A Polynomial Lower Bound on the Number of Rounds for Parallel Submodular Function Minimization and Matroid Intersection

Deeparnab Chakrabarty* Yu Chen† Sanjeev Khanna‡

November 16, 2021

Abstract

Submodular function minimization (SFM) and matroid intersection are fundamental discrete optimization problems with applications in many fields. It is well known that both of these can be solved making $\mathrm{poly}(N)$ queries to a relevant oracle (evaluation oracle for SFM and rank oracle for matroid intersection), where N denotes the universe size. However, all known polynomial query algorithms are highly adaptive, requiring at least N rounds of querying the oracle. A natural question is whether these can be efficiently solved in a highly parallel manner, namely, with $\mathrm{poly}(N)$ queries using only poly-logarithmic rounds of adaptivity.

An important step towards understanding the adaptivity needed for efficient parallel SFM was taken recently in the work of Balkanski and Singer who showed that any SFM algorithm making $\mathrm{poly}(N)$ queries necessarily requires $\Omega(\log N/\log\log N)$ rounds. This left open the possibility of efficient SFM algorithms in poly-logarithmic rounds. For matroid intersection, even the possibility of a constant round, $\mathrm{poly}(N)$ query algorithm was not hitherto ruled out.

In this work, we prove that any, possibly randomized, algorithm for submodular function minimization or matroid intersection making $\mathrm{poly}(N)$ queries requires[1] $\tilde{\Omega}\left(N^{1/3}\right)$ rounds of adaptivity. In fact, we show a polynomial lower bound on the number of rounds of adaptivity even for algorithms that make at most $2^{N^{1-\delta}}$ queries, for any constant $\delta > 0$. Therefore, even though SFM and matroid intersection are efficiently solvable, they are not highly parallelizable in the oracle model.

*Department of Computer Science, Dartmouth College. Email: deeparnab@dartmouth.edu. Supported in part by NSF award CCF-2041920.

†Department of Computer and Information Science, University of Pennsylvania. Email chenyu2@cis.upenn.edu

‡Department of Computer and Information Science, University of Pennsylvania. Email sanjeev@cis.upenn.edu Supported in part by NSF awards CCF-1910534, CCF-1926872, and CCF-2045128.

[1] Throughout the paper, we use the usual convention of using $\widetilde{\Omega}(f(n))$ to denote $\Omega(f(n)/\log^c f(n))$ and using $\tilde{O}(f(n))$ to denote $O(f(n)\cdot\log^c f(n))$, for some unspecified constant c

1 Introduction

A function $f : 2^U \to \mathbb{Z}$ defined over subsets of a ground set U of N elements is submodular if for any two sets $A \subseteq B$ and an element $e \notin B$, the *marginal* of e on A, that is, $f(A \cup e) - f(A)$ is at least $f(B \cup e) - f(B)$. The submodular function minimization (SFM) problem is to find a subset S minimizing $f(S)$ given access to an evaluation oracle for the function that returns the function value on any specified subset. SFM is a fundamental discrete optimization problem which generalizes classic problems such as minimizing global and s-t cuts in graphs and hypergraphs, and more recently has found applications in areas such as image segmentation [BK04, BVZ01, KKT08] and speech analysis [IB13, IJB13].

A remarkable fact is that SFM *can* be solved in polynomial time with polynomially many queries to the evaluation oracle. This was first established by Grötschel, Lovász, and Schrijver [GLLS81] using the ellipsoid method. Since then, a lot of work [Cun85, IFF01, Sch00, Orl09, IO09, CJK14, LJJ15, LSW15, CLSW17, DVZ18, ALS20, Jia21] has been done trying to understand the query complexity of SFM. The current best known algorithms are an $O(N^3)$-query polynomial-time and an $O(N^2 \log N)$-query exponential time algorithm by Jiang [Jia21] building on the works [LSW15, DVZ18], an $\tilde{O}(N^2 \log M)$-query and time algorithm by Lee, Sidford, and Wong [LSW15] where $|f(S)| \le M$ for all $S \subseteq U$, and an $\tilde{O}(NM^2)$ query and time algorithm by Axelrod, Liu, and Sidford [ALS20] improving upon [CLSW17].

Any SFM algorithm accesses the evaluation oracle in rounds, where the queries made in a certain round depend only on the answers to queries made in previous rounds. There is a trade-off between the number of queries (per round) made by the algorithm, and the number of rounds needed to find the answer : there is an obvious 1-round algorithm which makes all 2^N queries. All known efficient algorithms for SFM described above are *highly sequential*; all of them proceed in $\Omega(N)$ rounds. Can the number of rounds be substantially decreased (made poly-logarithmic in N) while still keeping the number of queries bounded by $\mathrm{poly}(N)$? In spirit, this is related to the $\mathbf{P}$ versus $\mathbf{NC}$ question which at a high-level asks whether problems with polynomial time algorithms be solved by poly-sized circuits with poly-logarithmic depth. From a practical standpoint, given the applications of SFM to problems involving huge data and the availability of computing infrastructure to perform parallel computation, the question of low-depth parallel SFM algorithms is timely.

A study of this question was initiated by Balkanski and Singer in [BS20] who proved that any polynomial query SFM algorithm must proceed in $\Omega(\frac{\log N}{\log \log N})$ rounds. This still leaves open the possibility of polynomial query poly-logarithmic round algorithms. Indeed for the related problem of submodular function *maximization* subject to cardinality constraint, in a different paper [BS18], Balkanski and Singer showed that the correct answer is indeed $\tilde{\Theta}(\log N)$. They proved that with polynomially many queries no constant factor approximation is possible with $o\left(\frac{\log N}{\log \log N}\right)$ rounds, while an $1/3$-approximation can be obtained in $O(\log N)$-rounds[2]. Can the situation be the same for SFM?

In this paper we answer this question in the negative. We prove a ***polynomial*** lower bound on the number of rounds needed by any polynomial query SFM algorithm.

Theorem 1. *For any constant $\delta > 0$ and any $1 \le c \le N^{1-\delta}$, any possibly randomized algorithm for SFM on an N element universe making $\le N^c$ evaluation oracle queries per round and succeeding with probability $\ge 2/3$ must have $\Omega\left(\frac{N^{1/3}}{(c \log N)^{1/3}}\right)$ rounds-of-adaptivity. This is true even when the range of the submodular function is $\{-N, -N+1, \ldots, N-1, N\}$, and even if the algorithm is only required to output the value of the minimum.*

[2] This result has since been improved [BRS19, CQ19a, CQ19b, EN19, ENV19, LLV20]; see Section 1.1 for details.

We note that a polynomial lower bound on the number of rounds holds even if the algorithm is allowed to make $2^{N^{1-\delta}}$ queries per round for any $\delta > 0$, and the lower bound on the number of rounds is $\widetilde{\Omega}(N^{1/3})$ for polynomial query algorithms. Our construction also proves lower bounds on the number of rounds required for *approximate* submodular function minimization. In this problem, one assumes via scaling that the function's range is in $[-1, +1]$ and the goal is to return a set whose value is within an additive ε from the minimum. We can prove an $\widetilde{\Omega}(1/\varepsilon)$-lower bound on the number of rounds required for approximate SFM. The only previous work ruling out ε-approximate minimizers is another work of Balkanski and Singer [BS17] who proved that *non-adaptive* algorithms, that is single round algorithms, cannot achieve any non-trivial approximation with polynomially many queries.

Matroid intersection is another fundamental combinatorial optimization problem generalizing the maximum cardinality bipartite matching problem and the problem of packing spanning trees and arborescences in graphs. In matroid intersection, we are given two matroids $\mathcal{M}_1 = (U, \mathcal{I}_1)$ and $\mathcal{M}_2 = (U, \mathcal{I}_2)$ over the same universe, and the objective is to find the largest cardinality independent set present in both matroids. There are two standard ways to access these matroids: one is via the independence oracle which says whether a set is independent in a given matroid or not, and the other is via the rank oracle, which when queried with a subset S returns the size of the largest independent subset of S. The rank oracle is stronger. It is known via Edmond's minimax [Edm70] result that matroid intersection (with access via rank oracles) is, in fact, a special case of submodular function minimization. The first algorithms for matroid intersection [AD71, Law75, Edm70] made $O(N^3)$ *independence oracle* queries, which was improved to $O(N^{2.5})$ by Cunningham [Cun86]. More recently, Chakrabarty *et al.* [CLS+19] and Nguyen [Ngu19] improved the number of queries to $\tilde{O}(N^2)$. The current record holder is a randomized algorithm by Blikstad *et al.* [BvdBMN21] making $\tilde{O}(N^{9/5})$ independence queries. The best algorithm using rank oracle queries is in [CLS+19] which gives an $\tilde{O}(N^{1.5})$-rank oracle query algorithm. As in the case of SFM, all these algorithms are sequential requiring $\Omega(N)$-rounds of adaptivity.

The submodular functions we construct to prove Theorem 1 are closely related to the rank functions of nested matroids, a special kind of laminar matroids. As a result, we prove a similar result as in Theorem 1 for matroid intersection.

Theorem 2. *For any constant $\delta > 0$ and any $1 \le c \le N^{1-\delta}$, any possibly randomized algorithm for matroid intersection on an N element universe making $\le N^c$* rank-oracle *queries per round and succeeding with probability $\ge 2/3$ must have $\Omega\left(\frac{N^{1/3}}{(c \log N)^{1/3}}\right)$ rounds-of-adaptivity. This is true even when the two matroids are* nested matroids*, a special class of laminar matroids, and also when the algorithm is only required to output the value of the optimum.*

In particular, any algorithm making polynomially many queries to the rank oracle must have $\widetilde{\Omega}(N^{1/3})$ rounds of adaptivity, even to figure out the size of the largest common independent set. That is, even the "decision" version of the question (is the largest cardinality at least some parameter K) needs polynomially many rounds of adaptivity.

Our results shows that in the general query model, SFM and matroid intersection cannot be solved in polynomial time in poly-logarithmic rounds, even with randomization. This is in contrast to *specific* explicitly described succinct SFM and matroid intersection problems. For instance, global minimum cuts in an undirected graph is in **NC** [KM97], finding minimum s-t-cuts with poly-bounded capacities is in **RNC** [KUW86], and linear and graphic matroid intersection is in **RNC** [NSV94]. More recently, inspired by some of these special cases, Gurjar and Rathi [GR20] defined a class of submodular functions called *linearly representable* submodular functions and gave **RNC** algorithms for the same.

Our lower bounding submodular functions fall in a class introduced by Balkanski and Singer [BS20] which we call *partition submodular functions*. Given a partition $\mathcal{P} = (P_1, \ldots, P_r)$ of the universe U, the value of a partition submodular function $f(S)$ depends only on the *cardinalities* of the $|S \cap P_i|$'s. In particular, $f(S) = h(\mathbf{x})$ where $\mathbf{x}$ is an r-dimensional non-negative integer valued vector with $\mathbf{x}_i := |S \cap P_i|$, and h is a discrete submodular function on a hypergrid. Note that when $r = 1$, the function h is a univariate concave function, and when $r = n$ we obtain general submodular functions. Thus, partition submodular functions form a nice way of capturing the complexity of a submodular function.

The [BS20] functions are partition submodular and they prove an $\Omega(r)$-lower bound for their specific functions. As we explain in Section 2, their construction idea has a bottleneck of $r = O(\log N)$, and thus cannot prove a polynomial lower bound. Our lower bound functions are also partition submodular, and we also prove an $\Omega(r)$ lower bound though we get r to be polynomially large in the size of the universe. Furthermore, our partition submodular functions turn out to be closely related to ranks of nested matroids which lead to our lower bound for parallel matroid intersection.

1.1 Related Work

For parallel algorithms, the depth required for the "decision" version and the "search" version may be vastly different. In a thought provoking paper [KUW88], Karp, Upfal and Wigderson considered this question. In particular, they proved that any efficient algorithm that finds a maximum independent set in a *single* (even a partition) matroid with access to an *independence oracle* must proceed in $\widetilde{\Omega}(N^{1/3})$ rounds. On the other hand, with access to a *rank* oracle which takes S and returns $r(S)$, the size of the largest independent set in S, there is a simple algorithm[3] which makes N queries in a single round and finds the optimal answer. Our lower bound shows that for matroid intersection, rank oracles also suffer a polynomial lower bound, even for the decision version of the problem. At this point, we should mention a very recent work of Ghosh, Gurjar, and Raj [GGR22] which showed that if there existed poly-logarithmic round algorithms for the (weighted) decision version for matroid intersection with rank-oracles, then in fact there exists *deterministic* polylogarithmic round algorithms for the *search* version. A similar flavor result is also present in [NSV94]. Unfortunately, our result proves that polylogarithmic depth is impossible for arbitrary matroids (even nested ones), even when access is via rank oracles.

The rounds-of-adaptivity versus query complexity question has seen a lot of recent work on submodular function *maximization*. As mentioned before, Balkanski and Singer [BS18] introduced this problem in the context of maximizing a non-negative monotone submodular function $f(S)$ subject to a cardinality constraint $|S| \leq k$. This captures $\mathbf{NP}$-hard problems, has a *sequential* greedy $(1 - \frac{1}{e})$-approximation algorithm [NWF78], and obtaining anything better requires [NW78, Von13] exponentially many queries. [BS18] showed that obtaining even an $O\left(\frac{1}{\log N}\right)$-approximation with polynomially many queries requires $\Omega\left(\frac{\log N}{\log \log N}\right)$ rounds, and gave an $O(\log N)$-round, polynomial query, $\frac{1}{3}$-approximation. Soon afterwards, several different groups [BRS19, EN19, FMZ19, CQ19b, CQ19a, ENV19] gave $\left(1 - \frac{1}{e} - \varepsilon\right)$-approximation algorithms making polynomially many queries which run in $\mathrm{poly}(\log N, \frac{1}{\varepsilon})$-rounds, even when the constraint on which S to pick is made more general. More recently, Li, Liu and Vondrák [LLV20] showed that the dependence of the number of rounds on ε (the distance from $1 - 1/e$) must be a polynomial. Also related is the question of maximizing a non-negative non-monotone submodular function without any constraints. It is known that a random set gives a $\frac{1}{4}$-approximation, and a sequential "double-greedy" $\frac{1}{2}$-approximation was given by Buchbinder, Feldman, Naor, and Schwartz [BFNS15], and this approximation

[3] Order elements as $e_1, \ldots, e_N$ and query $r(\{e_1, \ldots, e_i\})$ for all i, and return the points at which the rank changes.

factor is tight [FMV11]. Chen, Feldman, and Karabasi [CFK19] gave a nice parallel version obtaining an $\left(\frac{1}{2} - \varepsilon\right)$-approximation in $O(\frac{1}{\varepsilon})$-rounds.

In the continuous optimization setting, the question of understanding the "parallel complexity" of minimizing a non-smooth convex function was first studied by Nemirovski [Nem94]. In particular, the paper studied the problem of minimizing a bounded-norm convex (non-smooth) function over the unit ℓ_∞ ball in N-dimensions, and showed that any polynomial query (value oracle or gradient oracle) algorithm which comes ε-close must have $\widetilde{\Omega}(N^{1/3}\ln(1/\varepsilon))$ rounds of adaptivity. Nemirovski [Nem94] conjectured that the lower bound should be $\widetilde{\Omega}(N\ln(1/\varepsilon))$, and this is still an open question. When the dependence on ε is allowed to be polynomial, then the sequential vanilla gradient descent outputs an ε-minimizer in $O(1/\varepsilon^2)$-rounds (over Euclidean unit norm balls), and the question becomes whether parallelism can help over gradient descent in some regimes of ε. Duchi, Bartlett, and Wainwright [DBW12] showed an $O(N^{1/4}/\varepsilon)$-query algorithm which is better than gradient-descent when $\frac{1}{\varepsilon^2} > \sqrt{N}$. A matching lower bound in this regime was shown recently by Bubeck et al. [BJL$^+$19], and this paper also gives another algorithm which has better depth dependence in some regime of ε. It is worth noting that submodular function minimization can also be thought of as minimizing the Lovász extension which is a non-smooth convex function. Unfortunately, the domain of interest (the unit cube) has ℓ_2-radius $\sqrt{N}$, and the above algorithms do not imply "dimension-free" ε-additive approximations for submodular function minimization. Our work shows that $\Omega(1/\varepsilon)$-rounds are needed, and it is an interesting open question whether a $\mathrm{poly}(N, \frac{1}{\varepsilon})$-lower bound can be shown on the number of rounds, or whether one can achieve efficient ε-approximations in rounds independent of N.

The question of rounds-of-adaptivity versus query complexity has been asked for many other computational models, and also is closely related to other fields such as communication complexity and streaming. We note a few results which are related to submodular function minimization. Assadi, Chen, and Khanna [ACK19] considered the problem of finding the minimum s-t-cut in an undirected graph in the streaming setting. They showed that any p-pass algorithm must take $\widetilde{\Omega}(n^2/p^5)$-space, where n is the number of vertices. Their result also implied that any sub-polynomial round algorithm for the s-t-cut submodular function must make $\widetilde{\Omega}(n^2)$ queries; note that with $O(n^2)$ queries, the whole graph can be non-adaptively learned. Rubinstein, Schramm, and Weinberg [RSW18] considered the global minimum cut function in an undirected unweighted graph, and showed that $\tilde{O}(n)$ queries suffice, and their algorithm can be made to run in $O(1)$-rounds. Subsequently, Mukhopadhyay and Nanongkai [MN20] generalized this for weighted undirected graphs and gave an $\tilde{O}(n)$ query algorithm.

2 Technical Overview

In this section, we give a technical overview of our approach to proving a polynomial lower bound on the rounds of adaptivity. We start by describing the Balkanski-Singer [BS20] framework for proving rounds-of-adaptivity lower bounds as it serves as a starting point for our work. Our presentation will first briefly highlight why the approach taken in [BS20] cannot yield better than a logarithmic lower bound on the rounds of adaptivity and then describe the approach we take to sidestep the logarithmic bottleneck.

The Lower Bound Framework. Balkanski and Singer [BS20] consider a class of submodular functions which we call ***partition submodular functions***. Given a partition $\mathcal{P} = (P_1, \ldots, P_r)$ of the universe U, a set function is partition submodular if its value at a subset S depends only on the *cardinalities* of the number of elements it contains from each part. That is, $f_{\mathcal{P}}(S) = h(|S \cap P_1|, \ldots, |S \cap P_r|)$ for some function h whose domain is the set of r-dimensional non-negative integer vectors. The lower bound framework dictates the following three conditions on the functions h and the resulting partition submodular function $f_{\mathcal{P}}$.

(P1) The function h is defined such that $f_{\mathcal{P}}$ is *submodular*.

(P2) The last part P_r is the unique minimizer of f_P. We also assume $f_{\mathcal{P}}(\emptyset) = h(0, 0, \ldots, 0) = 0$, and thus $f_{\mathcal{P}}(P_r)$ is necessarily < 0.

(P3) For any $1 \leq i < r$, even if we know the identity of the parts $P_1, \ldots, P_{i-1}$, a single round of polynomially many queries tells us nothing about the identity of the parts P_{i+1} to P_r. More precisely, a random re-partitioning of the elements in $P_{i+1} \cup P_{i+2} \cup \cdots \cup P_r$ will, with high probability, give the same values to the polynomially many queries made in the current round.

(P3) is the key property for proving the lower bound. The function h is fixed. Let $\mathcal{P}$ be the uniform distribution over partitions with given sizes $|P_1|$ to $|P_r|$ which, along with h, induces a distribution over submodular functions. By Yao's lemma it suffices to show that any $(r-2)$-round deterministic algorithm making polynomially many queries fails to find the minimizer with any non-trivial probability. (P3) implies that after $(r-2)$ rounds of queries and obtaining their answers, the algorithm cannot distinguish between two functions f_P and $f_{P'}$ where the partitions P and P' agree on the first $(r-2)$ parts, but (P_{r-1}, P_r) and (P'_{r-1}, P'_r) are random re-partitioning of the elements of $P_{r-1} \cup P_r$. Since (P2) implies the minimizer of f_P is P_r and $f_{P'}$ is P'_r, and these will be different with high probability, any algorithm will make a mistake on one of them. The non-triviality is therefore in the construction of the "h" functions, and in particular for how large an r can one manage while maintaining (P1), (P2), and (P3).

The Balkanski-Singer Approach. For now, let us fix a *random* partition $P := (P_1, \ldots, P_r)$ of the universe U. Given a subset S, let $\mathbf{x} := (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r)$, where $\mathbf{x}_i := |S \cap P_i|$ be its *signature*. Before we describe Balkanski and Singer's construction approach, let us understand what one needs for establishing a condition like (P3). Consider the case $i = 1$, that is, the first round of queries. (P3) requires that the answers should not leak any information about $P_2, P_3, \ldots, P_r$.

Consider a query S. Since the partition P is random, we expect S's signature $\mathbf{x}$ to be random as well. More precisely, we expect $\frac{\mathbf{x}_i}{|P_i|}$ to be "roughly same" for all $i \in [r]$. Call such vectors *balanced*; we are deliberately not defining them precisely at this point. For (P3) to hold, we ***must*** have that $\partial_i h(\mathbf{x})$, the marginal increase in the function upon adding an element from P_i, is the same for all $2 \leq i \leq r$ for ***balanced*** vectors. Otherwise, the algorithm can distinguish between different parts. On the other hand, the marginals cannot be same for *all* vectors $\mathbf{x}$, as that would imply the sets P_2 to P_r have the same value, which would violate the constraint (P2) since P_r is the unique minimizer.

To orchestrate this, Balkanski and Singer use the idea of masking. All marginals $\partial_i h(\mathbf{x})$ are between $[-1, 1]$. In the first round, the masking is done via the first coordinate $\frac{\mathbf{x}_1}{|P_1|}$ of the signature. At a very high level, when $\frac{\mathbf{x}_1}{|P_1|}$ is "large", all the marginals $\partial_i h(\mathbf{x})$, for $2 \leq i \leq r$, take the value -1, while $\partial_1 h(\mathbf{x})$ takes the value 0. In plain English, if any set S contains a large fraction of elements from P_1, then *all* elements in $P_2 \cup \cdots \cup P_r$ have marginal -1; the preponderance of these P_1 elements masks all the other parts outs. Therefore, at the first round, after making polynomially many queries an algorithm can only perhaps detect P_1, but has no information about parts P_2 to P_r.

More generally one requires this kind of property to hold *recursively* as the algorithm discovers P_1, P_2, and so on in successive rounds. In any round i, if one considers a set S with $\frac{|S \cap P_i|}{|P_i|}$ "large" for some i, then for all elements e in parts P_j, $j > i$, the marginals are -1. In this way, they are able to maintain the property (P3). Of course, one has to be careful about what occurs when $|S \cap P_i|$'s are small, and the whole construction is rather technical, but this aspect described above is key to how they maintain indistinguishability.

A Logarithmic Bottleneck. Unfortunately, this powerful masking property is also a bottleneck. One can argue that the above construction ***cannot*** have $r = \omega(\log N)$. Consider the first round of queries. The Balkanski-Singer masking property asserts that if $\frac{|S \cap P_1|}{|P_1|}$ is "large" then *all* $e \in P_2 \cup \cdots \cup P_r$ give a marginal of -1. In particular, if one considers the the set $S = P_1$, then the marginal of all elements in $(P_2 \cup \cdots \cup P_r)$ to S is -1. This, along with submodularity, implies that $f_{\mathcal{P}}(U) \leq f(P_1) - (\sum_{i=2}^{r} |P_i|)$. Since U is not the minimizer, this needs to be $> f_{\mathcal{P}}(P_r)$, and since all marginals are in $[-1, +1]$, we get that

$$|P_1| \geq |P_2| + \cdots + |P_{r-1}|$$

That is, the first part is thus required to be bigger than the sum of the rest. And recursively, the second part is bigger than the sum of the rest. And so on. This implies[4] $r = O(\log N)$ and therefore the Balkanski-Singer masking idea **cannot** give a polynomial lower bound.

2.1 Ideas Behind Our Construction

Let us again focus on the first round of queries. In the Balkanski-Singer construction, whenever $\mathbf{x}_1$ is "large" ***irrespective*** of how the other $\mathbf{x}_i$'s look like, the marginals $\partial_i h(\mathbf{x}) = -1$ for $i \geq 2$. This strong masking property led to $|P_1|$ being much larger than the sum of the remaining parts so as to compensate for all the negative marginals coming from the elements in the other parts.

Our approach is not to set $\partial_i h(\mathbf{x})$ depending on just $\mathbf{x}_1$, but ***rather by looking at the whole suffix*** $\mathbf{x}_2 : \mathbf{x}_r$. More precisely, if $\mathbf{x}_1$ is "large" (say, even the whole part P_1), but all the rest are empty, even in that case we want *all* marginals $\partial_i h(\mathbf{x})$ to be in fact $+1$. Only when (almost) *all* coordinates $\mathbf{x}_i$ are "large", do we switch to $\partial_i h(\mathbf{x}) = -1$ for all $i \geq 2$. Therefore, in a sense, elements in any part contribute a negative marginal towards the function value, only after a significant number of elements from that part have already contributed positively, thus canceling out the negative conributions. This is what allows our construction to have all parts of equal size $n = N/r$, setting the stage for a polynomial lower bound.

Although deciding a marginal depending on the suffix may sound complicated, in the end our lower bound functions are simple to describe. Indeed, all marginals are in the set $\{-1, 0, +1\}$ and thus not only do we prove a polynomial lower bound on exact SFM, we also prove a $O(1/\varepsilon)$-lower bound even for ε-approximate SFM. Furthermore, as we explain below, our lower bounding functions are closely connected to rank functions of *nested* matroids, which are a special class of laminar matroids. Therefore, we also obtain lower bounds on the rounds-of-adaptivity of polynomial query matroid intersection algorithms with rank-oracle queries. In the rest of this subsection, we give more details on how the partition submodular functions are constructed. This discussion is still kept informal and is meant to help the reader understand the rationale behind the construction. The full formal details along with all the properties we need are deferred to Section 3, which the reader can feel free to skip to.

For our lower bound, we construct two partition submodular functions, $f_{\mathcal{P}}(S) = h(\mathbf{x})$ and $f^*_{\mathcal{P}} = h^*(\mathbf{x})$, where (a) the minimizer of f_P is the empty set and the minimizer of $f^*_{\mathcal{P}}$ is the set P_r (satisfying (P2)), and *both* these functions satisfy (P3) for $1 \leq i < r/2$, and furthermore, any, possibly randomized, algorithm distinguishing these functions and which uses only $o(N^{1/3}/\log^{1/3} N)$ rounds of adaptivity must make super-polynomial number of queries in some round. It is easier to understand the functions h and h^* via their marginals. Here are the properties we desire from these marginal functions.

[4] It is not easy to even orchestrate a $\Omega(\log N)$ lower bound this way. The masking functions that Balkanski-Singer constructed needs to be quite delicate to preserve submodularity, and in the end, the sets P_1 is in fact r times bigger than the rest. This leads to their $\Omega(\log N/\log\log N)$ lower bound.

- (Submodularity.) Both function's marginals should be monotonically decreasing. Thus, once $\partial_j h(\mathbf{x})$ or $\partial_j h^*(\mathbf{x})$ becomes -1, they should stay -1 for all $\mathbf{y}$ "larger" than $\mathbf{x}$.
- (Unique Minima.) The part P_r should be the unique minimizer for h^*. This restricts how often $\partial_j h^*(\mathbf{x})$ can be -1 when $j \neq r$. This is in tension with the previous requirement.
- (Suffix Indistinguishability.) For $i \leq r/2$ and for any $\mathbf{x}$ which is *i-balanced*, that is, $\mathbf{x}_i \approx \mathbf{x}_{i+1} \approx \cdots \approx \mathbf{x}_r$, we need that $\partial_j h^*(\mathbf{x})$ and $\partial_j h(\mathbf{x})$ for such $\mathbf{x}$'s should be the ***same*** for all $i+1 \leq j \leq r$. This is what we call suffix indistinguishability. This would also imply h and h^* would give the same values on all queried points with high probability.

At any point $\mathbf{x}$, let us first describe the r marginals $\partial_i h(\mathbf{x})$ for $1 \leq i \leq r$. As mentioned above, the marginals will be in the set $\{-1, 0, +1\}$. It is best to think of this procedure constructively as an algorithm. Initially, all the r marginals are set to $+1$. Next, we select up to two coordinates a and b in $\{1, 2, \ldots, r\}$, which depend on the query point $\mathbf{x}$. Given these coordinates, we decrement *all* marginals $a \leq i \leq r$ and *all* marginals $b \leq i \leq r$ by 1. For instance, if $r = 6$ and we choose the coordinates $a = 2$ and $b = 5$ at some $\mathbf{x}$, then the marginals $(\partial_1 h(\mathbf{x}), \ldots, \partial_6 h(\mathbf{x}))$ are $(1, 0, 0, 0, -1, -1)$. The 5th and 6th coordinate decrement twice and thus go from $+1$ to -1, while the 2nd, 3rd, and 4th coordinate only decrement once and thus go from $+1$ to 0. The first coordinate is never decremented in this example. Note that the vector of marginals when considered from 1 to r is always in decreasing order.

The crux of the construction is, therefore, in the choice of the a and the b at a certain point $\mathbf{x}$. These will clearly depend on $\mathbf{x}$, but how? Submodularity tells us that if we move from $\mathbf{x}$ to $\mathbf{y} = \mathbf{x} + \mathbf{e}_i$, then the a's and the b's should only *move left*, that is, become smaller; that would ensure decreasing marginals. This in turn implies that a and b should be defined by the ***suffix sums*** at $\mathbf{x}$. More precisely, if we decide to choose a and b as the coordinates which *maximize* some function $\phi(\cdot)$ which depends on the suffix sums $\sum_{i \geq t} \mathbf{x}_i$, t ranging from 1 to r, then increasing a coordinate can only move a's and b's to the left. This is precisely what we do, and now the crux shifts to the choice of this function $\phi(\cdot)$.

Consider an i-balanced vector $\mathbf{x}$. We need that when all the coordinates are "large", then the marginals of h^* should be -1; otherwise, P_r would not be the minimizer. Since h and h^* should be indistinguishable, the same should be true for h. On the other hand, when all the coordinates are "small", most marginals of both function should be $+1$, otherwise U would be the minimizer. In sum, when the coordinates of $\mathbf{x}$ are "large", we should have the a and b to the left, close to 1; this would make most marginals -1. And when they are small, a and b should be towards the right; this would make most marginals $+1$. This motivates the following rule that we formalize in the next section : we define r different functions (called $\ell_t(\mathbf{x})$ for $1 \leq t \leq r$) where the tth function $\ell_t(\mathbf{x})$ is the sum of $(\mathbf{x}_i - \tau)$ over all coordinates $t \leq i \leq r$ where τ is a "threshold" which is "close" to $n/2$. Here n is the size of each part $|P_i|$. After taking the sum over these coordinates, we further subtract an"offset" γ. In sum, the functions look like $\ell_t(\mathbf{x}) := (\sum_{i=t}^{r} (\mathbf{x}_i - \tau)) - \gamma$.

We choose a (respectively b) to be the ***odd*** (respectively, ***even***) coordinate t with the largest $\ell_t(\mathbf{x})$, *ignoring* them if this largest value is negative. That is, if all odd $\ell_t(\mathbf{x})$'s are negative, a is undefined; if all even $\ell_t(\mathbf{x})$'s are negative, b is undefined. Note that if both a and b are undefined, all marginals $\partial_i h(\mathbf{x})$ are $+1$; if one of them is undefined, then the marginals $\partial_i h(\mathbf{x})$ are $\{+1, 0\}$. Indeed, the function h which achieves such marginals can be succinctly stated as

$$h(\mathbf{x}) = \|\mathbf{x}\|_1 - \left(\max(0, \max_{a:\text{odd}} \ell_a(\mathbf{x})) + \max(0, \max_{b:\text{even}} \ell_b(\mathbf{x})) \right)$$

To see why $\mathbf{x}$ satisfies suffix indistinguishability, consider a balanced vector $\mathbf{x}$ with $\mathbf{x}_1 \approx \mathbf{x}_2 \cdots \approx \mathbf{x}_r$. If all of these entries $\mathbf{x}_i \gg \frac{n}{2}$ for all i, then note that the odd/even arg-maximizers are precisely $\{1, 2\}$. Thus, the marginals $\partial_i h(\mathbf{x})$'s are $(0,-1,-1,\ldots,-1)$. On the other hand if all $\mathbf{x}_i \ll \frac{n}{2}$, then due to our choice $\tau \approx \frac{n}{2}$, all $\ell_t(\mathbf{x})$'s will be negative, and thus $\{a,b\}$ will be ignored, implying that the marginals $\partial_i h(\mathbf{x})$ will be $(+1,+1,\ldots,+1)$. In either case, the marginals $\partial_i h(\mathbf{x})$ for $i \geq 2$ are the same, implying Suffix Indistinguishability . In reality, we must allow a wiggle room of "few standard deviations" in the $\approx$ between the $\mathbf{x}_i$'s since even a random set would exhibit such a behavior. To account for this, the same wiggle room needs to provided in the threshold τ and also in the offset γ. More precisely, we need to choose $\tau = \frac{n}{2} - g$, where $g \approx \tilde{\Theta}(\sqrt{n})$, and choose $\gamma \approx g \cdot r$.

Indeed the fact that this gap $g = \tilde{\Theta}(\sqrt{n})$ also is the reason why our construction cannot get better than $N^{1/3}$ lower bound. If we take the set $S = U = P_1 \cup \cdots \cup P_r$, that is, the signature $\mathbf{x} = \mathbf{n} = (n,n,\ldots,n)$, then one can evaluate $h(\mathbf{n}) = \frac{n}{2} - \tilde{\Theta}(r\sqrt{n})$. If we want $f(U) > 0$, we must have $n > \Theta(r\sqrt{n})$, implying $r = \tilde{O}(\sqrt{n})$. Since $N = nr$, this implies $r = \tilde{O}(N^{1/3})$.

The above was the description of the function h which is non-negative. The function h^* is simply the function h if $\mathbf{x}_r < \frac{n}{2} - \frac{g}{4}$, but if $\mathbf{x}_r \geq \frac{n}{2} - \frac{g}{4}$, the rth coordinate has marginal -1 irrespective of the other $\mathbf{x}_j$'s. This makes P_r become the minimizer of f^*_P with value $-\Theta(g)$. Since we only modify the behavior of the last index in going from h to h^*, in the beginning few rounds h and h^* behave similarly. Indeed, if $\mathbf{x}_r > \frac{n}{2} - \frac{g}{4}$, then any i-balanced vector for $i \leq r/2$, has half the coordinates $\geq \frac{n}{2} - O(g)$. The offset γ is chosen such that in this case h also has marginal -1 for the rth coordinate. Thus, h and h^* are indistinguishable in the first $r/2$ rounds. This, in turn, shows that if an algorithm runs for $< r/2$ rounds, then it cannot distinguish between these two functions, and therefore, cannot distinguish between the case when the minimum value is 0 and when the minimum value is $\approx -N^{1/3}$.

We end this informal description by stating how our results also imply lower bounds for *approximate* SFM. Since the marginals of our functions are $\{-1,0,+1\}$, the range of the function is $[-N,N]$. If we scale by a multiplicative factor N, we immediately get an $\tilde{\Omega}(1/\sqrt{\varepsilon})$-lower bound on the number of rounds needed to get an ε-additive approximation. However, we can boost this by a bit. The main idea is to not have P_r as the minimizer in h^*, but have the last $r/3$ parts together be the minimizer. This is done by simply having the last $r/3$ parts behave differently in h^*; and this boosts the minimum value to $\approx -\Theta(gr) \approx -N^{2/3}$. This implies an $\tilde{\Omega}(1/\varepsilon)$-lower bound on the depth required to obtain an ε-additive approximation.

Connection with Matroid Ranks and Matroid Intersection. The above description of $h(\mathbf{x})$ may seem a bit obscure. However, they are intimately connected to rank functions of matroids, in particular, nested matroids. Given a universe U, consider a nested family of subsets $\mathcal{C} := (U = C_1 \supseteq C_2 \supseteq \cdots \supseteq C_r)$. Furthermore, let each set C_i have a "capacity" cap_i. Then, the following family of subsets $\mathcal{I}_{\mathcal{C}} := \{I \subseteq U : |I \cap C_i| \leq \mathsf{cap}_i\}$ forms a matroid. Such matroids are called nested matroids, and they form a special class of laminar matroids. A nested matroid can also be described using a partition $P = (P_1,\ldots,P_r)$ where $P_r = C_r$ and $P_i := C_i \setminus C_{i+1}$ for all $1 \leq i < r$, and thresholds $\tau_r = \mathsf{cap}_r$ and $\tau_i := \mathsf{cap}_i - \mathsf{cap}_{i+1}$ for $1 \leq i < r$. It is not too hard to show (see Section 5) that the rank of the matroid is given by

$$\mathsf{rk}(S) := |S| - \max\left(0, \max_{1\leq a\leq r} \ell_a(S)\right), \quad \text{where} \quad \ell_t(S) := \sum_{i\geq t} (|S \cap P_i| - \tau_i)$$

The reader can see the connection between these rank functions and the partition submodular functions described above. Indeed, our partition submodular functions can be decomposed as $\mathsf{rk}_{\mathcal{M}_1}(S) + \mathsf{rk}_{\mathcal{M}_2}(U \setminus S)$ (plus a constant) for two nested matroids $\mathcal{M}_1$ and $\mathcal{M}_2$. Using Edmond's minimax relationship that the cardinality of the largest common independent set in $\mathcal{M}_1$ and $\mathcal{M}_2$ is precisely the minimum value of functions as above, our lower bounds for parallel SFM also prove a lower bound of $\tilde{\Omega}(N^{1/3})$ on the rounds of adaptivity required for efficient matroid intersection, even in the presence of rank oracle queries.

3 Description of our Lower Bound Functions

We begin by formally defining partition submodular functions and some properties of such functions. We then describe in detail the lower bound functions that we use in the proof of Theorem 1.

3.1 Partition Submodular Functions

Let U be a universe of elements and $\mathcal{P} = (P_1, \ldots, P_r)$ be a partition of the elements of U. Let $h : \mathbb{Z}^r_{\geq 0} \to \mathbb{R}$ be a function whose domain is the r-dimensional non-negative integer hypergrid. Given $(\mathcal{P}, h)$, one can define a set-function $f_{\mathcal{P}} : 2^U \to \mathbb{R}$ as follows:

$$f_{\mathcal{P}}(S) = h\left(|S \cap P_1|, \ldots, |S \cap P_r|\right) \tag{1}$$

In plain English, the value of $f_{\mathcal{P}}(S)$ is a function only of the *number* of elements of each part that is present in S. We say that $f_{\mathcal{P}}$ is induced by the partition P and h. A ***partition submodular function*** is a submodular function which is induced by some partition P and some hypergrid function h.

A function defined by (P, h) is submodular if and only if h satisfies the same decreasing marginal property as f. To make this precise, let us settle on some notation. Throughout the paper, for any integer k, we use $[k]$ to denote the set $\{0, 1, \ldots, k\}$. First, note that the domain of h is the r-dimensional hypergrid $[|P_1|] \times [|P_2|] \times \cdots \times [|P_r|]$. For brevity's sake, we call this $\mathbf{dom}(h)$. We use boldfaced letters like $\mathbf{x}, \mathbf{y}$ to denote points in $\mathbf{dom}(h)$. When we write $\mathbf{x} + \mathbf{y}$ we imply coordinate-wise sum. Given $i \in \{1, \ldots, r\}$, we use $\mathbf{e}_i$ to denote the r-dimensional vector having 1 at the ith coordinate and 0 everywhere else. The function h induces r different ***marginal*** functions defined as

$$\text{For } 1 \leq i \leq r, \quad \partial_i h(\mathbf{x}) := h(\mathbf{x} + \mathbf{e}_i) - h(\mathbf{x}) \tag{2}$$

The domain of $\partial_i h$ is $[|P_1|] \times [|P_2|] \times \cdots \times [|P_i| - 1] \times \cdots \times [|P_r|]$.

Definition 1. *We call a function $h : \mathbb{Z}^r \to \mathbb{R}$ defined over a integer hypergrid $\mathbf{dom}(h)$ (hypergrid) submodular if and only if for every $1 \leq i \leq r$, for every $\mathbf{x} \in \mathbf{dom}(h)$ with $\mathbf{x}_i < |P_i|$, and every $1 \leq j \leq r$, we have*

$$\partial_j h(\mathbf{x}) \geq \partial_j h(\mathbf{x} + \mathbf{e}_i) \tag{3}$$

Lemma 1. *A set function $f_{\mathcal{P}}$ defined by a partition P and hypergrid function h as in (1) is (partition) submodular if and only if h is (hypergrid) submodular.*

Proof. Let $A \subseteq U$ and let $\mathbf{x}$ be the r-dimensional integer vector with $\mathbf{x}_i := |A \cap P_i|$. Pick elements $e, e' \in U \setminus A$. Let $e \in P_i$ and $e' \in P_j$ for $1 \leq i, j \leq r$. Note that j could be the same as i. Then $f_{\mathcal{P}}$ is submodular is equivalent to $f_{\mathcal{P}}(A + e') - f_{\mathcal{P}}(A) \geq f_{\mathcal{P}}(A + e + e') - f_{\mathcal{P}}(A + e)$, which is equivalent to (3). □

The following lemma shows that minima of partition submodular functions can be assumed to take all or nothing of each part.

Lemma 2. *Let $f_{\mathcal{P}}$ be a partition submodular function induced by a partition $P = (P_1, \ldots, P_r)$ and hypergrid function h. Let O be a maximal by inclusion minimizer of f. Then, $O \cap P_i \neq \emptyset$ implies $O \cap P_i = P_i$.*

Proof. Let $\mathbf{x} \in \mathbf{dom}(h)$ be the vector induced by O, that is, $\mathbf{x}_i = |O \cap P_i|$ for all $1 \leq i \leq r$. For the sake of contradiction, assume $0 < \mathbf{x}_i < |P_i|$. Let e_1 and e_2 be two arbitrary elements in $O \cap P_i$ and $P_i \setminus O$ respectively. Since O is the minimizer, $f_{\mathcal{P}}(O) - f_{\mathcal{P}}(O - e_1) \leq 0$. Now note that the LHS is precisely $\partial_i h(\mathbf{x} - \mathbf{e}_i)$. And this is also equal to $f(O - e_1 + e_2) - f(O - e_1)$ and thus this is also ≤ 0. By submodularity, however, $f(O + e_2) - f(O) \leq f(O - e_1 + e_2) - f(O - e_1)$, and thus we obtain $f(O + e_2) \leq f(O)$ which contradicts that O was an inclusion-wise maximal minimizer. □

3.2 Suffix Functions

The lower bound functions we construct are partition submodular functions defined with respect to a partition $\mathcal{P} = (P_1, \ldots, P_r)$ of the universe U of N elements into r parts. The number of parts r is an odd integer whose value will be set to be $\tilde{\Theta}(N^{1/3})$. Each part P_i has the same size n, where n is an even positive integer such that $nr = N$. The hypergrid submodular function $h : [n]^r \to \mathbb{Z}$ which define the partition submodular function are themselves defined using *suffix* functions, which we describe below.

Let g be an integer which is divisible by 4 and which is $\tilde{\Theta}(\sqrt{n})$. That is, $\left(\frac{n}{2} - g\right)$ is "many standard deviations" away from $\frac{n}{2}$, and in particular, any random subset of an n-universe set has cardinality within $\pm g$ of the expected value with all but inverse polynomial probability. As described in the previous informal discussion, the following linear suffix functions play a key role in the description of the marginals. Define

$$\text{For any } 1 \leq t \leq r, \quad \ell_t(\mathbf{x}) := \sum_{s=t}^{r} \left(\mathbf{x}_s - \left(\frac{n}{2} - g\right)\right) - \frac{gr}{4} \tag{4}$$

Given $\mathbf{x}$, let $a := a(\mathbf{x}) \in [r]$ be the *odd*-coordinate $t \in [r]$ with the largest $\ell_t(\mathbf{x})$, breaking ties towards smaller indices in case of ties. Let $b := b(\mathbf{x}) \in [r]$ be the *even*-coordinate $t \in [r]$ with the largest $\ell_t(\mathbf{x})$, breaking ties towards smaller indices in case of ties. We call $\{a, b\}$ the largest odd-even index of $\mathbf{x}$.

Now we are ready to describe our lower bounding functions. First define the function $h : [n]^r \to \mathbb{Z}$ as follows

$$h(\mathbf{x}) = \|\mathbf{x}\|_1 - \Big(\max(0, \ell_a(\mathbf{x})) + \max(0, \ell_b(\mathbf{x}))\Big) \tag{5}$$

The above function contains the seed of the hardness, and satisfies (P1) and (P3). However, the above function, for the precise choice of g we will finally choose, will in fact be non-negative. To obtain the lower bounding functions which treats P_r specially, we define

$$h^*(\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{if } \mathbf{x}_r \leq \frac{n}{2} - \frac{g}{4} \\ h(\mathbf{x}_\downarrow) - \left(\mathbf{x}_r - \left(\frac{n}{2} - \frac{g}{4}\right)\right) & \text{otherwise} \end{cases} \text{where,} \, \mathbf{x}_\downarrow := \Big(\mathbf{x}_1, \ldots, \mathbf{x}_{r-1}, \min(\mathbf{x}_r, \frac{n}{2} - \frac{g}{4})\Big) \tag{6}$$

In Section 3.3, for completeness sake, we give a direct proof that both the functions, h and h^* are hypergrid submodular. However, as we show in Section 5, these functions arise as sum of rank functions of particular nested matroids, and thus give a more principled reason why these functions are submodular. In Section 3.4, we show that the function h is non-negative, while $h^*(0, 0, \ldots, 0, n)$ attains a negative value of $-g/2$. In Section 3.5, we show that i-balanced vectors, for $i < r/2$, cannot distinguish between h and h^*. This, in turn, is used in Section 4 to prove the lower bound for parallel SFM.

3.3 Submodularity

We first prove that $h : [n]^r \to \mathbb{Z}$ is submodular, and then use this to prove that $h^* : [n]^r \to \mathbb{Z}$ is submodular. We need to prove

Lemma 3. *Fix $\mathbf{x}$ and a coordinate $1 \leq i \leq r$. Let $\mathbf{y} := \mathbf{x} + \mathbf{e}_i$. Let j be any arbitrary coordinate. Then,*

$$\partial_j h(\mathbf{x}) \geq \partial_j h(\mathbf{y}) \tag{7}$$

The high-level reason why h is submodular is when one moves from $\mathbf{x}$ to $\mathbf{y} = \mathbf{x} + \mathbf{e}_i$, the odd-even index $\{a, b\}$ of $\mathbf{y}$ can only "move to the left", that is, become smaller. Formally,

Claim 1. *Let $\mathbf{x}$ be any point and let $\mathbf{y} := \mathbf{x} + \mathbf{e}_i$. Suppose a is the odd coordinate t with the largest $\ell_t(\mathbf{x})$ breaking ties towards smaller indices. Suppose a' is the odd coordinate t with the largest $\ell_t(\mathbf{y})$ breaking ties towards smaller indices. If $a' \neq a$, then (i) $a' \leq i < a$, and (ii) $\ell_{a'}(\mathbf{y}) = \ell_a(\mathbf{y})$. A similar statement is true for even coordinates.*

Proof. First from the definition, observation that $\ell_t(\mathbf{y}) = \ell_t(\mathbf{x})$ if $t > i$ and $\ell_t(\mathbf{y}) = \ell_t(\mathbf{x}) + 1$ if $t \leq i$. Thus, if $a' \neq a$, we must have that $a' \leq i < a$, establishing (i). Furthermore, since $a' < a$, we must have $\ell_a(\mathbf{x}) \geq \ell_{a'}(\mathbf{x}) + 1$ for otherwise a' would've been chosen with respect to $\mathbf{x}$. Since $\ell_{a'}(\mathbf{y}) \geq \ell_a(\mathbf{y})$, again by the observation of the first line, we establish (ii). □

To see how the claim helps in proving Lemma 3, it is instructive to first establish how the marginals of the function defined in (5) look like. To this end, define the following indicator functions. For any $1 \leq t \leq n$ and for any $1 \leq i \leq n$, define

$$\mathsf{C}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \ell_t(\mathbf{x}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \mathsf{C}_t^i(\mathbf{x}) = \mathsf{C}_t(\mathbf{x}) \cdot \mathbf{1}_{\{i \geq t\}}$$

where $\mathbf{1}_{\{i \geq t\}}$ is the indicator function taking the value 1 if $i \geq t$ and 0 otherwise. Using these notations, we can describe the r different marginals at $\mathbf{x}$ succinctly as

Lemma 4. *Fix $\mathbf{x}$ in the domain of h. Let $\{a, b\}$ be largest odd-even index of $\mathbf{x}$. Then,*

$$\forall 1 \leq i \leq r, \quad \partial_i h(\mathbf{x}) = 1 + \mathsf{C}_a^i(\mathbf{x}) + \mathsf{C}_b^i(\mathbf{x}) \tag{Marginals}$$

In plain English, given a point $\mathbf{x}$, one first finds the largest odd-even index $\{a, b\}$ of $\mathbf{x}$. If any of these function values are negative, throw them away from consideration: the suffixes aren't large enough. Next, given a coordinate i, the marginal $\partial_i h(\mathbf{x})$ depends on where i lies in respect to a and b (if they are still in consideration). If i is smaller than both, then the marginal is 1, if i is smaller than one, then the marginal is 0, if i is greater than or equal to both, the marginal is -1. Given this understanding of how the marginals look like, it is perhaps clear why Claim 1 implies submodularity : as $\{a, b\}$ move left the the marginal of any coordinate j can only decrease when one moves to $\mathbf{y}$.

Proof of Lemma 4. Fix an $\mathbf{x}$ and a coordinate i. Let $\mathbf{y} = \mathbf{x} + \mathbf{e}_i$. Let's consider $h(\mathbf{y}) - h(\mathbf{x})$ using (5), and then show it is precisely as asserted in (Marginals). First note that we can rewrite

$$h(\mathbf{x}) = \|\mathbf{x}\|_1 + \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x}) + \mathsf{C}_b(\mathbf{x})\ell_b(\mathbf{x}) \tag{8}$$

Consider the expression $\mathsf{C}_a(\mathbf{y})\ell_a(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x})$. If $i < a$, then $\ell_a(\mathbf{y}) = \ell_a(\mathbf{x})$, and thus $\mathsf{C}_a(\mathbf{y}) = \mathsf{C}_a(\mathbf{x})$, and thus the expression evaluates to 0. If $i \geq a$, then $\ell_a(\mathbf{y}) = \ell_a(\mathbf{x}) + 1$. For the expression to contribute anything non-zero, we must have $\ell_a(\mathbf{y}) \geq 1$ implying $\ell_a(\mathbf{x}) \geq 0$, or in other words, $\mathsf{C}_a(\mathbf{x}) = \mathsf{C}_a(\mathbf{y}) = -1$. And in that case, we get $\mathsf{C}_a(\mathbf{y})\ell_a(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x}) = -1$. To summarize,

$$\mathsf{C}_a(\mathbf{y})\ell_a(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x}) = \begin{cases} 0 & \text{if } i < a \text{ or if } \ell_a(\mathbf{x}) < 0 \text{, that is, } \mathsf{C}_a(\mathbf{x}) = 0 \\ -1 & \text{otherwise, that is, if } i \geq a \text{ and } \mathsf{C}_a(\mathbf{x}) = -1 \end{cases}$$

In other words,

$$\mathsf{C}_a(\mathbf{y})\ell_a(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x}) = \mathsf{C}^i_a(\mathbf{x}) \tag{9}$$

Now suppose $\{a', b'\}$ are the odd-even index of $\mathbf{y}$. The above discussion proves the claim when $\{a', b'\} = \{a, b\}$. Indeed, plugging (9) into (8), we get

$$h(\mathbf{y}) - h(\mathbf{x}) = \underbrace{(\|\mathbf{y}\|_1 - \|\mathbf{x}\|_1)}_{=1} + \mathsf{C}^i_a(\mathbf{x}) + \mathsf{C}^i_b(\mathbf{x})$$

A little more care is needed to take care of the case when $\{a', b'\} \neq \{a, b\}$. Suppose $a \neq a'$. Then, by Claim 1, we get that $a' < i \leq a$ and $\ell_{a'}(\mathbf{y}) = \ell_a(\mathbf{y})$. Thus, $\mathsf{C}_{a'}(\mathbf{y})\ell_{a'}(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x}) = \mathsf{C}_a(\mathbf{y})\ell_a(\mathbf{y}) - \mathsf{C}_a(\mathbf{x})\ell_a(\mathbf{x})$ and the proof follows as in the $a' = a$ case. The case $b' \neq b$ is similar. □

Proof of Lemma 3. Let $\{a_1, b_1\}$ be the odd-even index of $\mathbf{x}$. Let $\{a_2, b_2\}$ be the odd-even index of $\succ_{\mathbf{y}}$. From the definition of the marginals, what we need to show is

$$\mathsf{C}^j_{a_1}(\mathbf{x}) + \mathsf{C}^j_{b_1}(\mathbf{x}) \geq \mathsf{C}^j_{a_2}(\mathbf{y}) + \mathsf{C}^j_{b_2}(\mathbf{y}) \tag{10}$$

We will show this term by term, and focus on a_1, a_2. For any $1 \leq t \leq r$, observe that $\ell_t(\mathbf{y}) \geq \ell_t(\mathbf{x})$, and thus $\mathsf{C}_t(\mathbf{x}) \geq \mathsf{C}_t(\mathbf{y})$. Thus if $a_1 = a_2$, we are done.

If $a_1 \neq a_2$, then by Claim 1 $a_2 \leq i < a_1$ and $\ell_{a_2}(\mathbf{y}) = \ell_{a_1}(\mathbf{y}) \geq \ell_{a_1}(\mathbf{x})$. This implies $\mathsf{C}_{a_1}(\mathbf{x}) \geq \mathsf{C}_{a_2}(\mathbf{y})$. Since $a_2 < a_1$, we get that $\mathbf{1}_{\{j \geq a_2\}} \geq \mathbf{1}_{\{j \geq a_1\}}$. Since C is non-positive, we get $\mathsf{C}^j_{a_1}(\mathbf{x}) = \mathbf{1}_{\{j \geq a_1\}} \cdot \mathsf{C}_{a_1}(\mathbf{x}) \geq \mathbf{1}_{\{j \geq a_2\}} \cdot \mathsf{C}_{a_2}(\mathbf{y}) = \mathsf{C}^j_{a_2}(\mathbf{y})$. □

Lemma 5. *The function h^* as defined in (6) is submodular*

Proof. We recall the definition.

$$h^*(\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{if } \mathbf{x}_r \leq \frac{n}{2} - \frac{g}{4} \\ h(\mathbf{x}_\downarrow) - \left(\mathbf{x}_r - \left(\frac{n}{2} - \frac{g}{4}\right)\right) & \text{otherwise} \end{cases} \quad \text{where, } \mathbf{x}_\downarrow := \left(\mathbf{x}_1, \ldots, \mathbf{x}_{r-1}, \min(\mathbf{x}_r, \frac{n}{2} - \frac{g}{4})\right)$$

Observe,

- If $j \neq r$, then $\partial_j h^*(\mathbf{x}) = \partial_j h(\mathbf{x}_\downarrow)$.
- If $j = r$, then $\partial_r h^*(\mathbf{x}) = -1$ if $\mathbf{x}_r \geq \frac{n}{2} - \frac{g}{4}$, else $\partial_r h^*(\mathbf{x}) = \partial_r h(\mathbf{x})$.

Now pick $\mathbf{x} \in [n]^r$, $\mathbf{y} := \mathbf{x} + \mathbf{e}_i$. Since $\mathbf{x}_\downarrow$ is coordinate wise dominated by $\mathbf{y}_\downarrow$, we get that if $j \neq r$,

$$\partial_j h^*(\mathbf{x}) = \partial_j h(\mathbf{x}_\downarrow) \underbrace{\geq}_{\text{Lemma 3}} \partial_j h(\mathbf{y}_\downarrow) = \partial_j h^*(\mathbf{y})$$

If $j = r$, then either $\mathbf{y}_r \geq \frac{n}{2} - \frac{g}{4}$ and then $\partial_r h^*(\mathbf{x}) \geq \partial_r h^*(\mathbf{y})$ since the RHS is -1 and the LHS is at least that. Or, both $\mathbf{x}_r, \mathbf{y}_r < \frac{n}{2} - \frac{g}{4}$, and thus $\partial_r h^*(\mathbf{x}) = \partial_r h(\mathbf{x}) \underbrace{\geq}_{\text{Lemma 3}} \partial_r h(\mathbf{y}) = \partial_r h^*(\mathbf{y})$. □

3.4 Minimizers

Lemma 6. *Suppose the parameters n, g and r chosen such that $5gr \leq n$. Let $P = (P_1, \ldots, P_r)$ be any partition with $|P_i| = n$ for all i. Let $f_{\mathcal{P}}$ be the partition submodular function induced by $(P; h)$ and let f^*_P be the partition submodular function induced by $(P; h^*)$. Then, $\emptyset$ is the unique minimizer of $f_{\mathcal{P}}$ achieving the value 0, and*[a] $f^*_P(P_r) \leq -\frac{g}{2}$.

[a] In fact, one can show P_r is the unique minimizer of f^*_P, but that is not needed for the lower bound.

Proof. It is obvious that $f_{\mathcal{P}}(\emptyset) = f^*_P(\emptyset) = h(0, 0, \ldots, 0) = 0$. Next, observe that

$$f^*_P(P_r) = h^*(0, 0, \ldots, n) = h\left(0, 0, \ldots, 0, \frac{n}{2} - \frac{g}{4}\right) - \left(\frac{n}{2} + \frac{g}{4}\right)$$

If we let $\mathbf{z} = (0, 0, \ldots, 0, \frac{n}{2} - \frac{g}{4})$, then just using $h(\mathbf{z}) \leq \|\mathbf{z}\|_1$, we get $f^*_P(P_r) \leq -\frac{g}{2}$. Indeed, when $r \geq 3$, this is an equality since then $\ell_t(\mathbf{z}) \leq 0$ for all t and $h(\mathbf{z}) = \|\mathbf{z}\|_1$.

Next, we establish that if $5gr \leq n$, then the minimum value $f_{\mathcal{P}}$ takes is indeed 0. From Lemma 2, we know that the maximal minimizer of h is a vector $\mathbf{x}^*$ where $\mathbf{x}^*_i \in \{0, n\}$ for $1 \leq i \leq r$. Now fix an arbitrary $\mathbf{x}$ with $\mathbf{x}_i \in \{0, n\}$ which is different from the all zeros vector. We claim that $h(\mathbf{x}) > 0$, which would prove the lemma. Let the number of i's with $\mathbf{x}_i = n$ among the coordinates $\{1, 2, \ldots, r\}$ be $k \geq 1$.

Note that for any $t \leq r$,

$$\ell_t(\mathbf{x}) = \sum_{i \geq t} \left(\mathbf{x}_i - \left(\frac{n}{2} - g\right)\right) - \frac{gr}{4} \leq (k - t + 1) \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{4}$$

Therefore, if $\{a, b\}$ are the odd-even index of $\mathbf{x}$, we get that these ℓ_t values are at most $k \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{4}$ and $(k-1) \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{4}$, respectively, since a and b are distinct (and occurs when $a = 1$ and $b = 2$). Thus,

$$h^*(\mathbf{x}) = h(\mathbf{x}) > kn - \max\left(0, k \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{4}\right) - \max\left(0, (k-1) \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{4}\right)$$

If both the max terms in the expression for h turn out to be 0, then since $k \geq 1$, we get $h(\mathbf{x}) > n$. If only one of them is 0, then we get $h(\mathbf{x}) > k\left(\frac{n}{2} - g\right) + \frac{gr}{4} > 0$. Otherwise, we get that

$$h^*(\mathbf{x}) = h(\mathbf{x}) > kn - (2k - 1) \cdot \left(\frac{n}{2} + g\right) - \frac{gr}{2} \underbrace{\geq}_{\text{using } k \leq r} \frac{n}{2} - \frac{5gr}{2} + g \underbrace{>}_{\text{if } 5gr \leq n} 0$$

□

3.5 Suffix Indistinguishability

We now establish the key property about h and h^* which allows us to prove a polynomial lower bound on the rounds of adaptivity. To do so, we need a definition.

Definition 2. *For $1 \leq i < r$, a point $\mathbf{x} \in [n]^r$ is called **i-balanced** if $\mathbf{x}_i - \frac{g}{8} \leq \mathbf{x}_j \leq \mathbf{x}_i + \frac{g}{8}$ for all $j > i$.*

Suffix Indistinguishability asserts that two points $\mathbf{x}$ and $\mathbf{x}'$ which are i-balanced, have the same norm, and which agree on the first i coordinates have the same function value. More precisely,

Lemma 7 (Suffix Indistinguishability). *Let $i < \frac{r}{2}$. If $\mathbf{x}$ and $\mathbf{x}'$ are two i-balanced points with $\mathbf{x}_j = \mathbf{x}'_j$ for $j \leq i$ and $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$, then $h^*(\mathbf{x}) = h^*(\mathbf{x}') = h(\mathbf{x}) = h(\mathbf{x}')$.*

Proof. We first prove Suffix Indistinguishability for h, and then show that if $i < \frac{r}{2}$, then h and h^* take the same value on i-balanced points, which implies Suffix Indistinguishability for h^* as well (for $i < \frac{r}{2}$).

Claim 2. *Let $i \leq r - 2$. If $\mathbf{x}$ and $\mathbf{x}'$ are two i-balanced points with $\mathbf{x}_j = \mathbf{x}'_j$ for $j \leq i$ and $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$, then $h(\mathbf{x}) = h(\mathbf{x}')$.*

Proof. First note that for any $t \in \{1, 2, \ldots, i+1\}$, $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}')$; this follows from the fact that $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$ and that $\mathbf{x}$ and $\mathbf{x}'$ agree on the first i-coordinates.
Case 1: $\mathbf{x}_i = \mathbf{x}'_i < \frac{n}{2} - \frac{7g}{8}$. Since $\mathbf{x}$ and $\mathbf{x}'$ are both i-balanced, we have $\mathbf{x}_j, \mathbf{x}'_j < \frac{n}{2} - \frac{7g}{8} + \frac{g}{8} = \frac{n}{2} - \frac{3g}{4}$ for all $j \geq i$. This, in turn, implies that for any $t \geq i$, $\ell_t(\mathbf{x}), \ell_t(\mathbf{x}')$ are both $\leq \frac{gr}{4} - \frac{gr}{4} = 0$, since each summand in the definition (4) contributes at most $\frac{g}{4}$. So the largest odd (similarly, even) indexed $\ell_t(\mathbf{x})$ is either negative in which case it contributes 0 to $h(\mathbf{x})$, or $t \in \{1, \ldots, i+1\}$ in which case it subtracts $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}')$ from $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$. Furthermore, in the latter case, the same t is the maximize for $\mathbf{x}'$ as well. Therefore, in either case, $h(\mathbf{x}) = h(\mathbf{x}')$.
Case 2: $\mathbf{x}_i = \mathbf{x}'_i \geq \frac{n}{2} - \frac{7g}{8}$. Since $\mathbf{x}$ and $\mathbf{x}'$ are both i-balanced, we have $\mathbf{x}_j, \mathbf{x}'_j \geq \frac{n}{2} - g$ for all $j \geq i$. Thus each term in the summands of (4) is ≥ 0. This, in turn implies that both the odd and the even maximizers of $\ell_t(\mathbf{x}), \ell_t(\mathbf{x}')$, lie in $\{1, 2, \ldots, i+1\}$. Since $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}')$ for all such t's and $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$, we get that $h(\mathbf{x}) = h(\mathbf{x}')$. □

Next, we prove that when i is bounded way from r, for any i-balanced vector $\mathbf{x}$, we have $h^*(\mathbf{x}) = h(\mathbf{x})$. This lemma is useful to prove the indistinguishability of h^* and h.

Claim 3. *If $i < \frac{r}{2}$ and $\mathbf{x}$ is i-balanced, then $h^*(\mathbf{x}) = h(\mathbf{x})$.*

Proof. If $\mathbf{x}_r \leq \frac{n}{2} - \frac{g}{4}$, we have $h^*(\mathbf{x}) = h(\mathbf{x})$ by definition. So we only need to consider the case when $\mathbf{x}_r \geq \frac{n}{2} - \frac{g}{4}$. Let $k := \mathbf{x}_r - \left(\frac{n}{2} - \frac{g}{4}\right)$, by definition $\|\mathbf{x}\|_1 = \|\mathbf{x}_\downarrow\|_1 + k$ and $h^*(\mathbf{x}) = h(\mathbf{x}_\downarrow) - k$. For any $1 \leq t \leq r$, we have $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}_\downarrow) + k$, which means that the odd (respectively, even) index t with largest $\ell_t(\mathbf{x})$ is the same for $\ell_t(\mathbf{x}_\downarrow)$. That is the odd-even index $\{a, b\}$ is the same for $\mathbf{x}$ and $\mathbf{x}_\downarrow$.

Since $\mathbf{x}$ is i-balanced and $\mathbf{x}_r \geq \frac{n}{2} - \frac{g}{4}$, we have $\mathbf{x}_i \geq \frac{n}{2} - \frac{3g}{8}$, and thus, for any $j \geq i$, $\mathbf{x}_j \geq \frac{n}{2} - \frac{g}{2}$. Thus, all summands in (4) for $j \geq i$ give non-negative contribution. This means both a and b lie in $\{1, 2, \ldots, i+1\}$. On the other hand, both $\ell_i(\mathbf{x}_\downarrow)$ and $\ell_{i+1}(\mathbf{x}_\downarrow)$ are at least $(r - i - 1)\frac{g}{2} - \frac{gr}{4} \geq 0$ since $i \leq \frac{r}{2} - 1$. So both $\ell_a(\mathbf{x}_\downarrow)$ and $\ell_b(\mathbf{x}_\downarrow)$ are at least 0, which implies that both $\ell_a(\mathbf{x})$ and $\ell_b(\mathbf{x})$ are at least k (we only need they are ≥ 0). Therefore, we have

$$h^*(\mathbf{x}) = h(\mathbf{x}_\downarrow) - k = \left(\|\mathbf{x}_\downarrow\|_1 - \ell_a(\mathbf{x}_\downarrow) - \ell_b(\mathbf{x}_\downarrow)\right) - k = \|\mathbf{x}\|_1 - \ell_a(\mathbf{x}) - \ell_b(\mathbf{x}) = h(\mathbf{x}). \quad \square$$

Claim 2 and Claim 3 implies the Suffix Indistinguishability property of h^* and h. □

4 Parallel SFM Lower bound : Proof of Theorem 1

We now prove lower bounds on the rounds-of-adaptivity for algorithms which make $\leq N^c$ queries per round for some $1 \leq c \leq N^{1-\delta}$ where $\delta > 0$ is a constant. Let n be an even integer and g be an integer divisible by 4 such that $800\sqrt{cn \log n} \geq g \geq 200\sqrt{cn \log n}$. Let r be the largest odd integer such that $5gr \leq n$. Finally, let $N = nr$. Note that $g = \Theta(N^{1/3}(c\log N)^{2/3})$, $r = \Theta\left(\frac{N^{1/3}}{(c\log N)^{1/3}}\right)$, and $n = \Theta(N^{2/3}(c\log N)^{1/3})$. Since $c \leq N^{1-\delta}$, we get $n > cN^{2\delta/3} > c\log N$ and thus $g \geq 200c\log n$.

Remark 1. *It is perhaps worth reminding that we are allowing the algorithm to query $N^{N^{1-\delta}}$ sets. A reader may wonder with these many queries available won't one be able to find the minimizer by brute force even in a single round. In the "hard functions" we construct, the minimizer has $n \approx N^{1-\frac{\delta}{3}} \gg N^{1-\delta}$ elements. And thus $N^{N^{1-\delta}}$ queries would not be able to find the minimizer by enumeration over $\approx N^n$ sets.*

Let $\mathcal{P} = (P_1, \ldots, P_r)$ be a random equipartition of a universe U of N elements into parts of size n. Given a subset S, let the r-dimensional vector $\mathbf{x}$ defined as $\mathbf{x}_i := |S \cap P_i|$ be the signature of S with respect to $\mathcal{P}$. We say a query S is *i-balanced* with respect to $\mathcal{P}$ if the associated signature $\mathbf{x}$ is i-balanced. We use the following simple property of a random equipartition.

Lemma 8. *For any integer $i \in [1, \ldots, (r-1)]$, let $P_1, P_2, ..., P_{i-1}$ be a sequence of $(i-1)$ sets each of size n such that for $1 \leq j \leq (i-1)$, the set P_j is generated by choosing uniformly at random n elements from $U \setminus (P_1 \cup P_2 \cup ... P_{j-1})$. Let $S \subset U$ be any query that is chosen with possibly complete knowledge of $P_1, P_2, ..., P_{i-1}$. Then if we extend $P_1, P_2, ..., P_{i-1}$ to a uniformly at random equipartition $(P_1, ..., P_r)$ of U, with probability at least $1 - 1/n^{2c+3}$, the query S is i-balanced with respect to the partition $(P_1, P_2, ..., P_r)$; here the probability is taken over the choice of $P_i, P_{i+1}, ..., P_r$.*

Proof. Let $V = U \setminus (P_1 \cup P_2 \cup ... \cup P_{i-1})$. For $i \leq j \leq r$, let X_j be the random variable whose value equals $|S \cap P_j|$, and let $\mu = E[X_j] = |S \cap V|/(r - i + 1) \leq n$. To prove the assertion of the lemma, it is sufficient to show that with probability at least $1 - 1/n^{2c+3}$, we have $|X_j - \mu| \leq g/16$ for any j.

Note that each X_j is a sum of $|V|$ negatively correlated $0/1$ random variables. By Chernoff bound for negatively correlated random variables [DR98, IK10], the probability that X_j deviates from its expectation μ by more than $g/16$ is at most $2e^{\max\{-\frac{(g/16)^2}{3\mu}, -(g/16)\}} \leq 2e^{-10c\log n} \leq 1/n^{2c+4}$. By taking a union bound over all $i \leq j \leq r$, with probability at least $1 - 1/n^{2c+3}$, we have $|X_j - \mu| \leq g/16$ for all such j. □

To prove Theorem 1, we use Yao's minimax lemma. The distribution over hard functions is as follows. First, we sample a random equipartition $\mathcal{P}$ of the U into r parts each of size n. Given $\mathcal{P}$ and a subset S, let $f_{\mathcal{P}}(S) := h(\mathbf{x})$ and $f^*_{\mathcal{P}}(S) := h^*(\mathbf{x})$, where $\mathbf{x}$ is the signature of S with respect to $\mathcal{P}$. Select one of $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$ uniformly at random. This fixes the distribution over the functions, and this distribution is offered to a deterministic algorithm. We now prove that any s-round deterministic algorithm with $s < \frac{r}{2}$ fails to return the correct answer with probability $> 1/3$, and this would prove Theorem 1. In fact, we prove that with probability $\geq 1 - 1/n$, over the random equipartition $\mathcal{P}$, the deterministic algorithm cannot distinguish between $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$, that is, the answers to all the queries made by the algorithm is the same on both functions. This means that the deterministic algorithm errs with probability $\geq \frac{1}{2} \cdot (1 - \frac{1}{n}) > \frac{1}{3}$.

An s-round deterministic algorithm performs a collection of queries $\mathsf{Q}^{(\ell)}$ at every round $1 \leq \ell \leq s$ with $|\mathsf{Q}^{(\ell)}| \leq N^c \leq n^{2c}$. Let $\mathsf{Ans}^{(\ell)}$ denote the answers to the queries in $\mathsf{Q}^{(\ell)}$. The subsets queried in $\mathsf{Q}^{(\ell)}$ is a deterministic function of the answers given in $\mathsf{Ans}^{(1)}, \ldots, \mathsf{Ans}^{(\ell-1)}$. After receiving the answers to the sth round of queries, that is $\mathsf{Ans}^{(s)}$, the algorithm must return the minimizing set S. We now prove that when $\mathcal{P}$ is a random equipartition of U, then with probability $1 - \frac{1}{n}$, the answers $\mathsf{Ans}^{(\ell)}$ given to $\mathsf{Q}^{(\ell)}$ are the same for $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$, if $s < \frac{r}{2}$.

We view the process of generating the random equipartition as a game between an adversary and the algorithm where the adversary reveals the parts one-by-one. Specifically, the process of generating the random equipartition will be such that at the start of any round $\ell \in [1, \ldots, s]$, the adversary has only chosen and revealed to the algorithm the parts $P_1, P_2, ..., P_{\ell-1}$, and at this stage, $P_\ell, P_{\ell+1}, ..., P_r$ are equally likely to be any equipartition of $U \setminus (P_1 \cup P_2 \cup ... \cup P_{\ell-1})$ into $(r - \ell + 1)$ parts. By the end of round ℓ, the adversary has committed and revealed to the algorithm the part P_ℓ, and the game continues with one caveat. In each round, there will be a small probability (at most $1/n^2$) with which the adversary may "fail". This occurs at a round ℓ if any query made by the algorithm *on or before round* ℓ turns out to be not ℓ-balanced with respect to the sampled partition at round ℓ. In that case, the adversary reveals all remaining parts to the algorithm (consistent with the answers given thus far), and the game terminates in the current round ℓ itself with the algorithm winning the game (that is, the algorithm can distinguish between $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$). The probability of this failure event can be bound by $s/n^2 \leq 1/n$, summed over all rounds. In absence of this failure event, by Lemma 7, we know that the answers will be the same for $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$ at the end of the algorithm, concluding the proof. We now formally describe this process.

At the start of round 1, the adversary samples a uniformly at random equipartition of U, say, $\Gamma^{(1)} = (P_1^{(1)}, P_2^{(1)}, ..., P_r^{(1)})$. The algorithm reveals its set of queries for round 1, namely, $\mathsf{Q}^{(1)}$. The adversary answers all queries in $\mathsf{Q}^{(1)}$ in accordance with the partition $\Gamma^{(1)}$. By Lemma 8, since $|\mathsf{Q}^{(1)}| \leq n^{2c}$, every query in $\mathsf{Q}^{(1)}$ is 1-balanced with respect to the partition $\Gamma^{(1)}$, with probability at least $1 - 1/n^3$. If this event occurs, the adversary reveals $P_1^{(1)}$ to the algorithm, and continues to the next round. Otherwise, the adversary reveals the entire partition $\Gamma^{(1)}$ to the algorithm and the game terminates.

At the start of round 2, the adversary samples another uniformly at random equipartition of U, say, $\Gamma^{(2)} = (P_1^{(2)}, P_2^{(2)}, ..., P_r^{(2)})$ subject to the constraint $P_1^{(2)} = P_1^{(1)}$. Note that $\Gamma^{(2)}$ is a uniformly at random equipartition of U since $P_1^{(1)}$ was chosen uniformly at random. The algorithm reveals its set of queries for round 2, namely, $\mathsf{Q}^{(2)}$. Again by Lemma 8, we have that (i) every query in $\mathsf{Q}^{(1)}$ is 1-balanced with respect to the partition $\Gamma^{(2)}$, with probability at least $1 - 1/n^3$, and (ii) every query in $\mathsf{Q}^{(2)}$ is 2-balanced with respect to the partition $\Gamma^{(2)}$, with probability at least $1 - 1/n^3$. If this event occurs, the adversary answers all queries in $\mathsf{Q}^{(2)}$ in accordance with the partition $\Gamma^{(2)}$, and the game proceeds to the next round. The key insight here is that by Lemma 7, if a query $S \in \mathsf{Q}^{(i)}$ is i-balanced w.r.t. some partition $(P_1, ..., P_r)$, then the function value on the query S is completely determined by $P_1, P_2, ..., P_i$ and $|S|$, and does not require knowledge of $P_{i+1}, ..., P_r$. Furthermore, the value of $f_{\mathcal{P}}(S)$ *and* $f^*_{\mathcal{P}}(S)$ are the same. In other words, the function value on query S remains unchanged, for both f and f^*, if we replace $P := (P_1, ..., P_i, P_{i+1}, ..., P_r)$ with any other partition $P' := (P_1, ..., P_i, P'_{i+1}, .., P'_r)$ such that S remains i-balanced with respect to P'. So answers to all queries in $\mathsf{Q}^{(1)}$ are the same under both partitions $\Gamma^{(1)}$ and $\Gamma^{(2)}$. On the other hand, if either (i) or (ii) above does not occur, the adversary terminates the game and reveals the entire partition $\Gamma^{(1)}$ to the algorithm.

In general, if the game has successfully reached round $\ell \leq s$, then at the start of round ℓ, the adversary samples a uniformly at random equipartition of U, say, $\Gamma^{(\ell)} = (P_1^{(\ell)}, P_2^{(\ell)}, ..., P_r^{(\ell)})$ subject to the constraints $P_1^{(\ell)} = P_1^{(1)}, P_2^{(\ell)} = P_2^{(2)}, ..., P_{\ell-1}^{(\ell)} = P_{\ell-1}^{(\ell-1)}$. Once again, note that $\Gamma^{(\ell)}$ is a uniformly at random equipartition of U since $P_1^{(1)}$ was chosen uniformly at random, $P_2^{(2)}$ was chosen uniformly at random having fixed $P_1^{(1)}$, and so on. The algorithm now reveals its set of queries for round ℓ, namely, $\mathsf{Q}^{(\ell)}$. By Lemma 8, we have that for any fixed $i \in [1, \ldots, \ell]$, all queries in $\mathsf{Q}^{(i)}$ are i-balanced with respect to the partition $\Gamma^{(\ell)}$ with probability at least $1 - 1/n^3$ each. Thus with probability at least $1 - \ell/n^3$, for every $i \in [1, \ldots, \ell]$, all queries in $\mathsf{Q}^{(i)}$ are i-balanced with respect to the partition $\Gamma^{(\ell)}$. If this event occurs, the adversary answers all queries in $\mathsf{Q}^{(\ell)}$ with respect to the partition $\Gamma^{(\ell)}$, and once again, by Lemma 7, answers to all queries in $\mathsf{Q}^{(1)}, \mathsf{Q}^{(2)}, ..., \mathsf{Q}^{(\ell-1)}$ remain unchanged if we answer them using the partition $\Gamma^{(\ell)}$. The game then continues to the next round. Otherwise, with probability at most $\ell/n^3 \leq 1/n^2$, the game terminates and the adversary reveals the entire partition $\Gamma^{(\ell-1)}$ to the algorithm.

Summing up over all rounds 1 through $s \leq \frac{r}{2} - 1$, the probability that the game reaches round s is at least $1 - s/n^2 \geq 1 - 1/n$. This, in turn, implies that with probability $\geq 1 - \frac{1}{n}$, the random equipartition $\mathcal{P}$ satisfies the following property : all the queries in $\mathsf{Q}^{(i)}$ are i-balanced with respect to $\mathcal{P}$ for all $i \in [1..s]$. Now, since $s \leq \frac{r}{2}$, by Claim 3 we get that the answers $\mathsf{Ans}^{(1)}, \ldots, \mathsf{Ans}^{(s)}$ given to these queries are the same for $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$. Hence the algorithm cannot distinguish between these two cases. This completes the proof of Theorem 1.

4.1 Modification to boost gap : $\Omega(1/\varepsilon)$-lower bound for ε-approximate SFM

An inspection of the proof of Theorem 1 shows us that the minimum values of $f_{\mathcal{P}}$ and $f^*_{\mathcal{P}}$ are 0 and $-\frac{g}{2}$ for all $\mathcal{P}$'s (by Lemma 6). That is, any polynomial query algorithm making fewer than $\widetilde{\Omega}(N^{1/3})$ rounds of adaptivity cannot distinguish between the case when the minimum value is 0 and minimum value is $-g/2$. Since $g = \Theta(N^{1/3}(c \log N)^{2/3})$, we also rule out *additive* $O(N^{1/3})$-approximations for submodular functions whose range is $\{-N, -N+1, \ldots, N\}$. Scaling such that the range is $[-1, +1]$, we in fact obtain an $\widetilde{\Omega}(1/\sqrt{\varepsilon})$-dept lower bound to obtain ε-additive approximation algorithms.

In this section we show how a small modification leads to indistinguishability between functions with minimum value 0 and those with minimum value $-\Theta(N^{2/3})$ thus proving an $\widetilde{\Omega}(\frac{1}{\varepsilon})$ lower bound on the depth required for polynomial query ε-additive approximation algorithms for SFM.

The difference is in the definition of h^*; we redefine it such that the minimizer is not just P_r (or rather $(0, 0, \ldots, 0, n)$) but $P_{\frac{2r}{3}} \cup P_{\frac{2r}{3}+1} \cup \cdots \cup P_r$, and the minimum value becomes $-\frac{gr}{6} = -\Theta(N^{2/3})$. However, it still remains indistinguishable from h if the number of rounds is $< r/2$, and thus the proof of Theorem 1 carries word-to-word.

Define $\mathbf{x}_\downarrow := \Big(\mathbf{x}_1, \ldots, \mathbf{x}_{\frac{2r}{3}-1}, \min(\mathbf{x}_{\frac{2r}{3}}, \frac{n}{2} - \frac{g}{4}), \min(\mathbf{x}_{\frac{2r}{3}+1}, \frac{n}{2} - \frac{g}{4}) \ldots, \min(\mathbf{x}_r, \frac{n}{2} - \frac{g}{4})\Big)$. Then,

$$h^{**}(\mathbf{x}) = h(\mathbf{x}_\downarrow) - \sum_{i=\frac{2r}{3}}^{r} \max\left(0, \mathbf{x}_i - \left(\frac{n}{2} - \frac{g}{4}\right)\right) \tag{11}$$

Below we note the relevant changes. Let $f^{**}_{\mathcal{P}}$ be the partition submodular function induced by a partition $P = (P_1, \ldots, P_r)$ with $|P_i| = n$, and h^{**}.

- The proof of Lemma 5 generalizes to prove h^{**} is partition submodular. The two cases are $j < \frac{2r}{3}$ and $j \geq \frac{2r}{3}$. In the former case, $\partial_j h^{**}(\mathbf{x}) = \partial_j h(\mathbf{x}_\downarrow)$ and $\partial_j h^{**}(\mathbf{y}) = \partial_j h(\mathbf{y}_\downarrow)$, and submodularity follows from submodularity of h. If $j \geq \frac{2r}{3}$ and $\mathbf{y}_j \geq \frac{n}{2} - \frac{g}{4}$, then $\partial_j h^{**}(\mathbf{y}) = -1$ which implies it's $\leq \partial_j h^{**}(\mathbf{x})$. Otherwise, both $\mathbf{x}_j, \mathbf{y}_j < \frac{n}{2} - \frac{g}{4}$, and then submodularity again follows from that of h.

- In Lemma 6, we can now assert $f^{**}_{\mathcal{P}}(P_{\frac{2r}{3}} \cup \cdots \cup P_r) = -\frac{g}{2} \cdot \frac{r}{3} = -\frac{gr}{6}$.

- We assert that Lemma 7 still holds. To see this, note that the only changes are in the proof of Claim 3 (not the statement), and we sketch this below. Let $k := \sum_{i=\frac{2r}{3}}^{r} \max\Big(0, \mathbf{x}_{\frac{2r}{3}} - \big(\frac{n}{2} - \frac{g}{4}\big)\Big)$; we (still) have $\|\mathbf{x}\|_1 = \|\mathbf{x}_\downarrow\|_1 + k$ and $h^{**}(\mathbf{x}) = h(\mathbf{x}_\downarrow) - k$. Furthermore, for any $1 \leq t \leq \frac{2r}{3}$, we have $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}_\downarrow) + k$, and so if the odd-even index $\{a, b\}$ of $\mathbf{x}$ is in $\{1, \ldots, \frac{2r}{3}\}$, then $\{a, b\}$ is also the odd-even index for $\mathbf{x}_\downarrow$.

  Now, if $\mathbf{x}_t \leq \frac{n}{2} - \frac{g}{4}$ for all $\frac{2r}{3} \leq t \leq r$, then $\mathbf{x}_\downarrow = \mathbf{x}$ and $k = 0$ and $h^{**}(\mathbf{x}) = h(\mathbf{x})$. So, we may assume that some $\mathbf{x}_t > \frac{n}{2} - \frac{g}{4}$. And since $\mathbf{x}$ is i-balanced (for $i < t$), we get (just as in the previous proof) $\mathbf{x}_j \geq \frac{n}{2} - \frac{g}{2}$ for all $j \geq i$. And thus, the odd-even index $\{a, b\}$ of $\mathbf{x}$ lies in $\{1, 2, \ldots, i+1\}$. The rest of the proof now proceeds exactly as in Claim 3.

5 Suffix Functions, Nested Matroids, and Parallel Matroid Intersection

In this section we explain how our suffix functions, and as a result our partition submodular functions, arise in the context of matroid intersection. This is then used to prove Theorem 2 which states that any efficient matroid intersection algorithm, even with access to *rank* functions to the two matroids, must proceed in polynomially many rounds.

Matroids. A matroid $\mathcal{M} = (U, \mathcal{I})$ is a set-system over a universe U satisfying the following two axioms

- $I \in \mathcal{I}$ and $J \subseteq I$ implies $J \in \mathcal{I}$.
- For any $I, J \in \mathcal{I}$ with $|I| < |J|$, there exists $x \in J \setminus I$ such that $I + x \in \mathcal{I}$.

The sets in $\mathcal{I}$ are called *independent* sets of the matroid. A maximal independent set is called a *base*. It is well-known that all bases have the same cardinality. There are two usual oracles to access matroids. The first is the ***independence oracle*** which given a subset $S \subseteq U$ returns whether S is independent or not. The second stronger oracle, and we assume an algorithm has access to this, is the ***rank oracle*** which given a subset S returns $\mathsf{rk}_{\mathcal{M}}(S)$ which is the cardinality of the largest independent subset of S. It is well known that $\mathsf{rk}(S)$ is a submodular function whose marginals are in $\{0, +1\}$.

Nested Matroids. Let $\mathcal{C} = \{U = C_1 \supseteq C_2 \supseteq \cdots \supseteq C_r\}$ be a collection of nested subsets of the universe U. Let each set C_i have an associated non-negative integer capacity cap_i. Let $\vec{\mathsf{cap}} = (\mathsf{cap}_1, \ldots, \mathsf{cap}_r)$ be the capacity vector. Then $(\mathcal{C}, \vec{\mathsf{cap}})$ defines the following set family which is a matroid. Such matroids are called *nested matroids* (see, for example, [FO17]) and are a special class of laminar matroids.

$$\mathcal{M}_{\mathcal{C}} := \{I \subseteq U \; : \; |I \cap C_t| \leq \mathsf{cap}_t, \quad 1 \leq t \leq r\} \tag{Nested Matroids}$$

Given the nested family $\mathcal{C}$, there is an obvious associated partition $\mathcal{P} := (P_1, P_2, \ldots, P_r)$ of the universe U defined as $P_r := C_r$, the minimal subset in $\mathcal{C}$, and $P_j := C_j \setminus C_{j+1}$ for all $1 \leq j < r$. Similarly, we define "thresholds" for each part of the partition $\mathcal{P}$ as $\tau_r := \mathsf{cap}_r$, and $\tau_j := \mathsf{cap}_j - \mathsf{cap}_{j+1}$. We use $\vec{\tau}$ to denote the threshold vector $(\tau_1, \ldots, \tau_r)$.

Observe that these definitions are interchangeable : given $(\mathcal{P}, \vec{\tau})$ one gets the nested matroid defined by $(\mathcal{C}, \vec{\mathsf{cap}})$, where $C_j = \bigcup_{t \geq j} P_t$ for all $1 \leq j \leq r$, and $\mathsf{cap}_j = \sum_{t \geq j} \tau_t$.

Rank of a Nested Matroid. Given a nested matroid $\mathcal{M}$, let $\mathcal{P} = (P_1, \ldots, P_r)$ be the associated partition with thresholds τ_1 to τ_r. For simplicity, let us assume $|P_i| = n$ for all $1 \leq i \leq r$. Given a subset $S \subseteq U$, let $\mathbf{x} \in \mathbb{Z}^r_{\geq 0}$ be the signature of S where $\mathbf{x}_i := |P_i \cap S|$. Define

$$\text{for any } 1 \leq t \leq r, \;\; \ell_t(\mathbf{x}) := \sum_{s=t}^{r} (\mathbf{x}_s - \tau_s) \tag{12}$$

Note that a set S is independent if and only if $\ell_t(\mathbf{x}) \leq 0$ for all $1 \leq t \leq r$. Also note the connection with (4) when we set $\tau_1 = \cdots = \tau_{r-1} = \left(\frac{n}{2} - g\right)$ and $\tau_r = \left(\frac{n}{2} - g\right) + \frac{gr}{4}$. The next lemma shows how these functions define the rank of a nested matroid.

Lemma 9 (Rank of a Nested Matroid)**.**
Let $\mathcal{M}$ be a nested matroid defined by $(\mathcal{P} = (P_1, \ldots, P_r); \vec{\tau} = (\tau_1, \ldots, \tau_r))$ where $\tau_i \geq 0$ for all i.

Given any subset $S \subseteq U$ with signature $\mathbf{x}$, the rank of S is

$$\mathsf{rk}_{\mathcal{M}}(S) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1\le a\le r} \ell_a(\mathbf{x})\right)$$

where $\ell_t(\mathbf{x})$ is as defined in (12).

Proof. The rank $\mathsf{rk}_{\mathcal{M}}(S)$, which we also denote as $\mathsf{rk}_{\mathcal{M}}(\mathbf{x})$, is the cardinality of the largest independent subset of S. This value can be found by the following linear program, which is integral because the constraint matrix is totally unimodular.

$$\begin{array}{lll} \mathsf{rk}(\mathbf{x}) := \max \displaystyle\sum_{i=1}^r \mathbf{y}_i & & \\ \qquad\qquad \mathbf{y}_i \le \mathbf{x}_i, & \forall i \in [r] & \\ \qquad\qquad \displaystyle\sum_{i\ge t} \mathbf{y}_i \le \sum_{i\ge t} \tau_i, & \forall t \in [r] & \end{array} \quad \underbrace{=}_{\text{Duality}} \quad \begin{array}{ll} \min \displaystyle\sum_{i=1}^r \eta_i \mathbf{x}_i + \sum_{t=1}^r \mathbf{z}_t \cdot \left(\sum_{i\ge t} \tau_i\right) & \\ \displaystyle\sum_{t\le i} \mathbf{z}_t + \eta_i = 1, & \forall i \in [r] \\ \mathbf{z}, \eta \ge 0 & \end{array}$$

We do not impose non-negativity constraints on the $\mathbf{y}_i$ variables in the primal because the maximizing solution will indeed have non-negative $\mathbf{y}_i$'s. To see this, suppose $\mathbf{y}_j < 0$ and let $t \le j$ be the largest index such that $\sum_{i\ge t} \mathbf{y}_i = \sum_{i \ge t} \tau_i$. That is, the largest indexed constraint, among the ones containing $\mathbf{y}_j$, which is tight. There must be such a t for otherwise we could increase the objective by incrementing $\mathbf{y}_j$. Furthermore, $\mathbf{y}_t > 0$ for otherwise $\sum_{i\ge t+1} \mathbf{y}_i = \sum_{i\ge t+1} \tau_i$ and our t won't be largest; this argument uses $\tau_t \ge 0$. Now, increasing $\mathbf{y}_j$ and decreasing $\mathbf{y}_t$ by the same amount gives a feasible solution with the same optimum, and continuing the above procedure, we will get to a non-negative $\mathbf{y}$.

We can massage the dual as follows. Let $\mathsf{pref}_i(\mathbf{z}) := \sum_{t\le i} \mathbf{z}_t$. Thus, we can rewrite $\eta_i = 1 - \mathsf{pref}_i(\mathbf{z})$, and since $\eta_i \ge 0$, we get all $\mathsf{pref}_i(\mathbf{z})$'s, and in particular which is equivalent to, by the non-negativity of $\mathbf{z}$, the constraint $\|\mathbf{z}\|_1 \le 1$. Therefore, we can eliminate η's and get

$$\mathsf{rk}(\mathbf{x}) = \min_{\mathbf{z}:\|\mathbf{z}\|_1\le 1} \quad \sum_{i=1}^r \mathbf{x}_i \cdot (1 - \mathsf{pref}_i(\mathbf{z})) + \sum_{t=1}^r \mathbf{z}_t \left(\sum_{i\ge t} \tau_i\right)$$

Next, using the observation that $\sum_{t=1}^r \mathbf{z}_t \left(\sum_{i\ge t} \tau_i\right) = \sum_{i=1}^r \mathsf{pref}_i(\mathbf{z}) \cdot \tau_i$, we can further simplify to get

$$\mathsf{rk}(\mathbf{x}) = \min_{\mathbf{z}:\|\mathbf{z}\|_1\le 1} \quad \sum_{i=1}^r \mathbf{x}_i - \sum_{i=1}^r \mathsf{pref}_i(\mathbf{z}) \cdot (\mathbf{x}_i - \tau_i) = \|\mathbf{x}\|_1 - \max_{\mathbf{z}:\|\mathbf{z}\|_1\le 1} \sum_{t=1}^r \mathbf{z}_t \underbrace{\left(\sum_{i\ge t} (\mathbf{x}_i - \tau_i)\right)}_{\ell_t(\mathbf{x})}$$

The last summand $\max_{\mathbf{z}:\|\mathbf{z}\|_1\le 1} \sum_{t=1}^r \mathbf{z}_t \ell_t(\mathbf{x})$ is 0 if all $\ell_t(\mathbf{x}) \le 0$ (by setting $\mathbf{z} \equiv \mathbf{0}$), and otherwise, it is $\max_{1\le a\le t} \ell_a(\mathbf{x})$. This completes the proof. □

The reader should notice the similarity with (5). We will now make the connection more precise. Before doing so, we need another well known definition.

Duals of Matroids. Given a matroid $\mathcal{M}$, the dual matroid $\mathcal{M}^*$ is defined as follows

$$\mathcal{I}^* := \{S \subseteq U \ : \ U \setminus S \ \text{ contains a base of } \ \mathcal{M}\}$$

It is not too hard to check this is a matroid. The rank of any set in the dual matroid can be computed using the rank of the original matroid as follows.

Lemma 10 (e.g., Theorem 39.3 in [Sch03])**.** *Let $\mathcal{M}$ be a matroid with rank function rk. Let $\mathcal{M}^*$ be its dual with corresponding rank function rk^*. Then,*

$$\forall S \subseteq U: \quad \mathsf{rk}^*(S) = \mathsf{rk}(U \setminus S) + |S| - \mathsf{rk}(U)$$

It is not too hard to see that the dual of a nested matroid is another nested matroid whose nesting is from the "other end". More formally, one can prove the following.

Lemma 11. *Let $\mathcal{M}$ be a nested matroid defined by the partition $\mathcal{P} = (P_1, \ldots, P_r)$ and thresholds $\vec{\tau} := (\tau_1, \ldots, \tau_r)$. Then, $\mathcal{M}^*$ is another nested matroid defined by the* reverse *partition $\mathcal{P}' = (P_r, P_{r-1}, \ldots, P_2, P_1)$ and thresholds $\vec{\tau}' := (n_r - \tau_r, n_{r-1} - \tau_{r-1}, \ldots, n_1 - \tau_1)$, where $n_i := |P_i|$.*

Proof. Let S be a subset with signature $\mathbf{x}$ with respect to the original partition $\mathcal{P}$. S is independent in $\mathcal{M}^*$ if and only if $U \setminus S$ contains a base of $\mathcal{M}$. Equivalently, $\mathsf{rk}_{\mathcal{M}}(U \setminus S) = \mathsf{rk}_{\mathcal{M}}(U)$. Now, the latter is precisely $\|\mathbf{n}\|_1 - \ell_1(\mathbf{n})$ where $\mathbf{n} = (n_1, n_2, \ldots, n_r)$ is the signature of the universe U. Let $\mathbf{z}$ be the signature of $U \setminus S$; note that $\mathbf{z}_i = n_i - \mathbf{x}_i$. Thus, we get that S is independent in $\mathcal{M}^*$ if and only if

$$\|\mathbf{z}\|_1 - \max(0, \max_{1 \le a \le r} \ell_a(\mathbf{z})) = \|\mathbf{n}\|_1 - \ell_1(\mathbf{n}) \quad \underbrace{\Rightarrow}_{\text{Rearranging}} \quad \ell_1(\mathbf{z}) = \max(0, \max_{1 \le a \le r} \ell_a(\mathbf{z}))$$

$\ell_1(\mathbf{z})$ is largest suffix if and only if all the $(r-1)$ *prefix-sums* are non-negative, and $\ell_1(\mathbf{z}) \ge 0$ implies all prefix-sums are non-negative. Thus, we get

$$\forall 1 \le j \le r, \ \sum_{j \le t} (\mathbf{z}_j - \tau_j) \ge 0 \quad \equiv \quad \forall 1 \le j \le r, \ \sum_{j \le t} (\mathbf{x}_j - (n_j - \tau_j)) \le 0$$

which is precisely the signature of an independent set in the nested matroid defined by $(\mathcal{P}', \vec{\tau}')$. □

The Hard Matroid Intersection Set-up. Let $r = 2k+1$ be an odd number. Let $\mathcal{P} = (P_1, \ldots, P_r)$ be a partition with $|P_i| = n$. Each part will be associated with a parameter τ_i. These will be set to $\tau_1 = \cdots = \tau_{r-1} = \left(\frac{n}{2} - g\right)$ and $\tau_r = \left(\frac{n}{2} - g\right) + \frac{gr}{4}$, where g, $\frac{gr}{4}$ are as described in Section 4.

We define *three* coarsenings of this partition. The first is the *odd* coarsening containing $(k+1)$ parts defined as follows.

$$\mathcal{P}_{\mathsf{odd}} := (P_1 \cup P_2, \ \ P_3 \cup P_4, \ \ \ldots, \ \ P_{r-2} \cup P_{r-1}, \ \ P_r)$$

and the associated τ-values are, as expected, the sum of the relevant τ_j's. More precisely, they are $\vec{\tau}_{\mathsf{odd}} := (\tau_1 + \tau_2, \tau_3 + \tau_4, \ldots, \tau_{r-2} + \tau_{r-1}, \tau_r)$. Let $\mathcal{M}_{\mathsf{odd}}$ be the nested matroid defined by $(\mathcal{P}_{\mathsf{odd}}, \vec{\tau}_{\mathsf{odd}})$. The rank of $\mathcal{M}_{\mathsf{odd}}$ is given by Lemma 9 as follows; we only consider the odd indices since r is odd.

Claim 4. *Let $S \subseteq U$. Let $\mathbf{x}$ be the signature of S with respect to the $(2k+1)$-part partition $\mathcal{P}$. Then,*

$$\mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(\mathbf{x}) := \mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(S) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1 \le a \le r, \ a \text{ odd}} \ell_a(\mathbf{x})\right)$$

The second coarsening is the *even* coarsening containing $(k+1)$-parts defined as

$$\mathcal{P}_{\mathsf{even}} := (P_1, \;\; P_2 \cup P_3, \;\; P_4 \cup P_5, \;\; \ldots, \;\; P_{r-1} \cup P_r)$$

The associated τ-values are slightly different in that the first part is effectively "ignored". The vector of τ's are $\vec{\tau}_{\mathsf{even}} := (n, \tau_2 + \tau_3, \tau_4 + \tau_5, \ldots, \tau_{r-1} + \tau_r)$. Let $\mathcal{M}_{\mathsf{even}}$ be the corresponding nested matroid defined by $(\mathcal{P}_{\mathsf{even}}, \vec{\tau}_{\mathsf{even}})$. Note that any base of $\mathcal{M}_{\mathsf{even}}$ must contain the whole set P_1. Again using Lemma 9, the rank of this matroid is given as follows.

Claim 5. *Let $S \subseteq U$. Let $\mathbf{x}$ be the signature of S with respect to the $(2k+1)$-part partition $\mathcal{P}$. Then,*

$$\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) := \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(S) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1 \le a \le r, \; a \text{ even}} \ell_a(\mathbf{x})\right)$$

The reason the first part does not count is because $(\mathbf{x}_1 - n)$ is ≤ 0, and this cannot be the maximizer when we apply Lemma 9. And otherwise, it corresponds to an even index in the original partition.

Finally, the third coarsening is a refinement of $\mathcal{P}_{\mathsf{even}}$ where the last part $P_{r-1} \cup P_r$ is divided into two. That is,

$$\mathcal{P}'_{\mathsf{even}} := (P_1, \;\; P_2 \cup P_3, \;\; P_4 \cup P_5, \;\; \ldots, \;\; P_{r-1}, \;\; P_r)$$

The associated τ vector is $\vec{\tau}'_{\mathsf{even}} := (n, \tau_2 + \tau_3, \tau_4 + \tau_5, \ldots, \tau_{r-1} + \tau_r - \theta, \theta)$ for some parameter θ, which is set to $\left(\frac{n}{2} - \frac{g}{4}\right)$. Let $\mathcal{M}'_{\mathsf{even}}$ be the nested matroid defined by $(\mathcal{P}'_{\mathsf{even}}, \vec{\tau}'_{\mathsf{even}})$.

Claim 6. *Let $S \subseteq U$. Let $\mathbf{x}$ be the signature of S with respect to the $(2k+1)$-part partition $\mathcal{P}$. Then,*

$$\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) := \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(S) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow})$$

where, $\mathbf{x}_{\downarrow} = (\mathbf{x}_1, \ldots, \mathbf{x}_{r-1}, \min(\mathbf{x}_r, \theta))$.

Proof. First observe that for any t, $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}_{\downarrow}) + \max(0, (\mathbf{x}_r - \theta))$. Therefore, for any $\mathbf{x}$, the t maximizing $\ell_t(\mathbf{x})$ also is the one maximizing $\ell_t(\mathbf{x}_{\downarrow})$.

When computing $\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x})$ as $\|\mathbf{x}\| - \max(0, \max_a \ell_a(\mathbf{x}))$, the maximization over a is over all even indices and also r. This leads to two cases.

Case 1: This maximizer is at $a = r$, that is, $\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) = \|\mathbf{x}\|_1 - \max(0, \mathbf{x}_r - \theta)$. In that case, we have $\ell_a(\mathbf{x}) \le (\mathbf{x}_r - \theta)$ for all other a's. Which implies $\ell_a(\mathbf{x}_{\downarrow}) \le 0$. Therefore, $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow}) = \|\mathbf{x}_{\downarrow}\|_1 = \|\mathbf{x}\|_1 - \max(0, \mathbf{x}_r - \theta) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x})$.

Case 2: This maximizer at $a \ne r$, that is, $\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) = \|\mathbf{x}\|_1 - \max(0, \ell_a(\mathbf{x}))$ for some even a. Note that this a is also the maximizer when computing $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow})$. Therefore,

$$\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow}) = \|\mathbf{x}_{\downarrow}\|_1 - \max(0, \ell_a(\mathbf{x}_{\downarrow})) = \|\mathbf{x}_{\downarrow}\|_1 - \max(0, \ell_a(\mathbf{x}) - \max(0, \underbrace{(\mathbf{x}_r - \theta)}_{\ell_r(\mathbf{x})}))$$

If $\mathbf{x}_r \le \theta$, we get $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow}) = \|\mathbf{x}_{\downarrow}\|_1 - \max(0, \ell_a(\mathbf{x})) = \|\mathbf{x}\|_1 - \max(0, \ell_a(\mathbf{x})) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x})$, where the second equality follows because $\mathbf{x}_{\downarrow} = \mathbf{x}$ when $\mathbf{x}_r \le \theta$.

If $\mathbf{x}_r > \theta$, then $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow}) = \|\mathbf{x}_{\downarrow}\|_1 - (\ell_a(\mathbf{x}) - (\mathbf{x}_r - \theta))$ since $\ell_a(\mathbf{x}) \ge \ell_r(\mathbf{x}) \ge 0$ as a is the maximizer. Now observe that $\|\mathbf{x}_{\downarrow}\|_1 = \|\mathbf{x}\|_1 - (\mathbf{x}_r - \theta)$, and so $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_{\downarrow}) = \|\mathbf{x}\| - \ell_a(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x})$. □

Claim 7. $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(U)$.

Proof. Let $\mathbf{n}$ be the $(n, n, \ldots, n)$ vector. $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U) = \|\mathbf{n}\|_1 - \ell_2(\mathbf{n})$, and $\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(U) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{n}_\downarrow)$. This, in turn, is $\|\mathbf{n}_\downarrow\|_1 - \ell_2(\mathbf{n}_\downarrow) = (\|\mathbf{n}\| - (n - \theta)) - (\ell_2(\mathbf{n}) - (n - \theta)) = \|\mathbf{n}\|_1 - \ell_2(\mathbf{n})$. □

The following lemma connects matroid intersection with submodular function minimization for the functions described in Section 3.

Lemma 12. *The size of the largest cardinality independent set in $\mathcal{M}_{\mathsf{odd}} \cap \mathcal{M}^*_{\mathsf{even}}$ is precisely $C + \min_{S \subseteq U} f(S)$ where $C = |U| - \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U)$ and $f(S) = h(\mathbf{x})$ with*

$$h(\mathbf{x}) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1 \le a \le r,\ a\ odd} \ell_a(\mathbf{x})\right) - \max\left(0, \max_{1 \le a \le r,\ a\ even} \ell_a(\mathbf{x})\right)$$

and the size of the largest cardinality independent set in $\mathcal{M}_{\mathsf{odd}} \cap (\mathcal{M}'_{\mathsf{even}})^*$ is precisely $C + \min_{S \subseteq U} f^*(S)$ where $C = |U| - \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(U) = |U| - \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U)$ and $f^*(S) = h^*(\mathbf{x})$ with

$$h^*(\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \textit{if } \mathbf{x}_r \le \theta \\ h(\mathbf{x}_\downarrow) - (\mathbf{x}_r - \theta) & \textit{otherwise} \end{cases} \quad \textit{where}, \mathbf{x}_\downarrow := (\mathbf{x}_1, \ldots, \mathbf{x}_{r-1}, \min(\mathbf{x}_r, \theta))$$

Proof. From Edmond's theorem [Edm70], we know that for any two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, one has

$$\max_{I \in \mathcal{M}_1 \cap \mathcal{M}_2} |I| \;=\; \min_{S \subseteq U} \left(\mathsf{rk}_{\mathcal{M}_1}(S) + \mathsf{rk}_{\mathcal{M}_2}(U \setminus S)\right)$$

Fix a set S with signature $\mathbf{x}$ with respect to the $(2k+1)$-part partition $\mathcal{P}$. By Claim 4, we have $\mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(S) = \mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(\mathbf{x}) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{odd}} \ell_a(\mathbf{x})\right)$. By Lemma 10, we have $\mathsf{rk}_{\mathcal{M}^*_{\mathsf{even}}}(U \setminus S) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(S) + |U| - \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U) - |S| = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) + C - \|\mathbf{x}\|_1$. By Claim 5, we have $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(S) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \|\mathbf{x}\|_1 - \max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{even}} \ell_a(\mathbf{x})\right)$. And thus,

$$\mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(S) + \mathsf{rk}_{\mathcal{M}^*_{\mathsf{even}}}(U \setminus S) = C + h(\mathbf{x})$$

Similarly, by Lemma 10, we have $\mathsf{rk}_{(\mathcal{M}'_{\mathsf{even}})^*}(U \setminus S) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(S) + |U| - \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(U) - |S| = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) + C - \|\mathbf{x}\|_1$. By Claim 6, the RHS equals $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}_\downarrow) + C - \|\mathbf{x}\|_1$. And so,

$$\mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(S) + \mathsf{rk}_{(\mathcal{M}'_{\mathsf{even}})^*}(U \setminus S) = C + \|\mathbf{x}_\downarrow\|_1 - \max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{odd}} \ell_a(\mathbf{x})\right) - \max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{even}} \ell_a(\mathbf{x}_\downarrow)\right)$$

When $\mathbf{x}_r \le \theta$, the RHS is $C + h(\mathbf{x})$. When $\mathbf{x}_r > \theta$, we have $\ell_t(\mathbf{x}) = \ell_t(\mathbf{x}_\downarrow) + (\mathbf{x}_r - \theta)$ for all t, and as before, one can argue that $\max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{odd}} \ell_a(\mathbf{x})\right) = \max\left(0, \max_{1 \le a \le r,\ a\ \mathrm{odd}} \ell_a(\mathbf{x}_\downarrow)\right) + (\mathbf{x}_r - \theta)$. Which implies the RHS is $C + h(\mathbf{x}_\downarrow) - (\mathbf{x}_r - \theta)$. In sum, the RHS is $C + h^*(\mathbf{x})$. □

An Illustration. It is perhaps instructive to illustrate the difference in the two situations described in Lemma 12 with a concrete example which directly describes why the largest cardinality common independent sets are different in the two different cases. Take $r = 3$. Fix a partition (P_1, P_2, P_3) with each part having n elements each, and the size of the universe is $3n$. The τ values are $(\frac{n}{2} - g, \frac{n}{2} - g, \frac{n}{2} - 0.25g)$.

Let us understand what $\mathcal{M}_{\mathsf{odd}}$ is in this case. This is generated by $(P_1 \cup P_2, P_3)$ and the threshold vector $(n - 2g, \frac{n}{2} - 0.25g)$. So, a subset I is independent in $\mathcal{M}_{\mathsf{odd}}$ iff (a) it contains $\le \frac{n}{2} - 0.25g$ elements from P_3, and (b) $\le \frac{3n}{2} - 2.25g$ elements overall.

Similarly, the matroid $\mathcal{M}_{\mathsf{even}}$ is generated by $(P_1, P_2 \cup P_3)$ with the threshold vector $(n, n - 1.25g)$. We are interested in its dual, which is also a nested matroid which, by Lemma 11 is generated by the partition

$(P_2 \cup P_3, P_1)$ with thresholds $(n + 1.25g, 0)$. That is, a subset I is independent in $\mathcal{M}^*_{\mathsf{even}}$ iff (a) it contains 0 elements from P_1, and (b) $\leq n + 1.25g$ elements overall.

Notice that any set I^* which contains $\frac{n}{2} - 0.25g$ elements from P_3, $\frac{n}{2} + 1.5g$ elements from P_2, and 0 elements from P_1 is a ***base*** of $\mathcal{M}_{\mathsf{even}}$ which is independent in $\mathcal{M}_{\mathsf{odd}}$. All that is needed is that $1.5g \leq \frac{n}{2}$ so that there are enough items in P_2 to pick from.

Finally, let us consider the matroid $(\mathcal{M}'_{\mathsf{even}})$ and its dual. The former is a nested matroid generated by (P_1, P_2, P_3) with thresholds $(n, \frac{n}{2} - g, \frac{n}{2} - 0.25g)$. Which, in turn, implies that its dual is a nested matroid generated by (P_3, P_2, P_1) with thresholds $(\frac{n}{2} + 0.25g, \frac{n}{2} + g, 0)$. That is, an independent set cannot contain more than $\frac{n}{2} + g$ elements from P_2, thus ruling out the I^* described in the previous paragraph. Indeed, since $\mathcal{M}_{\mathsf{odd}}$ forces at most $\frac{n}{2} - 0.25g$ elements from P_3, the largest common independent set in $\mathcal{M}_{\mathsf{odd}}$ and $(\mathcal{M}'_{\mathsf{even}})^*$ is at most of size $n + 0.75g$ elements. Which is exactly $-g/2$ less, as predicted by Lemma 12 and Lemma 6. Note, however, that the size of the largest independent set in $(\mathcal{M}'_{\mathsf{even}})^*$ is the same as that in $\mathcal{M}^*_{\mathsf{even}}$, that is $n + 2.75g$; that set picks more elements from P_3. It is the intersection with $\mathcal{M}_{\mathsf{odd}}$ which prevents picking such a base of $(\mathcal{M}'_{\mathsf{even}})^*$.

Proof of Theorem 2. To complete the proof of Theorem 2, we need one more thing. In SFM, we have access to evaluation oracle for the function. In particular, if $\mathbf{x}$ is the signature of a set S with respect to a partition, then we have access to $h(\mathbf{x})$. In the matroid intersection problem, we have access to the individual ranks of each matroid. Therefore, we need to establish suffix-indistinguishability for each of the individual ranks. Since the rank of the dual matroid can be simulated by the rank of the original matroid, the suffix indistinguishability of both matroids is established by the following lemma whose proof is very similar to that of Lemma 7.

Lemma 13. *A signature $\mathbf{x}$ (with respect to the original $(2k+1)$-part partition) is i-balanced if $\mathbf{x}_i - \frac{g}{8} \leq \mathbf{x}_j \leq \mathbf{x}_i + \frac{g}{8}$. Let $i < \frac{r}{2}$. If $\mathbf{x}$ and $\mathbf{x}'$ are two i-balanced points with $\mathbf{x}_j = \mathbf{x}'_j$ for $j \leq i$ and $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1$, then (a) $\mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}_{\mathsf{odd}}}(\mathbf{x}')$, and (b) $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}') = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}')$*

Proof. As in the proof of Lemma 7, we proceed in two claims. First, we claim that for any $i \leq r-2$, if $\mathbf{x}$ and $\mathbf{x}'$ are i-balanced, then $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}')$. If $\mathbf{x}_i = \mathbf{x}'_i < \frac{n}{2} - \frac{7g}{8}$, then just as in Claim 2, all $\mathbf{x}_j, \mathbf{x}'_j$, for $j \geq i$, are $\leq \frac{n}{2} - \frac{3g}{4}$, implying that the even-index with the largest $\ell_t(\cdot)$ must lie in $\{1, 2, \ldots, i+1\}$. And this, due to the premise of the lemma, implies (using Claim 5) $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}')$. A similar argument using odd-index and Claim 4 proves part (a).

The proof of the second and third equality in part(b) follows as in Claim 3. We have $\theta = \frac{n}{2} - \frac{g}{4}$. If $\mathbf{x}_r \leq \theta$, then the two ranks are the same by Claim 6. If $\mathbf{x}_r > \theta$, then since $\mathbf{x}$ is i-balanced, all $\mathbf{x}_j \geq \frac{n}{2} - \frac{g}{2}$ for $j \geq i$. This means the even index with the largest $\ell_t(\mathbf{x})$ lies in $\{1, \ldots, i+1\}$. And since $i \leq r/2$, which implies that both $\ell_i(\mathbf{x}_\downarrow)$ and $\ell_{i+1}(\mathbf{x}_\downarrow)$ (we look at both for we don't know which is even, but one of them is) are ≥ 0. Therefore, $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \|\mathbf{x}\|_1 - \ell_a(\mathbf{x})$ for some even $a \leq i+1$, and $\mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x}) = \|\mathbf{x}_\downarrow\|_1 - \ell_a(\mathbf{x}_\downarrow)$ for the same a. Since $a \leq i+1$, we get that $\|\mathbf{x}_\downarrow\|_1 = \|\mathbf{x}\|_1 - k$ and $\ell_a(\mathbf{x}_\downarrow) = \ell_a(\mathbf{x}) - k$, where $k = \mathbf{x}_r - \theta$. In sum, we get $\mathsf{rk}_{\mathcal{M}_{\mathsf{even}}}(\mathbf{x}) = \mathsf{rk}_{\mathcal{M}'_{\mathsf{even}}}(\mathbf{x})$, and this, together with the previous paragraph, implies part (b). □

The proof of Theorem 2 then follows almost word-to-word as the proof of Theorem 1. The hard distributions over the pairs of matroids are as follows. First one samples a random equipartition P of U into $(2k+1)$ parts. Given P, the "odd" matroid $\mathcal{M}_{\mathsf{odd}}$ is one nested matroid. The other nested matroid is either $\mathcal{M}^*_{\mathsf{even}}$ or $(\mathcal{M}'_{\mathsf{even}})^*$. Note that by Lemma 11, these duals are also nested matroids. We give the algorithm rank-oracle access to these two matroids. As in the proof of Theorem 1, armed with Lemma 13, one can show that for any s-round deterministic algorithm for $s \leq \frac{r}{2} - 1$, with probability $\geq 1 - \frac{1}{n}$, the answers given in the case of $(\mathcal{M}_{\mathsf{odd}}, \mathcal{M}^*_{\mathsf{even}})$ and the answers given in the case of $(\mathcal{M}_{\mathsf{odd}}, (\mathcal{M}'_{\mathsf{even}})^*)$ are exactly the same. Since the *sizes* of the largest common independent sets in both cases are different, one gets the proof of Theorem 2. □

6 Concluding Remarks

The main finding of this paper is that submodular function minimization and matroid intersection, two fundamental discrete optimization problems which have efficient algorithms, are not highly parallelizable in the oracle model. More precisely, if the access to the submodular function is via an evaluation oracle, or if the access to the matroids is via rank oracles, then any, possibly randomized, algorithm making at most $\mathrm{poly}(N)$ queries to these oracles must proceed in $\tilde{\Omega}(N^{1/3})$ rounds, where N is the number of elements in the universe the functions/matroids are defined on. It is an interesting question if the lower bound can be improved to $\tilde{\Omega}(N)$, or if there can be $o(N)$-round $\mathrm{poly}(N)$ query algorithms for either of these problems. As remarked in Section 2, our constructions have a bottleneck at $N^{1/3}$, and a new idea is needed if one wants to prove better lower bounds.

Figuring out what the query complexity of SFM and matroid intersection, regardless of adaptivity, is an intriguing question. Currently, the best known upper bounds on the query complexity for SFM is $\tilde{O}(N^2)$ [Jia21]. For matroid intersection, the best known upper bounds using rank-oracles is $\tilde{O}(N^{1.5})$ [CLS+19] and with independence oracles it is $\tilde{O}(N^{9/5})$ [BvdBMN21]. For both SFM and matroid intersection, even with independence oracles, the best known lower bounds [GPRW20, Har10] are only linear in N. At this juncture we mention that the submodular functions we construct in this paper can indeed be minimized in $O(N)$ queries. The main idea is that in the ith round one can find $O(g)$ elements from the ith part making only $O(gr)$ queries, and once these $O(g)$ elements are known, one can repeat the same for the next round. This gives an $O(gr^2) = O(N)$ query algorithm. Similar ideas also give $\tilde{O}(N)$-independence query algorithms for the matroid intersection problem, for the matroids we consider. One would need more ideas to obtain a super-linear lower bound.

We believe that *partition* submodular functions deserve further study in their own right. When $r = N$ they encompass every submodular function, and at the other extreme when $r = 1$, they contain the functions which are concave functions of the cardinality. In this sense, the number of parts r behaves as a measure of complexity of such functions. Can r-partition submodular functions be minimized in $O(N + \mathrm{poly}(r))$ or $O(N \cdot \mathrm{poly}(r))$ queries? Can partition submodular functions be minimized in $\mathrm{poly}(r)$ rounds, independent of N? We believe these questions are worthy of study.

References

[ACK19] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial pass lower bounds for graph streaming algorithms. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 265–276, 2019. 4

[AD71] Martin Aigner and Thomas A Dowling. Matching theory for combinatorial geometries. *Trans. Amer. Math. Soc.*, 158(1):231–245, 1971. 2

[ALS20] Brian Axelrod, Yang P. Liu, and Aaron Sidford. Near-optimal approximate discrete and continuous submodular function minimization. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 837–853, 2020. 1

[BFNS15] Niv Buchbinder, Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing (SICOMP)*, 44(5):1384–1402, 2015. 3

[BJL$^+$19] Sébastien Bubeck, Qijia Jiang, Yin-Tat Lee, Yuanzhi Li, and Aaron Sidford. Complexity of highly parallel non-smooth convex optimization. In *Adv. in Neu. Inf. Proc. Sys. (NeurIPS)*, pages 13900–13909, 2019. 4

[BK04] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9):1124 – 1137, 2004. 1

[BRS19] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 283–302, 2019. 1, 3

[BS17] Eric Balkanski and Yaron Singer. Minimizing a submodular function from samples. In *Adv. in Neu. Inf. Proc. Sys. (NeurIPS)*, pages 814–822, 2017. 2

[BS18] Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 1138–1151, 2018. 1, 3

[BS20] Eric Balkanski and Yaron Singer. A lower bound for parallel submodular minimization. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 130–139, 2020. 1, 3, 4

[BvdBMN21] Joakim Blikstad, Jan van den Brand, Sagnik Mukhopadhyay, and Danupon Nanongkai. Breaking the quadratic barrier for matroid intersection. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 421–432, 2021. 2, 24

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization viagraph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222 – 1239, 2001. 1

[CFK19] Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 102–113, 2019. 4

[CJK14] Deeparnab Chakrabarty, Prateek Jain, and Pravesh Kothari. Provable submodular minimization using Wolfe's algorithm. In *Adv. in Neu. Inf. Proc. Sys. (NeurIPS)*, pages 802–809, 2014. 1

[CLS$^+$19] Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, Sahil Singla, and Sam Chiu-wai Wong. Faster matroid intersection. In *Proc., IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019. To appear. 2, 24

[CLSW17] Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic submodular function minimization. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 1220–1231, 2017. 1

[CQ19a] Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 78–89, 2019. 1, 3

[CQ19b] Chandra Chekuri and Kent Quanrud. Submodular function maximization in parallel via the multilinear relaxation. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 303–322, 2019. 1, 3

[Cun85] William H. Cunningham. On submodular function minimization. *Combinatorica*, 5:185 – 192, 1985. 1

[Cun86] William H. Cunningham. Improved bounds for matroid partition and intersection algorithms. *SIAM Journal on Computing (SICOMP)*, 15(4):948–957, 1986. 2

[DBW12] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012. 4

[DR98] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998. 15

[DVZ18] Daniel Dadush, László A. Végh, and Giacomo Zambelli. Geometric rescaling algorithms for submodular function minimization. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 832–848, 2018. 1

[Edm70] J. Edmonds. Submodular Functions, Matroids, and Certain Polyhedra. In R. Guy, H. Hanam, and J. Schonheim, editors, *Combinatorial structures and their applications*, pages 69–85, New York, 1970. Gordon and Breach. 2, 22

[EN19] Alina Ene and Huy L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 274–282, 2019. 1, 3

[ENV19] Alina Ene, Huy L. Nguyen, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 90–101, 2019. 1, 3

[FMV11] Uriel Feige, Vahab Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing (SICOMP)*, 40(4):1133 – 1153, 2011. 4

[FMZ19] Matthew Fahrbach, Vahab Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 255–273, 2019. 3

[FO17] Tara Fife and James Oxley. Laminar matroids. *European Journal of Combinatorics*, 62:206–216, 2017. 18

[GGR22] Sumanta Ghosh, Rohit Gurjar, and Roshan Raj. A deterministic parallel reduction from weighted matroid intersection search to decision. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page to appear, 2022. 3

[GLLS81] Martin Grötschel, László Lovasz, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169 – 197, 1981. 1

[GPRW20] Andrei Graur, Tristan Pollner, Vidhya Ramaswamy, and S. Matthew Weinberg. New query lower bounds for submodular function minimization. In *Proc., Innovations in Theoretical Computer Science (ITCS)*, pages 64:1–64:16, 2020. 24

[GR20] Rohit Gurjar and Rajat Rathi. Linearly representable submodular functions: An algebraic algorithm for minimization. In *Proc., International Colloquium on Automata, Languages and Programming (ICALP)*, pages 61:1–61:15, 2020. 2

[Har10] Nicholas J. A. Harvey. Query lower bounds for matroid intersection. *RIMS Kokyuroku Bessatsu*, B23:81–105, 2010. 24

[IB13] Rishabh Iyer and Jeff A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. *Adv. in Neu. Inf. Proc. Sys. (NeurIPS)*, 2013. 1

[IFF01] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001. 1

[IJB13] Rishabh Iyer, Stefanie Jegelka, and Jeff A. Bilmes. Fast semidifferential-based submodular function optimization. In *Proc., International Conference on Machine Learning (ICML)*, pages 855–863, 2013. 1

[IK10] Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *Proc., International Workshop on Randomization and Computation (RANDOM)*, pages 617–631, 2010. 15

[IO09] Satoru Iwata and James B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1230–1237, 2009. 1

[Jia21] Haotian Jiang. Minimizing convex functions with integral minimizers. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 976–985, 2021. 1, 24

[KKT08] Pushmeet Kohli, M. Pawan Kumar, and Philip H. S. Torr. P3 and beyond: Move making algorithms for solving higher order functions. *IEEE Trans. Pattern Anal. and Machine Learning*, 31:1–8, 2008. 1

[KM97] David R. Karger and Rajeev Motwani. An NC algorithm for minimum cuts. *SIAM J. Comput.*, 26(1):255–272, 1997. 2

[KUW86] Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986. 2

[KUW88] Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel search. *J. Comput. System Sci.*, 36(2):225–253, 1988. 3

[Law75] Eugene L. Lawler. Matroid intersection algorithms. *Math. Programming*, 9(1):31–56, 1975. 2

[LJJ15] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Adv. in Neu. Inf. Proc. Sys. (NeurIPS)*, 2015. 1

[LLV20] Wenzheng Li, Paul Liu, and Jan Vondrák. A polynomial lower bound on adaptive complexity of submodular maximization. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 140–152, 2020. 1, 3

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-Wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. *Proc., IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1049–1065, 2015. 1

[MN20] Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 496–509, 2020. 4

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

[Ngu19] Huy L. Nguyen. A note on cunningham's algorithm for matroid intersection. *arXiv e-prints*, 2019. 2

[NSV94] H. Narayanan, Huzur Saran, and Vijay V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arborescences and edge-disjoint spanning trees. *SIAM J. Comput.*, 23(2):387–397, 1994. 2, 3

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

[NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Math. Programming*, 14(1):265–294, 1978. 3

[Orl09] James B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Math. Programming*, 118(2):237–251, 2009. 1

[RSW18] Aviad Rubinstein, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In *Proc., Innovations in Theoretical Computer Science (ITCS)*, pages 39:1–39:16, 2018. 4

[Sch00] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combin. Theory Ser. B*, 80(2):346–355, 2000. 1

[Sch03] Alexander Schrijver. *Combinatorial Optimization*. Springer, New York, 2003. 20

[Von13] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013. 3