bound for cardinality-constrained maximization. Our construction consists of a sequence of "layers" of elements $X_1, X_2, \ldots, X_\ell, Y$ such that one can learn only one layer in one round of queries. At this level, our construction is analogous to that of Balkanski and Singer [5]. However, a key difference is that the number of layers in [5] is limited to $\Theta(\log n / \log\log n)$ since the construction forces the size of each layer to shrink by a *minimum* factor of $\Theta(\log^c n)$ compared to the previous one. In our construction, the layers shrink by constant factors. Depending on the shrinkage rate of the layers, we obtain a trade-off between the approximation ratio and the adaptive complexity. When the layers shrink by a constant factor, there are $\Theta(\log n)$ layers and the best solution obtained after peeling away r layers is at most $1 - 1/e - \Theta(1/r)$. To increase the number of layers to $r > \log n$, the shrinkage factor is going to be $1 + \Theta\left(\frac{\log n}{r}\right)$ and one cannot achieve a factor better than $1 - 1/e - \Omega\left(\frac{\log^2 n}{r^3}\right)$ in r rounds.

The main difficulty is how to design the instance so that (i) the layers shrink by constant factors, and (ii) given $X_1, \ldots, X_r$, we can "see" only X_{r+1} but not the further layers. In [5], this is achieved by arguing that for non-trivial queries, the number of elements we can possibly catch in X_{r+2} etc. is so small that these layers do not make any difference. However, this argument doesn't work when the layers shrink by a constant factor; in this setting it is easy to catch many elements in X_{r+2}, X_{r+3}, etc.

We resolve this issue by an adaption of the *symmetry gap* technique: We design a monotone submodular function f as a composition of functions h_k inspired by the symmetry gap construction of [21, 35]; h_k treats a pair of layers (X_k, X_{k+1}) in a *symmetric way* so that a typical query cannot distinguish the elements of X_k from the elements of X_{k+1}. However, given X_k, it is possible to use h_k to determine X_{k+1}. This leads to the desired effect of not being able to determine X_{k+1} until one round after we have learned X_k. The indistinguishability argument does not rely on the inability to find many elements in a certain layer, but on the inability to find sets which are significantly *asymmetric*, or unbalanced with respect to different layers.

In addition, we need to design the function so that solutions found after a limited number of rounds are worse than $(1 - 1/e)OPT$. We achieve this by applying the symmetrization in a way different from previous applications: asymmetric solutions are *penalized* here compared to symmetric solutions (except for the optimal solution on Y). In addition, an initial penalty is imposed on the set X_1, which makes it disadvantageous to pick a uniformly random solution (and this also enables an adaptive algorithm to get started, by distinguishing X_1 from the other sets). Finally, Y contains a hard instance showing the optimality of the factor $1 - 1/e$. In other words, even if we learn all the layers, we still cannot achieve a factor better than $1 - 1/e$.

This construction can be extended to the setting where the shrinkage factor is closer to 1, $|X_{i+1}| = |X_i|/(1 + \frac{\log n}{r})$, with some additional technicalities. The construction in this case is a separate one for each value of r; the number of layers is $r + 1$ and we argue about the quality of solutions that can be found in $r - 1$ rounds. For technical reasons, the hardness factor here approaches $1 - 1/e$ proportionally to $\log^2 n / r^3$ rather than $1/r$. For ease of exposition, we describe the construction for $r = O(\log n)$ in the main body of the paper and defer the more general construction to the appendix.

Improved approximation for unconstrained submodular maximization. In the non-monotone case, it is known that a random set gets at least 1/4 of the optimal solution in expectation [21]. Furthermore, a 1/2-approximation is best possible [21, 35], no matter how many rounds of queries are allowed. Thus generally, the value of a random set is expected to be between $\frac{1}{4}OPT$ and $\frac{1}{2}OPT$. However, for the known cases (e.g. directed cut instances) where a random set achieves exactly $\frac{1}{4}OPT$, it is actually easy to find the optimal solution (for example by the "double greedy" algorithm). Conversely, for known hard instances, where double greedy is close to $\frac{1}{2}OPT$, a random set also gets roughly $\frac{1}{2}OPT$. We prove that there is a trade-off between the performance of the random set and a variant of the double greedy algorithm, by relating them to integrals that can be compared using the Cauchy-Schwarz inequality. Consequently, we prove that the gain of the double greedy algorithm over $\frac{1}{2}OPT$ grows at least quadratically in $\delta = \frac{1}{2} - \frac{\mathbf{E}[f(R)]}{OPT}$.

1.3 Paper organization

The rest of the paper is organized in the following way. In Section 2 we give a detailed construction of our lower bound for $r = O(\log n)$ adaptive rounds. In Section 3, we outline a similar construction that extends to $r = O(n^c)$, with full proofs deferred to the appendix. Section 4 presents our improved result for unconstrained submodular maximization. Finally, the appendix contains some basic results as well as a new hardness instance for cardinality-constrained submodular maximization that may be of independent interest.

2 LOG-ROUND LOWER BOUND FOR MONOTONE SUBMODULAR MAXIMIZATION

In this section we describe our hard instance which proves the following result.

Theorem 2.1. *For any n, there is a family of instances of monotone submodular maximization subject to a cardinality constraint on a ground set of size n, such that for any $r < \frac{1}{3}\log_2 n$, any algorithm using r rounds of queries achieves at most a $(1 - 1/e - \Theta(1/r))$-approximation.*

We note that this construction works only for $r = O(\log n)$; we provide a more general construction later in the paper, which works for $r = O(n^c)$ for some constant $c > 0$. The two constructions are in fact quite similar. An advantage of the construction for $r = O(\log n)$, apart from ease of exposition, is that it provides a single instance for which it is hard to achieve better than $(1 - 1/e - \Theta(1/r))$-approximation ir r rounds for any $r = O(\log n)$, and hard to achieve a better than $(1 - 1/e)$-approximation in any number of rounds. We discussed in the introduction why this is interesting vis-a-vis the unconstrained optimization problem.

2.1 Construction of the objective function

Let $E = X_1 \cup \ldots \cup X_\ell \cup Y_1 \cup \ldots Y_{\ell'}$ where $|E| = 2^{3\ell}$, $|X_i| = 2^{3\ell - i}$ and $k = |Y_i| = \frac{1}{\ell'}2^{2\ell} = n^{5/8}$, $\ell' = 2^{\ell/8} = n^{1/24}$ (ℓ is a multiple of 8). The partition is uniformly random among all partitions with these parameters. We define the objective function as a function of real variables $f(x_1, x_2, \ldots, x_\ell, y_1, \ldots, y_{\ell'})$ where the interpretation of x_i, y_i is that for a set $S \subseteq E$, $x_i = \frac{1}{k}|S \cap X_i|$ and $y_i = \frac{1}{k}|S \cap Y_i|$. We consider a function in the following form:

$$f_1(x_1, \ldots, x_\ell, y_1, \ldots, y_{\ell'}) = 1 - (1 - h(0, x_1))(1 - h(x_1, x_2)) \cdots (1 - h(x_{\ell-1}, x_\ell))(1 - h_1(x_\ell))(1 - g(y_1, \ldots, y_{\ell'})).$$

We design g, h, h_1 to be "smooth monotone submodular" functions (with non-negative and non-increasing partial derivatives), which means that the same properties are inherited by f_1. The actual objective function is going to be obtained by discretization of the function $f = \min\{f_1, 1 - \varepsilon\}$ ($\varepsilon > 0$ to be specified later). This defines a monotone submodular function (we refer the reader to Appendix A for details).

Before proceeding to technical details, we wish to make the following points:

- $h(x, x')$, $h_1(x)$ and $g(y_1, \ldots, y_{\ell'})$ are going to be non-decreasing continuous functions with range $[0, 1]$ and non-increasing first partial derivatives. It is well-known that this corresponds to monotone submodular functions in the discrete setting. Also, it is easy to verify that $f(x_1, \ldots, x_\ell, y_1, \ldots, y_{\ell'})$ defined as above inherits the same property of its partial derivatives, and for $x_i = \frac{1}{k}|S \cap X_i|$, $y_j = \frac{1}{k}|S \cap Y_j|$, this defines a monotone submodular function of S. We recap this in Appendix A.
- We set $h_1(x_\ell) = 1 - e^{-\frac{1}{2}x_\ell}$. This is the missing contribution of x_ℓ due to the fact that it appears only in one pair.
- $g(y_1, \ldots, y_{\ell'})$ encodes a hard instance demonstrating the impossibility of beating the factor of $1 - 1/e$. That is, a solution like $g(\frac{1}{\ell'}, \frac{1}{\ell'}, \ldots, \frac{1}{\ell'})$ should have value $1 - 1/e$, while the optimum should be close to 1.
- $h(x_k, x_{k+1})$ is a *symmetry gap* instance which makes it hard to distinguish the sets associated with the variables x_k, x_{k+1}. More specifically, it should hold that for any solution that roughly satisfies

$x_k = 2x_{k+1}$ (which is true for a random solution), we have $h(x_k, x_{k+1}) = 1 - e^{-(x_k+x_{k+1})/2}$. I.e., the contributing factor depends only on $x_k + x_{k+1}$ which means that it does not distinguish between the elements of X_k and X_{k+1}.

- For solutions deviating from $x_k = 2x_{k+1}$, the value decreases. Hence, the solutions using only the variables $x_1, \dots, x_r$ that achieve a value of $1 - 1/e$ would need to satisfy $x_k = 2x_{k+1}$ (roughly). However, this would force x_1 to be a non-trivially large value, which makes f suffer from the $1 - h(0, x_1)$ factor. This is essentially the reason why good solutions cannot be found in a limited number of rounds, although the precise analysis is a bit more complicated.

Next, we specify the function $h(x, x')$ in more detail. In the following, we work with a general parameter $\varepsilon > 0$, which will be eventually chosen to be $\varepsilon = 2n^{-1/24}$.

Lemma 2.2. *For $\varepsilon > 0$, define $h : [0, \infty) \times [0, \infty) \to [0, 1]$ as*

- $h(x, x') = 1 - e^{-\frac{1}{2}(x+x')}$, *if* $|x - 2x'| \le \varepsilon$.
- $h(x, x') = 1 - (\frac{2}{3}e^{-\frac{3}{4}x+\frac{1}{4}\varepsilon} + \frac{1}{3}e^{-\frac{3}{2}x'-\frac{1}{2}\varepsilon})$, *if* $x - 2x' \ge \varepsilon$,
- $h(x, x') = 1 - (\frac{2}{3}e^{-\frac{3}{4}x-\frac{1}{4}\varepsilon} + \frac{1}{3}e^{-\frac{3}{2}x'+\frac{1}{2}\varepsilon})$, *if* $x - 2x' \le -\varepsilon$.

Then $h(x, x')$ is well-defined (the values coincide for $|x - 2x'| = \varepsilon$), its first partial derivatives are continuous, positive and decreasing in both variables.

(Note: the asymmetry between x and x' in the expressions comes from the fact that for a random solution we expect $x = 2x'$, which defines the "symmetric region" here.)

Proof. For $x - 2x' = \varepsilon$, the first definition gives $h(x, x') = 1 - e^{-\frac{1}{2}(x+x')} = 1 - e^{-\frac{3}{2}x'-\frac{1}{2}\varepsilon}$, while the second definition gives $h(x, x') = 1 - (\frac{2}{3}e^{-\frac{3}{4}x+\frac{1}{4}\varepsilon} + \frac{1}{3}e^{-\frac{3}{2}x'-\frac{1}{2}\varepsilon}) = 1 - e^{-\frac{3}{2}x'-\frac{1}{2}\varepsilon}$, so the two definitions are consistent. Let us verify the partial derivatives now. The first definition gives $\frac{\partial h}{\partial x} = \frac{1}{2}e^{-\frac{1}{2}(x+x')}$ while the second definition gives $\frac{\partial h}{\partial x} = \frac{1}{2}e^{-\frac{3}{4}x+\frac{1}{4}\varepsilon}$. Just like above, it is easy to verify that the two expressions coincide for $x - 2x' = \varepsilon$, and hence $\frac{\partial h}{\partial x}$ is continuous there. Similarly, we can verify that for $x - 2x' = -\varepsilon$ the two definitions give continuous partial derivatives. Finally, the partial derivatives are obviously positive and decreasing in each of the two variables. $\square$

Next, we state the conditions that we require for the function $g(y_1, \dots, y_{\ell'})$. We prove the following lemma in Appendix B.

Lemma 2.3. *For any $0 < \varepsilon < 1$ and $\ell' \ge 2/\varepsilon^2$, there is a function $g : [0, 1]^{\ell'} \to [0, 1]$ such that*

- *g is continuous, non-decreasing in each coordinate, and it has first partial derivatives $\frac{\partial g}{\partial y_i}$ almost everywhere*[1] *which are non-increasing in every coordinate.*
- *We have $g(1, 0, 0, \dots, 0) = 1 - \varepsilon$.*
- *If $|y_i - \frac{1}{\ell'}\sum_{i=1}^{\ell'} y_i| \le \varepsilon/2$ for all $i \in [\ell']$, then $g(\mathbf{y}) = \min\{1 - e^{-\sum_{j=1}^{\ell'} y_j}, 1 - \varepsilon\}$.*

Recall that we use $g(y_1, \dots, y_{\ell'})$ and $h(x, x')$ to define the function f_1 and $f = \min\{f_1, 1 - \varepsilon\}$ as above. The conditions stated in Lemma 2.2 and Lemma 2.3 imply that a function obtained from f by discretization is monotone submodular (see e.g. [35] for more details).

Next, we prove a quantitative bound on how much the value of h (and henceforth f) decreases when the variables deviate from the symmetric region $x_k = 2x_{k+1}$.

Lemma 2.4. *Suppose $|x - 2x'| = \delta \ge \varepsilon$. Then for $h(x, x')$ defined as in Lemma 2.2,*

$$h(x, x') \le 1 - e^{-\frac{1}{2}(x+x')+\frac{1}{16}(\delta-\varepsilon)^2}.$$

[1] To be more precise, the partial derivatives are defined almost everywhere on any axis-aligned line, in the sense of Lemma A.1.

Proof. For $x - 2x' = \delta \geq \varepsilon$, we have

$$h(x,x') = 1 - \left(\frac{2}{3}e^{-\frac{3}{4}x+\frac{1}{4}\varepsilon} + \frac{1}{3}e^{-\frac{3}{2}x'-\frac{1}{2}\varepsilon}\right) = 1 - e^{-\frac{1}{2}(x+x')}\left(\frac{2}{3}e^{-\frac{1}{4}(\delta-\varepsilon)} + \frac{1}{3}e^{\frac{1}{2}(\delta-\varepsilon)}\right).$$

We use two elementary bounds: $e^t \leq 1 + t + t^2$ for $|t| \leq 1$, and $e^t \leq 1 + 2t$ for $0 \leq t \leq 1$. Hence,

$$\begin{aligned} h(x,x') &\leq 1 - e^{-\frac{1}{2}(x+x')}\left(\frac{2}{3}\left(1 - \frac{1}{4}(\delta-\varepsilon) + \frac{1}{16}(\delta-\varepsilon)^2\right) + \frac{1}{3}\left(1 + \frac{1}{2}(\delta-\varepsilon) + \frac{1}{4}(\delta-\varepsilon)^2\right)\right) \\ &= 1 - e^{-\frac{1}{2}(x+x')}\left(1 + \frac{1}{8}(\delta-\varepsilon)^2\right) \leq 1 - e^{-\frac{1}{2}(x+x')+\frac{1}{16}(\delta-\varepsilon)^2}. \end{aligned}$$

Symmetrically, we get the same bound for $x - 2x' = -\delta \leq -\varepsilon$. □

Lemma 2.5. *Suppose that an algorithm uses $r-1$ rounds of adaptivity, $r < \ell$. Then with high probability, the only solutions it can find are, up to additive error $O\left(n^{-1/24}\right)$ in each coordinate, in the form $(x_1, \ldots, x_r, \frac{1}{2}x_r, \frac{1}{4}x_r, \ldots, \frac{1}{2^{\ell-r}}x_r, y_1, \ldots, y_{\ell'})$, where $y_i = \frac{1}{\ell' 2^{\ell-r}}x_r$.*

Proof. We prove the following by induction: With high probability, the computation path of the algorithm and the queries it issues in the r-th round are determined by $X_1, X_2, \ldots, X_{r-1}$ (and do not depend on the way that $X_r \cup X_{r+1} \cup \ldots \cup X_\ell \cup Y_1 \cup \ldots \cup Y_{\ell'}$ is partitioned into $X_r, \ldots, X_\ell, Y_1, \ldots, Y_{\ell'}$).

As a first step, we assume the algorithm is deterministic by fixing its random bits and choose the partition of E into X_i and Y_i uniformly at random.

To prove the inductive claim, let $\mathcal{E}_r$ denote the "atypical event" that the algorithm issues any query Q in round r such that the answer is *not* in the form $f = \min\{\tilde{f}_1, 1-\varepsilon\}$,

$$\tilde{f}_1 = 1 - (1 - h(0,x_1))(1 - h(x_1,x_2))\ldots(1 - h(x_{r-1},x_r))e^{-\frac{1}{2}x_r - \sum_{i=r+1}^{\ell} x_i - \sum_{j=1}^{\ell'} y_j}, \tag{1}$$

where $x_i = \frac{1}{k}|Q \cap X_i|$ and $y_j = \frac{1}{k}|Q \cap Y_j|$. Assuming that $\mathcal{E}_r$ does not occur, all answers to queries in round r are in this form, and in particular they depend only on Q and the sets $X_1, \ldots, X_r$. (The summation $\sum_{i=r+1}^{\ell} x_i + \sum_{j=1}^{\ell'} y_j$ is determined by $|Q \setminus (X_1 \cup \ldots \cup X_r)|$.) Assuming that the queries in round r depend only on $X_1, \ldots, X_{r-1}$, and $\mathcal{E}_r$ does not occur, this means that the entire computation path in round r is determined by $X_1, \ldots, X_r$. By induction, we conclude that if none of $\mathcal{E}_1, \ldots, \mathcal{E}_r$ occurs, the computation path in round r is determined by $X_1, \ldots, X_r$.

In the following we focus on the analysis of the event $\mathcal{E}_r$. Let $\mathcal{Q}_r$ denote the queries in round r, assuming that none of $\mathcal{E}_1, \ldots, \mathcal{E}_{r-1}$ occurred so far. $\mathcal{Q}_r$ is determined by $X_1, \ldots, X_{r-1}$. Conditioned on $X_1, \ldots, X_{r-1}$, the partitioning of $X_r \cup X_{r+1} \cup \ldots \cup X_\ell \cup Y_1 \cup \ldots \cup Y_{\ell'}$ is uniformly random. This implies that for each query Q, the set $Q \setminus (X_1 \cup \ldots \cup X_{r-1})$ is partitioned randomly into $Q \cap X_r, \ldots, Q \cap X_\ell, Q \cap Y_1, \ldots, Q \cap Y_{\ell'}$ and the cardinalities $|Q \cap X_i|, |Q \cap Y_j|$ are concentrated around their expectations. We have $\mathbf{E}[|Q \cap X_{i+1}|] = \frac{1}{2}\mathbf{E}[|Q \cap X_i|]$ and $\mathbf{E}[|Q \cap Y_j|] = \frac{1}{\ell'}\mathbf{E}[|Q \cap Y|] = \frac{1}{\ell'}\mathbf{E}[|Q \cap X_\ell|]$ for any $r \leq i < \ell$ and $1 \leq j \leq \ell'$. By Hoeffding's bound[2], for $x_i = \frac{1}{k}|Q \cap X_i|$, $i \geq r$, and conditioned on the choice of $X_1, \ldots, X_{r-1}$,

$$\Pr(|x_i - \mathbf{E}x_i| > \alpha) \leq 2\exp\left(-\alpha^2 k^2/|X_i|\right) \leq 2\exp\left(-\alpha^2 n^{1/4}\right), \tag{2}$$

where we used $k = n^{2/3}/\ell' = n^{5/8}$ and $|X_i| \leq n/2$. Similarly, $|y_j - \mathbf{E}y_j| > \alpha$ with probability at most $2e^{-\alpha^2 n^{1/4}}$. We set $\alpha = \frac{1}{2}\varepsilon = n^{-1/24}$ to obtain a high probability bound in the form $1 - e^{-\Omega(n^{1/6})}$.

If $|x_i - \mathbf{E}x_i| \leq \varepsilon/2$ and $|y_j - \mathbf{E}y_j| \leq \varepsilon/2$ for all $i \geq r$ and $j \leq \ell$, this means that the query Q is in the symmetric region for all the relevant evaluations of $h(x_i, x_{i+1})$ and $g(y_1, \ldots, y_{\ell'})$. Therefore, by construction the answer will be in the form (1), which only depends on Q and $X_1, \ldots, X_r$.

[2] Technically, Hoeffding's bound does not apply directly, since elements do not appear in $X_1, X_2, \ldots$ independently. However, due to the cardinality constraints, the appearances of elements are negatively correlated, so Hoeffding's bound still applies; see [33]

Let us bound the probability of $\mathcal{E}_r \setminus (\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots \cup \mathcal{E}_{r-1})$. If we condition on $X_1, \ldots, X_{r-1}$, assuming that none of $\mathcal{E}_1, \ldots, \mathcal{E}_{r-1}$ occurred, the query set Q_r in round r is fixed. By a union bound over Q_r, the probability that any of them violates (1) is $e^{-\Omega(-n^{1/6})}$. Hence,

$$\Pr[\mathcal{E}_r \setminus (\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots \cup \mathcal{E}_{r-1}) \mid X_1, X_2, \ldots, X_{r-1}] \leq \mathrm{poly}(n)\, e^{-\Omega(n^{-1/6})} = e^{-\Omega(n^{-1/6})}.$$

Now we can average over the choices of $X_1, \ldots, X_{r-1}$ and still obtain

$$\Pr[\mathcal{E}_r \setminus (\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots \cup \mathcal{E}_{r-1})] = e^{-\Omega(n^{-1/6})}.$$

Therefore, by induction,

$$\Pr\left[\bigcup_{i=1}^{r} \mathcal{E}_i\right] = \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2 \setminus \mathcal{E}_1] + \ldots + \Pr[\mathcal{E}_r \setminus (\mathcal{E}_1 \cup \ldots \cup \mathcal{E}_{r-1})] = re^{-\Omega(n^{-1/6})} = e^{-\Omega(n^{-1/6})}.$$

This implies that with high probability, the computation path in round r is determined by $X_1, \ldots, X_{r-1}$.

Consequently, a solution returned after $r-1$ rounds is determined by $X_1, \ldots, X_{r-1}$ with high probability. By the same Chernoff-Hoeffding bounds, the solution is with high probability in the form

$$\left(x_1, x_2, \ldots, x_r, \frac{1}{2}x_r, \frac{1}{4}x_r, \ldots, \frac{1}{2^\ell}x_r, y_1, \ldots, y_{\ell'}\right), \quad y_i = \frac{1}{\ell' 2^{\ell-r}} x_r,$$

up to additive error $O\left(n^{-1/24}\right)$ in each coordinate.

Finally, we note that by allowing the algorithm to use random bits, the results are a convex combination of the bounds above, so the same high probability bounds are satisfied. □

We remark an adaptive algorithm can indeed learn the identity of $X_1, \ldots, X_r$ in the first r rounds, so in this sense our analysis is tight. In the r-th round, we can determine X_r by examining the marginal values of elements with respect to a random subset of $E \setminus (X_1 \cup \ldots \cup X_{r-1})$. After the r rounds, an adaptive algorithm can completely determine $\{X_i\}_{i \leq r}$ and is free to choose the values of $x_1, \ldots, x_r$ in a query, but not the further variables.

2.2 Analysis of an r-round algorithm

Here we bound the value that an algorithm can possibly achieve in r rounds.

Lemma 2.6. *The minimum of the following optimization problem has value at least* $\frac{1}{4r}$*:*

$$\min_{\mathbf{x}} f(\mathbf{x}) := 4x_1^2 + \sum_{i=2}^{r} (2x_i - x_{i-1})^2 : \quad \sum_{i=1}^{r-1} x_i + 2x_r \geq \frac{1}{2},\ x_i \geq 0. \tag{3}$$

Proof. By the convexity of x^2, we have

$$f(\mathbf{x}) \geq r\left(\frac{1}{r}2x_1 + \frac{1}{r}\sum_{i=2}^{r}(2x_i - x_{i-1})\right)^2 = \frac{1}{r}\left(\sum_{i=1}^{r-1} x_i + 2x_r\right)^2 \geq \frac{1}{4r}.$$

□

Theorem 2.7. *Any r-round adaptive algorithm for monotone submodular optimization can achieve at most a $1 - 1/e - \Omega(1/r)$ approximation, for $r \leq \frac{1}{3}\log_2 n$ where n is the number of elements.*

Proof. By Lemma 2.5, in r rounds we can only find solutions of the form

$$(x_1, x_2, \ldots, x_r, \frac{1}{2}x_r, \frac{1}{4}x_r, \ldots, \frac{1}{2^\ell}x_r, y_1, \ldots, y_{\ell'}), \quad y_i = \frac{1}{\ell' 2^{\ell-r}} x_r$$

up to $\pm n^{-1/24}$ error in each coordinate. Choosing $\varepsilon = 2n^{-1/24}$, any solution found after r rounds has w.h.p. a value consistent with the function $f = \min\{\tilde{f}_1, 1-\varepsilon\}$,

$$\tilde{f}_1 = 1 - (1-h(0,x_1))(1-h(x_1,x_2))\cdots(1-h(x_{r-2},x_{r-1}))\exp\left(-\frac{1}{2}x_{r-1} - \sum_{i=r}^{\ell} x_i - \sum_{i=1}^{\ell'} y_i\right).$$

By Lemma 2.4, $h(x_{i-1},x_i) \leq 1 - e^{-\frac{1}{2}(x_{i-1}+x_i)+\frac{1}{16}(|2x_i-x_{i-1}|-\varepsilon)^2}$ and so

$$\begin{aligned}
\tilde{f}_1 &\leq 1 - \exp\left(-\sum_{i=1}^{\ell} x_i - \sum_{i=1}^{\ell'} y_i\right)\exp\left(\frac{1}{16}(2x_1-\varepsilon)^2 + \frac{1}{16}\sum_{i=2}^{r}(|2x_i - x_i| - \varepsilon)^2\right) \\
&\leq 1 - \exp\left(-\sum_{i=1}^{\ell} x_i - \sum_{i=1}^{\ell'} y_i\right)\exp\left(\frac{1}{16}\left(4x_1^2 + \sum_{i=2}^{r}(2x_i - x_{i-1})^2\right) - \frac{1}{8}\varepsilon\left(2x_1 + \sum_{i=2}^{r}|2x_i - x_{i-1}|\right) + \varepsilon^2 r\right) \qquad (4) \\
&\leq 1 - \exp\left(-\sum_{i=1}^{\ell} x_i - \sum_{i=1}^{\ell'} y_i\right)\exp\left(\frac{1}{64r} - \varepsilon\right) \\
&\leq 1 - 1/e - \Omega(1/r).
\end{aligned}$$

Since $\tilde{f}_1$ is monotone, we can assume that the solution has maximum possible cardinality, which means $\sum_{i=1}^{\ell} x_i + \sum_{j=1}^{\ell'} y_j = 1$. We then use the cardinality constraint to bound $\sum_{i=1}^{r-1} x_i + 2x_r \geq \sum_{i=1}^{\ell} x_i + \sum_{j=1}^{\ell'} y_j - O(\ell'\varepsilon) \geq 1/2$ w.h.p., Lemma 2.6 to estimate $4x_1^2 + \sum_{i=2}^{r}(2x_i - x_{i-1})^2 \geq \frac{1}{4r}$, and the fact that $\varepsilon = O(n^{-1/24})$ which is negligible compared to $1/r$. The optimal solution is any Y_i, which gives $f(0,0,0,\ldots,0,1) = 1-\varepsilon = 1 - O(n^{-1/24})$. Thus in r rounds we get an approximation factor of $1 - 1/e - \Omega(1/r)$. □

3 POLY-ROUND LOWER BOUND FOR MONOTONE SUBMODULAR MAXIMIZATION

In this section we show a variation on the lower bound in Section 2, choosing blocks decreasing by factors of $1 + O\left(\frac{\log n}{r}\right)$ instead of 2. This will allow us to extend our result to $r = O(n^c)$ for some constant c but with a weaker dependence of $\varepsilon = \Theta\left(\frac{\log^2 n}{r^3}\right)$ rather than $\varepsilon = \Theta(1/r)$.

Theorem 3.1. *For any n and r satisfying $\Omega(\log n) < r < O(n^c)$, where $c > 0$ is some absolute constant, there is no algorithm using r rounds of queries and achieving better than a $\left(1 - 1/e - \Omega\left(\frac{\log^2 n}{r^3}\right)\right)$-approximation for monotone submodular maximization subject to a cardinality constraint (on a ground set of size n).*

Instead of modifying the presentation of the previous section, we show an alternative construction that – on the surface – looks quite different. Our main reason for showing this alternate construction is that it is technically simpler, although less intuitive to derive. To avoid spending time on the technicalities of the construction, we outline the main ideas of the construction below and delay precise proofs to the appendix.

3.1 Construction of the objective function

One difference in this construction is that a separate instance is needed each value of r, since the shrinkage rate between the blocks depend on r.

Let $E = X_1 \cup \ldots \cup X_r \cup Y_1 \cup \ldots Y_{\ell'}$ where $|X_{i+1}| = |X_i|/(1+\delta)$ and $|Y_i| = \frac{1}{\ell'}|X_r| = k$.[3] In the construction, we require $k = \Omega(n^{2/3})$, so we choose $\ell' = \Theta(n^{1/5})$ and $\delta = \gamma\frac{\log n}{r}$ with $0 < \gamma < 2/15$.

For the lower bound, we consider functions in the following form:

$$f(x_1,\ldots,x_r,y_1,\ldots,y_{\ell'}) = 1 - (1 - q(x_1,\ldots,x_r))(1 - g(y_1,\ldots,y_{\ell'}))$$

where $x_i = \frac{1}{k}|S \cap X_i|$ and $y_i = \frac{1}{k}|S \cap Y_i|$.

[3] We ignore the issue of rounding number to the nearest integer. It is easy to verify that this does not affect the analysis significantly.

By Lemmas proven in Appendix A, f is monotone submodular so long as q and g are monotone submodular. As in the previous section, g will be the $1 - 1/e$ hard instance constructed in Appendix B. Furthermore, the actual objective function will be $\max\{1 - \epsilon, f\}$ for a parameter ε to be specified later. Now we specify the function $q(x_1, \ldots, x_r)$.

$$q(x_1, \ldots, x_r) = 1 - \exp\left(-\sum_{i=1}^{r} x_i + \sum_{i=0}^{r-1} h((1+\delta)x_{i+1} - x_i)\right), \text{where } x_0 = 0.$$

$$h(x) = \begin{cases} 0 & x \leq \epsilon \\ \alpha(x - \epsilon)^2 & \epsilon < x \leq 2 + \epsilon \\ 4\alpha(x - 1 - \epsilon) & x > 2 + \epsilon \end{cases}$$

where α is a small enough constant.

Though somewhat unintuitive, this choice of q yields a hard instance f with proper choice of α and ε. We show in Appendix D that this instance is both monotone and submodular. Before we explain how to choose the constants, we first explain the connection between the two constructions.

3.2 Connection between the two constructions

Ignoring for now the third case of h, we see that q can be more succintly phrased as

$$q(x_1, \ldots, x_r) = 1 - \exp\left(-\sum_{i=1}^{r-1} x_i + \alpha((1+\delta)x_1 - \varepsilon)_+^2 + \alpha \sum_{i=1}^{r-1}((1+\delta)x_{i+1} - x_i - \varepsilon)_+^2\right), \tag{5}$$

where $(x)_+ = \max(x, 0)$.

Supposing for a moment that we set $\alpha = \frac{1}{16}$ and $\delta = 1$, this closely mimics the bound on the penalty function from Lemma 2.4 and Equation 4. The main difference is that only the case $(1+\delta)x_{i+1} - x_i > \varepsilon$ is penalized, as opposed to the range of $(1+\delta)x_{i+1} - x_i < -\varepsilon$.

For a query Q with $|Q| = k$ with no knowledge of any of the partitions, we expect for $(1+\delta)x_{i+1} - x_i$ to be less than ε, hiding all but the first term of the penalty function. As we learn more parts, the algorithm can spread the penalty terms among the learned layers X_i, thus lessening the penalty (as in Section 2).

3.3 Overview of lower bound

Unfortunately, some technicalities remain in the construction, which requires case 3 in the definition of h as well as a judicious choice of α. We outline the lemmas required below (with full proofs in the appendix). The following properties are true for h and q, for some constant $\alpha > 0$.

Lemma 3.2.

- *h is continuous, non-decreasing and differentiable.*
- *The derivative of h is continuous and at most 4α.*
- *$h(x) \leq 4\alpha x$ when $x \geq 0$.*

In particular, case 3 of h comes into play when proving bounds on the derivative of h. This boundedness is required to show that q has non-increasing first partial derivatives.

Lemma 3.3.

- $q(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'}) \in [0, 1]$.
- *q is continuous and it has first partial derivatives which are continuous in every coordinate.*
- *q is non-decreasing and its first partial derivative are non-increasing in each coordinate.*

These two lemmas along with Lemma A.2 imply that f is monotone submodular. Next, we show that in r rounds, the best approximation we can achieve is $1 - 1/e - \Omega\left(\frac{\log^2 n}{r^3}\right)$.

3.4 Analysis of an r-round algorithm

The following analogue of Lemmas 2.5, 2.6, and Theorem 2.7 can be shown (full details in Appendix).

Lemma 3.4. *Suppose that an algorithm uses $s-1$ rounds of adaptivity, $s<r$. Then with high probability, the only solution it can find $(x_1,\ldots,x_r,y_1,\ldots,y_{\ell'})$ satisfies the properties that $(1+\delta)x_{i+1}-x_i\le\epsilon$ for any $s\le i\le r-1$, $|y_i-\bar{y}|\le\epsilon/2$ for any $1\le i\le\ell'$ and $|x_r-\ell'\bar{y}|>\epsilon$.*

Next we bound the value that an algorithm can possibly achieve in $r-1$ rounds.

Lemma 3.5. *The minimum of the following optimization problem has value $\Omega(\delta^2 r^{-1})$:*

$$\begin{aligned}
&\min_{\mathbf{x}} f(\mathbf{x}) := (1+\delta)^2 x_1^2 + \sum_{i=2}^{r}\left((1+\delta)x_i - x_{i-1}\right)^2 : \\
&\sum_{i=1}^{r-1} x_i + x_r \ge \frac{1}{3},\ x_i \ge 0.
\end{aligned} \tag{6}$$

Theorem 3.6. *Any r-round adaptive algorithm for monotone submodular optimization can achieve at most a $1-1/e-\Omega\left(\frac{\log^2 n}{r^3}\right)$ approximation, for $r=O(n^c)$ where n is the number of elements and $c>0$ is some absolute constant.*

4 IMPROVED ANALYSIS FOR UNCONSTRAINED NON-MONOTONE MAXIMIZATION

In this section we show the following result:

Theorem 4.1. *Let $R\subseteq E$ be a uniformly random subset and $f:2^E\to\mathbb{R}_+$ be a non-monotone submodular function with maximum value OPT. If $\mathbf{E}f(R)\le(1/2-\delta)OPT$, then the low-adaptivity continuous double greedy algorithm (Algorithm 1) achieves value at least $(1/2+\Omega(\delta^2))OPT$. Furthermore, the algorithm achieves this value in $O(1/\delta^2)$ rounds.*

As previously mentioned, the current state-of-the-art algorithm for unconstrained non-monotone maximization takes $O(1/\varepsilon)$ rounds to get a $(1/2-\varepsilon)$-approximation [11, 16]. While we do not improve this result, we show that for an instance with given $\mathbf{E}[f(R)]=(1/2-\delta)OPT$, the number of rounds doesn't blow up arbitrarily as we approach the factor of 1/2 (in contrast to the monotone cardinality-constrained problem). It takes $O(\min\{\frac{1}{\delta^2},\frac{1}{\epsilon}\})$ rounds to achieve a $(1/2-\epsilon)$-approximation, and in fact a strictly *better* than 1/2-approximation in $O(1/\delta^2)$ rounds.

The main intuition for this result is that in some sense, the worst possible sequence of steps in double greedy returns exactly the $\frac{1}{2}\mathbb{1}$ point. In this case, the analysis of existing algorithms show no gain over $OPT/2$ [16]. However, this is also exactly the value of a random set, which we can evaluate in just one adaptive round.

One might expect that the closer a random set is to $OPT/2$, the easier it is to improve its value via some iterative procedure. However, we show below that there are instances where a random solution being close to 1/2 means that its difficult for any polynomial round algorithm to get much better than 1/2.

Lemma 4.2. *For any $\delta>0$, there exist submodular functions f for which $\mathbf{E}[f(R)]=(1/2-O(\delta))OPT$ and no algorithm can achieve better than a $1/2+O(\delta)$-approximation in a polynomial number of rounds.*

Proof. In fact, one can construct such instances quite easily using the hardness instances of Vondrak et al. [21]. Let $f_{1/2}$ be a 1/2-hardness instance for non-monotone submodular optimization in the value query model, as defined in Section 4.2 of [21]. The main properties of $f_{1/2}$ is that (1) any random set will have $\mathbf{E}[f(R)]=\frac{1}{2}OPT$, and (2) any sequence of polynomially many queries to f will not return a result larger than $(1/2+o(1))OPT$.

To construct our hardness instance, first partition the ground set E randomly into two halves X_1 and X_2, and let $f_\delta(S) = \delta x_1(1-x_2)OPT$, where $x_i = |S \cap X_i|/|X_i|$. Our hardness instance is then simply

$$f(S) = f_{1/2}(S) + f_\delta(S).$$

For a uniformly random subset R, we have $\mathbf{E}[f(R)] = (1/2 + \delta/4)OPT$. On the other hand, the optimal solution to f has value $(1+\delta)OPT$. However, the properties of $f_{1/2}$ guarantees that no polynomial sequence of queries can obtain a value better than $(1/2 + \delta + o(1))OPT$. Thus, relative to the optimum of f, any random set obtains a $\frac{1/2+\delta/4}{1+\delta} = (1/2 - O(\delta))$-approximation, and no polynomial-round adaptive algorithm can do better than a $(1/2 + O(\delta))$-approximation. □

4.1 Continuous double greedy

As a preliminary, we review the low-adaptivity continuous double greedy (Algorithm 1) of Ene et al. [16] with some modifications. The algorithm assumes access to the multilinear extension of f (this assumption can be removed through standard sampling techniques). We use the notation $\mathbf{v}_+$ and $\mathbf{v}_-$ to denote $\max(\mathbf{v}, \mathbf{0})$ and $\min(\mathbf{v}, \mathbf{0})$ respectively. Other arithmetic operations in Algorithm 1 is assumed to be done elementwise when applicable. Our presentation differs from that of Ene et al. in two ways. (1) we use a different update rule in the line search (simplifying and removing a logarithmic factor from the round complexity), and (2) we simplify the special cases for when $\nabla_i f(\mathbf{x}) \leq 0$ or $\nabla_i f(\mathbf{x}) \geq 0$ with the max and min operators. The update rule is derived from the double greedy algorithm of Chen et al. [11].

Discretization and implementation details. When the line searches are inexact, the analysis of Ene et al. can be applied to show that the errors incurred are at most $O(\gamma OPT)$ in total. Further discretization error is incurred by the termination condition of the while loop. This causes $\mathbf{x}$ and $\mathbf{y}$ in the discretized version to not meet exactly at the end of the algorithm. The error for this is at most $\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbb{1}\rangle \leq \gamma OPT$. Thus for an additive $O(\gamma OPT)$ error, we can assume in our analysis that the line searches are exact and the points $\mathbf{x}$ and $\mathbf{y}$ meet exactly. The algorithm also requires the exact value of OPT. This estimate can be obtained to accuracy $(1 + o(\gamma))OPT$ via $\log(1/\gamma)$ parallel runs of the algorithm. First, a constant factor approximation of OPT is obtained by sampling a random set. Then, this approximation is multiplied by successive powers of $(1 + o(\gamma))$ and the algorithm is run with all guesses in parallel. More details can be found in Ene et al. [16]. This incurs error at most $o(\gamma OPT)$, so we assume the algorithm knows OPT exactly.

Through analysis similar to Chen et al. [11] and Ene et al. [16], one can show that Algorithm 1 has the following properties:

(1) The algorithm terminates with a solution DG in $O(1/\gamma)$ rounds.
(2) The returned solution DG satisfies

$$DG \geq (1-\gamma/2)\frac{OPT}{2} + \frac{1}{4}\sum_{s>0}\eta_s \sum_{i\in E}\frac{(\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-)^2}{\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-}. \tag{7}$$

The proof of property 1 gives us a useful inequality for discretizing the algorithm, so we show it below. For property 2, we give a full analysis in the appendix.

Let η_i be the step size returned by the line search on iteration i of the while loop and $t_i = \sum_{j=0}^{i-1}\eta_i$ with $t_0 = 0$. Let η_0 be the result of the line search before the while loop.

Theorem 4.3. *Algorithm 1 terminates in $O(1/\gamma)$ rounds.*

Proof. Let $\Phi_n = \langle \nabla f(\mathbf{x}(t_n)) - \nabla f(\mathbf{y}(t_n)), \mathbb{1}\rangle$. We show that each iteration of the while loop decreases Φ_n by at least γOPT.

Algorithm 1: Low adaptivity continuous greedy algorithm. Input is a submodular function f and a parameter γ.

1 Line search for the smallest $\eta_0 \in (0, 1/2]$ such that $\langle \nabla f(\eta_0 \mathbb{1}) - \nabla f((1-\eta_0)\mathbb{1}), \mathbb{1} \rangle \leq 4OPT$
2 **if** *η_0 does not exist* **then**
3   **return** $f(\frac{1}{2}\mathbb{1})$
4 **end**
5 $\mathbf{x} = \eta_0 \mathbb{1}, \mathbf{y} = (1-\eta_0)\mathbb{1}$
6 **while** $\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbb{1} \rangle \geq \gamma OPT$ **do**
7     $\Delta_i^{(x)} = \frac{\nabla_i f(\mathbf{x})_+}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}, \forall i \in E$
8     $\Delta_i^{(y)} = \frac{\nabla_i f(\mathbf{y})_-}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}, \forall i \in E$
9   Line search for the smallest $\eta > 0$ such that

$$\left\langle \nabla f(\mathbf{x} + \eta \Delta^{(x)}), \Delta^{(x)} \right\rangle + \left\langle \nabla f(\mathbf{y} + \eta \Delta^{(y)}), \Delta^{(y)} \right\rangle \leq \left\langle \nabla f(\mathbf{x}), \Delta^{(x)} \right\rangle + \left\langle \nabla f(\mathbf{y}), \Delta^{(y)} \right\rangle - \gamma OPT$$

10     $\mathbf{x} = \mathbf{x} + \eta \Delta^{(x)}, \mathbf{y} = \mathbf{y} + \eta \Delta^{(y)}$
11 **end**
12 **return** $\max(f(\mathbf{x}), f(\mathbf{y}))$

Since $\Delta^{(x)} - \Delta^{(y)} = \mathbb{1}$, the line search condition can be rewritten as

$$\left\langle \nabla f(\mathbf{x} + \eta \Delta^{(x)}) - \nabla f(\mathbf{y} + \eta \Delta^{(y)}), \mathbb{1} \right\rangle + \left\langle \nabla f(\mathbf{y} + \eta \Delta^{(y)}), \Delta^{(x)} \right\rangle + \left\langle \nabla f(\mathbf{x} + \eta \Delta^{(x)}), \Delta^{(y)} \right\rangle \leq$$
$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbb{1} \rangle + \left\langle \nabla f(\mathbf{y}), \Delta^{(x)} \right\rangle + \left\langle \nabla f(\mathbf{x}), \Delta^{(y)} \right\rangle - \gamma OPT.$$

By submodularity, $\nabla f(\mathbf{x}) \geq \nabla f(\mathbf{x} + \eta \Delta^{(x)})$. Since $\Delta^{(y)} \leq 0$, this implies that $\left\langle \nabla f(\mathbf{x} + \eta \Delta^{(x)}), \Delta^{(y)} \right\rangle \geq \left\langle \nabla f(\mathbf{x}), \Delta^{(y)} \right\rangle$. Similarly, $\left\langle \nabla f(\mathbf{y} + \eta \Delta^{(y)}), \Delta^{(x)} \right\rangle \geq \left\langle \nabla f(\mathbf{y}), \Delta^{(x)} \right\rangle$. Thus $\Phi_{n+1} \leq \Phi_n - \gamma OPT$.

Since $\Phi_1 \leq 4OPT$, the algorithm terminates in at most $4/\gamma$ iterations of the while loop. □

Now we have all the ingredients to prove Theorem 4.1. To get some intuition for Theorem 4.1, note that whenever $\nabla f(x)_+ + \nabla f(y)_- \neq 0$, we get some gain over $OPT/2$ in DG (Equation 7). More precisely, the gain over $OPT/2$ is proportional to the L_2 norm of $\nabla f(x)_+ + \nabla f(y)_-$ under a certain non-uniform scaling. The norm of the scaling vector can be related again to the performance of the double greedy algorithm. And, we show that the gap between DG and the expected value of a uniformly random subset R is bounded by the L_1 norm of $\nabla f(x)_+ + \nabla f(y)_-$. The Cauchy-Schwarz inequality connects these three quantities.

Proof. Let $RND := \mathbf{E} f(R)$ and suppose $RND \leq (1/2 - \delta)OPT$. For the sake of brevity, we omit the t argument on x and y when it is clear from context.

First we note that if the algorithm terminates on line 2, then $f(\frac{1}{2}\mathbb{1})$ has value at least OPT and we are done. Suppose for the remainder of the proof that the algorithm has progressed past line 2.

We first estimate the gap between DG and RND, by considering the evolution of the point $\frac{\mathbf{x}(t)+\mathbf{y}(t)}{2}$ from $\frac{1}{2}\mathbb{1}$ to the output of DG (up to an additive $O(\gamma OPT)$):

$$\begin{aligned}
DG - RND &= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \mathrm{d}t \\
&= \int_0^1 \left\langle \nabla f\left(\frac{\mathbf{x}(t)+\mathbf{y}(t)}{2}\right), \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{x}(t)+\mathbf{y}(t)}{2}\right)\right\rangle \mathrm{d}t \\
&\le \frac{1}{2}\int_0^1 \sum_{i\in E} \left|\nabla_i f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\right| \left|\frac{\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}\right| \mathrm{d}t \\
&\le \frac{1}{2}\int_0^1 \sum_{i\in E} |\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-| \left|\frac{\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}\right| \mathrm{d}t \\
&= \frac{1}{2}\int_0^1 \sum_{i\in E} |\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-| \,\mathrm{d}t
\end{aligned} \tag{8}$$

where the bound on line 4 is due to the fact that the positive coordinates of $\nabla f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)$ are upper bounded by $\nabla f(\mathbf{x})_+$ and the negative coordinates are lower bounded by $\nabla f(\mathbf{y})_-$.

Next we focus on the discretization of the integral. For $\eta \in [0, \eta_i)$,

$$\begin{aligned}
&\sum_{i\in E} \left|\nabla_i f(\mathbf{x}(t_i) + \eta\Delta^{(x(t_i))})_+ + \nabla_i f(\mathbf{y}(t_i) + \eta\Delta^{(y(t_i))})_-\right| - |\nabla_i f(\mathbf{x}(t_i))_+ + \nabla_i f(\mathbf{y}(t_i))_-| \\
\le &\sum_{i\in E} \left(\nabla_i f(\mathbf{x}(t_i))_+ - \nabla_i f(\mathbf{x}(t_i) + \eta\Delta^{(x(t_i))})_+\right) - \left(\nabla_i f(\mathbf{y}(t_i)))_- - \nabla_i f(\mathbf{y}(t_i) + \eta\Delta^{(y(t_i))})_-\right) \\
\le &\left\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{x}(t_i) + \eta\Delta^{(x(t_i))}), \mathbb{1}\right\rangle - \left\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{y}(t_i) + \eta\Delta^{(y(t_i))}), \mathbb{1}\right\rangle.
\end{aligned}$$

where line 2 uses the fact that $\nabla f(\mathbf{x}) \ge \nabla f(\mathbf{x} + \eta\Delta^{(x)}) \ge \nabla f(\mathbf{y} + \eta\Delta^{(y)}) \ge \nabla f(\mathbf{y})$ by submodularity.

From the line search and the proof of Theorem 4.3, we have for $\eta < \eta_i$:

$$\left\langle \nabla f(\mathbf{x}(t_i) + \eta\Delta^{(x(t_i))}) - \nabla f(\mathbf{y}(t_i) + \eta\Delta^{(y(t_i))}), \mathbb{1}\right\rangle \ge \langle \nabla f(\mathbf{x}(t_i)) - \nabla f(\mathbf{y}(t_i)), \mathbb{1}\rangle - \gamma OPT.$$

$$\left\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{x}(t_i) + \eta\Delta^{(x(t_i))}), \mathbb{1}\right\rangle - \left\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{y}(t_i) + \eta\Delta^{(y(t_i))}), \mathbb{1}\right\rangle \le \gamma OPT.$$

Combining the inequalities from above, we have

$$\begin{aligned}
\int_0^1 \sum_{i\in E} |\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-| \,\mathrm{d}t &\le \sum_{s>0} \int_{t_s}^{t_s+\eta} \left(\gamma OPT + \sum_{i\in E} |\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-|\right) \mathrm{d}t \\
&\le \gamma OPT + \sum_{s>0} \eta_s \sum_{i\in E} |\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-| \\
&\le \gamma OPT + \sqrt{\sum_{s>0} \eta_s \sum_{i\in E} \frac{(\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-)^2}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}} \sqrt{\sum_{s>0} \eta_s \sum_{i\in E} (\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-)}.
\end{aligned}$$

where the third line comes from applying Cauchy-Schwarz in the Euclidean norm (over the combined summation over s and i).

Next, we show a bound relating DG and the scaling vector $\nabla f(\mathbf{x})_+ - \nabla f(\mathbf{y})_-$.

$$
\begin{aligned}
DG &\geq \int_0^1 \frac{1}{2} \left(\langle \nabla f(\mathbf{x}), \dot{\mathbf{x}} \rangle + \langle \nabla f(\mathbf{y}), \dot{\mathbf{y}} \rangle\right) \mathrm{d}t \\
&= \frac{1}{2} \int_0^1 \sum_{i \in E} \frac{(\nabla_i f(\mathbf{x}) \nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y}) \nabla_i f(\mathbf{y})_-)}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-} \mathrm{d}t \\
&= \frac{1}{2} \int_0^1 \sum_{i \in E} \frac{(\nabla_i f(\mathbf{x})_+^2 + \nabla_i f(\mathbf{y})_-^2)}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-} \mathrm{d}t \\
&\geq \frac{1}{4} \int_0^1 \sum_{i \in E} \left(\frac{(\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-)^2}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-} + \frac{(\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-)^2}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-} \right) \mathrm{d}t \\
&= \frac{1}{4} \sum_{s>0} \int_{t_s}^{t_s+\eta_s} \sum_{i \in E} (\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-) \mathrm{d}t \\
&\geq \frac{1}{4} \sum_{s>0} \int_{t_s}^{t_s+\eta_s} \left(-\gamma OPT + \sum_{i \in E} (\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-) \right) \mathrm{d}t \\
&\geq -\frac{\gamma}{4} OPT + \frac{1}{4} \sum_{s>0} \eta_s \sum_{i \in E} (\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-)
\end{aligned}
$$

Consequently,

$$
\sum_{s>0} \eta_s \sum_{i \in E} (\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-) \leq (4+\gamma) DG \leq (4+\gamma) OPT.
$$

Combining everything together, we have

$$
\begin{aligned}
DG - RND &\leq \frac{1}{2} \int_0^1 \sum_{i \in E} |\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-| \, \mathrm{d}t \\
&\leq \frac{\gamma}{2} OPT + \frac{1}{2} \sqrt{\sum_{s>0} \eta_s \sum_{i \in E} \frac{(\nabla_i f(\mathbf{x})_+ + \nabla_i f(\mathbf{y})_-)^2}{\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-}} \sqrt{\sum_{s>0} \eta_s \sum_{i \in E} (\nabla_i f(\mathbf{x})_+ - \nabla_i f(\mathbf{y})_-)} \qquad (9) \\
&\leq \frac{\gamma}{2} OPT + \frac{1}{2} \sqrt{4DG - 2(1-\gamma/2) OPT} \sqrt{(4+\gamma) OPT}.
\end{aligned}
$$

Thus if $RND = (1/2 - \delta) OPT$, then

$$
(DG - RND)^2 \leq (4+\gamma)(DG - (1-\gamma/2) OPT/2) OPT.
$$

This inequality is quadratic in DG, and solving for a lower bound on DG yields $DG = (1/2 + \Omega(\delta^2) - O(\gamma)) OPT$.

Finally, the result on the number of rounds follows from setting $\gamma = O(\delta^2)$. □

A SOME BASICS ON SUBMODULAR FUNCTIONS

We use the following facts in our constructions of instances of submodular maximization. These properties have been used in previous work, see e.g. [34].

Lemma A.1. *Suppose that $F : \mathbb{R}_+^n \to \mathbb{R}_+$ is a continuous function, such that*

- *for every $\mathbf{x} \in \mathbb{R}_+^n$ and $i \in [n]$, the partial derivative $\left.\frac{\partial F}{\partial x_i}\right|_{\mathbf{x}+t\mathbf{e}_i}$ is defined and continuous almost everywhere as a function of $t \in \mathbb{R}_+$,*
- *the partial derivative $\frac{\partial F}{\partial x_i}$ (wherever defined) is non-negative and non-increasing in all coordinates.*

Let $k \geq 1$ and $[n] = E_1 \cup \ldots \cup E_n$ any partition of the ground set. Then

$$f(S) = F\left(\frac{|S \cap E_1|}{k}, \frac{|S \cap E_2|}{k}, \ldots, \frac{|S \cap E_n|}{k}\right)$$

is a monotone submodular function.

Sketch of proof. If we denote $x_i = \frac{1}{k}|S \cap X_i|$, the marginal values of f are

$$\begin{aligned} f(S+i) - f(S) &= F(x_1, \ldots, x_i + 1/k, \ldots, x_n) - F(x_1, \ldots, x_i, \ldots, x_n) \\ &= \int_0^{1/k} \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}+t\mathbf{e}_i} dt \end{aligned}$$

considering that $\frac{\partial F}{\partial x_i}$ is defined almost everywhere as a function of t. The non-negativity of $\frac{\partial F}{\partial x_i}$ implies that f in monotone, and the non-increasing property of $\frac{\partial F}{\partial x_i}$ implies that f is submodular. □

Lemma A.2. *If $F_1 : \mathbb{R}_+^n \to [0,1]$ and $F_2 : \mathbb{R}_+^n \to [0,1]$ satisfy the assumptions of Lemma A.1, then so does $F : \mathbb{R}_+^n \to [0,1]$,*

$$F(\mathbf{x}) = 1 - (1 - F_1(\mathbf{x}))(1 - F_2(\mathbf{x})).$$

Sketch of proof. Assuming that the values of F_i are in $[0,1]$, the same holds true for F. The partial derivatives of F are

$$\frac{\partial F}{\partial x_i} = (1 - F_1(\mathbf{x}))\frac{\partial F_2}{\partial x_i} + (1 - F_2(\mathbf{x}))\frac{\partial F_1}{\partial x_i}.$$

Assuming that $\frac{\partial F_i}{\partial x_i}$ are non-negative and non-increasing, and hence F_1 and F_2 are non-decreasing, $\frac{\partial F}{\partial x_i}$ is non-negative and non-increasing as well. □

B HARD INSTANCE FOR $1 - 1/e$

Here we prove Lemma 2.3, which describes a function implying the hardness of achieving any factor better than $1 - 1/e$ (which is embedded in the set Y as discussed above). Although such instances are well known by now, we need a new variant, which proves the hardness of achieving a factor better than $1 - 1/e + 1/n^c$ for some constant $c > 0$, as opposed to $1 - 1/e + 1/\log^c n$, which would follow for example from [35]. Also, we need the property that in the symmetric region it takes the form $g(y_1, \ldots, y_{\ell'}) = 1 - e^{-\sum y_j}$ (as opposed to $g(y_1, \ldots, y_{\ell'}) = 1 - (1 - \frac{1}{\ell'}\sum y_j)^{\ell'}$). We provide a self-contained construction here.

Lemma B.1. *For any $0 < \varepsilon < 1$ and $\ell' \geq 2/\varepsilon^2$, there is a function $g : [0,1]^{\ell'} \to [0,1]$ such that*

- *g is continuous, non-decreasing in each coordinate, and it has first partial derivatives $\frac{\partial g}{\partial y_i}$ almost everywhere which are non-increasing in every coordinate.*
- *We have $g(1, 0, 0, \ldots, 0) = 1 - \varepsilon$.*
- *If $|y_i - \frac{1}{\ell'}\sum_{i=1}^{\ell'} y_i| \leq \varepsilon/2$ for all $i \in [\ell']$, then $g(\mathbf{y}) = \min\{1 - e^{-\sum_{j=1}^{\ell'} y_j}, 1 - \varepsilon\}$.*

Proof. We start by defining a function of a single variable:

- For $x \in [0, \varepsilon]$, $\gamma(x) = 1 - e^{-x}$.
- For $x \in [\varepsilon, 1]$, $\gamma(x) = 1 - e^{-\varepsilon}(1 - x + \varepsilon)$.

This function is continuous ($\gamma(\varepsilon) = 1 - e^{-\varepsilon}$ according to both definitions), non-decreasing, and its derivative is continuous and non-increasing ($\gamma'(x) = e^{-x}$ for $x \in [0, \varepsilon]$, and $\gamma'(x) = e^{-\varepsilon}$ for $x \in [\varepsilon, 1]$). For $x \in [0,1]$, $\gamma(x) \in [0,1]$.

Then we define

$$g(y_1, \ldots, y_{\ell'}) = \min\left\{1 - \prod_{i=1}^{\ell'}(1 - \gamma(y_i)), 1 - \varepsilon\right\}.$$

Clearly, this is a continuous non-decreasing function. As long as $\prod_{i=1}^{\ell'}(1-\gamma(y_i)) > \varepsilon$, the partial derivatives are

$$\frac{\partial g}{\partial y_i} = \gamma'(y_i) \prod_{j \neq i} (1 - \gamma(y_j))$$

which is non-increasing in each coordinate y_j (since $\gamma'(y_i)$ is non-increasing and $\gamma(y_j)$ is non-decreasing). For $\prod_{i=1}^{\ell'}(1 - \gamma(y_i)) < \varepsilon$, the partial derivatives are 0. ($\frac{\partial g}{\partial y_i}$ is discontinuous at $\prod_{i=1}^{\ell'}(1 - \gamma(y_i)) = \varepsilon$ but that is at most one point t on any line $\mathbf{y}(t) = \mathbf{x} + t\mathbf{e}_i$).)

Consider $y_1 = 1$ and $y_2 = \ldots = y_n = 0$. We have $\gamma(0) = 0$ and $\gamma(1) = 1 - e^{-\varepsilon}\varepsilon$. Therefore $g(1, 0, \ldots, 0) = \min\{1 - e^{-\varepsilon}\varepsilon, 1 - \varepsilon\} = 1 - \varepsilon$.

Finally, let $\bar{y} = \frac{1}{\ell'} \sum_{i=1}^{\ell'} y_i$ and suppose that $|y_i - \bar{y}| \leq \varepsilon/2$ for all $i \in [\ell']$. Then we distinguish two cases:

- If $\bar{y} \leq \varepsilon/2$, then $y_i \leq \varepsilon$ for all i. Therefore, $\gamma(y_i) = 1 - e^{-y_i}$ and we obtain

$$g(y_1, \ldots, y_{\ell'}) = \min\left\{1 - \prod_{i=1}^{\ell'} e^{-y_i}, 1 - \varepsilon\right\} = \min\left\{1 - e^{-\sum_{i=1}^{\ell'} y_i}, 1 - \varepsilon\right\}.$$

- If $\bar{y} > \varepsilon/2$, then $g(\mathbf{y})$ must be close to 1: by the AMGM inequality,

$$1 - \prod_{i=1}^{\ell'}(1 - \gamma(y_i)) \geq 1 - \left(\frac{1}{\ell'}\sum_{i=1}^{\ell'}(1 - \gamma(y_i))\right)^{\ell'} \geq 1 - \left(\frac{1}{\ell'}\sum_{i=1}^{\ell'}(1 - y_i(1 - \varepsilon))\right)^{\ell'} = 1 - (1 - \bar{y}(1 - \varepsilon))^{\ell'}$$

  where we used the fact that $\gamma(y_i) \geq y_i(1 - \varepsilon)$ for all $y_i \in [0, 1]$. Hence, using the assumptions that $\bar{y} > \varepsilon/2$ and $\ell' \geq 2/\varepsilon^2$,

$$1 - \prod_{i=1}^{\ell'}(1 - \gamma(y_i)) \geq 1 - e^{-\ell'\bar{y}(1-\varepsilon)} \geq 1 - e^{-(1-\varepsilon)/\varepsilon} \geq 1 - \varepsilon$$

  where we used $e^{-t} \leq \frac{1}{1+t}$ with $t = \frac{1-\varepsilon}{\varepsilon}$. Consequently, $g(y_1, \ldots, y_{\ell'}) = 1 - \varepsilon$.

□

As a corollary, we present the following implication for the submodular maximization problem. We are not aware of a prior hardness result showing a hardness factor better than $1 - 1/e + 1/\log^c n$.

Theorem B.2. *For the monotone submodular maximization problem subject to a cardinality constraint,* $\max\{f(S) : |S| \leq k\}$*, any* $(1 - 1/e + \Omega(1/n^{1/4}))$*-approximation algorithm on instances with n elements would require exponentially many value queries.*

Proof. Consider $n = 4^\ell$, $k = \ell' = \sqrt{n} = 2^\ell$ and $\varepsilon = 2/n^{1/4}$. Let $[n] = Y_1 \cup Y_2 \cup \ldots \cup Y_{\ell'}$ be a uniformly random partition of $[n]$ into $\ell' = \sqrt{n}$ blocks of size $k = \sqrt{n}$. We consider a monotone submodular function based on Lemma B.1,

$$f(S) = g\left(\frac{|S \cap Y_1|}{k}, \frac{|S \cap Y_2|}{k}, \ldots, \frac{|S \cap Y_{\ell'}|}{k}\right).$$

The optimization problem $\max\{f(S) : |S| \leq k\}$ has the solution $S = Y_1$ (or any other Y_i), which has value

$$f(Y_1) = g(1, 0, \ldots, 0) = 1 - \varepsilon$$

by Lemma B.1.

We claim that an algorithm cannot find a solution of value better than $1 - 1/e$, by arguments which are quite standard by now. For any fixed query Q, the fractions $y_i = \frac{|Q \cap Y_i|}{k}$ are well concentrated around their expectation which is $\bar{y} = \frac{1}{k\ell'}|Q| = \frac{1}{n}|Q|$. $Q \cap Y_i$ is a binomial random variable in the range $\{0, \ldots, \sqrt{n}\}$ and hence by Chernoff-Hoeffding bounds, the standard deviation for $Q \cap Y_i$ is $O(n^{1/4})$ and hence $|y_i - \bar{y}| > \frac{1}{n^{1/4}} = \varepsilon/2$ with exponentially small probability. Unless the algorithm issues exponentially many queries, with high probability it will never learn any information about the partition $(Y_1, \dots, Y_{\ell'})$ and it will return a solution which again satisfies $|y_i - \bar{y}| \le \varepsilon/2$ for all i with high probability. The value of any such solution, under the constraint that $\sum y_i \le 1$, is at most $1 - 1/e$ by Lemma B.1.

□

C ANALYSIS OF CONTINUOUS DOUBLE GREEDY

To show the guarantees of Algorithm 1, we borrow from the analysis of Chen et al. [11] and Ene et al. [16]. Let η_i be the step size returned by the line search on iteration i of the while loop and $t_i = \sum_{j=0}^{i-1} \eta_i$ with $t_0 = 0$. Let η_0 be the result of the line search before the while loop. Furthermore, let $\mathbf{x}^*$ be the optimal solution and let $\mathbf{p}(t) = \mathrm{Proj}_{[\mathbf{x}(t),\mathbf{y}(t)]}\mathbf{x}^*$ be the projection of x^* into the box defined by $\mathbf{x}$ and $\mathbf{y}$.

Theorem C.1. *The returned solution DG satisfies*

$$DG \ge (1-\gamma/2)OPT/2 + \frac{1}{4}\sum_s \eta_s \sum_{i\in E} \frac{(\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-)^2}{\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-}$$

To show Theorem C.1, we borrow the following lemmas of Ene et al. (proof of Lemma 7)[4] and Chen et al. (Corollary 3.19):

Lemma C.2. *With our choice of* $\dot{\mathbf{x}}$ *and* $\dot{\mathbf{y}}$*, we have the following inequalities:*

$$\int_0^{\eta_0} \frac{1}{2}\left(\langle \nabla f(\mathbf{x}), \dot{\mathbf{x}}\rangle + \langle \nabla f(\mathbf{y}), \dot{\mathbf{y}}\rangle\right) + \langle \nabla f(\mathbf{p}), \dot{\mathbf{p}}\rangle \,\mathrm{d}t \ge 0,$$

$$\int_{t_i}^{t_i+\eta_i} \langle \nabla f(\mathbf{p}), \dot{\mathbf{p}}\rangle \,\mathrm{d}t \ge \eta_i \sum_{i\in S(t)} \frac{\nabla_i f(\mathbf{x}(t_i))\nabla_i f(\mathbf{y}(t_i))}{\nabla_i f(\mathbf{x}(t_i)) - \nabla_i f(\mathbf{y}(t_i))}\ \forall i \in S(t_i)$$

where $S(t) = \{i \in E \mid \nabla_i f(\mathbf{x}(t)) \ge 0 \text{ and } \nabla_i f(\mathbf{y}(t)) \le 0$.

We are now ready to prove Theorem C.1.

Proof. Then we have the following sequence of inequalities:

$$\begin{aligned}
2DG - OPT &\ge \int_0^1 \frac{1}{2}\frac{\mathrm{d}f(\mathbf{x}) + f(\mathbf{y})}{\mathrm{d}t} + \frac{\mathrm{d}f(\mathbf{p})}{\mathrm{d}t}\mathrm{d}t \\
&= \sum_s \int_{t_s}^{t_s+\eta_s} \frac{1}{2}\left(\left\langle \nabla f(\mathbf{x}+\eta\Delta^{(x)}), \Delta^{(x)}\right\rangle + \left\langle \nabla f(\mathbf{y}+\eta\Delta^{(y)}), \Delta^{(y)}\right\rangle\right) + \langle \nabla f(\mathbf{p}), \dot{\mathbf{p}}\rangle \,\mathrm{d}t \\
&\ge \sum_{s>0} \int_{t_s}^{t_s+\eta_s} \frac{1}{2}\left(-\gamma OPT + \left\langle \nabla f(\mathbf{x}(t_s)), \Delta^{(x(t_s))}\right\rangle + \left\langle \nabla f(\mathbf{y}(t_s)), \Delta^{(y(t_s))}\right\rangle\right) + \langle \nabla f(\mathbf{p}), \dot{\mathbf{p}}\rangle \,\mathrm{d}t \\
&\ge \sum_{s>0} \eta_s \left(\frac{1}{2}\left(-\gamma OPT + \left\langle \nabla f(\mathbf{x}(t_s)), \Delta^{(x(t_s))}\right\rangle + \left\langle \nabla f(\mathbf{y}(t_s)), \Delta^{(y(t_s))}\right\rangle\right) + \sum_{i\in S(t)} \frac{\nabla_i f(\mathbf{x}(t_s))\nabla_i f(\mathbf{y}(t_s))}{\nabla_i f(\mathbf{x}(t_s)) - \nabla_i f(\mathbf{y}(t_s))}\right) \\
&\ge -\gamma OPT/2 + \sum_{s>0} \eta_s \left(\sum_{i\in S(t)} \frac{1}{2}\frac{(\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-)^2}{\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-} + \sum_{i\notin S(t)} \frac{1}{2}\frac{\nabla_i f(\mathbf{x}(t_s))_+^2 + \nabla_i f(\mathbf{y}(t_s))_-^2}{\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-}\right). \\
&\ge -\gamma OPT/2 + \sum_{s>0} \eta_s \sum_{i\in E} \frac{1}{2}\frac{(\nabla_i f(\mathbf{x}(t_s))_+ + \nabla_i f(\mathbf{y}(t_s))_-)^2}{\nabla_i f(\mathbf{x}(t_s))_+ - \nabla_i f(\mathbf{y}(t_s))_-}
\end{aligned}$$

where line 3 comes from the analysis of the line search in Theorem 4.3, and line 5 comes from completing the square for terms summed over $S(t)$. □

[4]The analysis actually sets $\dot{x} = \dot{y} = 0$ for i such that $\nabla_i f(x) \le 0$ or $\nabla_i f(y) \ge 0$, thus differing from our definition slightly for these i. However, their analysis is easily extended to all of E for our choice of $\dot{x}$ and $\dot{y}$ with little modifications.

D FULL ANALYSIS OF SECTION 3

Lemma 3.2.

- *h is continuous, non-decreasing and differentiable.*
- *The derivative of h is continuous and at most 4α.*
- *$h(x) \le 4\alpha x$ when $x \ge 0$.*

Proof. The first property is easy to verify. Since the derivative is non-decreasing, the derivative is bounded by the third case, which is 4α. For the third property, note that $4\alpha x$ coincides with $h(x)$ at $x = 0$, but has derivative greater or equal to $h(x)$ for all x. □

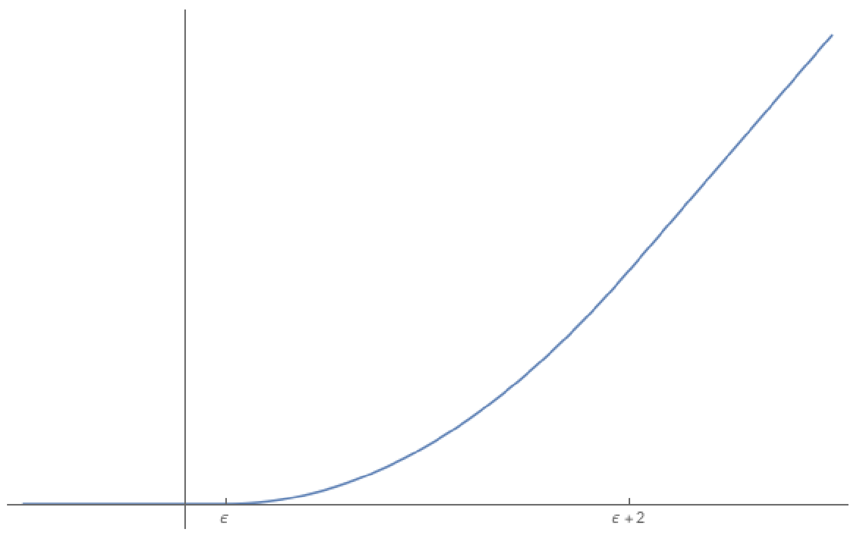

Fig. 1. An example of the function h.

Lemma 3.3.

- $q(x_1, \dots, x_r, y_1, \dots, y_{\ell'}) \in [0, 1]$.
- *q is continuous and it has first partial derivatives which are continuous in every coordinate.*
- *q is non-decreasing and its first partial derivative are non-increasing in each coordinate.*

Proof. Let us denote $q(x_1, \dots, x_r) = 1 - e^{-p(x_1,\dots,x_r)}$,

$$p(x_1, \dots, x_r) = \sum_{i=1}^{r} x_i - h((1+\delta)x_1) - \sum_{i=1}^{r-1} h((1+\delta)x_{i+1} - x_i).$$

First we prove that when each coordinate is non-negative, $q(x_1, \dots, x_r) \in [0, 1]$. The right-hand side of this inequality is obvious. In order to prove $1 - \exp(-p(x_1, \dots, x_r))$ is non-negative, we only need to prove that $p(x_1, \dots, x_r)$ is non-negative. By Lemma 3.2, h is non-decreasing and $h(x) \le 4\alpha x$ when $x \ge 0$, so we have

$$\begin{aligned} p(x_1, \dots, x_r) &= \sum_{i=1}^{r} x_i - h((1+\delta)x_1) - \sum_{i=1}^{r-1} h((1+\delta)x_{i+1} - x_i) \\ &\ge \sum_{i=1}^{r} x_i - h((1+\delta)x_1) - \sum_{i=1}^{r-1} h((1+\delta)x_{i+1}) \end{aligned}$$

$$\geq \sum_{i=1}^{r} x_i - 4\alpha(1+\delta) \sum_{i=1}^{r} x_i \geq 0$$

where we use a small enough constant $\alpha > 0$ and $\delta \leq 1$ in the last inequality.

Next we prove $q(x_1, \ldots, x_r)$ is continuous and its first partial derivatives are continuous in each coordinate. That is actually a direct corollary of Lemma 3.2. More specifically, by Lemma 3.2, h is continuous, differentiable and its first order derivative is continuous. Thus, by definition $q(x_1, \ldots, x_r)$ is also continuous and its first partial derivative is continuous in each coordinate.

Finally we prove that $q(x_1, \ldots, x_r)$ is non-decreasing and its first partial derivatives are non-increasing in every coordinate. By Lemma 3.2, the first order derivative of h is at most 4α. So by definition of p, we have

$$\frac{\partial p}{\partial x_i} \geq 1 - 4\alpha(1+\delta)$$

For the first-order partial derivative of q,

$$\frac{\partial q}{\partial x_i} = \frac{\partial p}{\partial x_i} \cdot e^{-p(x_1,\ldots,x_r)} \geq (1 - 4\alpha(1+\delta)) \cdot e^{-p(x_1,\ldots,x_r)} \geq 0$$

where we use α is a small enough constant and $\delta \leq 1$ in the last inequality. Thus q is non-decreasing. By definition of p, the second partial derivative $\frac{\partial^2 p}{\partial x_i \partial x_j}$ is non-zero if and only if x_i, x_j share the same h and that h is quadratic. Therefore $\frac{\partial^2 p}{\partial x_i \partial x_j} = 4\alpha(1+\delta)$ or 0. So for the second partial derivative of q,

$$\frac{\partial^2 q}{\partial x_i \partial x_j} = \left(\frac{\partial^2 p}{\partial x_i \partial x_j} - \frac{\partial p}{\partial x_i} \cdot \frac{\partial p}{\partial x_j}\right) \cdot e^{-p(x_1,\ldots,x_r)} \leq (4\alpha(1+\delta) - (1 - 4\alpha(1+\delta))^2) \cdot e^{-p(x_1,\ldots,x_r)} \leq 0$$

for $0 < \alpha \leq \frac{1}{24} \leq \frac{1}{12(1+\delta)}$ (for $0 < \delta \leq 1$). Thus the first partial derivatives of q are non-increasing. □

Lemma 3.4. *Suppose that an algorithm uses $s-1$ rounds of adaptivity, $s < r$. Then with high probability, the only solution $(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'})$ it can find satisfies the properties that $(1+\delta)x_{i+1} - x_i \leq \epsilon$ for any $s \leq i \leq r-1$, $|y_i - \bar{y}| \leq \epsilon/2$ for any $1 \leq i \leq \ell'$ and $|x_r - \ell'\bar{y}| > \epsilon$.*

The proof of this lemma will mimic the proof of Lemma 2.5

Proof. We first remind the reader that the parameters of our construction are $\delta = 1 + \frac{2}{15}\frac{\log n}{r}$ and $\ell' = \Theta(n^{1/5})$. This means that $k = \frac{n}{\ell'(1+\delta)^r} = \Omega(n^{2/3})$.

Continuing with the proof, we can assume by Yao's principle that the algorithm is deterministic.

We prove the following by induction: With high probability, the computation path of the algorithm and the queries it issues in the k-th round are determined by $X_1, X_2, \ldots, X_{s-1}$ (and do not depend on the way that $X_s \cup X_{s+1} \cup \ldots \cup X_r \cup Y_1 \cup \ldots \cup Y_{\ell'}$ is partitioned into $X_s, \ldots, X_r, Y_1, \ldots, Y_{\ell'}$).

To prove the inductive claim, let $\mathcal{E}_s$ denote the "atypical event" that the algorithm issues any query Q in round r such that the answer is *not* in the form $\min\{\tilde{f}, 1-\epsilon\}$,

$$\tilde{f} = 1 - \exp\left(-\sum_{i=1}^{r} x_i - \sum_{i=1}^{\ell'} y_i + h((1+\delta)x_1) + \sum_{i=1}^{s-1} h((1+\delta)x_{i+1} - x_i)\right), \tag{10}$$

where $x_i = \frac{1}{k}|Q \cap X_i|$ and $y_j = \frac{1}{k}|Q \cap Y_j|$. Assuming that $\mathcal{E}_s$ does not occur, all answers to queries in round r are in this form, and in particular they depend only on Q and the sets $X_1, \ldots, X_s$. (The summation $\sum_{i=s+1}^{r} x_i + \sum_{j=1}^{\ell'} y_j$ is determined by $|Q \setminus (X_1 \cup \ldots \cup X_s)|$.) Assuming that the queries in round r depend only on $X_1, \ldots, X_{s-1}$, and $\mathcal{E}_s$ does not occur, this means that the entire computation path in round k is determined by $X_1, \ldots, X_s$. By induction, we conclude that if none of $\mathcal{E}_1, \ldots, \mathcal{E}_s$ occurs, the computation path in round k is determined by $X_1, \ldots, X_s$.

In the following we focus on the analysis of the event $\mathcal{E}_s$. Let $\boldsymbol{Q}_s$ denote the queries in round s, assuming that none of $\mathcal{E}_1, \ldots, \mathcal{E}_{r-1}$ occurred so far. $\boldsymbol{Q}_s$ is determined by $X_1, \ldots, X_{s-1}$. Conditioned on $X_1, \ldots, X_{r-1}$, the partitioning of $X_k \cup X_{s+1} \cup \ldots \cup X_{\ell'} \cup Y_1 \cup \ldots \cup Y_{\ell'}$ is uniformly random. This implies that for each query Q, the set $Q \setminus (X_1 \cup \ldots \cup X_{s-1})$ is partitioned randomly into $Q \cap X_s, \ldots, Q \cap X_r, Q \cap Y_1, \ldots, Q \cap Y_{\ell'}$ and the cardinalities $|Q \cap X_i|, |Q \cap Y_j|$ are concentrated around their expectations. We have $(1+\delta)\mathbf{E}[|Q \cap X_{i+1}|] = \mathbf{E}[|Q \cap X_i|]$ and $\mathbf{E}[|Q \cap Y_j|] = \frac{1}{\ell'}\mathbf{E}[|Q \cap Y|] = \frac{1}{\ell'}\mathbf{E}[|Q \cap X_r|]$ for any $k \leq i < r$ and $1 \leq j \leq \ell'$. By Hoeffding's bound, for $x_i = \frac{1}{k}|Q \cap X_i|$, $i \geq s$, and conditioned on the choice of $X_1, \ldots, X_{r-1}$,

$$\begin{aligned} \Pr((1+\delta)x_{i+1} - x_i > \epsilon) &\leq \exp\left(-\frac{2\epsilon^2 k^2}{|X_i| + (1+\delta)^2 |X_{i+1}|}\right) \\ &\leq \exp\left(-n^{0.1}\right), \end{aligned} \tag{11}$$

where we use $k > \Omega(n^{2/3})$, $|X_i| + (1+\delta)^2|X_{i+1}| \leq 2n$ and $\epsilon = n^{-0.1}$. Similarly, we can prove the same bound for $|y_j - \bar{y}| > \epsilon/2$ and $|x_r - \ell'\bar{y}| > \epsilon$ for similiar reason,

$$\Pr[|y_j - \bar{y}| > \epsilon/2] \leq 2\exp\left(-\frac{-\epsilon^2 k^2}{2|Y|}\right) \leq \exp(-n^{0.1}),$$

$$\Pr[|x_r - \ell'\bar{y}| > \epsilon] \leq 2\exp\left(-\frac{-2\epsilon^2 k^2}{|Y| + |X_r|}\right) \leq \exp(-n^{0.1}),$$

For $s \leq i \leq r-1$, $h((1+\delta)x_{i+1} - x_i) = 0$ because $(1+\delta)x_{i+1} - x_i \leq \epsilon$, $g(y_1, \ldots, y_{\ell'}) = \max\{1-\epsilon, 1 - \exp\left(\sum_{i=1}^{\ell'} y_i\}\right)$ because $|y_i - \bar{y}| \leq \epsilon/2$ for any $1 \leq i \leq \ell'$. If $g(y_1, \ldots, y_{\ell'}) = \epsilon$, then $f(x_1, \ldots, x_r, y_1, \ldots, y'_\ell) \geq 1 - \epsilon$. Therefore, by construction the answer will be the form 10, which only depends on Q and $X_1, \ldots, X_s$.

Let us bound the probability of $\mathcal{E}_r \cap \overline{\mathcal{E}_1} \cap \overline{\mathcal{E}_2} \cap \ldots \cap \overline{\mathcal{E}_{s-1}}$. If we condition on $X_1, \ldots, X_{s-1}$, assuming that none of $\mathcal{E}_1, \ldots, \mathcal{E}_{s-1}$ occurred, the query set $\boldsymbol{Q}_s$ in round s is fixed. By a union bound over $\boldsymbol{Q}_s$, the probability that any of them violates (10) is $e^{-\Omega(-n^{-0.1})}$. Hence,

$$\Pr[\mathcal{E}_s \setminus (\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots \cup \mathcal{E}_{s-1}) \mid X_1, X_2, \ldots, X_{s-1}] = \mathrm{poly}(n)\ e^{-\Omega(n^{-0.1})} = e^{-\Omega(n^{-0.1})}.$$

Now we can average over the choices of $X_1, \ldots, X_{s-1}$ and still obtain

$$\Pr[\mathcal{E}_s \setminus (\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots \cup \mathcal{E}_{s-1})] = e^{-\Omega(n^{-0.1})}.$$

Therefore, by induction,

$$\Pr\left[\bigcup_{i=1}^{s} \mathcal{E}_i\right] = \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2 \setminus \mathcal{E}_1] + \ldots + \Pr[\mathcal{E}_s \setminus (\mathcal{E}_1 \cup \ldots \cup \mathcal{E}_{s-1})] = se^{-\Omega(n^{-0.1})} = e^{-\Omega(n^{-0.1})}.$$

This implies that with high probability, the computation path in round s is determined by $X_1, \ldots, X_{s-1}$.

Consequently, a solution returned after $s-1$ rounds is determined by $X_1, \ldots, X_{s-1}$ with high probability. By the same Chernoff-Hoeffding bounds, the solution, with high probability, satisfies the properties that $(1+\delta)x_{i+1} - x_i \leq \epsilon$ for any $s \leq i \leq r-1$, $|y_i - \bar{y}| \leq \epsilon/2$ for any $1 \leq i \leq \ell'$ and $|x_r - \ell'\bar{y}| > \epsilon$. ◻

Lemma 3.5. *The minimum of the optimization problem has value $\Omega(\frac{\delta^2}{r})$:*

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) := (1+\delta)^2 x_1^2 + \sum_{i=2}^{r} \left((1+\delta)x_i - x_{i-1}\right)^2 : \\ &\sum_{i=1}^{r} x_i \geq \frac{1}{3},\ x_i \geq 0. \end{aligned} \tag{12}$$

Proof. By the convexity of x^2,

$$
\begin{aligned}
((1+\delta)x_1)^2 + \sum_{i=2}^{r}((1+\delta)x_i - x_{i-1})^2 &\geq \frac{1}{r}\left((1+\delta)x_1 + \sum_{i=2}^{r}((1+\delta)x_i - x_{i-1})\right)^2 \\
&= \frac{1}{r}\left(\delta\sum_{i=1}^{r} x_i + (1-\delta)x_r\right)^2 \\
&\geq \frac{1}{r}\left(\frac{\delta}{3}\right)^2 = \Omega\left(\frac{\delta^2}{r}\right)
\end{aligned}
$$

□

Theorem 3.6. *Any r-round adaptive algorithm for monotone submodular optimization can achieve at most a $1 - 1/e - \Omega(\frac{\log^2 n}{r^3})$ approximation, for $r < n^c$ where n is the number of elements and $c > 0$ is some absolute constant.*

Proof. Since the function we construction $\max\{1-\epsilon, f((x_1, \ldots, x_r, y_1, \ldots, y_{\ell'}))\}$ is non-decreasing, without loss of generality we can assume $\sum_{i=1^r} x_i + \sum_{i=1}^{\ell'} y_i = 1$ for the solution. By Lemma 3.4, we know that, with high probability, the answer output by an $r-1$-round algorithm will be in the form $(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'})$ such that $|x_r - \ell'\bar{y}| > \epsilon$ for any $1 \leq j \leq \ell'$. Thus we have

$$
\sum_{i=1}^{r} x_i \geq \frac{1}{2}\left(\sum_{i=1}^{r} x_i + \sum_{i=1}^{\ell'} y_i - \ell'\epsilon\right) = \frac{1}{2}(1-\epsilon) \geq 1/3.
$$

Since for every i, $x_i \leq 1 < \frac{2+\epsilon}{1+\delta}$, we can bound each $h(x)$ by $\alpha x^2 - \alpha\epsilon^2$. As a result,

$$
\begin{aligned}
&p(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'}) \\
&= \sum_{i=1}^{r} x_i - h((1+\delta)x_1) - \sum_{i=1}^{\ell'-1} h((1+\delta)x_{i+1} - x_i) \\
&\leq \sum_{i=1}^{r} x_i - \alpha((1+\delta)x_1)^2 - \sum_{i=1}^{\ell'-1} \alpha((1+\delta)x_{i+1} - x_i)^2 + \alpha r\epsilon^2 \\
&\leq \sum_{i=1}^{r} x_i - \Omega\left(\frac{\delta^2}{r}\right) + \alpha r\epsilon^2 = \sum_{i=1}^{r} x_i - \Omega\left(\frac{\delta^2}{r}\right)
\end{aligned}
$$

where we use Lemma 3.5 in the last inequality. And we can further bound $f(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'})$,

$$
\begin{aligned}
f(x_1, \ldots, x_r, y_1, \ldots, y_{\ell'}) &= 1 - \exp(-p(x_1, \ldots, x_r)) \cdot (1 - g(y_1, \ldots, y_{\ell'})) \\
&= 1 - \exp(-p(x_1, \ldots, x_r)\} \cdot \exp\left(-\sum_{i=1}^{\ell'} y_i)\right) \\
&\leq 1 - \exp\left(-\sum_{i=1}^{r} x_i - \sum_{i=1}^{\ell'} y_i + \Omega\left(\frac{\delta^2}{r}\right)\right) \\
&= 1 - e^{-1} - \Omega\left(\frac{\delta^2}{r}\right)
\end{aligned}
$$

where $\delta = O(\log n/r)$. By Lemma B.1, we know that $OPT \geq f(0, \ldots, 0, 1) = 1 - \epsilon$. Since $\epsilon = o\left(\frac{\log^2 n}{r^3}\right)$, no algorithm can achieve approximation ratio better than $1 - 1/e - \Omega\left(\frac{\log^2 n}{r^3}\right)$ in $r - 1$ rounds with constant probability.

□

REFERENCES

[1] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: massive data summarization on the fly. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 671–680, 2014. doi:10.1145/2623330.2623637.

[2] Eric Balkanski, Adam Breuer, and Yaron Singer. Non-monotone submodular maximization in exponentially fewer iterations. *CoRR*, abs/1807.11462, 2018. URL: http://arxiv.org/abs/1807.11462, arXiv:1807.11462.

[3] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. *CoRR*, abs/1804.06355, 2018. URL: http://arxiv.org/abs/1804.06355, arXiv:1804.06355.

[4] 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, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 66–77, 2019. doi:10.1145/3313276.3316304.

[5] 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, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1138–1151, 2018. doi:10.1145/3188745.3188752.

[6] Eric Balkanski and Yaron Singer. Approximation guarantees for adaptive sampling. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 393–402, 2018. URL: http://proceedings.mlr.press/v80/balkanski18a.html.

[7] Adam Breuer, Eric Balkanski, and Yaron Singer. The FAST algorithm for submodular maximization. *CoRR*, abs/1907.06173, 2019. URL: http://arxiv.org/abs/1907.06173, arXiv:1907.06173.

[8] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM J. Comput.*, 44(5):1384–1402, 2015. doi:10.1137/130929205.

[9] 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, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 78–89, 2019. doi:10.1145/3313276.3316406.

[10] 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, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 303–322, 2019. doi:10.1137/1.9781611975482.20.

[11] 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, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 102–113, 2019. doi:10.1145/3313276.3316327.

[12] Vasek Chvatal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4(3):233–235, August 1979.

[13] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1236–1244, 2015. URL: http://proceedings.mlr.press/v37/barbosa15.html.

[14] Rafael da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. A new framework for distributed submodular maximization. In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science*, 2016.

[15] Alina Ene and Huy L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 274–282, 2019. doi:10.1137/1.9781611975482.18.

[16] Alina Ene, Huy L. Nguyen, and Adrian Vladu. A parallel double greedy algorithm for submodular maximization. *CoRR*, abs/1812.01591, 2018. URL: http://arxiv.org/abs/1812.01591, arXiv:1812.01591.

[17] Alina Ene, Huy L. Nguyen, 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, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 90–101, 2019. doi:10.1145/3313276.3316389.

[18] Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1833–1842, 2019. URL: http://proceedings.mlr.press/v97/fahrbach19a.html.

[19] Matthew Fahrbach, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 255–273, 2019. doi:10.1137/1.9781611975482.17.

[20] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

[21] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011. doi:10.1137/090779346.

[22] Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

[23] Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for MapReduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 938–948, 2010. doi:10.1137/1.9781611973075.76.

[24] Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3311–3320, 2019. URL: http://proceedings.mlr.press/v97/kazemi19a.html.

[25] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, pages 71–104, 2014.

[26] Fabian Kuhn and Roger Wattenhofer. On the complexity of distributed graph coloring. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 7–15, 2006. URL: http://doi.acm.org/10.1145/1146381.1146387, doi:10.1145/1146381.1146387.

[27] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in MapReduce and streaming. *TOPC*, 2(3):14:1–14:22, 2015. doi:10.1145/2809814.

[28] Paul Liu and Jan Vondrák. Submodular optimization in the mapreduce model. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 18:1–18:10, 2019. doi:10.4230/OASIcs.SOSA.2019.18.

[29] Andrew McGregor and Hoa T. Vu. Better streaming algorithms for the maximum coverage problem. In *20th International Conference on Database Theory, ICDT 2017, March 21-24, 2017, Venice, Italy*, pages 22:1–22:18, 2017. doi:10.4230/LIPIcs.ICDT.2017.22.

[30] Vahab S. Mirrokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *ACM Symposium on Theory of Computing (STOC)*, pages 153–162, 2015.

[31] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2049–2057, 2013. URL: http://papers.nips.cc/paper/5039-distributed-submodular-maximization-identifying-representative-elements-in-massive-data.

[32] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978. doi:10.1007/BF01588971.

[33] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997. doi:10.1137/S0097539793250767.

[34] Jan Vondrák. Symmetry and approximability of submodular maximization problems. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 651–670, 2009. doi:10.1109/FOCS.2009.24.

[35] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013. doi:10.1137/110832318.

[36] Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.