The One-Way Communication Complexity of Submodular Maximization with Applications to Streaming and Robustness

Moran Feldman
moranf@cs.haifa.ac.il
University of Haifa, Israel

Ashkan Norouzi-Fard
ashkannorouzi@google.com
Google Research, Swizterland

Ola Svensson
ola.svensson@epfl.ch
EPFL, Swizterland

Rico Zenklusen
ricoz@math.ethz.ch
ETH Zurich, Swizterland

ABSTRACT
We consider the classical problem of maximizing a monotone submodular function subject to a cardinality constraint, which, due to its numerous applications, has recently been studied in various computational models. We consider a clean multi-player model that lies between the offline and streaming model, and study it under the aspect of one-way communication complexity. Our model captures the streaming setting (by considering a large number of players), and, in addition, two player approximation results for it translate into the robust setting. We present tight one-way communication complexity results for our model, which, due to the above-mentioned connections, have multiple implications in the data stream and robust setting.

Even for just two players, a prior information-theoretic hardness result implies that no approximation factor above 1/2 can be achieved in our model, if only queries to feasible sets, i.e., sets respecting the cardinality constraint, are allowed. We show that the possibility of querying infeasible sets can actually be exploited to beat this bound, by presenting a tight 2/3-approximation taking exponential time, and an efficient 0.514-approximation. To the best of our knowledge, this is the first example where querying a submodular function on infeasible sets leads to provably better results. Through the above-mentioned link to the robust setting, both of these algorithms improve on the current state-of-the-art for robust submodular maximization, showing that approximation factors beyond 1/2 are possible. Moreover, exploiting the link of our model to streaming, we settle the approximability for streaming algorithms by presenting a tight 1/2 + \( \epsilon \) hardness result, based on the construction of a new family of coverage functions. This improves on a prior 1 – 1/\( e + \epsilon \) hardness and matches, up to an arbitrarily small margin, the best known approximation algorithm.

CCS CONCEPTS
- Mathematics of computing → Combinatorial optimization.

1 INTRODUCTION
A set function \( f: 2^W \rightarrow \mathbb{R} \) over a finite ground set \( W \) is submodular if
\[
\forall v, X \subseteq W \quad f(v | X) \geq f(v | Y) \quad \text{for all } X \subseteq Y \subseteq W \text{ and } v \in W \setminus Y,
\]
where, for a subset \( S \subseteq W \) and an element \( v \in W \), the value \( f(v | S) = f(S \cup \{v\}) - f(S) \) is the marginal contribution of \( v \) with respect to \( S \). The definition of submodular functions captures the natural property of diminishing returns, and submodular functions have a rich history in optimization with numerous applications (see, e.g., Schrijver’s book [Sch03]).

Already in 1978, Nemhauser, Wolsey, and Fisher [NWF78] analyzed the following algorithm, which we refer to as Greedy, for selecting the most valuable set \( S \subseteq W \) of cardinality at most \( k \).

(i) Initially, let \( S = \emptyset \).
(ii) For \( i = 1, \ldots, k \), choose any \( v \in \arg \max_{w \in W} f(w | S) \) and set \( S = S \cup \{v\} \).

In words, the algorithm greedily picks in each iteration an element with the largest marginal contribution with respect to the already selected elements \( S \). Assuming that \( f \) is non-negative \( (f(X) \geq 0 \text{ for all } X \subseteq W) \) and monotone \( (f(X) \leq f(Y) \text{ if } X \subseteq Y) \), Nemhauser et al. [NWF78] showed that Greedy returns a \( (1 - 1/e) \)-approximate solution. Moreover, the approximation guarantee of \( 1 – 1/e \) is known to be tight [NW78, Fei98].

In recent years, submodular function maximization has found several applications in problems related to data science and machine learning, including feature selection, sensor placement, and image collection summarization [Kra, GK11, Bac10, DDK12, DK11, ZJCP14, BBR15]. These applications are often modeled as a maximization of a non-negative and monotone submodular function. For example, if we wish to summarize an image collection, we would like to select \( k \) images that cover different topics, and this objective can be modeled as a (non-negative and
monotone) submodular function. While Greedy gives the best possible guarantee for solving this problem in traditional computing (when the entire instance is accessible to the algorithm at all times), the requirements stipulated by modern applications, often involving huge data sets, make such algorithms inadequate.

This motivates, together with the inherent theoretical interest, the study of submodular function maximization in new models of computation. Indeed, in recent years, there has been substantial interest in submodular function maximization with respect to limited memory (so-called data stream algorithms) [BMKK14, BFS15, FKK18, NTM*18, KMZ*19, AEF*20], robustness [OSU16, BSMC17, MKK17, MBNP*17, KZK18], parallel computation (in the map-reduce model) [dPBENW15, MZ15, BENW16], and most recently with respect to adaptivity [BS18, BRS19a, BRS19b, CFK19, CQ19, EN19, ENV19, FMZ19a, FMZ19b]. In each of these models, the central benchmark problem has been the basic cardinality-constrained problem studied in [NWF78], namely that of finding a set \( S \subseteq W \) of cardinality \( k \) that maximizes \( f(S) \), where \( f \) is a non-negative and monotone submodular function. We refer to this problem as Max-Card-\( k \).

While tight algorithms are known for Max-Card-\( k \) (and even for the more general problem where the cardinality constraint is replaced by a matroid) in the map-reduce model [BENW16, LV19] and the adaptive model [BRS19a, EN19, FMZ19b], it has remained an open problem to give tight results for the data stream and robust settings. In this paper, we resolve this question for data stream algorithms and make progress on the robust problem. These results are obtained by considering the one-way communication complexity of Max-Card-\( k \), the study of which highlights several interesting aspects of submodular functions. We first discuss our results in the clean communication model, and then give more detail about the connection to data stream and robust algorithms.

### 1.1 One-way Communication Complexity of Max-Card-\( k \)

We first study the one-way communication complexity of Max-Card-\( k \) in the presence of two players. An informal description of the model is as follows (see Section 2 for the formal definition). The first player Alice has only access to a subset \( V_A \subseteq W \) of the ground set and the second player Bob has access to \( V_B \subseteq W \) with \( V_A \cap V_B = \emptyset \). In the first phase, Alice can query the value of a submodular objective function \( f \) on any subset of her elements; and then, at the end of the phase, she can send an arbitrary message to Bob based on the information that she has. Then, in the second phase, Bob gets the message of Alice and the elements of \( V_B \). He can then query \( f \) on any subset of the elements, and his objective is to produce a subset of \( V_A \cup V_B \) of size at most \( k \) that approximately maximizes \( f \) among all such subsets.

A trivial protocol that allows Bob to always output the optimal solution is for Alice to send all the elements in \( V_A \); and for Bob to then output \( \arg \max_{S \subseteq V_A \cup V_B} |S| \leq k f(S) \). While this protocol has an optimal approximation guarantee of 1, it has a very large communication complexity since it requires Alice to send all the elements of \( V_A \), which may be as many as \( N = |W| \) elements.

A protocol of lower communication complexity is for Alice to calculate her “optimal” solution \( S_A = \arg \max_{S \subseteq V_A} |S| \leq k f(S) \) and send only those (at most \( k \) many) elements in \( S_A \) to Bob. Bob then outputs either his “optimal” solution \( S_B = \arg \max_{S \subseteq V_B} |S| \leq k f(S) \) or \( S_A \), whichever set attains the larger value. It is not hard to see that this protocol has an approximation guarantee of \( 1/2 \), i.e., that \( \max(f(S_A), f(S_B)) \geq 1/2 \cdot \max_{S \subseteq V_A \cup V_B} |S| \leq k f(S) \) for any \( V_A, V_B \subseteq W \).

The above examples indicate a natural trade-off between the amount of communication and the approximation guarantee, with the central question being to understand the optimal relationship between these two quantities. If one further restricts Alice to only query the submodular function on sets of cardinality at most \( k \), then the hardness result in [NTM*18] for streaming algorithms implies that, in any (potentially randomized) protocol with an approximation guarantee of \( (1/2 + \epsilon) \), Alice must send a message of length \( \Omega(\epsilon N/k) \). In other words, under this restriction on Alice, the two basic protocols described above achieve the optimal trade-off up to lower order terms (in this, as in previous work, we think of \( k \ll N \)).

Restricting Alice to evaluate \( f \) only on sets of cardinality at most \( k \) may appear like a mere technical assumption to make the arguments in [NTM*18] work; especially since, to the best of our knowledge, there are no known examples where querying the submodular function on infeasible sets leads to provably better results. Perhaps surprisingly, we prove this intuition wrong and give a protocol that crucially exploits the possibility to query the value of infeasible sets.

**Theorem 1.1.** There exists a two-player protocol for Max-Card-\( k \) with an approximation guarantee of \( 2/3 \) in which Alice sends a message consisting of \( O(k^2) \) elements.

We present this protocol in Section 4.1, where we also show that we can further reduce the message size of Alice down to \( O(k \log(k)/\epsilon) \) elements while still obtaining an approximation guarantee of \( 2/3 - \epsilon \). By allowing Alice to query \( f \) on sets of cardinality larger than \( k \), we can thus improve the approximation guarantee of \( 1/2 \) to \( 2/3 \) while still maintaining an (almost) linear-sized message in \( k \). In Section 4.2, we further show that the guarantee of \( 2/3 \) is tight in the following strong sense. In any protocol that achieves a better guarantee, Alice must send a message of roughly the same size as the trivial protocol mentioned above that achieves an approximation guarantee of \( 1 \).

**Theorem 1.2.** In any (potentially randomized) two-player protocol for Max-Card-\( k \) that has an approximation guarantee of \( 2/3 + \epsilon \) for \( \epsilon > 0 \), Alice sends a message of length at least \( \Omega(\epsilon N/k) \).

The fact that we can beat the approximation guarantee of \( 1/2 \) using little communication in the presence of two players, gives hope that a similar result may hold for many players, and more generally in the streaming model, which can be thought of as having one player per element. The definition of the \( p \)-player setting, with \( p \geq 2 \), is the natural generalization of the two-players setting. Informally (again, see Section 2 for the formal definition), the \( i \)-th player receives a private subset \( V_i \subseteq W \) of the ground set and, upon reception of a message from the previous player, she computes and sends a message to the following player. Finally, the last player’s
task is to output a subset of \( V_1 \cup V_2 \cup \cdots \cup V_p \) of cardinality at most \( k \) that approximately maximizes \( f \) among all such subsets.

One can observe that the streaming algorithm of [KMZ+19] yields, for any integer \( p \geq 2 \), a \( p \)-player protocol that has an approximation guarantee of \( (1/2 - \varepsilon) \) and where each player sends a message consisting of at most \( O(k/\varepsilon) \) elements. Moreover, the obtained protocol only queries \( f \) on sets of size at most \( k \) and is, thus, tight with respect to such protocols (even in the two-player setting). Similar to the two-player case, the naturally arising question is whether protocols querying \( f \) on sets of cardinality larger than \( k \) can improve over this approximation guarantee. Our most technical result shows that this is not the case as \( p \) and \( k \) tends to infinity.

**Theorem 1.3.** For every \( \varepsilon > 0 \), there is an integer \( p_0 \geq 2 \) such that the following holds for any (potentially randomized) \( p \)-player protocol for Max-Card-\( k \) with \( k = p \geq p_0 \). If the protocol has an approximation guarantee of \( 1/2 + \varepsilon \), then one of the players sends a message of length at least \( \Omega \left( \varepsilon N/p^3 \right) \).

The proof of the above theorem is given in Section 5. It is based on a new construction of a family of coverage functions that hides the optimal solution while guaranteeing that no solution that does not contain elements from the optimal solution can provide an approximation significantly better than \( 1/2 \). This result immediately implies a tight hardness result in the streaming model that we explain in the next section.

### 1.2 Applications to Data Stream and Robustness

One key reason for the success of communication complexity is that results for the models it motivates, which are on their own right interesting models capturing the essence of trade-offs involving message sizes, are often widely applicable to other models of computation. This is also the case for submodular functions. As we show below, our results yield both new hardness and algorithmic results in the context of data streams and robustness.

*Data stream algorithms.* We first discuss the well-known and direct connection to data stream algorithms. In the data stream model, the elements of the (unknown) ground set arrive one element at a time, rather than being available all at once, and the algorithm is restricted to only use a small amount of memory. A semi-streaming algorithm is an algorithm for this model whose memory size has only a nearly-linear dependence on the parameter \( k \) (the output size) and at most a logarithmic dependence on the size of the ground set. The goal is to output, at the end of the stream, a subset of the elements in the stream of cardinality at most \( k \) that approximately maximizes \( f \) among all such subsets.

The first result in this setting was given by Chakrabarti and Kale [CK14], who described a semi-streaming algorithm for Max-Card-\( k \) with an approximation guarantee of \( 1/4 \). Badanidiyuru et al. [BMKK14] proposed later a different semi-streaming algorithm which provides a better approximation ratio of \( (1/2 - \varepsilon) \) and maintains at most \( O \left( k \log(k)/\varepsilon \right) \) elements in memory. This memory footprint was recently improved by [KMZ+19], who obtained the same approximation guarantee while only maintaining at most \( O \left( k/\varepsilon \right) \) elements in memory.

The two last algorithms share the approximation guarantee of \( 1/2 - \varepsilon \). This was improved to \( 1 - 1/e - \varepsilon \) by Agrawal et al. [ASS18], but only under the assumption that the elements of the stream arrive in a uniformly random order. In contrast, Huang et al. [HKMY20] showed that without this assumption one cannot obtain an approximation ratio better than \( 2 - \sqrt{2} \approx 0.586 \) (improving over a previous inapproximability result of \( 1 - 1/e \) due to [MV19]). Hence, prior to the current work, it remained an open question whether the approximation guarantee of \( 1/2 - \varepsilon \) obtained by the state-of-the-art algorithms is optimal. However, a direct consequence of Theorem 1.3 settles this question. Specifically, any algorithm that achieves a better approximation guarantee than \( 1/2 \) must (up to a polynomial factor in \( k \)) essentially store all the elements of the stream.

**Theorem 1.4.** For any \( \varepsilon > 0 \), a data stream algorithm for Max-Card-\( k \) with an approximation guarantee of \( 1/2 + \varepsilon \) must use memory \( \Omega \left( \varepsilon s/k^3 \right) \), where \( s \) denotes the number of elements in the stream.

The proof of the above theorem is almost immediate given Theorem 1.3 and the well-known connection between data stream algorithms and one-way communication. Thus, it is deferred to the full version of this work [FNSZ20].

*Robust submodular function maximization.* The work on algorithms for Max-Card-\( k \) has been partially motivated by the desire to extract small summaries of huge data sets. In many settings, the extracted summary is also required to be robust. That is, the quality of the summary should degrade by as little as possible when some elements of the ground set are removed. Such removals may arise for many reasons, such as failures of nodes in a network, or user preferences which the model failed to account for; they could even be adversarial in nature. Recently, this topic has attracted special attention due to its importance in privacy and fairness constraints. The robust summaries enable us to remove sensitive data without incurring much loss in performance, giving us the ability to protect personal information (the right to be forgotten) and avoid biases (e.g., gender, measurement, and design biases).

The first attempts to design algorithms that generate robust summaries assumed that the summary is simply a set of size \( k \), and the algorithm should guarantee that the value of this set is competitive against the best possible such set even when some elements are deleted (from both the ground set and the solution set). Naturally, this objective makes sense only when the number \( d \) of deleted elements is significantly smaller than \( k \). Accordingly, [OSU16] provided the first constant \((0.387)\) factor approximation result to this problem for \( d = o(\sqrt{k}) \), and Buzaglo et al. [BMSC17] improved the restriction on number of deletions to \( d = o(k) \) while keeping the approximation guarantee unchanged.

More recent works studied a more general variant of the above problem where an algorithm consists of two procedures: a summary procedure and a query procedure. The summary procedure first generates a summary \( M \subseteq W \) of the ground set \( W \) with few, but

---

1We recall that the hardness result of [NTM+18] only applies to the restricted case when the value of the submodular function is queried on sets of cardinality at most \( k \).
We do this via the following theorem, which shows that one can robust submodular maximization.

This result improved over previous results by Mirzasoleiman et al. [MKK17] and Mitrovic et al. [MBNF17] (iii) if \( \alpha \) is at least as good as the approximation guarantee of \( \mathcal{A} \) for robust submodular maximization such that (i) the approximation guarantee of \( \mathcal{A} \) is at least as good as the approximation guarantee of \( \mathcal{P} \); (ii) the number of elements in the summary of \( \mathcal{A} \) is larger than the communication complexity of \( \mathcal{P} \) (in elements) only by an \( O(d) \) factor; (iii) if \( \mathcal{P} \) runs in polynomial time, then so is \( \mathcal{A} \).

As all the protocols we use to prove our results in this paper obey the natural properties required by Theorem 1.5, one can combine this theorem with Theorem 1.1 to get the following corollary.

Corollary 1.6. There exists an algorithm for robust submodular maximization returning a 2/3-approximate solution and using summaries of \( O(dk^2) \) elements.

An instance of our model consists both of global information known upfront to both Alice and Bob, and private information that is available only to either Alice or Bob. The global information includes the upper bound \( k \) on the size of the solution (which is a positive integer), a ground set \( W \) of elements and a partition of \( W \) into two disjoint sets \( W_A \) and \( W_B \). One should think of the sets \( W_A \) and \( W_B \) as all elements that Alice and Bob, respectively, could potentially get. We denote by \( V_A \subseteq W_A \) the set of elements that Alice actually gets, and by \( V_B \subseteq W_B \) the set of elements that Bob actually gets. Both these sets are private information available only to their respective players. Finally, the instance also includes a non-negative monotone submodular function \( f : 2^W \rightarrow \mathbb{R}_{\geq 0} \) defined over all the subsets of \( W \). Alice has access to this function through an oracle that can evaluate \( f \) on any set \( S \subseteq W_A \) (in other words, given such a set \( S \), the oracle returns \( f(S) \)). Bob, in contrast, has access to \( f \) through a more powerful oracle that can evaluate \( f \) on any subset of \( W \). Intuitively, the reason for the difference between the powers of the oracles is that Alice only needs to evaluate sets consisting of elements that she might get, while Bob must also be able to evaluate \( f \) on subsets that include elements sent by Alice (nevertheless, one can observe that the oracle of Bob does not leak information about the elements of \( W_A \) that Alice actually got, i.e., the elements that ended up in \( V_A \)). The objective of Alice and Bob

The last theorem is proved in the full version of this work [FNFSZ20]. Moreover, we show in this full version that combining this theorem with a reduction to the robust setting yields the following result for robust submodular maximization.

Corollary 1.8. There exists a polynomial time 0.514-approximation algorithm for robust submodular maximization using summaries of \( O(dk) \) elements.

2 FORMAL STATEMENT OF MODEL AND RESULTS

In this section we formally present the model that we assume in this paper, and restate in a formal way the results that we prove for this model. We begin by discussing the model for the two-player setting. It is natural to formulate a simple model for this setting in which Alice forwards some elements to Bob, and then Bob can access only these elements and the elements he receives directly. All the protocols we present fit into this simple model. However, one could imagine more involved protocols in which Alice passes coded information about the elements she received, rather than simply forwarding a subset of these elements. To make our impossibility results apply also to protocols of this kind, we formulate below a somewhat more involved model in which the message sent from Alice to Bob is an arbitrary string of bits. We note that there is no unique “right” way to cast the problem we consider into a model, and one can think of multiple natural ways to do so, each corresponding to a different intuitive viewpoint. Fortunately, it seems that our results are mostly independent of the particular formulation used (up to minor changes in the exact bounds), and thus, we chose a model that we believe is both intuitive and allows for a nice presentation of the results. Nevertheless, for completeness, we present a sketch of an alternative model that we also found attractive in the full version of this work [FNFSZ20].

An instance of our model consists both of global information known upfront to both Alice and Bob, and private information that is available only to either Alice or Bob. The global information includes the upper bound \( k \) on the size of the solution (which is a positive integer), a ground set \( W \) of elements and a partition of \( W \) into two disjoint sets \( W_A \) and \( W_B \). One should think of the sets \( W_A \) and \( W_B \) as all elements that Alice and Bob, respectively, could potentially get. We denote by \( V_A \subseteq W_A \) the set of elements that Alice actually gets, and by \( V_B \subseteq W_B \) the set of elements that Bob actually gets. Both these sets are private information available only to their respective players. Finally, the instance also includes a non-negative monotone submodular function \( f : 2^W \rightarrow \mathbb{R}_{\geq 0} \) defined over all the subsets of \( W \). Alice has access to this function through an oracle that can evaluate \( f \) on any set \( S \subseteq W_A \) (in other words, given such a set \( S \), the oracle returns \( f(S) \)). Bob, in contrast, has access to \( f \) through a more powerful oracle that can evaluate \( f \) on any subset of \( W \). Intuitively, the reason for the difference between the powers of the oracles is that Alice only needs to evaluate sets consisting of elements that she might get, while Bob must also be able to evaluate \( f \) on subsets that include elements sent by Alice (nevertheless, one can observe that the oracle of Bob does not leak information about the elements of \( W_A \) that Alice actually got, i.e., the elements that ended up in \( V_A \)). The objective of Alice and Bob
is to find a set $S \subseteq V_A \cup V_B$ maximizing $f$ among all such sets of size at most $k$.

A communication protocol $P = (\mathcal{A}_A, \mathcal{A}_B)$ for this model consists of two (possibly randomized) algorithms for Alice and Bob. The protocol proceeds in two phases. In the first phase, the algorithm $\mathcal{A}_A$ of Alice computes a message $m$ for Bob based on the global information and the private information available to Alice. Then, in the second phase, the algorithm $\mathcal{A}_B$ of Bob computes an output set based on (i) the global information, (ii) the private information available to Bob, and (iii) the message $m$ received from Alice. Formally, the communication complexity of protocol $P$ is the maximum length in bits of the message $m$, where the maximum is taken over all the possible inputs and the randomness of the algorithms. However, since the message $m$ in our protocols consists mostly of elements that Alice sends to Bob, we state the communication complexity of these protocols, for simplicity, in elements instead of bits. The real communication complexity of these protocols in bits is larger than the stated bound in elements, and furthermore, both algorithms in our protocols in elements instead of bits. The real communication complexity of these protocols, for simplicity, in elements instead of bits. The real communication complexity of these protocols, for simplicity, in elements instead of bits. The real communication complexity of these protocols, for simplicity, in elements instead of bits. The real communication complexity of these protocols, for simplicity, in elements instead of bits.

We can now restate our results for the two-player model in a more formal way. Note that the first of these theorems uses the $\tilde{O}$ notation, which suppresses poly-logarithmic terms, and the second of these theorems refers by $N$ to the size of the ground set $W$.

**Theorem 1.1.** For every $\epsilon > 0$, there exists a two-player protocol for Max-Card-$k$ with an approximation guarantee of $(2/3 - \epsilon)$ whose communication complexity is $\tilde{O}(k/\epsilon)$ elements. Moreover, there exists such a protocol achieving an approximation guarantee of $2/3$ whose communication complexity is $O(k^2)$ elements.

**Theorem 1.2.** For every $\epsilon \in (0, 1/4)$, any two-player (randomized) protocol with an approximation guarantee of $(2/3 + \epsilon)$ must have a communication complexity of $\Omega(N/n)$ bits in the regime $k \geq \epsilon^{-1}$.

**Theorem 1.7.** There exists a two-player protocol for Max-Card-$k$ with an approximation guarantee of $0.514$ whose communication complexity is $O(k)$ elements, and furthermore, both algorithms in this protocol run in polynomial time.

Let us now explain how the above model can be generalized to the $p$-player setting for $p \geq 2$. In this setting, the ground set $W$ is partitioned into $p$ disjoint sets $W_1, W_2, \ldots, W_p$, rather than just two; and the global information available to all the players is (i) the upper bound $k$ on the size of the solutions, (ii) the ground set $W$, and (iii) the partition of this ground set. Every player also has private information. In particular, the private information available to player $i$ is $[p]$ (recall that $[p]$ is a shorthand for the set $\{1, 2, \ldots, p\}$) is a subset $V_i \subseteq W_i$ and an oracle that can evaluate the objective function $f$ on every subset of $\bigcup_{j=1}^p W_j$. The objective of the players is to find a set $S \subseteq \bigcup_{i=1}^p V_i$ maximizing $f$ among all such sets of size at most $k$.

A communication protocol $P = (A_1, A_2, \ldots, A_p)$ for this $p$-player model consists of $p$ (possibly randomized) algorithms for the $p$ players. The protocol proceeds in $p$ phases. In the first phase the algorithm $A_1$ of the first player computes a message $m_1$ based on the global information and the private information available to this player. The next $p - 2$ phases are devoted to players $2$ up to $p - 1$. In particular, in phase $i \in \{2, 3, \ldots, p - 1\}$, the algorithm $A_i$ of player $i$ computes a message $m_i$ based on the global information, the private information available to this player, and the message $m_{i-1}$ produced by the previous player. Finally, in the last phase, the algorithm $A_p$ of the last player computes an output set based on the global information, the private information available to this player, and the message $m_{p-1}$ produced by the penultimate player.

The communication complexity of the protocol $P$ is the maximum length in bits of any one of the messages $m_1, m_2, \ldots, m_{p-1}$, where, like in the two-player model, the maximum is taken over all the possible inputs and the randomness of the algorithms. We can now restate our result for the $p$-player model in a more formal way. Recall that $N = |W|$, and for $p \in \mathbb{N}$, let $H_p = 1 + \frac{1}{3} + \frac{1}{3} + \ldots + \frac{1}{p}$ be the $p$-th harmonic number.

**Theorem 1.3.** For every $\epsilon > 0$, any $p$-player (randomized) protocol for Max-Card-$k$ with an approximation guarantee of

$$p + (H_p)^2 - 2p - H_p \cdot (1 + \epsilon)$$

must have a communication complexity of $\Omega\left(\frac{N\epsilon}{p^2}\right)$. Furthermore, this is true even in the special case in which the objective function $f$ is a coverage function and $k = p$.

### 3 PRELIMINARIES: THE INDEX AND CHAIN$_p$ PROBLEM

The impossibility results that we prove in this paper are based on reductions from problems which are known to require high communication complexity. The first of these problems is the well-known INDEX problem. In this two-player problem, Alice gets a string $x \in \{0, 1\}^n$ of $n$ bits, and can then send a message to Bob. Bob gets the message of Alice and an index $i \in [n]$, and based on these two pieces of information alone should output the value of $x_i$. Clearly, Bob can produce the correct answer with probability $1/2$ by outputting a random bit. However, it is known that Bob cannot guarantee any larger constant probability of success, unless the message he gets from Alice is of linear (in $n$) size (see, e.g., [BYJKS02, JK90]).

The second problem we reduce from is CHAIN$_p(n)$, a multi-player generalization of INDEX recently introduced by Cormode et al. [GC19], which is closely related to the Pointer Jumping problem (see [Cha07]). In CHAIN$_p(n)$, the index $p$ indicates the number of players and $n$ is a parameter that regulates the size of the bit string given to each player. The definition is as follows. There are $p$ players $P_1, P_2, \ldots, P_p$. For every $i \in [p - 1]$, player $P_i$ has as input a bit string $x^i \in \{0, 1\}^n$ of length $n$, and, for every $i \in \{2, 3, \ldots, p\}$, player $P_i$ (also) has as input an index $i^f \in \{1, 2, \ldots, n\}$ (note that the convention in this terminology is that the superscript of a string/index indicates the player receiving it). Furthermore, it is promised that either $x_{i^f}^i = 0$ for all $i \in [p - 1]$ or $x_{i^f}^i = 1$ for all $i$ values. We refer to these cases as the 0-case and 1-case, respectively. The objective of the players in CHAIN$_p(n)$ is to decide whether the input instance belongs to the 0-case or the 1-case.

In CHAIN$_p(n)$, we are interested in the communication complexity of a one-way protocol that guarantees a success probability of at least $2/3$. Such a protocol $P = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_p)$

2In [GC19], the problem was simply named CHAIN$_p$, keeping the parameter $n$ implicit.
consists of \( p \) (possibly randomized) algorithms corresponding to the \( p \) players. The protocol proceeds in \( p \) phases. In phase \( i \in \{ p - 1 \} \), the algorithm \( A_i \) of player \( i \) computes a message \( m_i \) based on the input of this player and the message \( m_{i-1} \) computed by \( A_{i-1} \) in the previous phase (unless \( i = 1 \), in which case the computation done by \( A_1 \) depends only on the input of player 1).

In the last phase, algorithm \( A_p \) of player \( p \) decides between the 0-case and the 1-case based on the input of player \( p \) and the message \( m_{p-1} \). The communication complexity of the protocol is defined as the maximum size (in bits) of any one of the messages \( m_1, m_2, \ldots, m_{p-1} \), where the maximum is taken over all the possible inputs and the randomness of the protocol’s algorithms. Furthermore, the success probability of the protocol is the probability that the case indicated by \( A_p \) matches the real case of the input instance.

Note that CHAIN\(_p\)(\( n \)) is indeed a generalization of the INDEX problem since the last problem is equivalent to CHAIN\(_2\)(\( n \)). In [GC19], the following communication complexity lower bound was shown for CHAIN\(_p\)(\( n \)).

**Theorem 3.1 ([GC19]).** Any protocol for CHAIN\(_p\)(\( n \)) with success probability of at least \( 2/3 \) must communicate at least \( \Omega(n/p^2) \) bits in total.

Moreover, the following stronger result, for a restricted range of \( p \), was announced in [GC19] without proof.

**Theorem 3.2 ([GC19]).** There is a constant \( C > 0 \) such that any protocol for CHAIN\(_p\)(\( n \)), where \( p \leq C \cdot \left( \frac{n}{\log n} \right)^{1/4} \), with success probability of at least \( 2/3 \) must communicate at least \( \Omega(n/p) \) bits in total.

We highlight that the above lower bounds are both for the total number of bits communicated and not the maximum message size. Because there are \( p \) messages, this immediately translates to lower bounds on the maximum message size of \( \Omega(n/p^2) \) and \( \Omega(n/p^2) \), respectively. For completeness, we show in the full version of this work [FNSZ20] how proofs of standard results for the INDEX problem can get the following impossibility result for CHAIN\(_p\)(\( n \)), which provides a lower bound of \( \Omega(n/p^2) \) on the maximum message size without restrictions on the range of \( p \).

**Theorem 3.3.** For any positive integers \( n \) and \( p \geq 2 \), any (potentially randomized) protocol for CHAIN\(_p\)(\( n \)) with success probability of at least \( 2/3 \) must have a communication complexity of at least \( n/(36p^2) \).

## 4 TWO PLAYER SUBMODULAR MAXIMIZATION

In this section we consider Max-Card-\( k \) in the two-player model, while ignoring the computational cost, i.e., we are only interested here in the relationship between the communication complexity and the approximation guarantee that can be obtained for this problem. Below, we restate the formal theorems that we prove in the section. The proofs of these theorems can be found in Sections 4.1 and 4.2, respectively. In a nutshell, the two theorems show together that an approximation guarantee of \( 2/3 \) is tight for the problem under a natural assumption on the communication complexity.

**Theorem 1.1.** For every \( \varepsilon > 0 \), there exists a two-player protocol for Max-Card-\( k \) with an approximation guarantee of \( (2/3 - \varepsilon) \) whose communication complexity is \( O(k/\varepsilon^2) \) elements. Moreover, there exists such a protocol achieving an approximation guarantee of \( 2/3 \) whose communication complexity is \( O(k^2) \) elements.

**Theorem 1.2.** For every \( \varepsilon \in (0, 1/4) \), any two-player (randomized) protocol with an approximation guarantee of \( (2/3 + \varepsilon) \) must have a communication complexity of \( \Omega(\frac{k^2}{\varepsilon}) \) bits in the regime \( k \geq \varepsilon^{-1} \).

### 4.1 Algorithms for Two Players

In this section we prove Theorem 1.1. For that purpose, let us present Protocol 1, which is a protocol for Max-Card-\( k \)-in the two-player model that uses exponential computation. In this protocol, Alice finds for every \( i \in \{0, 1, \ldots, 2k\} \) the maximum value subset \( S_i \) of \( V_A \) of size at most \( i \), and forwards all the sets she has found to Bob. Then, Bob finds the best solution over the elements that Alice has sent and \( V_B \).

**Protocol 1 Repeated Solving with Varying Sizes**

**Alice’s Algorithm**

1. For \( i = 0 \) to \( 2k \) do
2. Let \( S_i \) be the set maximizing \( f \) among all subsets of \( V_A \) of size at most \( i \).
3. Send all the sets \( \{S_i\}_{i=0}^{2k} \) as the message to Bob (we note that \( S_0 \) is always the empty set, so sending it is technically redundant. However, it allows us to treat all \( i \) values in the same way in the analysis).

**Bob’s Algorithm**

1. Let \( \tilde{S} \) be the subset of \( V_B \cup \cup_{i=0}^{2k} S_i \) of size at most \( k \) maximizing \( f \).
2. Return \( \tilde{S} \).

It is easy to see that Protocol 1 always outputs a feasible set; and moreover, the number of elements Alice sends to Bob is \( O(k^2) \) because she sends \( 2k + 1 \) sets of size at most \( 2k \) each. Thus, to prove that Protocol 1 obeys all the properties guaranteed by the second part of Theorem 1.1, it remains to show that it produces a \( 2/3 \)-approximation, which is our main objective in the rest of this section.

Let us denote by \( O \) a subset of \( V_A \cup V_B \) of size at most \( k \) maximizing \( f \) among all such subsets, and let \( OPT = f(O) \). Also, let \( M = V_B \cup \cup_{i=0}^{2k} S_i \) be the set of elements that Bob either receives from Alice or receives directly. Note that \( M \) is also the set of elements in which Bob looks for \( S_i \). Using this notation, we can now describe the intuitive idea behind our first observation.

Our analysis of Protocol 1 is based on two sets \( S_{k-|O\cap M|} \) and \( S_{2k-|O\cap M|} \). Observe that one candidate for \( S_{k-|O\cap M|} \) is the part of \( O \) that Alice got and did not forward to Bob. Thus, we know that \( S_{k-|O\cap M|} \) is as valuable as \( O \setminus M \). The following observation formalizes this fact.

**Observation 4.1.** \( f(S_{k-|O\cap M|}) \geq f(O \setminus M) \).

**Proof**. The set \( O \setminus M \) is a subset of \( (V_A \cup V_B) \setminus M \subseteq V_A \) of size \( |O \setminus M| = |O| - |O \cap M| \leq k - |O \cap M| \). Thus, the observation follows from the choice of \( S_{k-|O\cap M|} \) by Protocol 1. □
Despite the fact that $S_{k-|O\cap M|}$ is as valuable as $O \setminus M$, it is not clear to what extent the values of the two sets “overlap” (more formally, how far is the value of their union from the sum of their individual values). If the overlap is large, then this means that $S_{k-|O\cap M|}$ is a good replacement for $O \setminus M$, and thus, Bob can construct a good solution by combining $S_{k-|O\cap M|}$ with $O \setminus M$. In contrast, if the overlap between $S_{k-|O\cap M|}$ and $O \setminus M$ is small, then they can be combined into a single large value set, which guarantees a large value for $S_{2(k-|O\cap M|)}$. Thus, there is a trade-off between the values of the sets $\hat{S}$ and $S_{2(k-|O\cap M|)}$. Lemma 4.2 formally captures this trade-offs.

**Lemma 4.2.** $f(S_{2(k-|O\cap M|)}) \geq \text{OPT} + f(S_{k-|O\cap M|}) - f(\hat{S})$.

**Proof.** To prove the lemma, we have to show that $V_A$ includes a set of size at most $2(k - |O \cap M|) \leq 2k$ whose value is at least $\text{OPT} + f(S_{k-|O\cap M|}) - f(\hat{S})$. In particular, we will show that the set $(O \setminus M) \cup S_{k-|O\cap M|}$ has these properties. (Note that this set is a subset of $V_A$ because $V_B \subseteq M$.) Clearly, the size of this set is at most $((|O| - |O \cap M|) + (k - |O \cap M|)) \leq 2(k - |O \cap M|)$.

Our next goal is to lower bound the value of the above set. Towards this goal, we note that $(O \setminus M) \cup S_{k-|O\cap M|}$ is a subset of $M$ of size at most $k$, and thus, $f(\hat{S}) \geq f((O \setminus M) \cup S_{k-|O\cap M|})$ by the definition of $\hat{S}$. Using the last inequality, we get

\[
f((O \setminus M) \cup S_{k-|O\cap M|}) \geq f((O \setminus M) \cup S_{k-|O\cap M|}) + f(O\cup S_{k-|O\cap M|}) - f(\hat{S})
\]

where the second inequality follows from the submodularity of $f$, and the last inequality from its monotonicity. □

If $f(\hat{S})$ is large, then we are done. Otherwise, the previous lemma guarantees that $S_{2(k-|O\cap M|)}$ is a very valuable set. While this set might be infeasible (unless $|O\cap M| \geq k/2$), its value can be exploited by adding half of this set to $O \setminus M$. The following lemma gives the lower bound on $f(\hat{S})$ that can be obtained in this way.

**Lemma 4.3.** $2f(\hat{S}) \geq f(O \setminus M) + f(S_{2(k-|O\cap M|)})$.

**Proof.** Let us define $S_{1}^{H_{2(k-|O\cap M|)}}$ and $S_{2}^{H_{2(k-|O\cap M|)}}$ as an arbitrary disjoint partition of the set $S_{2(k-|O\cap M|)}$ into two subsets of size at most $k - |O \cap M|$ each. Then, the submodularity of $f$ implies

\[
2 \sum_{h=1}^{2} f((O \setminus M) \cup S_{h}^{H_{2(k-|O\cap M|)})} \\
\geq f(O \setminus M) + f((O \setminus M) \cup S_{2(k-|O\cap M|)}) \\
\geq f(O \setminus M) + f(S_{2(k-|O\cap M|)}) ,
\]

where the second inequality follows from the monotonicity of $f$. The lemma now follows by the definition of $\hat{S}$ and the observation that both $(O \setminus M) \cup S_{1}^{H_{2(k-|O\cap M|)}}$ and $(O \setminus M) \cup S_{2}^{H_{2(k-|O\cap M|)}}$ are subsets of $M$ of size at most $k$. □

We are now ready to prove the approximation guarantee of Protocol 1 (and thus, complete the proof of the second part of Theorem 1.1).

**Corollary 4.4.** Protocol 1 is a $2/3$-approximation protocol.

**Proof.** Combining Lemmas 4.2 and 4.3, we get

\[
2 \cdot f(\hat{S}) \geq f(O \setminus M) + f(S_{2(k-|O\cap M|)}) \\
\geq f(O \setminus M) + \text{OPT} + f(S_{k-|O\cap M|}) - f(\hat{S}) .
\]

Rearranging this inequality, and then plugging into it the lower bound on $f(S_{k-|O\cap M|})$ given by Observation 4.1, yields

\[
f(\hat{S}) \geq \frac{f(O \setminus M) + \text{OPT} + f(O \setminus M)}{3} \geq \frac{2}{3} \cdot \text{OPT} ,
\]

where the second inequality follows from the submodularity and non-negativity of $f$. Because $\hat{S}$ is the output of Protocol 1, this concludes the proof. □

To prove also the first part of Theorem 1.1, we need to reduce the number of elements forwarded from Alice to Bob by Protocol 1. This can be done by applying geometric grouping to the sizes of the sets in $(S_i)_{i=1}^{2k}$. More precisely, Alice only forwards the sets $S_i$ for either $i = 0, i = \lfloor (1 + \epsilon)j \rfloor$, or $i = 2 \lfloor (1 + \epsilon)j \rfloor$ for some integer $0 \leq j \leq \log_{1+\epsilon} k$, where $\epsilon$ is the parameter from the theorem. This reduces the number of elements forwarded to $O(k/\epsilon)$, and it is not difficult to argue that the above analysis of the approximation ratio still works after this reduction, but its guarantee becomes worse by a factor of $1 - O(\epsilon)$. A formal proof of this can be found in the full version of this work [FNFSZ20].

### 4.2 Hardness of Approximation for Two Players

In this section we prove the impossibility result stated in Theorem 1.2. We do that by using a reduction from a problem known as the INDEX problem, which is presented in Section 2. The same section also states an impossibility result for a generalization of this problem (Theorem 3.3), which in the context of the INDEX problem implies that any protocol guaranteeing a success probability of at least $2/3$ for this problem must have a communication complexity of at least $n/144$.

Our plan in this section is to assume the existence of a protocol named PRT for Max-Card-$k$ in the two-players model with an approximation guarantee of $2/3 + \epsilon$, and show that this leads to a protocol $\text{PRINDEX}$ for the INDEX problem whose communication complexity depends on the communication complexity of PRT. This allows us to translate the communication complexity lower bound for protocols for INDEX to a communication complexity lower bound for PRT.

Before getting to the protocol $\text{PRINDEX}$ mentioned above, let us first present a simpler protocol for the the INDEX problem, which is given as Protocol 2 and is used as a building block for $\text{PRINDEX}$. Protocol 2 refers to $n$ possible objective functions that we denote by $f_1, f_2, \ldots, f_n$. (Recall that $n$ is the length of the string that Alice receives in the INDEX problem.) To define these functions, we first need to define a set of $n$ other functions. Let $W = \{w\} \cup \{u_i \mid i \in \}$
work \[FNFSZ20\]. In a nutshell, it shows by a straightforward case which shows that the objective function this protocol passes to

\[
y(\mathcal{P}_i) = \begin{cases} 
1 & \text{if } S = \{w\} \\
\min \left\{ \frac{1}{2} |S \setminus \{w\}|, 1 \right\} & \text{otherwise}.
\end{cases}
\]

The multilinear extension of \( g_i \) is the function \( G_i: [0,1]^W \to \mathbb{R}_{\geq 0} \) defined by \( G_i(y) = \mathbb{E} [g_i(R(y))] \), where \( R(y) \) is a random subset of \( W \) including every element \( v \in W \) with probability \( y_v \), independently.\(^3\) In the context of \( G_i \), given an element \( v \in W \), we occasionally use the notation \( 1_v \) to denote the characteristic vector of the singleton set \( \{v\} \), i.e., the vector in \([0,1]^W\) containing 1 in the \( v \)-coordinate and 0 in all other coordinates.

Let us now define the ground set \( W = W_A \cup W_B \), where \( W_A = \{u^j_i \mid i \in [n] \text{ and } j \in [k-1]\} \) and \( W_B = \{w\} \). Then, for every \( i \in [n] \), the function \( f_i: 2^W \to \mathbb{R}_{\geq 0} \) is defined as

\[
f_i(S) = G_i(y^S) \quad \forall S \subseteq V ,
\]

where the vector \( y^S \in [0,1]^W \) is defined by

\[
y^S_{uv'} = \frac{|S \cap \{u^j_i \mid j \in [k-1]\}|}{k-1} \quad \forall u' \in [n]
\]

and

\[
y^S_w = |\{w\} \cap S| .
\]

### Protocol 2 Reduction from INDEXE to Max-Card-k in the Two-Player Model

#### Alice’s Algorithm

1. The set of elements Alice of PRT gets is \( V_A = \{u^j_i \mid i \in [n] \text{ with } x_i = 1, \text{ and } j \in [k-1]\} \). Notice that this is indeed a subset of \( W_A \), and it intuitively corresponds to the 1-bits of the vector \( x \) given in Alice in the INDEXE problem.
2. The objective function for PRT is one of the functions \( f_1, f_2, \ldots, f_n \). Since these functions are identical when restricted to \( W_A \), the Alice part of PRT can execute without knowing which one of them is the real objective function.
3. Send to Bob the same message sent by the Alice of PRT.

#### Bob’s Algorithm

1. The set of elements Bob of PRT gets is \( V_B = W_B = \{w\} \).
2. The objective function for PRT can now be determined to be \( f_i \), where \( i \) is the index received by Bob.
3. If PRT returns a set of value at most \( \frac{2k}{3(k-1)} \), output \( x_t = 0 \); otherwise, output \( x_t = 1 \).

We begin the analysis of Protocol 2 with the following lemma, which shows that the objective function this protocol passes to PRT has all the necessary properties. The proof of this lemma is simple and technical, and thus, we defer it to the full version of this work [FNFSZ220].

#### Lemma 4.5

For every \( i \in [n] \), the functions \( g_i \) and \( f_i \) are non-negative, monotone, and submodular.

Our next step is analyzing the output distribution of Protocol 2.

#### Lemma 4.6

If \( x_t = 0 \), where \( t \) is the index received by Bob, then Protocol 2 always produces the correct answer.

Proof. Let \( S \) denote the output of PRT. We need to show that \( f_t(S) \leq \frac{2k}{3(k-1)} \). There are two cases to consider. The first one is when \( w \notin S \). In this case,

\[
f_t(S) = G_t(y^S) \leq g_t(\emptyset) + \sum_{v \in W} [G_t(y^S_v \cdot 1_v) - g_t(\emptyset)]
\]

\[
= g_t(\emptyset) + \sum_{v \in W} y^S_v \cdot g_t(\{v\}) \leq \frac{2}{3} \cdot \sum_{v \in W} y^S_v \leq \frac{2k}{3(k-1)}
\]

where the first inequality holds by submodularity of \( g_t \), the second equality holds by the multilinearity of \( G_t \), and the last inequality holds since the fact that \( S \) contains up to \( k \) elements guarantees that the sum of the coordinates of \( y^S \) is at most \( k/(k-1) \).

The other case we need to consider is when \( w \in S \). In this case,

\[
f_t(S) = G_t(y^S) \leq g_t(\{w\}) + \sum_{v \in W \setminus \{w\}} [G_t(y^S_v \cdot 1_v + 1_w) - g_t(\{w\})]
\]

\[
= g_t(\{w\}) + \sum_{v \in W \setminus \{w\}} y^S_v \cdot g_t(\{v\}) \leq \frac{1}{3} + \frac{1}{3} \cdot \sum_{v \in W \setminus \{w\}} y^S_v \leq \frac{1}{3}.
\]

where the last inequality holds this time since the fact that \( S \) contains up to \( k-1 \) elements in addition to \( w \) guarantees that the sum of all coordinates of \( y^S \) except for the \( w \)-coordinate is at most \( 1 \). To see why the third equality holds as well, note that the fact that \( x_t = 0 \) implies that none of the elements \( u^j_i \), \( j \in [k-1] \), belong to \( V_A \), and thus, \( y^S_{u^j_i} = 0 \).

#### Lemma 4.7

For \( k \geq \varepsilon^{-1} \geq 4 \), if \( x_t = 1 \), where \( t \) is the index received by Bob, then Protocol 2 always produces the correct answer with probability at least \( \varepsilon \).

Proof. We first observe that if \( x_t = 1 \), then the maximum value that the submodular function \( f_t \) achieves over subsets of \( V_A \cup V_B \) of cardinality \( k \) is \( 1 \). Clearly, the function \( g_t \) does not take values larger than \( 1 \) for any set, and therefore the same holds for its multilinear extension \( G_t \) and the function \( f_t \) defined using this multilinear extension. Thus, it remains to show that there exists a set \( S \subseteq V_A \cup V_B \) of size at most \( k \) with \( f_t(S) = 1 \). Since \( x_t = 1 \), all the elements \( u^j_i \), \( j \in [k-1] \), belong to \( V_A \). Thus, the set \( S = \{w\} \cup \{u^j_i \mid j \in [k-1] \} \) is a subset of \( V_A \cup V_B \) of size \( k \) whose value is

\[
f_t(S) = G_t(y^S) = g_t(\{w, u^j_i\}) = 1.
\]

Let us define now \( X \) to be a random variable corresponding to the value of the solution returned by PRT. Since we assumed that PRT is a \((2/3 + \varepsilon)\)-approximation algorithm, and we already proved that the highest value of a feasible set is 1, we get \( E[X] \geq 2/3 + \varepsilon \). To complete the proof of the lemma, we have to show that the probability \( \alpha = \Pr[X \geq \frac{2k}{3(k-1)}] \), which is the probability that

\[\frac{2k}{3(k-1)}\]
Protocol 2 correctly returns \( x_t = 1 \), is at least \( \epsilon \). We upper bound \( E[X] \) through the following simple variation of Markov’s inequality:

\[
\frac{2}{3} + \epsilon \leq E[X] \leq \Pr\left[ X \leq \frac{2k}{3(k-1)} \right] + \Pr\left[ X > \frac{2k}{3(k-1)} \right] \cdot 1
\]

\[
= (1 - \alpha) \cdot \frac{2k}{3(k-1)} + \alpha \leq (1 - \alpha) \cdot \frac{2}{3(1-\epsilon)} + \alpha ,
\]

where in the second inequality we used the fact that \( x \) is the largest value \( X \) can take, and the last inequality follows from \( k \geq \epsilon^{-1} \).

Rearranging the above inequality leads to

\[
(1 - 3\epsilon) \cdot \alpha \leq (1 - 3\epsilon) \cdot \alpha ,
\]

which, by using \( \epsilon^{-1} \geq 4 \), implies \( \alpha \geq \epsilon \), as desired. \( \square \)

At this point we are ready to present the promised algorithm \( \text{PRINDEX} \), which simply executes \([2\epsilon^{-1}] \) parallel copies of Protocol 2, and then outputs \( x_t = 1 \) if and only if at least one of the executions returned this answer.

**Corollary 4.8.** For \( k \geq \epsilon^{-1} \geq 4 \), \( \text{PRINDEX} \) always answers correctly when \( x_t = 0 \), and answers correctly with probability at least \( 2/3 \) when \( x_t = 1 \).

**Proof.** The first part of the corollary is a direct consequence of Lemma 4.6. Additionally, by Lemma 4.7, the probability that \( \text{PRINDEX} \) answers \( x_t = 0 \) when in fact \( x_t = 1 \) is at most

\[
(1 - \epsilon) \cdot [2\epsilon^{-1}] \leq (1 - \epsilon) \cdot 2\epsilon^{-1} \leq e^{-\epsilon^{-2}} \leq e^{-2} < \frac{1}{3} .
\]

Using the last corollary, we can now complete the proof of Theorem 1.2.

**Proof of Theorem 1.2.** Since Corollary 4.8 shows that \( \text{PRINDEX} \) is an algorithm for the INDEX problem that succeeds with probability at least \( 2/3 \), Theorem 3.3 guarantees that its communication complexity is at least \( n/144 \). Observe now that the message of \( \text{PRINDEX} \) consists of \([2\epsilon^{-1}] \) messages of Protocol 2, and thus, the communication complexity of Protocol 2 must be of size at least

\[
\frac{n}{144} \geq \frac{n}{432} = \frac{ne}{432}.
\]

We now recall that the message of Protocol 2 is simply the message generated by \( \text{PRINDEX} \) given the instance of Max-Card-\( k \) generated for it by Protocol 2. Since this instance has a ground set of size \( N = |W| = 1 + n(k - 1) \), the communication complexity of \( \text{PRINDEX} \) must be at least

\[
\frac{ne}{432} = \frac{e(N - 1)}{432(k - 1)} = \Omega\left( \frac{eN}{k} \right).
\]

\( \square \)

5 HARDNESS FOR MANY PLAYERS

In this section we prove that, in the case of many players, any protocol with reasonable communication complexity has an approximation guarantee upper bounded by an expression that tends to \( 1/2 \) as the number of players tends to infinity. Specifically, we show the following (where, for \( p \in \mathbb{Z}_{\geq 0} \), \( H_p = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{p} \) is the \( p \)-th harmonic number).

**Theorem 1.3.** For every \( \epsilon > 0 \), any \( p \)-player (randomized) protocol for Max-Card-\( k \) with an approximation guarantee of

\[
\frac{p + (H_p)^2}{2p - H_p} \cdot (1 + \epsilon)
\]

must have a communication complexity of \( \Omega\left( \frac{N}{p^\alpha} \right) \). Furthermore, this is true even in the special case in which the objective function \( f \) is a coverage function and \( k = p \).

We highlight that a (weighted) coverage function \( f : 2^V \to \mathbb{R}_{\geq 0} \) is defined as follows. There is a finite universe \( U \) with non-negative weights \( a : U \to \mathbb{R}_{\geq 0} \), and \( V \subseteq 2^U \) is a family of subsets of \( U \). Then, for any \( S \subseteq V \), we have \( f(S) = \sum_{u \in \cup S} a(u) \). We also remark that our hardness construction applies to the related maximum set coverage problem. In that problem the stream consists of \( N \) sets \( S_1, S_2, \ldots, S_N \) of some universe \( U \) and each \( S_i \) is encoded as the list of elements in that set. In other words, the submodular function \( f \) is given explicitly by the sets of the underlying universe. Prior work showed that, even in this setting, any streaming algorithm with a better approximation guarantee than \((1 - 1/e)\) requires memory \( \Omega(N) \) [MV19]. Our techniques also apply to this setting,\(^5\) and hence we improve the hardness factor for the maximum set coverage problem to the tight factor \( 1/2 \).

The heart of the proof of the above theorem is the construction of a family \( \mathcal{F} \) of submodular coverage functions on a common ground set \( W \), partitioned into sets \( W_1, \ldots, W_p \), one for each player. All the sets \( W_i \) have the same cardinality, which we denote by \( n \), and thus, \( N = |W| = n \cdot p \). The family \( \mathcal{F} \) contains a weighted coverage function \( f_{o_1, \ldots, o_p} \) for every \( o_1 \in W_1, o_2 \in W_2, \ldots, o_p \in W_p \). The intuition is that \( \{o_1, \ldots, o_p\} \) will be the “hidden” optimal solution for \( f_{o_1, \ldots, o_p} \) when we set \( k = p \).

For the hardness result, there are two crucial properties that the construction should satisfy:

- **Indistinguishability:** The \( i \)-th player should not be able to obtain any information about \( o_i \) by querying the submodular function on subsets of \( W_1 \cup W_2 \cup \cdots \cup W_i \).
- **Value gap:** The value of the solution \( \{o_1, \ldots, o_p\} \) is roughly twice the value of any solution of cardinality \( k = p \) that does not contain any of these elements.

The first property intuitively ensures that the players must use much communication to identify the special elements \( \{o_1, \ldots, o_p\} \); and the second property implies that, if they fail to do so, then the last player can only output a \( 1/2 \)-approximate solution. The following lemma formalizes these two properties that our family \( \mathcal{F} \) satisfies.

\(^4\)In some texts, the term coverage function is used for its unweighted version, i.e., \( a(u) = 1 \) for \( u \in U \). Our statements and proofs are described in terms of weighted coverage functions. However, this is merely a matter of convenience because any weighted coverage function can be approximated arbitrarily well through a scaled version of an unweighted one.

\(^5\)To see that this is the case, it is sufficient to observe that the intuitive description of the family \( \mathcal{F} \) in Section 5.1 is equivalent to the formal definition in the full version of this work [FNS20] when the underlying universe \( U \) is of infinite size; and, from that point of view, it is clear that the algorithm receives no advantage if given the explicit representation of the sets compared to having an oracle access to the coverage function. Furthermore, by standard Chernoff concentration inequalities (see, e.g., the proof of Lemma 8 in [MV19]), the family can be approximated up to any desired accuracy for feasible sets of cardinality at most \( k \) by selecting \( |U| = \Theta(k \log N) \).
Lemma 5.1. Let $W$ be partitioned into $p$ sets $W_1, \ldots, W_p$ of cardinality $n$. There is a family $\mathcal{F} = \{ f_{a_1, a_2, \ldots, a_p} \mid a_1 \in W_1, a_2 \in W_2, \ldots, a_p \in W_p \}$ of coverage functions on the ground set $W$ that satisfies:

- Indistinguishability: For $i \in [p]$, any two functions $f_{a_1, \ldots, a_p}, f'_{a'_1, \ldots, a'_p} \in \mathcal{F}$ with $a_1 = a'_1, \ldots, a_{i-1} = a'_{i-1}$ are identical when restricted to the ground set $W_1 \cup \cdots \cup W_i$.
- Value gap: For any $f_{a_1, \ldots, a_p} \in \mathcal{F}$, we have $f_{a_1, \ldots, a_p}(W) = f_{a_1, \ldots, a_p}(\{a_1, \ldots, a_p\}) \leq 2p$ and
  \[
  \max_{S \subseteq W \setminus \{a_1, \ldots, a_p\}} \frac{|W|}{|S|} f_{a_1, \ldots, a_p}(S) \leq \left(\frac{p + (Hp)^2}{2p - Hp}\right) f_{a_1, \ldots, a_p}(\{a_1, \ldots, a_p\}).
  \]

Equipped with the above lemma, we prove Theorem 1.3 in the full version of this work [FNFSZ20] by a rather direct reduction from the CHANDAG problem. We note that the reduction is similar to the one presented in Section 4.2 for the two-player case.

The core part of this section is the construction of $\mathcal{F}$ and the proof of Lemma 5.1. The outline is as follows. We first give an intuitive description of the main ideas in Section 5.1. The family $\mathcal{F}$ is then formally defined in the full version of this work [FNFSZ20], along with the proof of value gap and indistinguishability properties of Lemma 5.1.

5.1 Intuitive Description of Our Construction

In this section, we highlight our main ideas for constructing the family $\mathcal{F}$ satisfying the properties of Lemma 5.1. We do so by presenting three families of coverage functions $\mathcal{H}, \mathcal{G}$, and finally $\mathcal{F}$. Family $\mathcal{H}$ is a natural adaptation of coverage functions that have previously appeared in hardness constructions (see, e.g., [MV19]). We then highlight our main ideas for overcoming issues with those functions by first refining $\mathcal{H}$ to $\mathcal{G}$, and then by refining $\mathcal{G}$ to obtain our final construction $\mathcal{F}$.

To convey the intuition, we work with unweighted coverage functions. However, to provide a clean and concise technical presentation later on, we use weighted coverage functions to formally realize the construction plan described here.

The first attempt: family $\mathcal{H}$. The construction of the family $\mathcal{H} = \{ h_{a_1, \ldots, a_p} \mid a_1 \in W_1, \ldots, a_p \in W_p \}$ is inspired by the coverage functions constructed in the NP-hardness result of [Fei98]. In those coverage functions every element corresponds to a subset of the underlying universe of size $|U|/p$. Furthermore, the optimal solution $\{a_1, \ldots, a_p\}$ forms a disjoint cover of $U$, whereas any other element behaves like a random subset of the universe of size $|U|/p$.

Inspired by this, we let $h_{a_1, \ldots, a_p} \in \mathcal{H}$ be the coverage function where

- the subsets of $U$ corresponding to $a_1, \ldots, a_p$ form a partition of equal-sized sets, i.e., of size $|U|/p$ each;
- every other element corresponds to a randomly selected subset of $U$ of size $|U|/p$.

While the above definition is randomized, we assume for the sake of simplicity in this overview that the value of a subset equals its expected value. This can intuitively be achieved by selecting the underlying universe $U$ to be large enough so as to ensure concentration. For a subset $S \subseteq W \setminus \{a_1, \ldots, a_p\}$, we thus have that $h_{a_1, \ldots, a_p}(S)$ equals the expected number of elements of $S$ covered by $|S|$ random subsets of cardinality $|U|/p$. Hence

\[
 h_{a_1, \ldots, a_p}(S) = \left(1 - \left(1 - \frac{|S|}{|U|}\right)^{|U|/p}\right) |U|.
\]

which is at least $(1 - 1/e)|U|$ if $|S| \approx p$. This already highlights the first issue of the construction: the value gap between the optimal solution $\{a_1, \ldots, a_p\}$, whose value is $|U|$, and a solution disjoint from this optimal solution is only $1 - 1/e$; while we need it to approach $1/2$ as $p$ tends to infinity.

The second and perhaps more significant issue is the indistinguishability. First, we can observe that the value of any subset $S \subseteq W_1$ only depends on $|S|$, and thus the selection of $a_1 \in W_1$ is indistinguishable when querying the submodular function restricted to $W_1$. However, the same does not hold for $a_2$ when querying the submodular function restricted to the set $W_1 \cup W_2$. To see this, note that $a_1$ and $a_2$ are the only elements of $W_1 \cup W_2$ whose corresponding subsets of $U$ are disjoint. In other words, $\{a_1, a_2\}$ is the unique maximizer to

\[
 \max_{S \subseteq W_1 \cup W_2 : |S| = 2} h_{a_1, \ldots, a_p}(S),
\]

and $a_2$ can thus be identified by querying the submodular function on $W_1 \cup W_2$. A natural idea for addressing this issue is to make all elements of $W_2$, and not only $a_2$, correspond to subsets of $U$ that are disjoint of the subset corresponding to $a_1$. Making this modification for all $W_2, \ldots, W_p$ results in the refined family $\mathcal{G}$ that we now describe. We note that a similar approach was used in [Kap13] to guarantee indistinguishability.

The first refinement: family $\mathcal{G}$. Motivated by the idea to make every element in $W_i$ correspond to a subset of $U$ disjoint from the subsets of $a_1, \ldots, a_{i-1}$, we define the family $\mathcal{G} = \{ g_{a_1, \ldots, a_p} \mid a_1 \in W_1, \ldots, a_p \in W_p \}$ of coverage functions. Specifically, we let $g_{a_1, \ldots, a_p} \in \mathcal{G}$ be the coverage function where

- the subsets of $U$ corresponding to $a_1, \ldots, a_p$ form a partition of equal-sized sets, i.e., of size $|U|/p$ each;
- for $i = 1, \ldots, p$, every element in $W_i \setminus \{a_i\}$ corresponds to a randomly selected subset of $U$ of size $|U|/p$ that is disjoint from the subsets corresponding to $a_1, \ldots, a_{i-1}$.

The above description of $\mathcal{G}$ is given in a way that highlights the changes compared to $\mathcal{H}$. Another equivalent definition of $g_{a_1, \ldots, a_p}$ is that it is the coverage function where

- the elements of $W_i$ form random subsets of $U$ of size $|U|/p$;
- for $i = 2, \ldots, p$, every element in $W_i$ corresponds to a randomly selected subset of $U$ of size $|U|/p$ that is disjoint from the subsets corresponding to $a_1, \ldots, a_{i-1}$.

From this viewpoint, it is clear that we now have the indistinguishability property of Lemma 5.1. Indeed, for $i \in [p]$, the only subsets of $U$ that depend on $a_i$ in the above construction are those corresponding to elements in $W_{i+1}, \ldots, W_p$. It follows that the value of a subset $S \subseteq W_i \cup \cdots \cup W_p$, which is a function of the subsets of $U$ corresponding to the elements in $S$, is independent of the selection of $a_i$.
Having verified indistinguishability, let us consider the value gap. First, note that we still have that the optimal solution \( \{a_1, \ldots, a_p\} \) covers the whole universe, and thus has value \(|U|\). Now consider a set \( S \subseteq W \setminus \{a_1, \ldots, a_p\} \). It will be instructive to first consider the case when \( S = \{v_1, \ldots, v_p\} \) with \( v_i \in W_i \) for all \( i \in [p] \), i.e., \( S \) contains exactly one element from each of the sets \( W_i \). Abbreviating \( g_{o_1, \ldots, o_p} \) by \( g \), we have in this case that
\[
g_{o_1, \ldots, o_p}(S) = g(o_1) + g(v_2 \mid \{v_1\}) + g(v_3 \mid \{v_1, v_2\}) + \ldots + g(v_p \mid \{v_1, \ldots, v_{p-1}\})
\]
equals
\[
\frac{|U|}{p} \left( 1 + \left(1 - \frac{1}{p}\right) + \left(1 - \frac{1}{p}\right)^2 + \ldots \right)
= \frac{|U|}{2} \left( 1 + \frac{1}{p} \right) .
\]
which in turn solves to
\[
\frac{|U|}{2} \left( 1 + \frac{1}{p} \right) .
\]
Hence, for sets \( S \subseteq W \setminus \{a_1, \ldots, a_p\} \) that contain one element from each \( W_i \), we have a value gap that approaches the desired constant \( 1/2 \) as \( p \) tends to infinity. The issue is that there are other subsets of \( W \setminus \{a_1, \ldots, a_p\} \) of significantly higher value. To see this, note that any element \( e \in W_i \setminus \{v_1\} \) has a marginal value with respect to \( \{v_1, \ldots, v_{p-1}\} \) that is much higher than the marginal value of \( v_p \) with respect to the same set. In particular, the value of a subset \( W_i \setminus \{v_1, \ldots, v_p\} \) of cardinality \( p \) is equivalent for functions in \( \mathcal{G} \) and \( \mathcal{H} \), and is at least \( (1 - 1/e)|U| \). To overcome this issue (i.e., the fact that elements of \( W_i \) are more “valuable” than other elements), we modify the above construction to let the elements from different \( W_i \)‘s correspond to subsets of different sizes.

The second and last refinement: family \( \mathcal{F} \). The family \( \mathcal{F} = \{f_{o_1, \ldots, o_p} \mid o_1 \in W_1, \ldots, o_p \in W_p\} \) is obtained from \( \mathcal{G} \) by selecting subsets of \( U \) of \( \text{non-uniform sizes} \). Specifically, we carefully select numbers \( 1 = a_1 < a_2 < \cdots < a_p \), and make the elements of \( W_i \) correspond to subsets of \( U \) of size \( a_i \); then we let the total size of \( U \) be \( a_1 + a_2 + \ldots + a_k \). We now let \( f_{o_1, \ldots, o_p} \in \mathcal{F} \) be the coverage function where
- the elements of \( W_i \) form random subsets of \( U \) of size \( a_i = 1 \);
- for \( i = 2, \ldots, p \), every element in \( W_i \) corresponds to a randomly selected subset of \( U \) of size \( a_i \) that is disjoint from the subsets corresponding to \( o_1, \ldots, o_{i-1} \).

The family \( \mathcal{F} \) satisfies the indistinguishability property of Lemma 5.1 for the exact same reasons \( \mathcal{G} \) satisfies it. We now explain how the values \( a_1, \ldots, a_p \) are selected so as to obtain the value gap. Consider a set \( S \subseteq W \setminus \{v_1, \ldots, v_p\} \) obeying \( S = \{v_1, \ldots, v_p\} \) for some choice of \( v_i \in W_i \) for every \( i \in [p] \). Abbreviating \( f_{o_1, \ldots, o_p} \) by \( f \), we thus have
\[
f_{o_1, \ldots, o_p}(S) = f(o_1) + f(v_2 \mid \{v_1\}) + f(v_3 \mid \{v_1, v_2\}) + \ldots + f(v_p \mid \{v_2, \ldots, v_{p-1}\}) .
\]
The numbers \( a_1, \ldots, a_p \) are selected so that each term of this sum equals \( 1 \), and hence, \( f_{o_1, \ldots, o_p}(S) = p \). Notice that this is in stark contrast to the functions in \( \mathcal{G} \) where the contributions to (1) were highly unequal. The intuitive reason why we set the numbers so that these marginal contributions are the same is that we want to prove that one cannot form a subset of \( W \setminus \{o_1, \ldots, o_p\} \) of cardinality at most \( p \) of significantly higher value by increasing the number of elements selected from one of the partitions \( W_i \). Formally, this is proved in the full version of this work [FNFSZ20] by considering the linear extension of a concave function at the point corresponding to such a set \( S \) that contains a single element from each \( W_i \). This allows us to upper bound the value of any subset \( W \setminus \{o_1, \ldots, o_p\} \) of cardinality at most \( p \) by \( p + (H_p)^2 \). The value gap then follows from basic calculations (see the full version of this work [FNFSZ20]) which show that \( f_{o_1, \ldots, o_p}(o_1, \ldots, o_p) = |U| = \sum_{i=1}^{p} a_i \) is at least \( 2p - H_p \) and at most \( 2p \).

**ACKNOWLEDGMENTS**

The first author was supported in part by the Israel Science Foundation (ISF) under grant no. 1357/16. The third author was supported by the Swiss National Science Foundation under grant no. 200021-184656 “Randomness in Problem Instances and Randomized Algorithms.” The last author was supported by Swiss National Science Foundation grants 200021_184622 and 200021_165866, and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 817750).

**REFERENCES**

[AES*20] Naor Alaluf, Alina Ene, Moran Feldman, Huy L. Nguyen, and Andrew Suh. Optimal streaming algorithms for submodular maximization with cardinality constraints. CoRR abs/1911.12959, 2020.

[ASS18] Shipra Agrawal, Mohammad Shadravan, and Cliff Stein. Submodular secretary problem with shortlists. CoRR, abs/1809.05082, 2018.

[Bac10] Francis R. Bach. Structured sparsity-inducing norms through submodular functions. In Proceedings of Advances in Neural Information Processing Systems (NIPS), volume 23, pages 118–126, 2010.

[BENW16] Rafael Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. A new framework for distributed submodular maximization. In Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 645–654, 2016.

[BFS15] Niv Buchbinder, Moran Feldman, and Roy Schwartz. Online submodular maximization with preemption. In Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1202–1216, 2015.

[BIRB15] Ramakrishna Bairi, Rishabh K. Iyer, Ganesh Ramakrishnan, and Jeff A. Bilmes. Summarization of multi-document topic hierarchies using submodular mixtures. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL | IJCNLP), volume 1, pages 553–563, 2015.

[BMKK14] Ashwin Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In Proceedings of the 26th ACM Conference on Knowledge Discovery and Data Mining (KDD), pages 671–680, 2014.

[BMSCT17] Bija Bgoumovic, Shlobodan Mitrovic, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In Proceedings of the 34th International Conference on Machine Learning (ICML), pages 508–516, 2017.

[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 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 283–302, 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 Symposium on Theory of Computing (STOC), pages 66–77, 2019.
[BS18] Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC), pages 1138–1151, 2018.

[BYJKS02] Ziv Bar-Yossef, Thathachar S. Jayram, Ravi Kumar, and D. Sivakumar. Communication theory methods in communication complexity. In Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC), pages 93–102, 2002.

[CCPV11] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. SIAM Journal on Computing, 40(6):1740–1766, 2011.

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

[Chat07] A. Chatrabhuti. Lower bounds for multi-player pointer jumping. Electronic Colloquium on Computational Complexity, 14, 2007.

[CK14] Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming. Matchings, matroids, and more. In Jon Lee and Jens Vygen, editors, Proceedings of the 17th Conference on Integer Programming and Combinatorial Optimization (IPCO), pages 210–221, 2014.

[CQ19] Chandra Chekuri and Kent Quanrud. Submodular function maximization in parallel via the multilinear relaxation. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 303–322, 2019.

[DDK12] Abhimanyu Das, Anirban Dasgupta, and Ravi Kumar. Selecting diverse features via spectral regularization. In Advances in Neural Information Processing Systems (NIPS), volume 25, pages 1592–1600, 2012.

[DK11] Abhamanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for suboptimal set selection, sparse approximation and dictionary selection. In Proceedings of the 26th International Conference on Machine Learning (ICML), pages 1057–1064, 2011.

[dPENW15] 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), pages 1236–1244, 2015.

[EN19] Alina Ene and Huy L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 274–282, 2019.

[ENV19] Alina Ene, Huy L. Nguyen, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC), pages 90–101, 2019.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. Journal of the ACM (JACM), 45(4):634–652, 1998.

[FKK18] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with subsampling. CoRR, abs/1802.07098, 2018.

[FMZ19a] Matthew Fahrbach, Vahab S. Mirrokni, and Mortezza Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In Proceedings of the 36th International Conference on Machine Learning (ICML), pages 1835–1842, 2019.

[FMZ19b] Matthew Fahrbach, Vahab S. Mirrokni, and Mortezza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 255–273, 2019.

[FNSZ20] Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness, 2020.

[GC19] C. Konrad G. Coromode, J. Dark. Independent sets in vertex-arrival streams. In Proceedings of 46th International Colloquium on Automata, Languages and Programming (ICALP), pages 45:1–45:14, 2019.

[GDV11] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research, 42(3):427–486, 2011.

[HKMY20] Chien-Chung Huang, Naonori Kakimura, Simon MOUR, and Yuichi Yoshida. Approximability of monotone submodular function maximization under cardinality and matroid constraints in the streaming model. CoRR, abs/2002.05477, 2020.

[JKS08] Thathachar S Jayram, Ravi Kumar, and D. Sivakumar. The one-way communication complexity of hamming distance. Theory of Computing, 4(1):129–135, 2008.

[Kap13] Michael Kapralov. Better bounds for matchings in the streaming model. In Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1679–1697, 2013.

[KMZ*19] Ehsan Kazemi, Marko Mitrovic, Mortezza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptivity complexity. In Proceedings of the 36th International Conference on Machine Learning (ICML), pages 3311–3320, 2019.

[Kra] Andreas Krause. Submodularity in machine learning. http://submodularity.org/.

[KKKZ18] Ehsan Kazemi, Mortezza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 2544–2553, 2018.

[LV19] Paul Liu and Jan Vondrák. Submodular optimization in the mapreduce model. In 2nd Symposium on Simplicity in Algorithms (SOCSA), pages 18:1–18:10, 2019.

[MNF*17] Slobodan Mitrovic, Ilya Bogunovic, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. Streaming robust submodular maximization: A partitioned thresholding approach. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), pages 4560–4569, 2017.

[MKK17] Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In Proceedings of the 34th International Conference on Machine Learning (ICML), pages 2449–2458, 2017.

[MV19] Andrew McGregor and Hoa T. Vu. Better streaming algorithms for the maximum coverage problem. Theory Comput. Syst., 63(3):1595–1619, 2019.

[MZ15] Vahab S. Mirrokni and Mortezza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC), pages 153–162, 2015.

[NTM*18] Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavi, and Ola Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 3826–3835, 2018.

[NW78] George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. Mathematics of Operations Research, 3(3):177–188, 1978.

[NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions— I. Mathematical Programming, 14(1):265–294, 1978.

[OSU16] James B. Orlin, Andreas S. Schulz, and Rajan Udwni. Robust monotone submodular function maximization. In Proceedings of 19th International Conference on Integer Programming and Combinatorial Optimization (IPCO), pages 312–324, 2016.

[Sch03] A. Schrijver. Combinatorial Optimization - Polyhedra and Efficiency. Springer, 2003.

[ZCP14] Jingjing Zheng, Zhixin Li, Rama Chellappa, and P. Jonathon Phillips. Submodular attribute selection for action recognition in video. In Proceedings of Advances in Neural Information Processing Systems (NIPS), volume 27, pages 1341–1349, 2014.