Submodular Maximization beyond Non-negativity: Guarantees, Fast Algorithms, and Applications

Christopher Harshaw [1] **Moran Feldman** [2] **Justin Ward** [3] **Amin Karbasi** [4]

Abstract

It is generally believed that submodular functions—and the more general class of γ-weakly submodular functions—may only be optimized under the non-negativity assumption $f(S) \geq 0$. In this paper, we show that once the function is expressed as the difference $f = g - c$, where g is monotone, non-negative, and γ-weakly submodular and c is non-negative modular, then strong approximation guarantees may be obtained. We present an algorithm for maximizing $g - c$ under a k-cardinality constraint which produces a random feasible set S such that $\mathbb{E}\left[g(S) - c(S)\right] \geq (1 - e^{-\gamma} - \epsilon) g(OPT) - c(OPT)$, whose running time is $O(\frac{n}{\epsilon} \log^2 \frac{1}{\epsilon})$, independent of k. We extend these results to the unconstrained setting by describing an algorithm with the same approximation guarantees and faster $O(\frac{n}{\epsilon} \log \frac{1}{\epsilon})$ runtime. The main techniques underlying our algorithms are two-fold: the use of a surrogate objective which varies the relative importance between g and c throughout the algorithm, and a geometric sweep over possible γ values. Our algorithmic guarantees are complemented by a hardness result showing that no polynomial-time algorithm which accesses g through a value oracle can do better. We empirically demonstrate the success of our algorithms by applying them to experimental design on the Boston Housing dataset and directed vertex cover on the Email EU dataset.

[1]Department of Computer Science, Yale University, New Haven, USA [2]Department of Mathematics and Computer Science, Open University of Israel, Raanana, Israel [3]School of Mathematical Sciences, Queen Mary University of London, London, UK [4]Department of Electrical Engineering, Yale University, New Haven, USA. Correspondence to: Christopher Harshaw <christopher.harshaw@yale.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

1. Introduction

From summarization and recommendation to clustering and inference, many machine learning tasks are inherently discrete. Submodularity is an attractive property when designing discrete objective functions, as it encodes a natural diminishing returns condition and also comes with an extensive literature on optimization techniques. Submodular optimization techniques have been successfully applied in a wide variety of machine learning tasks, including sensor placement (Krause & Guestrin, 2005), document summarization (Lin & Bilmes, 2011), speech subset selection (Wei et al., 2013) influence maximization in social networks (Kempe et al., 2003), information gathering (Golovin & Krause, 2011), and graph-cut based image segmentation (Boykov et al., 2001; Jegelka & Bilmes, 2011), to name a few. However, in instances when the objective function is not submodular, existing techniques for submodular optimization many perform arbitrarily poorly, motivating the need to study broader function classes. While several notions of approximate submodularity have been studied, the class of γ-weakly submodular functions have (arguably) enjoyed the most practical success. For example, γ-weakly submodular optimization techniques have been used in feature selection (Das & Kempe, 2011; Khanna et al., 2017), anytime linear prediction (Hu et al., 2016), interpretation of deep neural networks (Elenberg et al., 2017), and high dimensional sparse regression (Elenberg et al., 2018).

Here, we study the constrained maximization problem

$$\max_{|S| \leq k} g(S) - c(S) \ , \tag{1}$$

where g is a non-negative monotone γ-weakly submodular function and c is a non-negative modular function. Problem (1) has various interpretations which may extend the current submodular framework to apply to more tasks in machine learning. For instance, the modular cost c may be added as a penalty to existing submodular maximization problems to encode a cost for each element. Such a penalty term may play the role of a regularizer or soft constraint in a model. When g models the revenue of some collection of products S and c models the cost of each item, then (1) corresponds to maximizing profits.

While Problem 1 has promising modeling potential, existing optimization techniques fail to provide nontrivial approximation guarantees. The main reason is that most existing techniques require the objective function to take only non-negative values, while $g(S) - c(S)$ may take both positive and negative values. Moreover, $g(S) - c(S)$ might be non-monotone, and thus, the definition of γ-weak submodularity does not even apply to it when $\gamma < 1$.

Our Contributions We provide several fast algorithms for solving Problem (1) as well as a matching hardness result and experimental validation of our methods. In particular,

1. **Algorithms.** In the case where γ is known, we provide a deterministic algorithm which uses $O(nk)$ function evaluations and returns a set S such that $g(S)-c(S) \geq (1 - e^{-\gamma})g(OPT) - c(OPT)$. If g is regarded as revenue and c as a cost, then this guarantee intuitively states that the algorithm will return a solution whose total profit is at least as much as would be obtained by paying the same cost as the optimal solution while gaining at least $(1 - e^{-\gamma})$ as much revenue. We extend this to a randomized variant which uses $O(n \log \frac{1}{\epsilon})$ function evaluations and has a similar approximation guarantee in expectation, but with an ϵ additive loss in the approximation factor. We also provide a randomized algorithm for the unconstrained setting (when $k = n$) which achieves the same $1 - e^{-\gamma}$ approximation factor in expectation using only $O(n)$ function evaluations. When γ is unknown, we give a meta-algorithm for guessing γ that loses a δ additive factor in the approximation ratio and increases the run time by a multiplicative $O(\frac{1}{\delta} \log \frac{1}{\delta})$ factor.
2. **Hardness of Approximation.** To complement our algorithms, we provide a matching hardness result which shows that no algorithm which makes polynomially many queries in the value oracle model may do better. To the best of our knowledge, this is the first hardness result of this kind for γ-weakly submodular functions.
3. **Experimental Evaluation.** We demonstrate the effectiveness of our algorithm on experimental design on the Boston Housing dataset and directed vertex cover on the Email EU dataset, both with costs.

Prior Work The celebrated result of Nemhauser et al. (1978) showed that the greedy algorithm achieves a $(1-1/e)$ approximation for maximizing a nonnegative monotone submodular function subject to a cardinality constraint. Das & Kempe (2011) showed the more general result that the greedy algorithm achieves a $(1 - e^{-\gamma})$ when g is γ-weakly submodular. At the same time, an extensive line of research has lead to the development of algorithms to handle non-monotone submodular objectives under more complicated constraints (see, e.g., (Buchbinder & Feldman, 2016; Chekuri et al., 2014; Ene & Nguyen, 2016; Feldman et al., 2017; Lee et al., 2010; Sviridenko, 2004)). The $(1 - 1/e)$ approximation was shown to be optimal in the value oracle model (Nemhauser & Wolsey, 1978), but until this work, no stronger hardness result was known for constrained γ-weakly submodular maximization. The problem of maximizing $g + \ell$ for non-negative monotone submodular g and an (arbitrary) modular function ℓ under cardinality constraints was first considered in (Sviridenko et al., 2017), who gave a randomized polynomial time algorithm which outputs a set S such that $g(S) + \ell(S) \geq (1-1/e)g(OPT) + \ell(OPT)$ where OPT is the optimal set. This approximation was shown to be optimal in the value oracle model via a reduction from submodular maximization with bounded curvature. However, the algorithm of Sviridenko et al. (2017) is of mainly theoretical interest, as it requires continuous optimization of the multilinear extension and an expensive routine to guess the contribution of OPT to the modular term, yielding it practically intractable. Feldman (2018) suggested using a surrogate objective that varies with time, and showed that this removes the need for the guessing step. However, the algorithm of (Feldman, 2018) still requires expensive sampling as it is based on the multilinear extension. Moreover, neither of these approaches can currently handle γ-weakly submodular functions, as optimization routines that go through their multilinear extensions have not yet been developed.

Organization The remainder of the paper is organized as follows. Preliminary definitions are given in Section 2. The algorithms we present for solving Problem (1) are presented in Section 3. The hardness result is stated in Section 4. Applications, experimental set-up, and experimental results are discussed in Section 5. Finally, we conclude with a discussion in Section 6. Due to space considerations, most of the proofs have been omitted from the main paper and may be found in the supplementary material.

2. Preliminaries

Let Ω be a ground set of size n. For a real-valued set function $g : 2^\Omega \to \mathbb{R}$, we write the marginal gain of adding an element e to a set A as $g(e \mid S) \triangleq g(S \cup \{e\}) - g(S)$. We say that g is *monotone* if $g(A) \leq g(B)$ for all $A \subseteq B$. We say that g is *submodular* if for all sets $A \subseteq B \subseteq \Omega$ and element $e \notin B$,

$$g(e \mid A) \geq g(e \mid B) \ . \tag{2}$$

When g is interpreted as a utility function, (2) encodes a natural diminishing returns condition in the sense that the marginal gain of adding an element decreases as the current set grows larger. An equivalent definition is that $\sum_{e \in B} g(e \mid A) \geq g(A \cup B) - g(A)$, which allows for the following natural extension. A monotone set function g is γ-*weakly submodular* for $\gamma \in (0, 1]$ if

$$\sum_{e \in B \setminus A} g(e \mid A) \geq \gamma \left(g(A \cup B) - g(A) \right) \tag{3}$$

holds for all $A \subseteq B$. Here, γ is referred to as the *submodularity ratio*. Intuitively, such a function g may not have strictly diminishing returns, but the increase in the returns is bounded by the marginals. Note that g is submodular if and only if it is γ-weakly submodular with $\gamma = 1$. A real-valued set function $c : 2^{\Omega} \to \mathbb{R}$ is *modular* if (2) holds with equality. A modular function may always be written in terms of coefficients as $c(S) = \sum_{e \in S} c_e$ and is non-negative if and only if all of its coefficients are non-negative.

Our algorithms are specified in the *value oracle model*, namely under the assumption that there is an oracle that, given a set $S \subseteq \Omega$, returns the value $g(S)$. As is standard, we analyze the run time complexity of these algorithms in terms of the number of function evaluations they require.

3. Algorithms

In this section, we present a suite of fast algorithms for solving Problem 1. The main idea behind each of these algorithms is to optimize a surrogate objective, which changes throughout the algorithm, preventing us from getting stuck in poor local optima. Further computational speed ups are obtained by randomized sub-sampling of the ground set.[1] The first algorithms we present assume knowledge of the weak submodularity parameter γ. However, γ is rarely known in practice (unless it is equal to 1), and thus, we show in Section 3.4 how to adapt these algorithms for the case of unknown γ.

To motivate the distorted objective we use, let us describe a way in which the greedy algorithm may fail. Suppose there is a "bad element" $b \in \Omega$ which has the highest overall gain, $g(b) - c_b$ and so is added to the solution set; however, once added, the marginal gain of all remaining elements drops below the corresponding costs, and so the greedy algorithm terminates. This outcome is suboptimal when there are other elements e that, although their overall marginal gain $g(e \mid S) - c_e$ is lower, have much higher ratio between the marginal utility $g(e \mid S)$ and the cost c_e (see Appendix A for an explicit construction).

To avoid this type of situation, we design a distorted objective which initially places higher relative importance on the modular cost term c, and gradually increases the relative importance of the utility g as the algorithm progresses. Our analysis relies on two functions: Φ, the distorted objective,

[1] We note that these two techniques can be traced back to the works of (Feldman, 2018) and (Mirzasoleiman et al., 2015), respectively.

Algorithm 1 DISTORTED GREEDY

Input: utility g, weak γ, cost c, cardinality k
Initialize $S_0 \leftarrow \varnothing$
for $i = 0$ **to** $k - 1$ **do**
  $e_i \leftarrow \arg\max_{e \in \Omega} \left\{ \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\}$
  if $\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e_i \mid S_i) - c_{e_i} > 0$ **then**
    $S_{i+1} \leftarrow S_i \cup \{e_i\}$
  end if
end for

and Ψ, an important quantity in analyzing the trajectory of Φ. Let k denote the cardinality constraint, then for any iteration $i = 0, \ldots, k-1$ of our algorithm and any set T, we define

$$\Phi_i(T) \triangleq \left(1 - \frac{\gamma}{k}\right)^{k-i} g(T) - c(T) \ .$$

Additionally, for any $i = 0, \ldots, k$, a set $T \subseteq \Omega$, and an element $e \in \Omega$, let

$$\Psi_i(T, e) \triangleq \max\left\{0, \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid T) - c_e \right\} \ .$$

Most proofs in this section are deferred to Appendix B.

3.1. Distorted Greedy

Our first algorithm, DISTORTED GREEDY, is presented as Algorithm 1. At each iteration, this algorithm chooses an element e_i maximizing the increase in the distorted objective. The algorithm then only accepts e_i if it positively contributes to the distorted objective. The analysis consists mainly of two lemmas. First, Lemma 1 shows that the marginal gain in the distorted objective is lower bounded by a term involving Ψ. This fact relies on the non-negativity of c and the rejection step in the algorithm.

Lemma 1. *In each iteration of* DISTORTED GREEDY*,*

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \Psi_i(S_i, e_i) + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \ .$$

The second lemma shows that the marginal gain in the distorted objective is sufficiently large to ensure the desired approximation guarantees. This fact relies on the monotonicity and γ-weak submodularity of g.

Lemma 2. *In each iteration of* DISTORTED GREEDY*,*

$$\Psi_i(S_i, e_i) \geq \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - g(S_i)\right] - \frac{1}{k} c(OPT) \ .$$

Using these two lemmas, we present an approximation guarantee for DISTORTED GREEDY.

Theorem 3. DISTORTED GREEDY *makes $O(nk)$ evaluations of g and returns a set R of size at most k with*

$$g(R) - c(R) \geq \left(1 - e^{-\gamma}\right) g(OPT) - c(OPT) \ .$$

Proof. Since c is modular and g is non-negative, the definition of Φ gives

$$\Phi_0(S_0) = \left(1 - \frac{\gamma}{k}\right)^k g(\varnothing) - c(\varnothing) \geq 0$$

and

$$\Phi_k\left(S_k\right) = \left(1 - \frac{\gamma}{k}\right)^0 g(S_k) - c(S_k) = g(S_k) - c(S_k) \ .$$

Using this and the fact that the returned set R is in fact S_k, we get

$$\begin{aligned} g(R) - c(R) &\geq \Phi_k(S_k) - \Phi_0(S_0) \qquad (4) \\ &= \sum_{i=0}^{k-1} \Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) \ . \end{aligned}$$

Applying Lemmas 1 and 2, respectively, we have

$$\begin{aligned} &\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) \\ &= \Psi_i(S_i, e_i) + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \\ &\geq \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - g(S_i)\right] \\ &\quad - \frac{1}{k}c(OPT) + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \\ &= \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(OPT) - \frac{1}{k}c(OPT) \ . \end{aligned}$$

Finally, plugging this bound into (4) yields

$$\begin{aligned} &g(R) - c(R) \\ &\geq \sum_{i=0}^{k-1} \left[\frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(OPT) - \frac{1}{k}c(OPT)\right] \\ &= \left[\frac{\gamma}{k}\sum_{i=0}^{k-1}\left(1 - \frac{\gamma}{k}\right)^i\right] g(OPT) - c(OPT) \\ &= \left(1 - \left(1 - \frac{\gamma}{k}\right)^k\right) g(OPT) - c(OPT) \\ &\geq \left(1 - e^{-\gamma}\right) g(OPT) - c(OPT) \ . \end{aligned}$$

□

3.2. Stochastic Distorted Greedy

Our second algorithm, STOCHASTIC DISTORTED GREEDY, is presented as Algorithm 2. It uses the same distorted objective as DISTORTED GREEDY, but enjoys an asymptotically faster run time due to sampling techniques of (Mirzasoleiman et al., 2015). Instead of optimizing over the entire ground set at each iteration, STOCHASTIC DISTORTED GREEDY optimizes over a random sample $B_i \subseteq \Omega$ of size $O\left(\frac{n}{k}\log\frac{1}{\epsilon}\right)$. This sampling procedure ensures that sufficient potential gain occurs *in expectation*, which is true for the following reason. If the sample size is sufficiently large, then B_i contains at least one element of OPT with high probability. Conditioned on this (high probability) event, choosing the element with the maximum potential gain is at least as good as choosing an average element from OPT.

Algorithm 2 STOCHASTIC DISTORTED GREEDY

Input: utility g, weak γ, cost c, cardinality k, error ϵ
Initialize $S_0 \leftarrow \varnothing$, $s \leftarrow \lceil \frac{n}{k}\log(\frac{1}{\epsilon}) \rceil$
for $i = 0$ **to** $k-1$ **do**
  $B_i \leftarrow$ sample s elements uniformly & ind. from Ω
  $e_i \leftarrow \arg\max_{e \in B_i} \left\{ \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\}$
  if $\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e_i \mid S_i) - c_{e_i} > 0$ **then**
    $S_{i+1} \leftarrow S_i \cup \{e_i\}$
  end if
end for

Lemma 4. *In each step of* STOCHASTIC DISTORTED GREEDY,

$$\begin{aligned} \mathbb{E}\left[\Psi_i(S_i, e_i)\right] \geq (1-\epsilon)\Bigg(&\frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \Big[g(OPT) \\ &- \mathbb{E}\left[g(S_i)\right]\Big] - \frac{1}{k}c(OPT)\Bigg) \ . \end{aligned}$$

Theorem 5. STOCHASTIC DISTORTED GREEDY *uses $O(n\log\frac{1}{\epsilon})$ evaluations of g and returns a set R with*

$$\mathbb{E}\left[g(R) - c(R)\right] \geq \left(1 - e^{-\gamma} - \epsilon\right) g(OPT) - c(OPT) \ .$$

3.3. Unconstrained Distorted Greedy

In this section, we present UNCONSTRAINED DISTORTED GREEDY, an algorithm for the unconstrained setting (i.e., $k = n$), listed as Algorithm 3. UNCONSTRAINED DISTORTED GREEDY samples a *single* random element at each iteration, adding it to the current solution if the potential gain is sufficiently large. Note that this algorithm is faster than the previous two, as it requires only $O(n)$ evaluations of g. As before, the algorithm relies on the distorted objective and the heart of the analysis is showing that the potential increase is sufficiently large in each iteration.

Lemma 6. *In each step of* UNCONSTRAINED DISTORTED GREEDY,

$$\begin{aligned} \mathbb{E}\left[\Psi_i(S_i, e_i)\right] \geq \frac{\gamma}{n}\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} &\Big[g(OPT) \\ &- \mathbb{E}\left[g(S_i)\right]\Big] - \frac{1}{n}c(OPT) \ . \end{aligned}$$

In the same way that Theorem 3 follows from Lemma 2,

Algorithm 3 UNCONSTRAINED DISTORTED GREEDY

Input: utility g, weak γ, cost c, cardinality k
Initialize $S_0 \leftarrow \varnothing$
for $i = 0$ **to** $n-1$ **do**
  $e_i \leftarrow$ sample uniformly from Ω
  if $\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} g(e_i \mid S_i) - c_{e_i} > 0$ **then**
    $S_{i+1} \leftarrow S_i \cup \{e_i\}$
  end if
end for

the next theorem follows from Lemma 6 (and therefore, we omit its proof also from the appendix).

Theorem 7. UNCONSTRAINED DISTORTED GREEDY *requires $O(n)$ function evaluations and outputs a set R such that*

$$\mathbb{E}\left[g(R) - c(R)\right] \geq (1 - e^{-\gamma}) g(OPT) - c(OPT) \ .$$

3.4. Guessing Gamma: A Geometric Sweep

The previously described algorithms required knowledge of the submodularity ratio γ. However, it is very rare that the precise value of γ is known in practice—unless g is submodular, in which case $\gamma = 1$. Oftentimes, γ is data dependent and only a crude lower bound $L \leq \gamma$ is known. In this section, we describe a meta algorithm that "guesses" the value of γ. γ-SWEEP, listed as Algorithm 4, runs a maximization algorithm $\mathcal{A}$ as a subroutine with a geometrically decreasing sequence of "guesses" $\gamma^{(k)}$ for $k = 0, 1, \ldots, \lceil \frac{1}{\delta} \log \frac{1}{\delta} \rceil$. The best set obtained by this procedure is guaranteed to have nearly as good approximation guarantees as when the true submodularity ratio γ is known exactly. Moreover, fewer guesses are required if a good lower bound $L \leq \gamma$ is known, which is true for several problems of interest. In the following theorem, we assume that $\mathcal{A}(g, \gamma, c, k, \epsilon)$ is an algorithm which returns a set S with $|S| \leq k$ and $\mathbb{E}\left[g(S) - c(S)\right] \geq (1 - e^{-\gamma} - \epsilon)\, g(OPT) - c(OPT)$ when g is γ-weakly submodular, and $L \leq \gamma$ is known (one may always use $L = 0$).

Algorithm 4 γ-SWEEP

Input: utility g, cost c, alg. $\mathcal{A}$, lower bound L, $\delta \in (0, 1)$
$S_{-1} \leftarrow \varnothing$, $T \leftarrow \left\lceil \frac{1}{\delta} \ln\left(\frac{1}{\max\{\delta, L\}}\right) \right\rceil$
for $r = 0$ **to** T **do**
  $\gamma_r \leftarrow (1-\delta)^r$
  $S_r \leftarrow \mathcal{A}(g, \gamma_r, c, k, \delta)$
end for
$R \leftarrow \arg\max_{r=-1,\ldots,T} \{g(S_r) - c(S_r)\}$

Theorem 8. γ-SWEEP *requires at most $O\left(\frac{1}{\delta} \log \frac{1}{\delta}\right)$ calls to $\mathcal{A}$ and returns a set R with*

$$\mathbb{E}\left[g(R) - c(R)\right] \geq \left(1 - e^{-\gamma} - O(\delta)\right) g(OPT) - c(OPT) \ .$$

In our experiments, we see that STOCHASTIC DISTORTED GREEDY combined with the γ-SWEEP outperforms the DISTORTED GREEDY with γ-SWEEP, especially for larger values of k. Here, we provide some experimental evidence and explanation for why this may be occurring. Figure 1 shows the objective value of the sets $\{S_r\}_{r=0}^{T}$ produced by STOCHASTIC DISTORTED GREEDY and DISTORTED GREEDY during the γ-SWEEP for cardinality constraints $k = 5$, 10, and 20. Both subroutines return the highest objective value for similar ranges of γ. However, the STOCHASTIC DISTORTED GREEDY subroutine appears to be better in two ways. First, the average objective value is usually larger, meaning that an individual run of STOCHASTIC DISTORTED GREEDY is returning a higher quality set than DISTORTED GREEDY. This is likely due to the sub-sampling of the ground set, which might help avoiding the picking of a single "bad element", if one exists. Second, the variation in STOCHASTIC DISTORTED GREEDY leads to a higher chance of producing a good solution. For many values of γ, the DISTORTED GREEDY subroutine returns a set of the same value; thus, the extra guesses of γ are not particularly helpful. On the other hand, the variation within STOCHASTIC DISTORTED GREEDY subroutine means that these extra guesses are not wasted; in fact, they allow a higher chance of producing a set with good value. Figure 1 also shows that the objective function throughout the sweep is fairly well-behaved, suggesting the possibility of early stopping heuristics. However, that is outside the scope of this paper.

4. Hardness Result

In this section, we give a hardness result which complements our algorithmic guarantees. The hardness result shows that—in the case where $c = 0$—no algorithm making polynomially many queries to g can achieve a better approximation ratio than $1 - e^{-\gamma}$. Although this was known in the case when $\gamma = 1$ (i.e., g is submodular), the more general result for $\gamma < 1$ was unknown until this work.

Theorem 9. *For every two constants $\varepsilon > 0$ and $\gamma \in (0, 1]$, no polynomial time algorithm achieves $(1 - e^{-\gamma} + \varepsilon)$-approximation for the problem of maximizing a non-negative monotone γ-weakly submodular function subject to a cardinality constraint in the value oracle model.*

As is usual in hardness proofs for submodular functions, the proof is based on constructing a family of γ-weakly submodular functions on which any deterministic algorithm

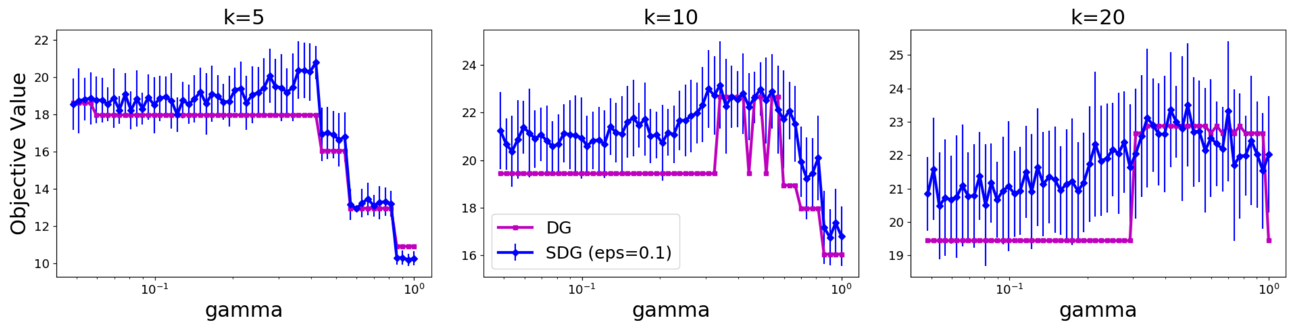

Figure 1: Results of the γ-SWEEP with DISTORTED GREEDY (DG) and STOCHASTIC DISTORTED GREEDY (SDG) as subroutines. For STOCHASTIC DISTORTED GREEDY, mean values with standard deviation bars are reported over 20 trials.

will perform poorly in expectation, and then applying Yao's principle. We defer details to Appendix C.

5. Experiments

To demonstrate the effectiveness of our proposed algorithms, we run experiments on two applications: Bayesian A-optimal design with costs and directed vertex cover with costs. The code was written using the `Julia` programming language, version 1.0.2. Experiments were run on a 2015 MacBook Pro with 3.1 GHz Intel Core i7 and 8 GB DDR3 SDRAM and the timing was reported using the `@timed` feature in `Julia`. [2]

5.1. Bayesian A-Optimal Design

We first describe the problem of Bayesian A-Optimal design. Suppose that $\theta \in \mathbb{R}^d$ is an unknown parameter vector that we wish to estimate from noisy linear measurements using least squares regression. Our goal is to choose a set S of linear measurements (the so-called experiments) which have low cost and also maximally reduce the variance of our estimate $\hat{\theta}$. More precisely, let $x_1, x_2, \dots x_n \in \mathbb{R}^d$ be a fixed set of measurement vectors, and let $X = [x_1, x_2, \dots x_n]$ be the corresponding $d \times n$ matrix. Given a set of measurement vectors $S \subseteq [n]$, we may run an experiment and obtain the noisy linear observation, $y_S = X_S^T \theta + \zeta_S$, where ζ_S is normal i.i.d. noise, i.e., $\zeta_1, \dots, \zeta_n \sim N(0, \sigma^2)$. We estimate θ using the least squares estimator $\hat{\theta} = (X_S X_S^T)^{-1} X_S^T y_S$. Assuming a normal Bayesian prior distribution on the unknown parameter, $\theta \sim N(0, \Sigma)$, the sum of the variance of the coefficients given the measurement set S is $r(S) = Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_S X_S^T\right)^{-1}$. We define $g(S) = r(\varnothing) - r(S)$ to be the *reduction in variance* produced by experiment set S. Bian et al. (2017) showed that g is γ-weakly submodular, providing a lower bound for γ in the case where $\Sigma = \beta I$. However, their bound relies rather unfavorably on the spectral norm of X, and does not extend to general Σ. Chamon & Ribeiro (2017) showed that g satisfies the stronger condition of γ-weak DR (Definition C.1), but their bound on the submodularity ratio γ depends on the cardinality of the sets. We give a tighter bound here, and the proof appears in Appendix D.

Claim 10. *g is a non-negative, monotone and γ-weakly submodular function with*

$$\gamma \geq \left(1 + \frac{s^2}{\sigma^2} \lambda_{\max}(\Sigma)\right)^{-1} \quad ,$$

where $s = \max_{i \in [n]} \|x_i\|_2$.

Suppose that each experiment x_i has an associated non-negative cost c_i. In this application, we seek to maximize the "revenue" of the experiment,

$$g(S) - c(S) = Tr\left(\Sigma\right) - Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_S X_S^T\right)^{-1} - c(S) \quad ,$$

which trades off the utility of the experiments (i.e., the variance reduction in the estimator) and their overall cost.

Unlike submodular functions, lazy evaluations (Minoux, 1978) of γ-weakly submodular g are generally not possible, as the marginal gains vary unpredictably. However, for specific functions, one can possibly speed up the greedy search. For the utility g considered here, we implemented a faster greedy search using the matrix inversion lemma. The naive approach of computing $g(e \mid S)$ by constructing $\Sigma^{-1} + X_S X_S^T$, explicitly computing its inverse, and summing the diagonal elements is not only expensive—inversion alone costs $O(d^3)$ arithmetic operations—but also memory-inefficient. Instead, one can use the matrix inversion lemma to show that

$$g(e \mid S) = \frac{\|z_e\|^2}{\sigma^2 + \langle x_e, z_e \rangle} \quad ,$$

where $z_e = M_S^{-1} x_e$ and $M_S = \Sigma^{-1} + X_S X_S^T$. Moreover, M_S^{-1} may be *stored and updated directly* without any matrix

[2] Source code available at https://github.com/crharshaw/submodular-minus-linear

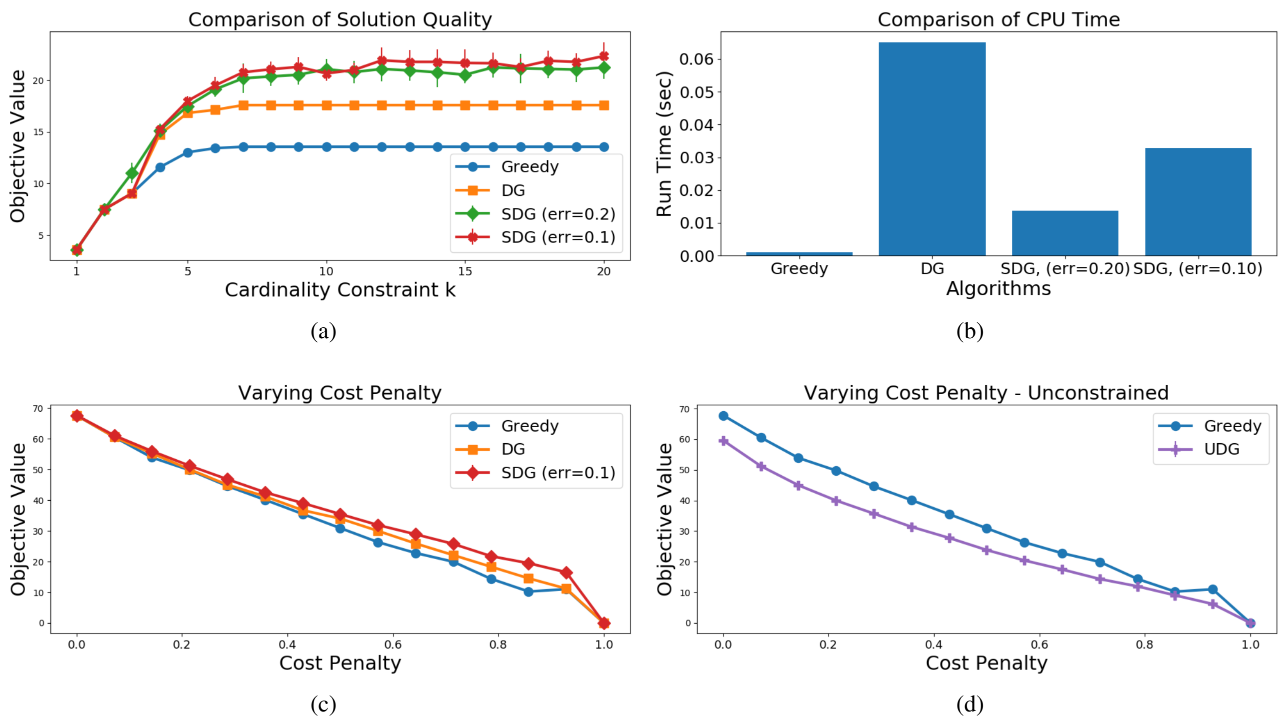

Figure 2: An algorithmic performance comparison for Bayesian A-Optimal design on the Boston Housing dataset. We report values for stochastic algorithms with mean and standard deviation bars, over 20 trials. (2a) objective values, varying the cardinality k, for a fixed cost penalty $\alpha = 0.8$. (2b) runtime for a fixed cardinality $k = 15$. (2c) objective values, varying the cost penalty α for a fixed cardinality $k = 15$. (2d) objective values, varying the cost penalty α in an unconstrained setting.

inversion. In this way, marginal gains $g(e \mid S)$ may be queried using only matrix-vector multiplication with a fixed M_S^{-1} and inner product computations, which requires $O(d^2)$ arithmetic operations and is more memory efficient. More details are given in Appendix D.

For this experiment, we used the Boston Housing dataset (Jr. & Rubenfield, 1978), a standard benchmark dataset containing $d = 14$ attributes of $n = 506$ Boston homes, including average number of rooms per dwelling, proximity to the Charles River, and crime rate per capita. We preprocessed the data by normalizing the features to have a zero mean and a standard deviation of 1. As there is no specified cost per measurement, we assigned costs proportionally to initial marginal gains in utility; that is, $c_e = \alpha g(e)$ for some $\alpha \in [0, 1]$. We set $\sigma = 1/\sqrt{d}$, and randomly generated a normal prior with covariance $\Sigma = ADA^T$, where A is randomly chosen as $A_{i,j} \sim N(0, 1)$ and D is diagonal with $D_{i,i} = (i/d)^2$. We choose not to use $\Sigma = \beta I$, as we found this causes g to be nearly modular along solution paths, yielding it an easy problem instance for all algorithms and not a suitable benchmark.

In our first experiment, we fixed the cost penalty $\alpha = 0.8$, and ran the algorithms for varying cardinality constraints from $k = 1$ to $k = 15$. We ran the greedy algorithm, DISTORTED GREEDY with γ-SWEEP (setting $\delta = 0.1$), and two instances of STOCHASTIC DISTORTED GREEDY with γ-SWEEP (with $\delta = \epsilon = 0.1$ and $\delta = \epsilon = 0.05$). All γ-SWEEP runs used $L = 0$. In Figure 2a, one can observe that the marginal gain obtained by the greedy algorithm is not non-increasing (at least for the first few elements), which is a result of the fact that g is weakly submodular with $\gamma < 1$. For small values of k, all algorithms produce comparable solutions; however, the greedy algorithm gets stuck in a local maximum of size $k = 7$, while our algorithms are able to produce larger solutions with higher objective value. Moreover, γ-SWEEP with STOCHASTIC DISTORTED GREEDY performs better than γ-SWEEP with DISTORTED GREEDY for larger values of k, for reasons discussed in Section 3.4. Figure 2b shows CPU times of each algorithm run with the single cardinality constraint $k = 20$. We see that the greedy algorithm runs faster than our algorithms. This difference in the runtime is a result of both the added complexity of the γ-SWEEP procedure, and that greedy terminates early, when a local maximum is reached. Figure 2b also shows that the sub-sampling step in STOCHASTIC DISTORTED GREEDY results in a faster runtime than DISTORTED GREEDY, as predicted by the theory. We did not display the number of function evaluations, as it exhibits nearly identical trends to the actual CPU run time. In our next experiment, we fixed the cardinality $k = 15$ and varied the cost penalty $\alpha \in [0, 1]$. Figure 2c shows that all algorithm return similar solutions for $\alpha = 0$ and $\alpha = 1$, which are the cases in which either $c = 0$ or the function $g - c$ is non-positive, respectively. For all other values of α, our algorithms yield improvements over greedy. In our final experiment, we varied the cost penalty $\alpha \in [0, 1]$, comparing the output of greedy and γ-SWEEP with UNCONSTRAINED DISTORTED GREEDY for the unconstrained setting. Figure 2d shows that greedy outperforms our algorithm in this instance, which can occur, especially in the absence of "bad elements" discussed in Section 3.

5.2. Directed Vertex Cover with Costs

The second experiment is directed vertex cover with costs. Let $G = (V, E)$ be a directed graph and let $w : V \to \mathbb{R}$ be a weight function on the vertices. For a vertex set $S \subseteq V$, let $N(S)$ denote the set of vertices which are pointed to by S, $N(S) \triangleq \{v \in V \mid (u, v) \in E \text{ for some } u \in S\}$. The weighted directed vertex cover function is $g(S) = \sum_{u \in N(S) \cup S} w_u$. We also assume that each vertex $v \in V$ has an associated nonnegative cost c_v. Our goal is to maximize the resulting revenue,

$$g(S) - c(S) = \sum_{u \in N(S) \cup S} w_u - \sum_{u \in S} c_u \ .$$

Because g is submodular, we can forgo the γ-SWEEP routine and apply our algorithms directly with $\gamma = 1$. Moreover, we implement lazy evaluations of g in our code.

For our experiments, we use the EU Email Core network, a directed graph generated using email data from a large European research institution (Yin et al., 2017; Leskovec et al., 2007). The graph has 1k nodes and 25k directed edges, where nodes represent people and a directed edge from u to v means that an email was sent from u to v. We assign each node a weight of 1. Additionally, as there are no costs in the dataset, we assign costs in the following manner. For a fixed q, we set $c(v) = 1 + \max\{d(v) - q, 0\}$, where $d(v)$ is the out-degree of v. Thus, all vertices with out-degree larger than q have the same initial marginal gain $g(v) - c(v) = q$.

In our first experiment, we fixed the cost factor $q = 6$, and ran the algorithms for varying cardinality constraints from $k = 1$ to $k = 130$. We see in Figure 3a that our methods outperform greedy. DISTORTED GREEDY achieves the highest objective value for each cardinality constraint, while STOCHASTIC DISTORTED GREEDY achieves higher objective values as the accuracy parameter ϵ is decreased. Figure 3b shows the number of function evaluations made by the algorithms when $k = 130$. We observe that STOCHASTIC DISTORTED GREEDY requires much fewer function evaluations, even when lazy evaluations are implemented.[3]

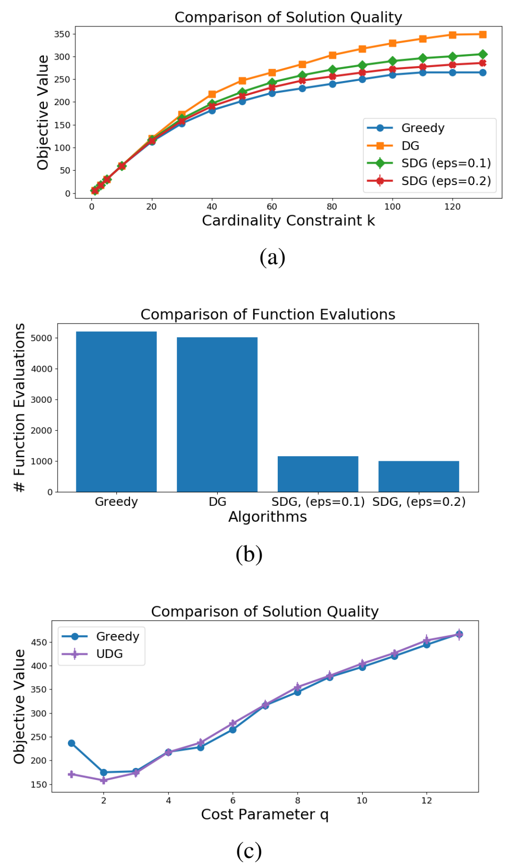

Figure 3: A performance comparison for directed vertex cover on the EU Email Core network. We report values for stochastic algorithms with mean and standard deviation bars, over 20 trials. (3a) objective values, varying the cardinality k, for a fixed cost factor $q = 6$. (3b) g evaluations for a fixed cardinality $k = 130$. (3c) objective values, varying the cost factor q in an unconstrained setting.

Finally, we ran greedy and UNCONSTRAINED DISTORTED GREEDY while varying the cost factor q from 1 to 12, and we note that in this setting (as can be seen in Figure 3c) our algorithm performs similarly to the greedy algorithm.

6. Conclusion

We presented a suite of fast algorithms for maximizing the difference between a non-negative monotone γ-weakly submodular g and a non-negative modular c in both the cardinality constrained and unconstrained settings. Moreover, we gave a matching hardness result showing that no algorithm can do better with only polynomially many oracle queries to g. Finally, we experimentally validated our algorithms on Bayesian A-Optimality and directed vertex cover with costs, demonstrating that they outperform the greedy heuristic.

[3] We do not report CPU time, which does not reflect function evaluations here. This is due to the lazy evaluation implementation.

Acknowledgments

Christopher Harshaw was supported in part by National Science Foundation GRFP (DGE1122492) and by ONR Award N00014-16-1-2374. Moran Feldman was supported in part by Israel Science Foundation (ISF) grant number 1357/16. Amin Karbasi was supported by AFOSR Young Investigator Award (FA9550-18-1-0160) and Grainger Award (PO 2000008083 2016012).

References

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 498–507, 2017.

Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11): 1222–1239, 2001.

Buchbinder, N. and Feldman, M. Constrained submodular maximization via a non-symmetric technique. *CoRR*, abs/1611.03253, 2016.

Chamon, L. F. O. and Ribeiro, A. Approximate supermodularity bounds for experimental design. In *Advances in Neural Information Processing Systems*, 2017.

Chekuri, C., Vondrák, J., and Zenklusen, R. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.*, 43(6): 1831–1879, 2014.

Chvátal, V. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

Das, A. and Kempe, D. Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection. In *International Conference on Machine Learning*, pp. 1057–1064, 2011.

Elenberg, E. R., Dimakis, A. G., Feldman, M., and Karbasi, A. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems*, 2017.

Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 46, 2018.

Ene, A. and Nguyen, H. L. Constrained submodular maximization: Beyond 1/e. In *FOCS*, pp. 248–257, 2016.

Feldman, M. Guess free maximization of submodular and linear sums. Technical Report arXiv:1810.03813v1 [cs.DS], ArXiV, October 2018.

Feldman, M., Harshaw, C., and Karbasi, A. Greed is good: Near-optimal submodular maximization via greedy optimization. In *COLT*, pp. 758–784, 2017.

Golovin, D. and Krause, A. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

Hu, H., Grubb, A., Bagnell, J. A., and Hebert, M. Efficient feature group sequencing for anytime linear prediction. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016.

Jegelka, S. and Bilmes, J. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1897–1904. IEEE, 2011.

Jr., D. H. and Rubenfield, D. L. Hedonic housing prices and the demand for clean air. *J. of Environmental Economics and Management*, 5(1):81–102, 1978.

Kempe, D., Kleinberg, J., and Tardos, E. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Khanna, R., Elenberg, E. R., Dimakis, A. G., Negahban, S., and Ghosh, J. Scalable Greedy Feature Selection via Weak Submodularity. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1560–1568, 2017.

Krause, A. and Guestrin, C. Near-optimal Nonmyopic Value of Information in Graphical Models. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 324–331, 2005.

Kuhnle, A., Smith, J. D., Crawford, V. G., and Thai, M. T. Fast maximization of non-submodular, monotonic functions on the integer lattice. In *ICML*, pp. 2791–2800, 2018.

Lee, J., Sviridenko, M., and Vondrák, J. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35(4):795–806, 2010.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th*

Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 510–520. Association for Computational Linguistics, 2011.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pp. 234–243, 1978.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1812–1818, 2015.

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

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294, 1978.

Skala, M. Hypergeometric tail inequalities: ending the insanity. *CoRR*, abs/1311.5939, 2013.

Sviridenko, M. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.

Sviridenko, M., Vondrák, J., and Ward, J. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Math. Oper. Res.*, 42(4):1197–1218, 2017.

Wei, K., Liu, Y., Kirchhoff, K., and Bilmes, J. Using Document Summarization Techniques for Speech Data Subset Selection. In *Proceedings of NAACL-HLT 2013*, pp. 721726, 2013.

Yin, H., Benson, A. R., Leskovec, J., and Gleich, D. F. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017.

A. Greedy Performs Arbitrarily Poorly

In this section, we describe an instance of Problem (1) where the greedy algorithm performs arbitrarily poorly. More specifically, the greedy algorithm does not achieve any constant factor approximation. Let G be a graph with n vertices and let $b \in V$ be a "bad vertex". The graph G includes a single directed edge (b, e) for every vertex $e \in V \setminus \{b\}$, and no other edges (i.e., G is a directed star with b in the center). Let g be the unweighted directed vertex cover function. Note that

$$g(\{e\}) = \begin{cases} 1 & \text{if } e \neq b \ , \\ n & \text{if } e = b \ . \end{cases}$$

Fix some $\epsilon > 0$, and let us define the nonnegative costs coefficients as

$$c_e = \begin{cases} 1/2 & \text{if } e \neq b \ , \\ n - (1/2 + \epsilon) & \text{if } e = b \ . \end{cases}$$

The initial marginal gain of a vertex e is now given by

$$g(\{e\}) - c_e = \begin{cases} 1/2 & \text{if } e \neq b \ , \\ 1/2 + \epsilon & \text{if } e = b \ . \end{cases}$$

Thus, the greedy algorithm chooses the "bad element" $b \in V$ in the first iteration. Note that after b is chosen, the greedy algorithm terminates, as $g(e \mid \{b\}) = 0$ and $c_e > 0$ for all remaining vertices e. However, for any set S of vertices which does not contain b, we have that

$$g(S) - c(S) = |S| - \frac{1}{2}|S| = \frac{1}{2}|S| \ .$$

Thus, for any $k < n$, the competitive ratio of greedy subject to a k cardinality constraint is at most

$$\frac{1/2 - \epsilon}{k/2} = \frac{1-\epsilon}{k} = O\left(\frac{1}{k}\right) \ .$$

B. Algorithm Proofs Omitted From the Main Body

B.1. Distorted Greedy

Proof of Lemma 1. By expanding the definition of Φ and rearranging, we get

$$\begin{aligned} \Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) &= \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_{i+1}) - c(S_{i+1}) - \left(1 - \frac{\gamma}{k}\right)^{k-i} g(S_i) + c(S_i) \\ &= \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_{i+1}) - c(S_{i+1}) - \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left(1 - \frac{\gamma}{k}\right) g(S_i) + c(S_i) \\ &= \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} [g(S_{i+1}) - g(S_i)] - [c(S_{i+1}) - c(S_i)] + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \ . \end{aligned}$$

Now let us consider two cases. First, suppose that the if statement in DISTORTED GREEDY passes, which means that $\Psi_i(S_i, e_i) = \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e_i \mid S_i) - c_{e_i} > 0$ and that e_i is added to the solution set. By the non-negativity of c, we can deduce in this case that $e_i \notin S_i$, and thus, $g(S_{i+1}) - g(S_i) = g(e_i \mid S_i)$ and $c(S_{i+1}) - c(S_i) = c_{e_i}$. Hence,

$$\begin{aligned} \Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = &\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e_i \mid S_i) - c_{e_i} + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \\ &= \Psi_i(S_i, e_i) + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \ . \end{aligned}$$

Next, suppose that the if statement in DISTORTED GREEDY does not pass, which means that $\Psi_i(S_i, e_i) = 0 \geq \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e_i \mid S_i) - c_{e_i}$ and the algorithm does not add e_i to its solution. In particular, $S_{i+1} = S_i$, and thus, $g(S_{i+1}) - g(S_i) = 0$ and $c(S_{i+1}) - c(S_i) = 0$. In this case,

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = 0 + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) = \Psi_i(S_i, e_i) + \frac{\gamma}{k}\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(S_i) \ . \qquad \square$$

Proof of Lemma 2. Observe that

$$
\begin{aligned}
k \cdot \Psi_i(S_i, e_i) &= k \cdot \max_{e \in \Omega} \left\{ 0, \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\} && \text{(definitions of } \Psi \text{ and } e_i) \\
&\geq |OPT| \cdot \max_{e \in \Omega} \left\{ 0, \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\} && (|OPT| \leq k) \\
&\geq |OPT| \cdot \max_{e \in OPT} \left\{ \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\} && \text{(restricting maximization)} \\
&\geq \sum_{e \in OPT} \left[\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right] && \text{(averaging argument)} \\
&= \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \sum_{e \in OPT} g(e \mid S_i) - c(OPT) \\
&\geq \gamma \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} [g(OPT) - g(S_i)] - c(OPT) \ . && (\gamma\text{-weak submodularity})
\end{aligned}
$$

□

B.2. Stochastic Distorted Greedy

Lemma 11. *In each step* $0 \leq i \leq k-1$ *of* STOCHASTIC DISTORTED GREEDY,

$$\Pr\left[B_i \cap OPT \neq \varnothing\right] \geq (1-\epsilon) \frac{|OPT|}{k} \ .$$

Proof.

$$\Pr\left[B_i \cap OPT = \varnothing\right] \leq \left(1 - \frac{|OPT|}{n}\right)^s \leq e^{-s\frac{|OPT|}{n}} = e^{-\frac{sk}{n}\frac{|OPT|}{k}} \ ,$$

where we used the known inequality $1 - x \leq e^{-x}$. Thus,

$$\Pr\left[B_i \cap OPT \neq \varnothing\right] \geq 1 - e^{-\frac{sk}{n}\frac{|OPT|}{k}} \geq \left(1 - e^{-\frac{sk}{n}}\right) \frac{|OPT|}{k} \geq (1-\epsilon) \frac{|OPT|}{k} \ ,$$

where the second inequality follows from $1 - e^{-ax} \geq (1 - e^{-a})x$ for $x \in [0, 1]$, and the last inequality follows from the choice of sample size $s = \lceil \frac{n}{k} \log \frac{1}{\epsilon} \rceil$. □

Conditioned on the fact that at least one element of OPT was sampled, the following lemma shows that sufficient potential gain is made.

Lemma 12. *In each step* $0 \leq i \leq k-1$ *of* STOCHASTIC DISTORTED GREEDY,

$$\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i,\ B_i \cap OPT \neq \varnothing\right] \geq \frac{\gamma}{|OPT|} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} [g(OPT) - g(S_i)] - \frac{1}{|OPT|} c(OPT) \ .$$

Proof. Throughout the proof, all expectations are conditioned on the current set S_i and the event that $B_i \cap OPT \neq \varnothing$, as in the statement of the lemma. For convenience, we drop the notations of these conditionals from the calculations below.

$$
\begin{aligned}
\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i)\right] &= \mathbb{E}\left[\max_{e \in B_i} \Psi_i(S_i, e_i)\right] && \text{(definition of } e_i) \\
&\geq \mathbb{E}\left[\max_{e \in B_i \cap OPT} \Psi_i(S_i, e_i)\right] && \text{(restricting max)} \\
&\geq \mathbb{E}\left[\max_{e \in B_i \cap OPT} \left\{ \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e \right\}\right] \ . && \text{(definition of } \Psi)
\end{aligned}
$$

We now note that $B_i \cap OPT$ is a subset of OPT that contains every element of OPT with the same probability. Moreover, this is true also conditioned on $B_i \cap OPT \neq \varnothing$. Thus, picking the best element from $B_i \cap OPT$ (when this set is not-empty) achieves gain at least as large as picking a random element from $B_i \cap OPT$, which is identical to picking a random element from OPT. Plugging this observation into the previous inequality, we get

$$\begin{aligned}
\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i)\right] &\geq \frac{1}{|OPT|} \sum_{e \in OPT} \left[\left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(e \mid S_i) - c_e\right] \\
&= \frac{1}{|OPT|} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \sum_{e \in OPT} g(e \mid S_i) - \frac{1}{|OPT|} c(OPT) \\
&\geq \frac{\gamma}{|OPT|} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT \cup S_i) - g(S_i)\right] - \frac{1}{|OPT|} c(OPT) \\
&\geq \frac{\gamma}{|OPT|} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - g(S_i)\right] - \frac{1}{|OPT|} c(OPT) \ ,
\end{aligned}$$

where the last two inequalities follows from the γ-weak submodularity and monotonicity of g, respectively. □

Using the last two lemmas, we can now prove the claims from the main paper.

Proof of Lemma 4. By the law of iterated expectation and the non-negativity of Ψ,

$$\begin{aligned}
\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i\right] &= \mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i,\ B_i \cap OPT \neq \varnothing\right] \Pr\left[B_i \cap OPT \neq \varnothing\right] \\
&\quad + \mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i,\ B_i \cap OPT = \varnothing\right] \Pr\left[B_i \cap OPT = \varnothing\right] \\
&\geq \mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i,\ B_i \cap OPT \neq \varnothing\right] \Pr\left[B_i \cap OPT \neq \varnothing\right] \\
&\geq \left(\frac{\gamma}{|OPT|} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - g(S_i)\right] - \frac{1}{|OPT|} c(OPT)\right) \left((1-\epsilon) \frac{|OPT|}{k}\right) \\
&= (1-\epsilon) \left(\frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - g(S_i)\right] - \frac{1}{k} c(OPT)\right) \ ,
\end{aligned}$$

where the second inequality holds by Lemmas 11 and 12. The lemma now follows since the law of iterated expectations also implies $\mathbb{E}\left[\Psi_i(S_i, e_i)\right] = \mathbb{E}_{S_i}\left[\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i\right]\right]$. □

Proof of Theorem 5. As discussed in the proof of Theorem 3, we have that

$$\mathbb{E}\left[g(R) - c(R)\right] \geq \mathbb{E}\left[\Phi_k(S_k) - \Phi_0(S_0)\right] = \sum_{i=0}^{k-1} \mathbb{E}\left[\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)\right] \ , \tag{5}$$

and so it is enough to lower bound each term in the rightmost side. To this end, we apply Lemma 1 and Lemma 4 to obtain

$$\begin{aligned}
\mathbb{E}\left[\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)\right] &\geq \mathbb{E}\left[\Psi_i(S_i, e_i)\right] + \frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \mathbb{E}\left[g(S_i)\right] \\
&\geq (1-\epsilon) \left(\frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \left[g(OPT) - \mathbb{E}\left[g(S_i)\right]\right] + \frac{1}{k} c(OPT)\right) + \frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \mathbb{E}\left[g(S_i)\right] \\
&= (1-\epsilon) \left(\frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(OPT) - \frac{1}{k} c(OPT)\right) + \epsilon \cdot \frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} \mathbb{E}\left[g(S_i)\right] \\
&\geq (1-\epsilon) \left(\frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(OPT) - \frac{1}{k} c(OPT)\right) \ ,
\end{aligned}$$

where the last inequality followed from non-negativity of g. Plugging this bound into (5) yields

$$\begin{aligned}
\mathbb{E}\left[g(R) - c(R)\right] &\geq (1-\epsilon) \sum_{i=0}^{k-1} \left[\frac{\gamma}{k} \left(1 - \frac{\gamma}{k}\right)^{k-(i+1)} g(OPT) - \frac{1}{k} c(OPT)\right] \\
&= (1-\epsilon) \left[\frac{\gamma}{k} \sum_{i=0}^{k-1} \left(1 - \frac{\gamma}{k}\right)^{i}\right] g(OPT) - (1-\epsilon) c(OPT) \\
&\geq (1-\epsilon) \left(1 - e^{-\gamma}\right) g(OPT) - c(OPT) && \text{(non-negativity of } g \text{ and } c \text{)} \\
&= \left(1 - e^{-\gamma} - \alpha\epsilon\right) g(OPT) - c(OPT) \ ,
\end{aligned}$$

where $\alpha = 1 - e^{-\gamma} \leq 0.65$.

To bound the number of function evaluations, observe that STOCHASTIC DISTORTED GREEDY has k rounds, each requiring $s = \lceil \frac{n}{k} \log \frac{1}{\epsilon} \rceil$ function evaluations. Thus, the total number of function evaluations is $k \times \lceil \frac{n}{k} \log \frac{1}{\epsilon} \rceil = O(n \log \frac{1}{\epsilon})$. □

B.3. Unconstrained Distorted Greedy

Proof of Lemma 6. We begin by analyzing the conditional expectation

$$\begin{aligned}
\mathbb{E}_{e_i}\left[\Psi_i(S_i, e_i) \mid S_i\right] &= \frac{1}{n} \sum_{e \in \Omega} \Psi_i(S_i, e) \\
&\geq \frac{1}{n} \sum_{e \in OPT} \Psi_i(S_i, e) && \text{(non-negativity of } \Psi) \\
&= \frac{1}{n} \sum_{e \in OPT} \max\left\{0, \left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} g(e \mid S_i) - c_e\right\} && \text{(by definition of } \Psi) \\
&\geq \frac{1}{n} \sum_{e \in OPT} \left\{\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} g(e \mid S_i) - c_e\right\} \\
&= \frac{1}{n}\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} \sum_{e \in OPT} g(e \mid S_i) - \frac{1}{n} c(OPT) && \text{(linearity of } c) \\
&\geq \frac{\gamma}{n}\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} \left[g(OPT \cup S_i) - g(S_i)\right] - \frac{1}{n} c(OPT) && (\gamma\text{-weak submodularity of } g) \\
&\frac{\gamma}{n}\left(1 - \frac{\gamma}{n}\right)^{n-(i+1)} \left[g(OPT) - g(S_i)\right] - \frac{1}{n} c(OPT) && \text{(monotonicity of } g) \ .
\end{aligned}$$

The lemma now follows by the law of iterated expectations. □

B.4. Guessing Gamma: A Geometric Sweep

Proof of Theorem 8. We consider two cases. First, suppose that $\gamma < \delta$. Under this assumption, we have

$$1 - e^{-\gamma} - \delta < 1 - e^{-\delta} - \delta \leq \delta - \delta = 0 \ ,$$

where the second inequality used the fact that $1 - e^{-x} \leq x$. Thus,

$$g(\varnothing) + \ell(\varnothing) \geq 0 \geq (1 - e^{-\gamma} - \delta) g(OPT) - c(OPT) \ ,$$

where the first inequality follows from non-negativity of g, and the second inequality follows from non-negativity of both c and g. Because Algorithm 4 sets $S^{(-1)} = \varnothing$ and R is chosen to be the best solution,

$$g(R) + \ell(R) \geq g(\varnothing) - c(\varnothing) \geq \left(1 - e^{-\gamma} - \delta\right) g(OPT) - c(OPT) \ .$$

For the second case, suppose that $\gamma \geq \delta$. Recall that $\gamma \geq L$ by assumption, and thus, $\gamma \geq B \triangleq \max\{\delta, L\}$. Now, we need to show that $(1-\delta)^T \leq B$. This is equivalent to $\left(\frac{1}{1-\delta}\right)^T \geq \frac{1}{B}$, and by taking $\ln$, this is equivalent to $T \geq \frac{\ln \frac{1}{B}}{\ln \frac{1}{1-\delta}}$. This is true since the inequality $\delta \leq \ln\left(\frac{1}{1-\delta}\right)$, which holds for all $\delta \in (0,1)$, implies

$$T = \left\lceil \frac{1}{\delta} \ln \frac{1}{B} \right\rceil \geq \frac{1}{\delta} \ln \frac{1}{B} \geq \frac{\ln \frac{1}{B}}{\ln \frac{1}{1-\delta}} \ .$$

Hence, we have proved that $(1-\delta)^T \leq B \leq \gamma$, which implies that there exists $t \in \{0, 1, \ldots, T\}$ such that $\gamma \geq \gamma^{(t)} \geq (1-\delta)\gamma$. For notational convenience, we write $\hat{\gamma} = \gamma^{(t)}$. Because g is γ-weakly submodular and $\gamma \geq \hat{\gamma}$, g is also $\hat{\gamma}$-weakly submodular. Therefore, by assumption, the algorithm $\mathcal{A}$ returns a set $S^{(t)}$ which satisfies

$$\mathbb{E}\left[g(S^{(t)}) + \ell(S^{(t)})\right] \geq \left(1 - e^{-\hat{\gamma}} - \delta\right) g(OPT) - c(OPT) \ .$$

From the convexity of e^x, we have $e^\delta \leq (1-\delta)e^0 + \delta e^1 = 1 + (e-1)\delta$ for all $\delta \in [0,1]$. Using this inequality, and the fact that $\hat{\gamma} \geq (1-\delta)\gamma$, we get

$$1 - e^{-\hat{\gamma}} \geq 1 - e^{-(1-\delta)\gamma} \geq 1 - e^{-\gamma}e^{\delta} \geq 1 - e^{-\gamma}(1 + (e-1)\delta) = 1 - e^{-\gamma} - e^{-\gamma}(e-1)\delta \geq 1 - e^{-\gamma} - \beta\delta$$

We remark that $\beta = e - 1 \approx 1.72$. Thus, by non-negativity of g and because the output set R was chosen as the set with highest value,

$$\mathbb{E}\left[g(R) - c(R)\right] \geq \mathbb{E}\left[g(S^{(t)}) - c(S^{(t)})\right] \geq \left(1 - e^{-\gamma} - (\beta\delta + \delta)\right) g(OPT) - c(OPT) \ . \qquad \square$$

C. Impossibility for Weakly Submodular Functions

It turns out that, instead of proving Theorem 9, it is easier to prove a stronger theorem given below as Theorem 13. However, before we can present Theorem 13, we need the following definition (this definition is related to a notion called the *DR-ratio* defined by Kuhnle et al. (2018) for functions over the integer lattice).

Definition C.1. *A function $f\colon 2^{\mathcal{N}} \to \mathbb{R}$ is γ-weakly DR if for every two sets $A \subseteq B \subseteq \mathcal{N}$ and element $u \in \mathcal{N} \setminus B$ it holds that $f(u \mid A) \geq \gamma \cdot f(u \mid B)$.*

Theorem 13. *For every two constants $\varepsilon > 0$ and $\gamma \in (0,1]$, no polynomial time algorithm achieves $(1 - e^{-\gamma} + \varepsilon)$-approximation for the problem of maximizing a non-negative monotone γ-weakly DR function subject to a cardinality constraint in the value oracle model.*

The following observation shows that every instance of the problem considered by Theorem 13 is also an instance of the problem considered by Theorem 9, and therefore, Theorem 13 indeed implies Theorem 9.

Observation 14. *A monotone γ-weakly DR set function $f\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ is also γ-weakly submodular.*

Proof. Consider arbitrary sets $A \subseteq B \subseteq \mathcal{N}$, and let us denote the elements of the set $B \setminus A$ by $u_1, u_2, \ldots, u_{|B\setminus A|}$ in a fixed arbitrary order. Then,

$$f(B \mid A) = \sum_{i=1}^{|B\setminus A|} f(u_i \mid A \cup \{u_1, u_2, \ldots, u_{i-1}\}) \geq \gamma \cdot \sum_{i=1}^{|B\setminus A|} f(u_i \mid A) \ . \qquad \square$$

The following proposition is the main technical component used in the proof of Theorem 13. To facilitate the reading, we defer its proof to Section C.1.

Proposition 15. *For every value $\varepsilon' \in (0, 1/6)$, value $\gamma \in (0,1]$ and integer $k \geq 1/\varepsilon'$, there exists a ground set $\mathcal{N}$ of size $\lceil 3k/\varepsilon' \rceil$ and a set function $f_T\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ for every set $T \subseteq \mathcal{N}$ of size at most k such that*

(P1) *f_T is non-negtive monotone and γ-weakly DR.*
(P2) *$f_T(S) \leq 1$ for every set $S \subseteq \mathcal{N}$, and the inequality holds as an equality for $S = T$ when the size of T is exactly k.*
(P3) *$f_\varnothing(S) \leq 1 - e^{-\gamma} + 12\varepsilon'$ for every set S of size at most k.*
(P4) *$f_T(S) = f_\varnothing(S)$ when $|S| \geq 3k - g$ or $|S \cap T| \leq g$, where $g = \lceil \varepsilon' k + 3k^2/|\mathcal{N}| \rceil$.*

At this point, let us consider some γ value and set $\varepsilon' = \varepsilon/20$. Note that Theorem 13 is trivial for $\varepsilon > 1$, and thus, we may assume $\varepsilon' \in (0, 1/6)$, which implies that there exists a large enough integer k for which γ, ε' and k obey all the requirements of Proposition 15. From this point on we consider the ground set $\mathcal{N}$ and the functions f_T whose existence is guaranteed by Proposition 15 for these values of γ, ε' and k. Let $\tilde{T}$ be a random subset of $\mathcal{N}$ of size k (such subsets exist because $|\mathcal{N}| > k$). Intuitively, in the rest of this section we prove Theorem 13 by showing that the problem $\max\{f_{\tilde{T}}(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ is hard in expectation for every algorithm.

Property (P2) of Proposition 15 shows that the optimal solution for the problem $\max\{f_{\tilde{T}}(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ is $\tilde{T}$. Thus, an algorithm expecting to get a good approximation ratio for this problem should extract information about the random set $\tilde{T}$. The question is on what sets should the algorithm evaluate $f_{\tilde{T}}$ to get such information. Property (P4) of the proposition shows that the algorithm cannot get much information about $\tilde{T}$ when querying $f_{\tilde{T}}$ on a set S that is either too large or has a too small intersection with $\tilde{T}$. Thus, the only way in which the algorithm can get a significant amount of information about $\tilde{T}$ is by evaluating $f_{\tilde{T}}$ on a set S that is small and not too likely to have a small intersection with $\tilde{T}$. Lemma 17 shows that such sets do not exist. However, before we can prove Lemma 17, we need the following known lemma.

Lemma 16 (Proved by Skala (2013) based on results of (Chvátal, 1979; Hoeffding, 1963))**.** *Consider a population of N balls, out of which M are white. Given a hypergeometric variable X measuring the number of white balls obtained by drawing uniformly at random n balls from this population, it holds for every $t \geq 0$ that $\Pr[X \geq nM/N + tn] \leq e^{-2t^2 n}$.*

Lemma 17. *For every set $S \subseteq \mathcal{N}$ whose size is less than $3k - g$, $\Pr[|S \cap \tilde{T}| \leq g] \geq 1 - e^{-\Omega(\varepsilon^3 |\mathcal{N}|)}$.*

Proof. The distribution of $|S \cap \tilde{T}|$ is hypergeometric. More specifically, it is equivalent to drawing k balls from a population of $|\mathcal{N}|$ balls, of which only $|S|$ are white. Thus, by Lemma 16, for every $t \geq 0$ we have

$$\Pr[|S \cap \tilde{T}| \geq k|S|/|\mathcal{N}| + tk] \leq e^{-2t^2 k} \enspace .$$

Setting $t = \varepsilon'$ and observing that $|S| \leq 3k - g \leq 3k$, the last inequality yields

$$\Pr[|S \cap \tilde{T}| \geq 3k^2/|\mathcal{N}| + \varepsilon' k] \leq \exp\left(-2(\varepsilon')^2 k\right) = \exp\left(-\frac{\varepsilon^2 k}{200}\right) \enspace .$$

The lemma now follows since $g \geq 3k^2/|\mathcal{N}| + \varepsilon' k$, and (by the definition of $\mathcal{N}$)

$$k \geq \frac{\varepsilon'(|\mathcal{N}| - 1)}{3} = \frac{\varepsilon(|\mathcal{N}| - 1)}{60} = \Omega(\varepsilon |\mathcal{N}|) \enspace . \qquad \square$$

Corollary 18. *For every set $S \subseteq \mathcal{N}$, $\Pr[f_\varnothing(S) = f_{\tilde{T}}(S)] \geq 1 - e^{-\Omega(\varepsilon^3 |\mathcal{N}|)}$.*

Proof. If $|S| \geq 3k - g$, then the corollary follows from Property (P4) of Proposition 15. Otherwise, it follows by combining this property with Lemma 17. □

Using the above results, we are now ready to prove an hardness result for deterministic algorithms.

Lemma 19. *Consider an arbitrary deterministic algorithm ALG for the problem $\max\{f(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ whose time complexity is bounded by some polynomial function $C(|\mathcal{N}|)$. Then, there is a large enough value k that depends only on $C(\cdot)$ and ε such that, given the random instance $\max\{f_{\tilde{T}}(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ of the above problem, the expected value of the output set of ALG is no better than $1 - e^{-\gamma} + \varepsilon$.*

Proof. Let $S_1, S_2, \ldots, S_\ell$ be the sets on which ALG evaluate $f_\varnothing$ when it is given the instance $\max\{f_\varnothing(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$, and $S_{\ell+1}$ be its output set given this instance. Let $\mathcal{E}$ be the event that $f_\varnothing(S_i) = f_{\tilde{T}}(S_i)$ for every $1 \leq i \leq \ell + 1$. By combining Corollary 18 with the union bound, we get that

$$\Pr[\mathcal{E}] \geq 1 - (\ell + 1) \cdot e^{-\Omega(\varepsilon^3 |\mathcal{N}|)} \geq 1 - [C(|\mathcal{N}|) + 1] \cdot e^{-\Omega(\varepsilon^3 |\mathcal{N}|)}) \enspace ,$$

where the second inequality holds since the time complexity of an algorithm upper bounds the number of sets on which it may evaluate $f_\varnothing$. Since $C(|\mathcal{N}|)$ is a polynomial function, by making k large enough, we can make $\mathcal{N}$ large enough to guarantee that $C(|\mathcal{N}|) \cdot e^{-\Omega(\varepsilon^3 |\mathcal{N}|)} \leq \varepsilon/20$, and thus, $\Pr[\mathcal{E}] \geq 1 - \varepsilon/20$.

When the event $\mathcal{E}$ happens, the values that ALG gets when evaluating $f_{\tilde{T}}$ on the sets $S_1, S_2, \ldots, S_\ell$ is equal to the values that it would have got if the objective function was $f_\varnothing$. Thus, in this case ALG follows the same execution path as when it gets $f_\varnothing$, and outputs $S_{\ell+1}$ whose value is

$$f_{\tilde{T}}(S_{\ell+1}) = f_\varnothing(S_{\ell+1}) \leq 1 - e^{-\gamma} + 12\varepsilon' = 1 - e^{-\gamma} + 3\varepsilon/5 \enspace ,$$

where the inequality holds by Property (P3) of Proposition 15 since the output set $S_{\ell+1}$ must be a feasible set, and thus, of size at most k. When the event $\mathcal{E}$ does not happen, we can still upper bound the value of the output set of ALG by 1 using Property (P2) of the same proposition. Thus, if we denote by R the output set of ALG, then, by the law of iterated expectations,

$$\begin{aligned}\mathbb{E}[f_{\tilde{T}}(R)] &= \Pr[\mathcal{E}] \cdot \mathbb{E}[f_{\tilde{T}}(S_{\ell+1}) \mid \mathcal{E}] + \Pr[\neg\mathcal{E}] \cdot \mathbb{E}[f_{\tilde{T}}(R) \mid \neg\mathcal{E}] \\ &\leq 1 \cdot (1 - e^{-\gamma} + 3\varepsilon/5) + (\varepsilon/20) \cdot 1 = 1 - e^{-\gamma} + 13\varepsilon/20 \leq 1 - e^{-\gamma} + \varepsilon \enspace . \end{aligned} \qquad \square$$

Lemma 19 shows that there is a single distribution of instances which is hard for every deterministic algorithm whose time complexity is bounded by a polynomial function $C(|\mathcal{N}|)$. Since a randomized algorithm whose time complexity is bounded by $C(|\mathcal{N}|)$ is a distribution over deterministic algorithms of this kind, by Yao's principle, Lemma 19 yields the next corollary.

Corollary 20. *Consider an arbitrary algorithm ALG for the problem* $\max\{f(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ *whose time complexity is bounded by some polynomial function* $C(|\mathcal{N}|)$*. Then, there is a large enough value* k *that depends only on* $C(\cdot)$ *such that, for some set* $T \subseteq \mathcal{N}$ *of size* k*, given the instance* $\max\{f_T(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ *of the above problem, the expected value of the output set of ALG is no better than* $1 - e^{-\gamma} + \varepsilon$*.*

Theorem 13 now follows from Corollary 20 because Property (P2) shows that the optimal solution for the instance $\max\{f_T(S) \mid S \subseteq \mathcal{N}, |S| \leq k\}$ mentioned by this corollary has a value of 1, and Property (P1) of the same proposition shows that this instance is an instance of the problem of maximizing a non-negative monotone γ-weakly-DR function subject to a cardinality constraint.

C.1. Proof of Proposition 15

In this section we prove Proposition 15. We begin the proof by defining the function f_T whose existence is guaranteed by the proposition. To define f_T, we first need to define the following four helper functions. Note that in $f_{T,2}$ we use the notation $[x]^+$ to denote the maximum between x and 0.

- $t_T(S) \triangleq |S \setminus T| + \min\{g, |S \cap T|\}$
- $f_{T,1}(S) \triangleq \left(1 - \dfrac{\gamma}{k-g}\right)^{\min\{t_T(S),k\}}$
- $f_{T,2}(S) \triangleq 1 - \dfrac{\min\{[t_T(S) - k]^+, k-g\}}{k-g}$
- $f_{T,3}(S) \triangleq 1 - \dfrac{\min\{|S| - t_T(S), k-g\}}{k-g}$.

Using these helper functions, we can now define f_T for every set $S \subseteq \mathcal{N}$ by

$$f_T(S) \triangleq 1 - f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) \ .$$

In the rest of this section we show that the function f_T constructed this way obeys all the properties guaranteed by Proposition 15. We begin with the following technical observation that comes handy in some of our proofs.

Observation 21. $g \leq 2\varepsilon' k + 1 \leq \min\{k-2, 3\varepsilon' k\}$.

Proof. The second inequality of the observation follows immediately from the assumptions of Proposition 15 regarding k and ε' (*i.e.*, the assumptions that $k \geq 1/\varepsilon'$ and $\varepsilon' \in (0, 1/6)$). Thus, we only need to prove the first inequality. Since $|\mathcal{N}| \geq 3k/\varepsilon'$,

$$g = \left\lceil \varepsilon' k + \frac{3k^2}{|\mathcal{N}|} \right\rceil \leq \varepsilon' k + \frac{3k^2}{3k/\varepsilon'} + 1 = 2\varepsilon' k + 1 \ . \qquad \square$$

The next three lemmata prove together Property (P1) of Proposition 15.

Lemma 22. *The outputs of the functions* $f_{T,1}$*,* $f_{T,2}$ *and* $f_{T,3}$ *are always within the range* $[0,1]$*, and thus,* f_T *is non-negative.*

Proof. We prove the lemma for every one of the functions $f_{T,1}$, $f_{T,2}$ and $f_{T,3}$ separately.

- Let $b = 1 - \gamma/(k-g)$. One can observe that $f_{T,1}$ is defined as b to the power of $\min\{t_T(S), k\}$. Thus, to show that the value of $f_{T,1}$ always belongs to the range $[0,1]$, it suffices to prove that $b \in (0,1]$ and $\min\{t_T(S), k\}$ is non-negative. The first of these claims holds since $\gamma \in (0,1]$ by assumption and $k - g \geq 2$ by Observation 21, and the second claim can be verified by looking at the definition of $t_T(S)$ and noting that g must be positive.
- Since $k - g \geq 0$ by Observation 21, $\min\{[t_T(S) - k]^+, k-g\} \in [0, k-g]$. Plugging this result into the definition of $f_{T,2}$ yields that the value of $f_{T,2}$ always belongs to $[0,1]$.
- Note that the definition of $t_T(S)$ implies $t_T(S) \leq |S|$. Together with the inequality $k - g \geq 0$, which holds by Observation 21, this guarantees $\min\{|S| - t_T(S), k-g\} \in [0, k-g]$. Plugging this result into the definition of $f_{T,3}$ yields that the value of $f_{T,3}$ always belongs to $[0,1]$. $\square$

We say that a set function $h\colon 2^{\mathcal{N}} \to \mathbb{R}$ is monotonically decreasing if $f(A) \geq f(B)$ for every two sets $A \subseteq B \subseteq \mathcal{N}$.

Lemma 23. *The functions f_T and $|S| - t_T(S)$ are monotone and the functions $f_{T,1}$, $f_{T,2}$ and $f_{T,3}$ are monotonically decreasing.*

Proof. It immediately follows from the definition of $t_T(S)$ that it is monotone. Additionally, $|S| - t_T(S)$ is a monotone function since it is equal to

$$|S| - t_T(S) = |S \cap T| - \min\{g, |S \cap T|\} = [|S \cap T| - g]^+ \enspace .$$

Plugging these observations into the definitions of $f_{T,1}$, $f_{T,2}$ and $f_{T,3}$ yields that these three functions are all monotonically decreasing. Since these three functions are also non-negative by Lemma 22, this implies that $f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S)$ is also a monotonically decreasing function, and thus, f_T is a monotone function since it is equal to 1 minus this product. $\square$

Lemma 24. *f_T is γ-weakly-DR.*

Proof. Consider arbitrary sets $A \subseteq B \subseteq \mathcal{N}$, and fix an element $u \in \mathcal{N} \setminus B$. We need to show that $f_T(u \mid A) \geq \gamma \cdot f_T(u \mid B)$, which we do by considering three cases.

The first case is when $t_T(A \cup \{u\}) = t_T(A) + 1$ and $t_T(B \cup \{u\}) = t_T(B) + 1$. Note that for every set S for which $t_T(S \cup \{u\}) = t_T(S) + 1$ and $t_T(S) < k$ we have

$$\begin{aligned} f_T(u \mid S) &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) - f_{T,1}(S \cup \{u\}) \cdot f_{T,2}(S \cup \{u\}) \cdot f_{T,3}(S \cup \{u\}) \\ &= f_{T,1}(S) \cdot f_{T,3}(S) - \left(1 - \frac{\gamma}{k-g}\right) \cdot f_{T,1}(S) \cdot f_{T,3}(S) = \frac{\gamma}{k-g} \cdot f_{T,1}(S) \cdot f_{T,3}(S) \enspace , \end{aligned} \tag{6}$$

and for every set S for which $t_T(S + u) = t_T(S) + 1$ and $t_T(S) \geq k$ we have

$$\begin{aligned} f_T(u \mid S) &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) - f_{T,1}(S \cup \{u\}) \cdot f_{T,2}(S \cup \{u\}) \cdot f_{T,3}(S \cup \{u\}) \\ &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) - f_{T,1}(S) \cdot \left[f_{T,2}(S) - \frac{\min\{[(k-g) - (t_T(S) - k)]^+, 1\}}{k-g}\right] \cdot f_{T,3}(S) \\ &= \frac{\min\{[(k-g) - (t_T(S) - k)]^+, 1\}}{k-g} \cdot f_{T,1}(S) \cdot f_{T,3}(S) \leq \frac{1}{k-g} \cdot f_{T,1}(S) \cdot f_{T,3}(S) \enspace , \end{aligned} \tag{7}$$

where the last inequality holds since $f_{T,1}$ and $f_{T,3}$ are non-negative by Lemma 22. Since f_1 and f_3 are monotonically decreasing functions (by Lemma 23), the above inequalities show $f_T(u \mid A) \geq \gamma \cdot f_T(u \mid B)$ whenever $t_T(A) < k$—if $t_T(B) < k$, then the inequalities in fact show $f_T(u \mid A) \geq f_T(u \mid B)$, but this implies $f_T(u \mid A) \geq \gamma \cdot f_T(u \mid B)$ because f_T is monotone and $\gamma \in (0, 1]$. It remains to consider the option $t(A) \geq k$. Note that when this happens, we also have $t_T(B) \geq k$ because $t_T(S)$ is a monotone function. Thus, $f_T(u \mid A) \geq f_T(u \mid B)$ because $f_{T,1}$, $f_{T,3}$ and $\min\{[(k-g) - (t_T(S) - k)]^+, 1\}$ are all non-negative monotone decreasing functions, and like in the above, this implies $f_T(u \mid A) \geq \gamma \cdot f_T(u \mid B)$.

The second case we consider is when $t_T(A \cup \{u\}) = t_T(A)$. Note that in this case we also have $t_T(B \cup \{u\}) = t_T(B)$ because the equality $t_T(A \cup \{u\}) = t_T(A)$ implies $g = \min\{|A \cap T|, g\} \leq \min\{|B \cap T|, g\} \leq g$, which implies in its turn $\min\{|B \cap T|, g\} = g$. For every set S for which $t_T(S \cup \{u\}) = t_T(S)$ and $u \notin S$ we have

$$\begin{aligned} f_T(u \mid S) &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) - f_{T,1}(S \cup \{u\}) \cdot f_{T,2}(S \cup \{u\}) \cdot f_{T,3}(S \cup \{u\}) \\ &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) - f_{T,1}(S) \cdot f_{T,2}(S) \cdot \left[f_{T,3}(S) - \frac{\min\{[(k-g) - (|S| - t_T(S))]^+, 1\}}{k-g}\right] \\ &= f_{T,1}(S) \cdot f_{T,2}(S) \cdot \frac{\min\{[(k-g) - (|S| - t_T(S))]^+, 1\}}{k-g} \enspace . \end{aligned} \tag{8}$$

Recall now that $f_{T,1}$ and $f_{T,3}$ are non-negative and monotonically decreasing functions by Lemmata 22 and 23. We additionally observe that the function $\min\{[(k-g) - (|S| - t_T(S)]^+, 1\}$ also has these properties because Lemma 23 shows that $|S| - t(S)$ is monotone. Combining these facts, we get that the expression we obtained for $f(u \mid S)$ is a monotonically decreasing function of S. Thus, $f(u \mid A) \geq f(u \mid B)$, which implies $f(u \mid A) \geq \gamma \cdot f(u \mid B)$.

The last case we need to consider is the case that $t_T(A \cup \{u\}) = t_T(A) + 1$ and $t_T(B \cup \{u\}) = t_T(B)$. The fact that $t_T(B \cup \{u\}) = t_T(B)$ implies that $u \in T$, and therefore, the fact that $t(A \cup \{u\}) = t(A) + 1$ implies that $|A \cap T| < g$ and $t_T(A) = |A|$, which induces in its turn $f_{T,3}(A) = 1$. There are now a few sub-cases to consider. If $t_T(A) < k$, then

$$f_T(u \mid A) = \frac{\gamma}{k-g} \cdot f_{T,1}(A) \geq \gamma \cdot f_{T,1}(B) \cdot f_{T,2}(B) \cdot \frac{\min\{[(k-g) - (|B| - t_T(B))]^+, 1\}}{k-g} = \gamma \cdot f_T(u \mid B) \enspace,$$

where the first equality holds by Equation (6), the last equality holds by Equation (8), and the inequality holds since $\min\{[(k-g) - (|B| - t_T(B))]^+, 1\} \in [0,1]$, $f_{T,1}(A) \geq f_{T,1}(B) \geq 0$ by Lemmata 22 and 23 and $f_{T,2}(B) \in [0,1]$ by Lemma 22 . The second sub-case we need to consider is when $k \leq t(A) \leq 2k - g - 1$. In this case

$$\begin{aligned} f_T(u \mid A) &= \frac{\min\{[(k-g) - (t_T(A) - k)]^+, 1\}}{k-g} \cdot f_{T,1}(A) = \frac{1}{k-g} \cdot f_{T,1}(A) \geq \frac{\gamma}{k-g} \cdot f_{T,1}(B) \\ &\geq \frac{\gamma \cdot \min\{[(k-g) - (|B| - t_T(B))]^+, 1\}}{k-g} \cdot f_{T,1}(B) \cdot f_{T,2}(B) = \gamma \cdot f_T(u \mid B) \enspace, \end{aligned}$$

where the first equality holds by Equation (7) and the second equality holds by Equation (8). The first inequality holds since $\gamma \in (0,1]$ and $f_{T,1}$ is non-negative and monotonicity decreasing, and the second inequality holds since $\min\{[(k-g) - (|B| - t_T(B))]^+, 1\}$ and $f_{T,2}(B)$ are both values in the range $[0,1]$ and $\gamma \cdot f_{T,1}(B)/(k-g)$ is non-negative. The final sub-case we consider is the case in which $t_T(A) \geq 2k - g - 1$. Since $|T \cap A| < g$ (but $|T \cap B| \geq g$), in this sub-case we must have $t_T(B) \geq 2k - g$, which implies $f_{T,2}(B) = 0$, and thus,

$$f_T(u \mid B) = f_{T,1}(B) \cdot f_{T,2}(B) \cdot \frac{\min\{[(k-g) - (|B| - t_T(B))]^+, 1\}}{k-g} = 0 \leq f_T(u \mid A) \enspace,$$

where the equality holds by Equation (8), and the inequality follows from the monotonicity of f_T. □

This completes the proof of Property (P1) of Proposition 15. The next lemma proves Property (P2) of this proposition.

Lemma 25. *$f_T(S) \leq 1$ for every set $S \subseteq \mathcal{N}$, and the inequality holds as an equality for $S = T$ when the size of T is exactly k.*

Proof. Since $f_{T,1}$, $f_{T,2}$ and $f_{T,3}$ all output only values within the range $[0,1]$ by Lemma 22, $f_T(S) = 1 - f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) \leq 1$. Additionally, since $g \leq k$ by Observation 21, $t_T(T) = g$ when $|T| = k$. Hence, for such T,

$$f_{T,3}(T) = 1 - \frac{\min\{k-g, k-g\}}{k-g} = 0 \enspace,$$

which implies, $f_T(T) = 1 - f_{T,1}(T) \cdot f_{T,2}(T) \cdot f_{T,3}(T) = 1$. □

The next lemma proves Property (P3) of Proposition 15.

Lemma 26. *$f_\varnothing(S) \leq 1 - e^{-\gamma} + 8\varepsilon'$ for every set S obeying $|S| \leq k$.*

Proof. Consider an arbitrary set S obeying $|S| \leq k$. Note that for such a set we have $t_\varnothing(S) = |S| \leq k$. Hence,

$$f_{\varnothing,2}(S) = f_{\varnothing,3}(S) = 1 - \frac{\min\{0, k-g\}}{k-g} = 1 \enspace.$$

Hence,

$$f_\varnothing(S) = 1 - f_{\varnothing,1}(S) \cdot f_{\varnothing,2}(S) \cdot f_{\varnothing,3}(S) = 1 - f_{\varnothing,1}(S) = 1 - \left(1 - \frac{\gamma}{k-g}\right)^{|S|} \leq 1 - \left(1 - \frac{\gamma}{k-g}\right)^k \enspace. \tag{9}$$

To prove the lemma, we need to upper bound the rightmost side of the last inequality. Towards this goal, observe that

$$\left(1 - \frac{\gamma}{k-g}\right)^k \geq \left(1 - \frac{\gamma}{k-3\varepsilon' k}\right)^k = \left(1 - \frac{\gamma}{k-3\varepsilon' k}\right)^{k-3\varepsilon' k} \cdot \left(1 - \frac{\gamma}{k-3\varepsilon' k}\right)^{3\varepsilon' k} \enspace, \tag{10}$$

where the first inequality holds since $g \leq 3\varepsilon' k$ by Observation 21. Let us now lower bound the two factors in the product on the rightmost side. First,

$$\left(1 - \frac{\gamma}{k - 3\varepsilon' k}\right)^{k - 3\varepsilon' k} \geq e^{-\gamma}\left(1 - \frac{\gamma^2}{k - 3\varepsilon' k}\right) \geq e^{-\gamma}\left(1 - 2\varepsilon'\right) \ ,$$

where the first inequality holds since the assumptions of Proposition 15 imply $k - 3\varepsilon' k \geq k/2 \geq 1$, and the second inequality holds since these assumptions include $k \geq 1/\varepsilon'$ and $\gamma \in (0, 1]$. Additionally,

$$\left(1 - \frac{\gamma}{k - 3\varepsilon' k}\right)^{3\varepsilon' k} \geq \left(1 - \frac{2}{k}\right)^{3\varepsilon' k} \geq 1 - \frac{2}{k} \cdot (3\varepsilon' k) = 1 - 6\varepsilon' \ ,$$

where the first inequality holds again since $\gamma \in (0, 1]$ and $k - 3\varepsilon' k \geq k/2$. Plugging the last two lower bounds into Inequality (10) and combining with Inequality (9), we get

$$f_{\varnothing}(S) \leq 1 - e^{-\gamma}(1 - 2\varepsilon') \cdot (1 - 6\varepsilon') \leq 1 - e^{-\gamma} \cdot (1 - 8\varepsilon') \leq 1 - e^{-\gamma} + 8\varepsilon' \ . \qquad \square$$

To complete the proof of Proposition 15, it remains to prove Property (P4), which is done by the next two observations.

Observation 27. *If $|S \cap T| \leq g$, then $f_T(S) = f_{\varnothing}(S)$.*

Proof. The only place in the definition of $f_T(S)$ in which the set T is used is in the definition of $t_T(S)$. Thus, $f_T(S) = f_{T'}(S)$ whenever $t_T(S) = t_{T'}(S)$. In particular, one can note that the condition $|S \cap T| \leq g$ implies $t_T(S) = |S| = t_{\varnothing}(S)$, and thus, f_T and $f_{\varnothing}$ must agree on the set S. $\square$

Observation 28. *The equality $f_T(S) = 1$ holds for every set S of size at least $3k - g$ and set T of size at most k.*

Proof. Note that $t_T(S) \geq |S \setminus T| \geq |S| - |T| \geq (3k - g) - k = 2k - g$. Thus,

$$f_{T,2}(S) = 1 - \frac{\min\{[t_T(S) - k]^+, k - g\}}{k - g} = 1 - \frac{k - g}{k - g} = 0 \ ,$$

which implies $f_T(S) = 1 - f_{T,1}(S) \cdot f_{T,2}(S) \cdot f_{T,3}(S) = 1$. $\square$

D. Details for Bayesian A-Optimal Design

Throughout this section, we make use of the matrix inversion lemma (also known as Woodbury Matrix Identity, Sherman-Morrison-Woodbury Formula) which is given below.

Lemma 29 (Woodbury)**.** *For matrices A, C, U, and V of the appropriate sizes,*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

In particular, for a matrix A, a vector x, and a number α, we have that

$$\left(A + \frac{1}{\alpha}xx^T\right)^{-1} = A^{-1} - \frac{A^{-1}xx^TA^{-1}}{\alpha + x^TA^{-1}x}$$

D.1. Improved Lower Bound on γ

Proof of Claim 10. Recall that

$$g(S) = Tr\left(\Sigma\right) - Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2}X_SX_S^T\right)^{-1} \ .$$

Let $A, B \subseteq \Omega$, and suppose without loss of generality that A and B are disjoint. Using Lemma 29, we show how to obtain a formula for $g(B \cup A) - g(A)$. Let $M_A = \Sigma^{-1} + \frac{1}{\sigma^2} X_A X_A^T$.

$$\begin{aligned}
g(B \cup A) - g(A) &= Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_A X_A^T\right)^{-1} - Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_{B\cup A} X_{B\cup A}^T\right)^{-1} && \text{(definition of } g\text{)} \\
&= Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_A X_A^T\right)^{-1} - Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_A X_A^T + \frac{1}{\sigma^2} X_B X_B^T\right)^{-1} \\
&= Tr\left(M_A\right)^{-1} - Tr\left(M_A + \frac{1}{\sigma^2} X_B X_B^T\right)^{-1} \\
&= Tr\left(M_A\right)^{-1} - Tr\left(M_A^{-1} - M_A^{-1} X_B \left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} X_B^T M_A^{-1}\right) && \text{(Lemma 29)} \\
&= Tr\left(M_A^{-1} X_B \left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} X_B^T M_A^{-1}\right) && \text{(linearity of trace)} \\
&= Tr\left(\left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} X_B^T M_A^{-2} X_B\right) \quad , && \text{(trace's cyclic property)}
\end{aligned}$$

where the identity matrix is of size $|B|$. From this formula, we can easily derive the marginal gain of a single element. In this case, $B = \{e\}$ and $X_B = x_e$, so the marginal gain is given by

$$g(e \mid A) = \frac{x_e^T M_A^{-2} x_e}{\sigma^2 + x_e^T M_A^{-1} x_e} \quad . \tag{11}$$

Note that $\Sigma^{-1} \preceq M_A$ (where $\preceq$ denotes the usual semidefinite ordering), and thus, M_A is positive definite. Hence, M_A^{-1} and M_A^{-2} are also positive definite, which means that their quadratic forms are non-negative. In particular, $x_e^T M_A^{-2} x_e \geq 0$ and $x_e^T M_A^{-1} x_e \geq 0$, which implies $g(e \mid A) \geq 0$. Also note that $g(\varnothing) = 0$. Combining this equality with the previous inequality, we get that g is non-negative and monotone increasing.

Now we seek to show the lower bound on γ. Again, let $A, B \subseteq \Omega$, and assume without loss of generality that A and B are disjoint. We seek to lower bound the ratio

$$\frac{\sum_{e \in B} g(e \mid A)}{g(B \cup A) - g(A)} \quad . \tag{12}$$

Let $s = \max_{e \in \Omega} \|x_e\|_2$. Observe that

$$\sigma^2 + x_e^T M_A^{-1} x_e = \sigma^2 + \|x_e\|^2 \left(\frac{x_e^T M_A^{-1} x_e}{\|x_e\|^2}\right) \leq \sigma^2 + s^2 \lambda_{\max}\left(M_A^{-1}\right) = \sigma^2 + s^2 \lambda_{\max}\left(\Sigma\right) \quad , \tag{13}$$

where the first inequality follows from the Courant-Fischer theorem, i.e., the variational characterization of eigenvalues. The second inequality is derived as follows: $M_A = \Sigma^{-1} + \frac{1}{\sigma} X_A X_A^T$ and so $M_A \succeq \Sigma^{-1}$. This means that $M_A^{-1} \preceq \Sigma$. Thus, $\lambda_{max}(M_A^{-1}) \leq \lambda_{max}(\Sigma)$. Using this, we may obtain a lower bound on the numerator in (12).

$$\begin{aligned}
\sum_{e \in B} g(e \mid A) &= \sum_{e \in B} \frac{x_e^T M_A^{-2} x_e}{\sigma^2 + x_e^T M_A^{-1} x_e} && \text{(by (11))} \\
&= \sum_{e \in B} \frac{Tr\left(x_e x_e^T M_A^{-2}\right)}{\sigma^2 + x_e^T M_A^{-1} x_e} && \text{(cyclic property of trace)} \\
&\geq \frac{1}{\sigma^2 + s^2 \lambda_{\min}\left(M_A\right)} \sum_{e \in B} Tr\left(x_e x_e^T M_A^{-2}\right) && \text{(by (13))} \\
&= \frac{Tr\left(X_B X_B^T M_A^{-2}\right)}{\sigma^2 + s^2 \lambda_{\min}\left(M_A\right)} && \text{(linearity of trace)} \\
&= \frac{Tr\left(X_B^T M_A^{-2} X_B\right)}{\sigma^2 + s^2 \lambda_{\min}\left(M_A\right)} \quad . && \text{(cyclic property of trace)}
\end{aligned}$$

Now, we will bound the denominator of (12). We have already shown that

$$g(B \cup A) - g(A) = Tr\left(\left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} X_B^T M_A^{-2} X_B\right) \quad .$$

Additionally, we have shown that M_A^{-1} is positive semidefinite, and thus, $X_B^T M_A^{-1} X_B$ is also positive semidefinite. Hence, $\sigma^2 I \preceq \sigma^2 I + X_B^T M_A^{-1} X_B$. This implies that $\left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} \preceq \left(\sigma^2 I\right)^{-1} = \frac{1}{\sigma^2} I$. Finally, we have shown that M_A^{-2} is positive semidefinite, and therefore, we have that $X_B^T M_A^{-2} X_B$ is also positive semidefinite. Thus,

$$g(B \cup A) - g(A) = Tr\left(\left(\sigma^2 I + X_B^T M_A^{-1} X_B\right)^{-1} X_B^T M_A^{-2} X_B\right) \leq \frac{1}{\sigma^2} Tr\left(X_B^T M_A^{-2} X_B\right) \quad .$$

Applying these bound on $\sum_{e \in B} g(e \mid A)$ and $g(A \cup B) - g(A)$, we obtain

$$\frac{\sum_{e \in B} g(e \mid A)}{g(B \cup A) - g(A)} \geq \left(\frac{\sigma^2}{\sigma^2 + s^2 \lambda_{max}(\Sigma)}\right) \frac{Tr\left(X_B^T M_A^{-2} X_B\right)}{Tr\left(X_B^T M_A^{-2} X_B\right)} = \left(1 + \frac{s^2}{\sigma^2} \lambda_{max}(\Sigma)\right)^{-1} \quad . \qquad \square$$

D.2. Faster Greedy Search Implementation

As discussed in Section 5.1, the Matrix Inversion lemma may be used to greatly speed up the greedy search in each algorithm. The naive approach of computing

$$g(e \mid S) = Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_S X_S^T\right)^{-1} - Tr\left(\Sigma^{-1} + \frac{1}{\sigma^2} X_{S \cup e} X_{S \cup e}^T\right)^{-1}$$

by constructing these matrices, explicitly computing inverses, and computing its trace is not only expensive (inversion alone takes $O(d^3)$ time), but also not memory-efficient. Instead, one can use the simplified formula given in the proof of Claim 10 to compute $g(e \mid S)$ quickly, namely that

$$g(e \mid S) = \frac{x_e^T M_S^{-2} x_e}{\sigma^2 + x_e^T M_S^{-1} x_e} \ ,$$

where $M_S = \Sigma^{-1} + \frac{1}{\sigma^2} X_S X_S^T$. In fact, M_S^{-1} may be stored and updated directly in each iteration using the matrix inversion lemma so that *no matrix inversion are required*. Note that $M_\varnothing^{-1} = \Sigma$, which is an input parameter. By the matrix inversion lemma,

$$M_{S \cup e}^{-1} = M_S^{-1} - \frac{M_S^{-1} x_e x_e^T M_S^{-1}}{\sigma^2 + x_e^T M_S^{-1} x_e} \ ,$$

which takes $O(d^2)$ arithmetic operations. Once M_S^{-1} is known explicitly, computing $g(e \mid S)$ is simply matrix-vector multiplication on a fixed matrix. We found that this greatly improved the efficiency of our code.