A. Proof of Lemma 3.1

Lemma A.1. *(usul, 2016) There exists an instance of a BP maximization problem that can not be approximately solved to any positive factor in polynomial time.*

Proof. We consider the BP problem with ground set n and a cardinality constraint $|X| \leq k = n/2$. Let $R \subseteq V$ be an arbitrary set with $|R| = k$. Let $f = 0$ and $g'(X) = \max(|X| - k, 0)$ so that $g'(X) = 0$ for all $|X| = k$. $g'(X)$ is clearly supermodular.

Let $g(X) = g'(X)$ for all $X \neq R$ but $g(R) = 0.5$. We notice that for $X \subset V$ and $v \notin X$, $g(v|X) = 0$ if $|X| \leq k-2$, $g(v|X) = 0$ or 0.5 if $|X| = k-1$, $g(v|X) = 0.5$ or 1 if $|X| = k$, and $g(v|X) = 1$ if $|X| \geq k+1$. Immediately, we have for all $X \subset Y \subset V$ and $v \notin Y$, $g(v|X) \leq g(v|Y)$. Therefore, $g(X)$ is also supermodular.

Next, we use a proof technique similar to (Svitkina & Fleischer, 2011). Note that $g'(X) = g(X)$ if and only if $X \neq R$. So for any algorithm maximizing $g(X)$, before it evaluates $g(R)$, all function evaluations are the same with maximizing $g'(X)$. Additionally, since $g'(X) = \max(|X| - k, 0)$, it is permutation symmetric. Therefore, the algorithm can only do random search to find R. If the algorithm acquires a polynomial number $O(n^m)$ of sets of size k, the probability of finding R is $\frac{O(n^m)}{\binom{n}{k}} \leq \frac{O(n^m)}{(n/k)^k} = \frac{O(n^m)}{2^{n/2}} \leq O(2^{-n/2+\epsilon n})$ for all $\epsilon > 0$. Therefore, no polynomial time algorithm can distinguish g and g' with probability greater than $1 - O(2^{-n/2+\epsilon n})$ and will return 0 in almost all cases.

Hence, we have $\max_{|X| \leq k} f(X) + g(X) = 0.5 > 0$ so no polynomial algorithm can do better than $\max_{|X| \leq k} f(X) + g'(X) = 0$ with high probability, or has any positive guarantee. □

B. Proof of Lemma 3.2

Lemma B.1. *There exists a monotonic non-decreasing set function h that is not BP decomposable.*

Proof. Let $h(X) = \min(\max(|X|, 1), 3) - 1$. This function is monotonic, and we wish to show it is not BP decomposable. Let $A \subset B$ be subsets of V with $|A| = 1$ and $|B| = 3$. Let $v \in V \setminus B$. We calculate that $h(v|\emptyset) = 0$, $h(v|A) = 1$, $h(v|B) = 0$. So $h(v|\emptyset) + h(v|B) < h(v|A)$.

Assume $h(X) = f(X) + g(X)$ where f is submodular, g is supermodular and both are monotonic non-decreasing. We have $f(v|\emptyset) + f(v|B) \geq f(v|\emptyset) \geq f(v|A)$ and $g(v|\emptyset) + g(v|B) \geq g(v|B) \geq g(v|A)$. Therefore $h(v|\emptyset) + h(v|B) \geq h(v|A)$ by summing the two inequalities, which is a contradiction. We thus have that h is not BP decomposable. □

C. Key Theorem Regarding Curvature

In this section, we state and then prove the following important theorem that holds for any chain of sets, not just those produced by the greedy algorithm.

Lemma C.1. *For any chain of solutions $\emptyset = S_0 \subset S_1 \subset \ldots \subset S_k$, where $|S_i| = i$, the following holds for all $i = 0 \ldots k-1$,*

$$h(X^*) \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + h(X^* \setminus S_i | S_i) \tag{10}$$

where $\{s_i\} = S_i \setminus S_{i-1}$, $a_i = h(s_i|S_{i-1})$ and X^* is the optimal set.

Proof. For any $i = 0, \ldots, k-1$, we focus on the term $h(X^* \cup S_i)$.

According to basic set operations,

$$h(X^* \cup S_i) = h(S_i) + h(X^*|S_i) \tag{11}$$

$$= \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + h(X^* \setminus S_i | S_i). \tag{12}$$

We can also express $h(X^* \cup S_i)$ the other way around, $h(X^* \cup S_i) = h(X^*) + h(S_i \setminus X^* | X^*)$. Since we already have an order of element in S_i, we can expand $h(S_i \setminus X^* | X^*)$. When adding s_j to the context $S_{j-1} \cup X^*$ we do not need add elements that are not in $S_i \setminus X^*$ since $h(s_j | X^* \cup S_{j-1}) = 0$ if $s_j \in X^*$. Thus, using Lemma C.2 (i), we get $h(X^* \cup S_i) = h(X^*) + \sum_{j: s_j \in S_i \setminus X^*} h(s_j | X^* \cup S_{j-1}) \geq h(X^*) + (1 - \kappa_f) \sum_{j: s_j \in S_i \setminus X^*} h(s_j | S_{j-1})$.

Therefore, we have inequalities on both sides of $h(X^* \cup S_i)$ and we can join them together to get:

$$h(X^*) + (1 - \kappa_f) \sum_{j: s_j \in S_i \setminus X^*} a_j \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + h(X^* \setminus S_i | S_i), \tag{13}$$

or

$$h(X^*) \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + h(X^* \setminus S_i | S_i). \tag{14}$$

□

We begin with the following four-part lemma,

Lemma C.2. *For a BP function $h(X) = f(X) + g(X)$, we have*

(i) $h(v|Y) \geq (1-\kappa_f)h(v|X)$ *for all* $X \subseteq Y \subset V$ *and* $v \notin Y$

(ii) $h(v|Y) \leq \frac{1}{1-\kappa^g}h(v|X)$ *for all* $X \subseteq Y \subset V$ *and* $v \notin Y$

(iii) $h(X|Y) \geq (1-\kappa_f)\sum_{v \in X \setminus Y} h(v|Y)$ *for all* $X, Y \subseteq V$

(iv) $h(X|Y) \leq \frac{1}{1-\kappa^g}\sum_{v \in X \setminus Y} h(v|Y)$ *for all* $X, Y \subseteq V$

Proof. (i) $\kappa_f = 1 - \min_{v \in V} \frac{f(v|V\setminus\{v\})}{f(v)}$, therefore, $f(v|V \setminus \{v\}) \geq (1-\kappa_f)f(v)$ for all v.

So we have $f(v|Y) \geq f(v|V\setminus\{v\}) \geq (1-\kappa_f)f(v) \geq (1-\kappa_f)f(v|X)$ and $g(v|Y) \geq g(v|X) \geq (1-\kappa_f)g(v|X)$ for all $X \subseteq Y \subset V$ and $v \notin Y$. Therefore, $h(v|Y) \geq (1-\kappa_f)h(v|X)$ for all $X \subset Y \subseteq V$ and $v \notin Y$.

(ii) $\kappa^g = 1 - \min_{v \in V} \frac{g(v)}{g(v|V\setminus\{v\})}$, therefore, $g(v|V \setminus \{v\}) \leq \frac{1}{1-\kappa^g}g(v)$ for all v.

So we have $g(v|Y) \leq g(v|V \setminus \{v\}) \leq \frac{1}{1-\kappa^g}g(v) \leq \frac{1}{1-\kappa^g}g(v|X)$ and $f(v|Y) \leq f(v|X) \leq \frac{1}{1-\kappa^g}f(v|X)$ for all $X \subseteq Y \subset V$ and $v \notin Y$. Therefore, $h(v|Y) \leq \frac{1}{1-\kappa^g}h(v|X)$ for all $X \subset Y \subseteq V$ and $v \notin Y$.

(iii) Let $X \setminus Y$ be $\{v_1, \ldots, v_m\}$, $h(X|Y) = \sum_{i=1,2,\ldots,m} h(v_i|Y \cup \{v_1\} \cup \{v_2\} \cup \ldots \cup \{v_{i-1}\}) \geq (1-\kappa_f)\sum_{i=1,2,\ldots,m} h(v_i|Y) = (1-\kappa_f)\sum_{v \in X\setminus Y} h(v|Y)$, according to (i).

(iv) Let $X \setminus Y$ be $\{v_1, \ldots, v_m\}$, $h(X|Y) = \sum_{i=1,2,\ldots,m} h(v_i|Y \cup \{v_1\} \cup \{v_2\} \cup \ldots \cup \{v_{i-1}\}) \leq \frac{1}{1-\kappa^g}\sum_{i=1,2,\ldots,m} h(v_i|Y) = \frac{1}{1-\kappa^g}\sum_{v \in X\setminus Y} h(v|Y)$, according to (ii).

□

D. Proof of Lemma 3.5

Lemma D.1. GREEDMAX *is guaranteed to obtain a solution $\hat{X}$ such that*

$$h(\hat{X}) \geq \frac{1}{\kappa_f}\left[1 - \left(1 - \frac{(1-\kappa^g)\kappa_f}{k}\right)^k\right] h(X^*) \tag{5}$$

where $X^* \in \operatorname{argmax}_{|X| \leq k} h(X)$, $h(X) = f(X) + g(X)$, κ_f is the curvature of submodular f and κ^g is the curvature of supermodular g.

Proof. According to Lemma C.1, for all $i = 0, \ldots, k-1$,

$$h(X^*) \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + h(X^* \setminus S_i|S_i) \tag{15}$$

Since GREEDMAX is choosing the feasible element with the largest gain, we have $h(v|S_i) \leq h(s_{i+1}|S_i)$ for all feasible $v \in X^*$. In fact, all elements in $X^* \setminus S_j$ are feasible since we are considering a cardinality constraint and $|S_j| \leq k-1$. Also, $|X^* \setminus S_j| = |X^*| - |X^* \cap S_j| = k - |X^* \cap S_j|$, and therefore from Lemma C.1 and Lemma C.2(iv), we have that:

$$h(X^*) \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + \frac{k - |X^* \cap S_i|}{1-\kappa^g} a_{i+1} \tag{16}$$

Next, we use a nested lemma, Lemma D.2, to get Equation (5).

Lemma D.2. *Given any chain of solutions $\emptyset = S_0 \subset S_1 \subset \ldots \subset S_k$ such that $|S_i| = i$, if the following holds for all $i = 0 \ldots k-1$:*

$$h(X^*) \leq \alpha \sum_{j: s_j \in S_i \setminus X^*} a_j + \sum_{j: s_j \in S_i \cap X^*} a_j + \frac{k - |X^* \cap S_i|}{1-\beta} a_{i+1} \tag{17}$$

where $0 \leq \alpha, \beta \leq 1$ and $s_i = S_i \setminus S_{i-1}$, and $a_i = h(s_i|S_{i-1})$, then we have

$$h(S_k) \geq \frac{1}{\alpha}\left[1 - \left(1 - \frac{(1-\beta)\alpha}{k}\right)^k\right] h(X^*). \tag{18}$$

Proof. Assume $\beta < 1$ as otherwise the bound is immediate. This lemma aims to show one inequality (Equation (18)) based on k other inequalities (Equation (17)) with k variables $a_1, \ldots, a_k$. In the inequalities, $s_j \in S_k \cap X^*$ and $s_j \in S_k \setminus X^*$ are not treated identically. We will, in fact, correspondingly treat the indices of the elements in $S_k \cap X^*$ as parameters. Recall, $S_k = \{s_1, s_2, \ldots, s_k\}$ is an ordered set and S_k has index set $\{1, 2, \ldots, k\} = [k]$. Let $B = \{b_1, \ldots, b_p\} \subseteq [k]$ be the set of indices of $S_k \cap X^*$ where b_i's are in increasing order (so $b_i < b_{i+1}$) and $p = |S_k \cap X^*|$. Thus, $i \in B$ means $s_i \in S_k \cap X^*$, and $i \in [k] \setminus B$ means $s_i \in S_k \setminus X^*$.

Our next step is to view this problem as a set of parameterized (by B) linear programming problems. Each linear programming problem is characterized as finding:

$$T(B) = T(b_1, b_2, \dots, b_p) = \min_{a_1, a_2, \dots, a_k} \sum_{i=1}^{k} a_i \tag{19}$$

subject to

$$h(X^*) \le \alpha \sum_{j \in [i-1] \setminus B_{i-1}} a_j + \sum_{j \in B_{i-1}} a_j + \frac{k - |B_{i-1}|}{1-\beta} a_i, \tag{20}$$

for $i = 1, \dots, k$,where $B_i = \{b \in B | b \le i\}$. In this LP problem, $a_1, \dots, a_k$ are non-negative variables, and k, α, β and $h(X^*)$ are fixed values. Different indices $B = \{b_1, b_2, \dots, b_p\}$ define different LP problems, and our immediate goal is to show that $T(\emptyset) \le T(b_1, b_2, \dots, b_p)$ for all $b_1, b_2, \dots, b_p$ and $p \ge 0$. In the below, we will use $\Upsilon(B, a, i)$ to refer to the right hand side of Equation (20) for a given set B, vector a, and index $i = 1, \dots, k$, and hence Equation (20) becomes $h(X^*) \le \Upsilon(B, a, i)$ for $i = 1, \dots, k$. Note that $\Upsilon(B, a, i)$ is linear in a with non-negative coefficients.

First, we show that there exists an optimal solution[3] $a_1, a_2, \dots, a_k$ s.t. for all $r \le k-1$ with $r \in B$, $a_r \le a_{r+1}$. Let r_a be the largest r s.t. $r \le k-1, r \in B$ and $a_r > a_{r+1}$; if such an r does not exist, let $r_a = 0$. Our goal here is equivalent to showing, for any feasible solution $\{a_i\}_{i=1}^k$ with $r_a > 0$, we can create another feasible solution $\{a'_i\}_{i=1}^k$ with $r_{a'} = 0$ and the objective $\sum_{i=1}^k a'_i \le \sum_{i=1}^k a_i$. We do this iteratively, by in each step showing that for any feasible solution $\{a_i\}_{i=1}^k$ with $r_a > 0$, we can create another feasible solution $\{a'_i\}_{i=1}^k$ with $r_{a'} \le r_a - 1$ and with objective having $\sum_{i=1}^k a'_i \le \sum_{i=1}^k a_i$. Repeating this argument leads ultimately to $r_{a'} = 0$.

Let $r = r_a$ for notational simplicity. Consider the r^{th} and $(r+1)^{\text{th}}$ inequalities:

$$h(X^*) \le \alpha \sum_{j \le [r-1] \setminus B_{r-1}} a_j + \sum_{j \in B_{r-1}} a_j + \frac{k - |B_{r-1}|}{1-\beta} a_r \tag{21}$$

and

$$h(X^*) \le \alpha \sum_{j \le [r-1] \setminus B_{r-1}} a_j + \sum_{j \in B_{r-1}} a_j + a_r \tag{22}$$

$$+ \frac{k - |B_{r-1}| - 1}{1-\beta} a_{r+1}. \tag{23}$$

Since $a_r > a_{r+1}$ and $\beta < 1$, $\frac{k-|B_{r-1}|}{1-\beta} a_r > \frac{k-|B_{r-1}|-1}{1-\beta} a_{r+1} + a_r$ and thus the r.h.s. of Eq. (21) is always strictly larger than the r.h.s. of Eq. (23).

[3] Optimal in this case means for the LP, distinct from the optimal BP maximization solution X^*.

Therefore, Eq. (21) is not tight and it is possible to decrease a_r a little bit. Let $\{a'_i\}$ be another set of solutions with $a'_i = a_i$ for all $i = 1, 2, \dots, r-1$; $a'_r = a_r - \epsilon$; $a'_i = a_i + \epsilon/(k - |B_r|)$ for $i = r+1, r+2, \dots, k$ and $\epsilon = \left[1 - \frac{1-\beta}{k-|B_{r-1}|}\right] [a_r - a_{r+1}]$. It is easy to see that $\epsilon > 0$ since $|B_{r-1}| \le r-1 \le k-2$.

Below, we show that $a'_r \le a'_{r+1}$. First, we notice $\sum_{i=1}^k a'_i \le \sum_{i=1}^k a_i$ since $|B_r| \le r$ and $-\epsilon + \frac{k-r}{k-|B_r|}\epsilon \le 0$. Next, we want to show that $a'_1, a'_2, \dots, a'_k$ is still feasible. As mentioned above, define $\Upsilon(B, a, i) = \alpha \sum_{j \in [i-1] \setminus B_{i-1}} a_j + \sum_{j \in B_{i-1}} a_j + \frac{k-|B_{i-1}|}{1-\beta} a_i$.

We examine if $h(X^*) \le \Upsilon(B, a', i)$ or not for each i.

1. For $i = 1, 2, \dots, r-1$, $\Upsilon(B, a', i) = \Upsilon(B, a, i) \ge h(X^*)$.
2. For $i = r$, $\Upsilon(B, a', r) - \Upsilon(B, a, r+1) = \frac{k-|B_{r-1}|}{1-\beta}[a_r - \epsilon] - a_r - \frac{k-|B_{r-1}|-1}{1-\beta} a_{r+1} \ge \frac{k-|B_{r-1}|}{1-\beta}[a_r - a_{r+1}] + a_{r+1} - a_r - \frac{k-|B_{r-1}|}{1-\beta}\epsilon = \left[\frac{k-|B_{r-1}|}{1-\beta} - 1\right][a_r - a_{r+1}] - \frac{k-|B_{r-1}|}{1-\beta}\left[1 - \frac{1-\beta}{k-|B_{r-1}|}\right][a_r - a_{r+1}] = 0$. So $\Upsilon(B, a', r) \ge \Upsilon(B, a, r+1) \ge h(X^*)$.
3. For $i = r+1, r+2, \dots, k$, we compare $\Upsilon(B, a', i)$ with $\Upsilon(B, a, i)$. Note that $\Upsilon(B, a, i) = \alpha \sum_{j \in [i-1] \setminus B_{i-1}} a_j + \sum_{j \in B_{i-1}} a_j + \frac{k-|B_{i-1}|}{1-\beta} a_i$ and it has three terms, that we consider individually.
   (a) The first term is not decreasing since $a'_i < a_i$ only if $i = r$, but $r \notin [i-1] \setminus B_{i-1}$. The increment therefore is at least 0.
   (b) a_r appears in the second term once, and when changing to a'_r, will decreases the value by ϵ. However, $a'_j = a_j + \epsilon/(k - |B_r|)$ for all $j = r+1, r+2, \dots, k$. Immediately, we notice the number of such a_j in the second term is $\sum_{j \in B_{i-1}, j \ge r+1} 1 = \sum_{j \in B_{i-1}, j \notin B_r} 1 = |B_{i-1}| - |B_r|$. So the increment of the second term is $\frac{|B_{i-1}| - |B_r|}{k - |B_r|}\epsilon - \epsilon$.
   (c) The third term is increased by $\frac{k-|B_{i-1}|}{(1-\beta)(k-|B_r|)}\epsilon \ge \frac{k-|B_{i-1}|}{k-|B_r|}\epsilon$.

   So overall, the increment is greater than or equal to $\frac{|B_{i-1}|-|B_r|}{k-|B_r|}\epsilon - \epsilon + \frac{k-|B_{i-1}|}{k-|B_r|}\epsilon \ge 0$, which means $\Upsilon(B, a', i) \ge \Upsilon(B, a, i) = h(X^*)$.

Therefore, $\{a'_i\}_{i=1}^k$ still satisfies all the constraints but $\sum_{i=1}^k a'_k \le \sum_{i=1}^k a_k$. Note that $r = r_a = \max(\{r' \in B | r' \le k-1, a_{r'} > a_{r'+1}\})$ by definition. And we have $a'_i = a_i + \frac{\epsilon}{k-|B_r|}$ for $i = r+1, r+

$2, \ldots, k$. Therefore, $a'_{r'} \le a'_{r'+1}$ for any $r' \in B \cap [r+1, k-1]$. Next we calculate $a'_r - a'_{r+1} = a_r - a_{r+1} - \epsilon - \frac{\epsilon}{k-|B_r|} = \left[1 - \left(1 + \frac{1}{k-|B_r|}\right)\left(1 - \frac{1-\beta}{k-|B_{r-1}|}\right)\right](a_r - a_{r+1}) \le 0$. Therefore, $a'_{r'} \le a'_{r'+1}$ for all $r' \in B \cap [r, k-1]$ which implies $r_{a'} \le r_a - 1$.

By repeating the above steps, we can get a feasible solution $\{a''\}$ s.t. $r_{a''} = 0$ and $\sum_{i=1}^k a''_k \le \sum_{i=1}^k a_k$. Therefore, from any optimal solution $\{a_i\}_{i=1}^k$, we can also create another optimal solution $\{a''_i\}$ s.t. for all $r \in B$ and $r \le k-1$, we have $a''_r \le a''_{r+1}$. W.l.o.g, we henceforth consider only the optimal solutions $\{a_i\}_{i=1}^k$ with $r_a = 0$.

Second, we assume $r \in B$ but $r+1 \notin B$ for some $r \le k-1$. We can create $B' = B \cup \{r+1\} \setminus \{r\}$ and show for all $\{a_i\}_{i=1}^k$ that satisfies the constraints of B, $\{a_i\}_{i=1}^k$ will also still satisfy the constraints of B' by showing that $\Upsilon(B', a, i) \ge \Upsilon(B, a, i)$ for $i = 1, \ldots, k$. We consider each i in turn.

1. For $i = 1, 2, \ldots, r$, $\Upsilon(B, a, i) = \Upsilon(B', a, i)$.
2. If $i = r+1$, we notice a_r moves from the second term to the first, and the third term is changed from $\frac{k-|B_r|}{1-\beta} a_{r+1}$ to $\frac{k-|B'_r|}{1-\beta} a_{r+1}$ and $|B'_r| = |B_r| - 1$. So the overall value is increased by $\Upsilon(B', a, i) - \Upsilon(B, a, i) = \frac{1}{1-\beta} a_{r+1} - (1-\alpha) a_r \ge 0$ since $a_r \le a_{r+1}$.
3. For $i = r+2, r+3, \ldots, k$, we notice that the third term does not change but a_r moves from the second term to the first and a_{r+1} moves from the first term to the second. Thus, the value is increased by $\Upsilon(B', a, i) - \Upsilon(B, a, i) = (1-\alpha)(a_{r+1} - a_r) \ge 0$ since $a_r \le a_{r+1}$.

Since $\Upsilon(B', a, i) \ge \Upsilon(B, a, i)$ for $i = 1, \ldots, k$, we have that $T(B') \le T(B)$. Therefore, if we see two indexes in B differ by at least 2, we can increase the first index by 1. Repeating this process, we get

$$T(B) \ge T(k-p+1, k-p+2, \ldots, k). \tag{24}$$

Third, if $\{a_i\}_{i=1}^k$ satisfies the constraints for $B = \{k-p+1, k-p+2, \ldots, k\}$and $a_{k-p+1} \le \ldots \le a_k$, then $\{a_i\}_{i=1}^k$ also must satisfy the constraints for $B' = \{k-p+2, k-p+3, \ldots, k\}$. We show that $\Upsilon(B', a, i) \ge \Upsilon(B, a, i)$ for $i = 1, \ldots, k$ and again consider each i in turn.

1. For $i = 1, 2, \ldots, k-p+1$, $\Upsilon(B', a, i) = \Upsilon(B, a, i)$.
2. For $i = k-p+2, k-p+3, \ldots, k$, the change of the value is $\Upsilon(B', a, i) - \Upsilon(B, a, i) = (\alpha - 1) a_{k-p+1} + \frac{1}{1-\beta} a_i$. We notice that $a_i \ge a_{k-p+1}$ since $k-p+1, k-p+2, \ldots, i-1 \in B$. Thus, we have $\Upsilon(B', a, i) - \Upsilon(B, a, i) \ge 0$ and correspondingly $T(B) \ge T(B')$.

Repeating this process, therefore, we have that

$$T(B) \ge T(\emptyset) \tag{25}$$

Next, we calculate $T(\emptyset)$. For $B = \emptyset$ and any feasible (for Equation (20)) $a_1, a_2, \ldots, a_k$, let T_i be the partial sum $T_i = \sum_{j=1}^i a_j$ for $i = 0, \ldots, k$ with $T_0 = 0$. We get, for $i = 1, \ldots, k$ that $h(X^*) \le \Upsilon(\emptyset, a, i)$ which takes the form

$$h(X^*) \le \alpha \sum_{j \in [i-1]} a_j + \frac{k}{1-\beta} a_i, \tag{26}$$

which is the same as

$$h(X^*) \le \alpha T_{i-1} + \frac{k}{1-\beta}(T_i - T_{i-1}), \tag{27}$$

and also, after multiplying both sides by $(1-\beta)/k$ and then adding $(1/\alpha)h(X^*)$ to both sides, the same as

$$\frac{1}{\alpha} h(X^*) - T_i \le \left(1 - \frac{(1-\beta)\alpha}{k}\right)\left(\frac{1}{\alpha} h(X^*) - T_{i-1}\right). \tag{28}$$

(29)

We then repeatedly apply all k inequalities from $i = k, \ldots, 1$, to get

$$\frac{1}{\alpha} h(X^*) - T_k \le \left(1 - \frac{(1-\beta)\alpha}{k}\right)^k \left(\frac{1}{\alpha} h(X^*) - T_0\right) \tag{30}$$

yielding

$$T_k \ge \frac{1}{\alpha}\left[1 - \left(1 - \frac{(1-\beta)\alpha}{k}\right)^k\right] h(X^*). \tag{31}$$

Let $\gamma = \frac{1}{\alpha}\left[1 - \left(1 - \frac{(1-\beta)\alpha}{k}\right)^k\right]$. So, for $B = \emptyset$ and any feasible $a_1, a_2, \ldots, a_k$, we have $\sum_{j=1}^k a_j = T_k \ge \gamma h(X^*)$. Therefore $T(\emptyset) = \min_{a_1, a_2, \ldots, a_k} \sum_{i=1}^k a_i \ge \gamma h(X^*)$.

Recall that $T(B) \ge T(\emptyset)$ for all B. We thus have, with $a_i = h(s_i | \{s_1, \ldots, s_{i-1}\})$ (which are also feasible for Equation (20) with B again the indices of $S_k \cap X^*$, which follows from Equation 16), $h(S_k) = \sum_i^k a_i \ge T(B) \ge T(\emptyset) \ge \gamma h(X^*)$. ☐

Lemma D.2 yields Equation (5) which shows the result for Lemma 3.5. ☐

E. Weaker bound in the cardinality constrained case

The bound in Equation (6) is one of the major contributions of this paper. Another bound can be achieved using a surrogate objective $h'(X) = f(X) + \sum_{v \in X} g(v)$, similar to an approach used in (Iyer et al., 2013a). We have that $h'(X) \leq h(X)$ thanks to the supermodularity of g, and we can apply GREEDMAX directly to h', the solution of which has a guarantee w.r.t. the original objective h. The proof of this bound is quite a bit simpler, so we first offer it here immediately. On the other hand, we also show that the bound obtained by this method is worse than Equation (6) for all $0 < \kappa_f, \kappa^g < 1$, sometimes appreciably.

Lemma E.1. Weak bound in cardinality constrained case. GREEDMAX *maximizing* $h'(X) = f(X) + \sum_{v \in X} g(v)$ *is guaranteed to obtain a solution* $\hat{X}$ *such that*

$$h(\hat{X}) \geq \frac{1-\kappa^g}{\kappa_f}\left[1 - e^{-\kappa_f}\right] h(X^*) \tag{32}$$

where $X^* \in \operatorname{argmax}_{|X| \leq k} h(X)$*,* $h(X) = f(X) + g(X)$*,* κ_f *is the curvature of submodular* f *and* κ^g *is the curvature of supermodular* g*.*

Proof. According to lemma C.2 (iv), $(1 - \kappa^g)h(X) \leq h'(X)$ for all $X \subseteq V$. Also we have $h'(X) \leq h(X)$. And h' is a monotone submodular function with $\kappa_{h'} = 1 - \min_{v \in V} \frac{h'(v|V\setminus\{v\})}{h'(v)} = 1 - \min_{v \in V} \frac{f(v|V\setminus\{v\}) + g(v)}{f(v)+g(v)} \leq 1 - \min_{v \in V} \frac{f(v|V\setminus\{v\})}{f(v)} = \kappa_f$ since $0 \leq f(v|V \setminus \{v\}) \leq f(v)$.

Using the traditional curvature bound for submodular maximization (Conforti & Cornuejols, 1984), the greedy algorithm to maximize h' provides a solution $\hat{X}$ s.t. $h'(\hat{X}) \geq \frac{1}{\kappa_{h'}}\left[1 - e^{-\kappa_{h'}}\right] h'(X^*)$ where $X^* \in \operatorname{argmax}_{|X| \leq k} h(X)$. Thus, we have

$$h(\hat{X}) \geq h'(\hat{X}) \geq \frac{1}{\kappa_{h'}}\left[1 - e^{-\kappa_{h'}}\right] h'(X^*) \tag{33}$$

$$\geq \frac{1}{\kappa_f}\left[1 - e^{-\kappa_f}\right] h'(X^*) \tag{34}$$

$$\geq \frac{1-\kappa^g}{\kappa_f}\left[1 - e^{-\kappa_f}\right] h(X^*) \tag{35}$$

□

Next, we show that this bound is almost everywhere worse than Equation (6).

Lemma E.2. $\frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] \geq \frac{1-\kappa^g}{\kappa_f}\left[1 - e^{-\kappa_f}\right]$ *for all* $0 \leq \kappa_f, \kappa^g \leq 1$ *where equality holds if and only if* $\kappa_f = 0$ *or* $\kappa^g = 0$ *or* $\kappa^g = 1$*. For simplicity, dividing by 0 is defined using limits, e.g.,* $\frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] = \lim_{\kappa_f \to 0^+} \frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] = 1 - \kappa^g$ *when* $\kappa_f = 0$*.*

Proof. Let $\phi(\kappa_f, \kappa^g) = \frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right]$ and $\psi(\kappa_f, \kappa^g) = \frac{1-\kappa^g}{\kappa_f}\left[1 - e^{-\kappa_f}\right]$. Specifically, $\phi(0, \kappa^g) = \lim_{\kappa_f \to 0^+} \phi(\kappa_f, \kappa^g) = 1 - \kappa^g$ and $\psi(0, \kappa^g) = \lim_{\kappa_f \to 0^+} \psi(\kappa_f, \kappa^g) = 1 - \kappa^g$. So if $\kappa_f = 0$, $\phi(\kappa_f, \kappa^g) = \psi(\kappa_f, \kappa^g)$.

When $0 < \kappa_f \leq 1$, we notice that $\phi(\kappa_f, \kappa^g) = \psi(\kappa_f, \kappa^g)$ when $\kappa^g = 0$ or $\kappa^g = 1$. When $0 < \kappa^g < 1$, we have $\phi(\kappa_f, \kappa^g) > \psi(\kappa_f, \kappa^g)$ since $\phi(\kappa_f, \kappa^g)$ is a strictly concave function in κ^g and $\psi(\kappa_f, \kappa^g)$ is linear in κ^g. □

A simple computation shows the maximum ratio of these two bounds is $1/(1 - e^{-1}) \approx 1.5820$ when $\kappa_f = 1$ and $\kappa^g \to 1$. As another example, with $\kappa_f = 1$ and $\kappa^g = \ln(e - 1) \approx 0.541$, the ratio is ≈ 1.2688.

F. Proof of Theorem 3.7

Theorem F.1. Theoretical guarantee in the p matroids case. GREEDMAX *is guaranteed to obtain a solution* $\hat{X}$ *such that*

$$h(\hat{X}) \geq \frac{1-\kappa^g}{(1-\kappa^g)\kappa_f + p} h(X^*) \tag{7}$$

where $X^* \in \operatorname{argmax}_{X \in \mathcal{M}_1 \cap \ldots \cap \mathcal{M}_p} h(X)$*,* $h(X) = f(X) + g(X)$*,* κ_f *is the curvature of submodular* f *and* κ^g *is the curvature of supermodular* g*.*

Proof. The greedy procedure produces a chain of solutions $S_0, S_1, \ldots, S_k$ such that $|S_i| = i$, $S_i \subset S_{i+1}$, where k is the iteration after which any addition to S_k is infeasible in at least one matroid, and hence[4] $|\hat{X}| = k$. Immediately, we notice all S_i and X^* are independent sets for all p matroids.

For $j = 0, \ldots, k$ and $l = 1, \ldots, p$, there exist at least $\max(|X^*| - j, 0)$ elements $v \in X^* \setminus S_j$ s.t. $v \notin S_j$ and $S_j + v \in \mathcal{I}(M_l)$, which follows from the third property in the matroid definition. Therefore, for $j = 0, \ldots, k-1$, $l = 1, \ldots, p$, there are at most j elements of X^* that can not be added to S_j.

We next consider the intersection of all p matroids. For $j = 0, \ldots, k$, since in each matroid, there are at most j elements of X^* that cannot be added to S_j, the total possible number of elements for which there exists at least one matroid preventing us from adding to S_j is jp (the case that the p sets of at most j elements are disjoint). In other words, there are at least $\max(|X^*| - pj, 0)$ different $v \in |X^*|$ s.t. $v \notin S_j$, $S_j \cup \{v\} \in \mathcal{M}_1 \cap \ldots \cap \mathcal{M}_p$.

[4] There should be no confusion here that the k we refer to in this section is not any cardinality constraint, but rather the size of the greedy solution.

We claim $|X^*| \le pk$ as otherwise, by setting $j = k$ above, there are still feasible elements in $X^* \setminus S_k$ in the context of S_k, which indicates that GREEDMAX has not ended at iteration k. Therefore, we are at liberty to create $pk - |X^*|$ dummy elements, that are always feasible (i.e., independent in all matroids) and that have $h(v|X) = 0$ for all $X \subset V$ for each dummy v. We add these dummy elements to X^* and henceforth assume, w.l.o.g., that $|X^*| = pk$.

We next form an ordered k-partition of $X^* = X_0 \cup X_1 \cup \ldots \cup X_{k-1}$. We show below that it is possible to form this partition so that it has the following properties for $j = 0, \ldots, k-1$:

1. $|X_j| = p$;
2. for all $v \in X_j$, we have $v \notin S_j$ and $S_j \cup \{v\} \in \mathcal{M}_1 \cap \ldots \cap \mathcal{M}_p$ (i.e., v can be added to S_j);
3. and for all j s.t. $s_{j+1} \in X^* \cap S_k$, we have $s_{j+1} \in X_j$.

Immediately, we notice that property 3 is compatible with property 2.

We construct this partition in an order reverse from that of the greedy procedure, that is we create X_j from $j = k-1$ to 0. Recall that, at each step with index $j = k-1, k-2, \ldots, 0$, there are at least $|X^*| - pj = p(k-j)$ elements in X^* can be added to S_j.

When $j = k-1$, there are at least p candidate elements[5] in X^* and we choose p of them to form X_{k-1}. The element s_k can be added to S_{k-1} because the greedy algorithm only adds feasible elements and hence, if also $s_k \in X^*$, then s_k can be one of the elements in X_{k-1}. Thus, abiding property 3 above, we place $s_k \in X_{k-1}$.

Continuing, for $j = k-2, k-3, \ldots, 0$, there are at least p candidate elements in $X^* \setminus [X_{k-1} \cup X_{k-2} \cup \ldots \cup X_{j+1}]$ since $|X_{k-1} \cup X_{k-2} \cup \ldots \cup X_{j+1}| = p(k-j-1)$ and we choose p of them for X_j. Moreover, if $s_{j+1} \in X^*$, we notice s_{j+1} may be one of those candidate elements because of the greedy properties and since $s_{j+1} \notin [X_{k-1} \cup X_{k-2} \cup \ldots \cup X_{j+1}]$ (this follows because $s_{j+1} \in S_{j'}$ for any $j' \ge j+1$, so s_{j+1} is not a candidate element at step $j' = k-2, \ldots, j+1$). Similar to what was done in step $k-1$, we again choose p candidate elements to form X_j, and, if $s_{j+1} \in X^*$, we place $s_{j+1} \in X_j$.

We then arrive at partition $X^* = X_0 \cup X_1 \cup \ldots \cup X_{k-1}$ with the aforementioned three properties.

Next, we order the elements in $X^* = \{x_1, \ldots, x_{pk}\}$ where $\{x_{jp+1}, x_{jp+2}, \ldots, x_{(j+1)p}\} = X_j$ for $j = 0, 1, \ldots, k-1$. According to greedy, we have $h(x_{jp+t}|S_j) \le h(s_{j+1}|S_j) = a_{j+1}$ for $t = 1, \ldots, p$. Recall that a_i is defined to be $h(s_i|S_{i-1})$. Moreover, if $x_{jp+t} \in X^* \cap S_k$, we have $x_{jp+t} = s_{j+1}$.

According to Lemma C.1 above,

$$
\begin{aligned}
h(X^*) \le \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j \\
& + \sum_{j: s_j \in S_k \cap X^*} a_j + h(X^* \setminus S_k | S_k) && (36) \\
= \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j + \sum_{j: s_j \in S_k \cap X^*} h(s_j|S_{j-1}) \\
& + \sum_{i=1}^{pk} h(x_i | S_k \cup \{x_1\} \ldots \cup \{x_{i-1}\}) \mathbf{1}_{\{x_i \in X^* \setminus S_k\}} && (37) \\
\le \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j + \frac{1}{1-\kappa^g} \sum_{j: s_j \in S_k \cap X^*} h(s_j|S_{j-1}) \\
& + \frac{1}{1-\kappa^g} \sum_{j=0}^{k-1} \sum_{t=1}^{p} h(x_{jp+t}|S_j) \mathbf{1}_{\{x_{jp+t} \in X^* \setminus S_k\}} && (38) \\
= \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j \\
& + \frac{1}{1-\kappa^g} \Big[\sum_{j: s_j \in S_k \cap X^*} h(s_j|S_{j-1}) \\
& \quad + \sum_{j=0}^{k-1} \sum_{t=1}^{p} h(x_{jp+t}|S_j) \\
& \quad - \sum_{j=0}^{k-1} \sum_{t=1}^{p} h(x_{jp+t}|S_j) \mathbf{1}_{\{x_{jp+t} \in X^* \cap S_k\}} \Big] && (39) \\
= \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j \\
& + \frac{1}{1-\kappa^g} \Big[\sum_{j: s_j \in S_k \cap X^*} h(s_j|S_{j-1}) \\
& \quad + \sum_{j=0}^{k-1} \sum_{t=1}^{p} h(x_{jp+t}|S_j) \\
& \quad - \sum_{j: s_j \in S_k \cap X^*} h(s_j|S_{j-1}) \Big] && (40) \\
\le \kappa_f & \sum_{j: s_j \in S_k \setminus X^*} a_j + \frac{1}{1-\kappa^g} \sum_{j=0}^{k-1} \sum_{t=1}^{p} a_{j+1} && (41) \\
\le & \left[\kappa_f + \frac{p}{1-\kappa^g} \right] \sum_{j=0}^{k-1} a_{j+1} && (42) \\
= & \left[\kappa_f + \frac{p}{1-\kappa^g} \right] h(\hat{X}) && (43)
\end{aligned}
$$

[5] Elements that can be added at the given step.

where $\mathbf{1}_{\{\text{condition}\}}$ equals 1 if the condition is met and is 0 otherwise. Line 37 to 38 hold because of Lemma C.2 (ii). As for Line 39 to 40, we notice $x_{jp+t} = s_{j+1}$ if $x_{jp+t} \in X^* \cap S_k$. Line 40 to line 41 follows via the greedy procedure.

Therefore, we have our result which is

$$h(\hat{X}) \geq \frac{1-\kappa^g}{(1-\kappa^g)\kappa_f + p} h(X^*). \tag{44}$$

□

G. Proof of Theorem 3.8

Theorem G.1. SEMIGRAD *initialized with the empty set is guaranteed to obtain a solution $\hat{X}$ for the cardinality constrained case such that*

$$h(\hat{X}) \geq \frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] h(X^*) \tag{8}$$

where $X^* \in \operatorname{argmax}_{|X|\leq k} h(X)$, $h(X) = f(X) + g(X)$, & κ_f (resp. κ^g) is the curvature of f (resp. g).

Proof. If SEMIGRAD is initialized by empty set, we need to calculate the semigradient of g at $\emptyset$. By definition, we have

$$m_{g,\emptyset,1}(Y) = m_{g,\emptyset,2}(Y) = \sum_{v \in Y} g(j) \tag{45}$$

So in the first step of SEMIGRAD, we are optimizing $h'(X) = f(X) + m_g(X) = f(X) + \sum_{v\in X} g(v)$ by GREEDMAX. We will focus elusively on this step as later iterations can only improve the objective value.

According to Lemma C.1, we have

$$\begin{aligned} h(X^*) \leq & \kappa_f \sum_{j:s_j \in S_i \setminus X^*} h(s_j|S_{j-1}) \\ & + \sum_{j:s_j\in S_i \cap X^*} h(s_j|S_{j-1}) + h(X^* \setminus S_j|S_j) \end{aligned} \tag{46}$$

Since GREEDMAX is choosing the feasible element with the largest gain, in the semigradient approximation we have $h'(v|S_i) \leq h'(s_{i+1}|S_i)$ instead of $h(v|S_i) \leq h(s_{i+1}|S_i)$. We get:

$$\begin{aligned} & h(X^* \setminus S_j|S_j) \\ &= f(X^* \setminus S_j|S_j) + g(X^* \setminus S_j|S_j) && (47) \\ &\leq \sum_{v\in X^*\setminus S_j} f(v|S_j) + \frac{1}{1-\kappa^g}\sum_{v\in X^*\setminus S_j} g(v) && (48) \\ &\leq \frac{1}{1-\kappa^g}\sum_{v\in X^*\setminus S_j} h'(v|S_j) && (49) \\ &\leq \frac{1}{1-\kappa^g}\sum_{v\in X^*\setminus S_j} h'(s_{j+1}|S_j) && (50) \\ &= \frac{1}{1-\kappa^g}\sum_{v\in X^*\setminus S_j} f(s_{j+1}|S_j) + g(s_{j+1}) && (51) \\ &\leq \frac{1}{1-\kappa^g}\sum_{v\in X^*\setminus S_j} f(s_{j+1}|S_j) + g(s_{j+1}|S_j) && (52) \\ &= \frac{|X^* \setminus S_j|}{1-\kappa^g} h(s_{j+1}|S_j) && (53) \end{aligned}$$

And hence,

$$\begin{aligned} h(X^*) \leq \kappa_f & \sum_{j:s_j\in S_i\setminus X^*} a_i + \sum_{j:s_j\in S_i\cap X^*} a_i \\ & + \frac{k - |X^*\cap S_i|}{1-\kappa^g} s_{i+1}. \end{aligned} \tag{54}$$

We can then use Lemma D.2 to h to finish the proof. □

H. Proof of Theorem 3.9

Theorem H.1. SEMIGRAD *initialized with the empty set is guaranteed to obtain a solution $\hat{X}$, feasible for the p matroid constraints, such that*

$$h(\hat{X}) \geq \frac{1-\kappa^g}{(1-\kappa^g)\kappa_f + p} h(X^*) \tag{9}$$

where $X^* \in \operatorname{argmax}_{X\in\mathcal{M}_1\cap\ldots\cap\mathcal{M}_p} h(X)$, $h = f + g$, & κ_f (resp. κ^g) is the curvature of f (resp. g).

Proof. If SEMIGRAD is initialized by empty set, we need to calculate the semigradient of g at $\emptyset$. By definition, we have

$$m_{g,\emptyset,1}(Y) = m_{g,\emptyset,2}(Y) = \sum_{v \in Y} g(j) \tag{55}$$

So in the first step of SEMIGRAD, we are optimizing $h'(X) = f(X) + m_g(X) = f(X) + \sum_{v\in X} g(v)$ by GREEDMAX. We will focus on this step.

According to Lemma C.1, we have

$$h(X^*) \leq \kappa_f \sum_{j: s_j \in S_i \setminus X^*} h(s_j | S_{j-1}) + \sum_{j: s_j \in S_i \cap X^*} h(s_j | S_{j-1}) + h(X^* \setminus S_j | S_j) \tag{56}$$

We then follow the proofs of Theorems 3.7 and 3.8. The only difference is that in Theorem 3.7 we have $h(v|S_i) \leq h(s_{i+1}|S_i)$ for all feasible v, but in this proof, we have $h'(v|S_i) \leq h'(s_{i+1}|S_i)$, which does not affect the proof as shown in the proof of Theorem 3.8. □

I. Proof of Theorem 4.1

Lemma I.1. *(lemma 4.1 from (Svitkina & Fleischer, 2011)) Let R be a random subset of V of size $\alpha = \frac{x\sqrt{n}}{5}$, let $\beta = \frac{x^2}{5}$, and let x be any parameter satisfying $x^2 = \omega(\ln n)$ and such that α and β are integer. Let $f_1(X) = \min(|X|, \alpha)$ and $f_2(X) = \min(\beta + |X \cap \bar{R}|, |X|, \alpha)$. Any algorithm that makes a polynomial number of oracle queries has probability $n^{-\omega(1)}$ of distinguishing the functions f_1 and f_2.*

Theorem I.2. Hardness for cardinality constrained case. *For all $0 \leq \beta \leq 1$, there exists an instance of a BP function $h = f + g$ with supermodular curvature $\kappa^g = \beta$ such that no poly-time algorithm solving Problem 1 with a cardinality constraint can achieve an approximation factor better than $1 - \kappa^g + \epsilon$, for any $\epsilon > 0$.*

Proof. $\kappa^g = \alpha = 0$ is trivial since no algorithm can do better than 1.

The case when $\kappa^g = 1$ can be proven using the example in Lemma 3.1. $g(X) = \max\{|X| - k, 0\}$, except for a special set R where $g(R) = 0.5$ and $|R| = k$.

For the other case, we prove this result using the hardness construction from (Goemans et al., 2009; Svitkina & Fleischer, 2011). The intuition is to construct two supermodular functions, g and g' both with curvature κ^g which are indistinguishable[6] with high probability in polynomially many function queries. Therefore, any polynomial time algorithm to maximize $g(X)$ can not find $\hat{X} \subseteq V$ with $|\hat{X}| \leq k$ s.t. $g(\hat{X}) > \max_{X \leq k} g'(X)$; otherwise we will have $g(\hat{X}) > \max_{X \leq k} g'(X) \geq g'(\hat{X})$ which contradicts the indistinguishability. In this case, the approximate ratio $\frac{g(\hat{X})}{OPT} \leq \frac{\text{OPT}'}{\text{OPT}}$ where $\text{OPT} = \max_{X \leq k} g(X)$ and $\text{OPT}' = \max_{X \leq k} g'(X)$. The guarantee, by definition, is the best case approximate ratio and, thus no greater than $\frac{\text{OPT}'}{\text{OPT}}$. If any polynomial algorithm has a guarantee greater than $\frac{\text{OPT}'}{\text{OPT}}$, then it contradicts the information theoretic hardness. This is meaningful if $\text{OPT}' < \text{OPT}$.

Let $g(X) = |X| - \beta \min\{\gamma + |X \cap \bar{R}|, |X|, \alpha\}$ and $g'(X) = |X| - \beta \min\{|X|, \alpha\}$, where $R \subseteq V$ is a random set of cardinality α. Let $\alpha = x\sqrt{n}/5$ and $\gamma = x^2/5$ and let x be any parameter satisfying $x^2 = \omega(\ln n)$ s.t. $\gamma < \alpha$ are positive integers and $\alpha \leq \frac{n}{2} - 1$.[7] g and g' are modular minus submodular functions, which implies supermodularity. Monotonicity follows from $g(v|X), g'(v|X) \geq 0$. Also, $\text{OPT} = \alpha - \beta\gamma > \text{OPT}' = \alpha(1 - \beta)$.

Next, we calculate the supermodular curvature. $g(\emptyset) = g'(\emptyset) = 0$. $g(v) = g'(v) = 1 - \beta$ for all $v \in V$ since $\alpha, \gamma \geq 1$. $g(V \setminus \{v\}) = g'(V \setminus \{v\}) = n - 1 - \beta\alpha$ and $g(V) = g'(V) = n - \beta\alpha$ for all $v \in V$ since $\alpha \leq \frac{n}{2} - 1$. Therefore, $\kappa^g = 1 - \min_{v \in V} \frac{g(v)}{g(v|V-v)} = \beta$. $\kappa^{g'} = 1 - \min_{v \in V} \frac{g'(v)}{g'(v|V-v)} = \beta$. So g and g' are monotone non-decreasing supermodular functions with curvature β. Let $f(X) = 0$ for all X and $h(X) = f(X) + g(X) = g(X)$ is the objective BP function.

Any algorithm that uses a polynomial number of queries can distinguish g and g' with probability only $n^{-\omega(1)}$ according to lemma I.1 (Svitkina & Fleischer, 2011). More precisely, $g(X) > g'(X)$[8] if and only if $\gamma + |X \cap \bar{R}| < |X|$ and $\gamma + |X \cap \bar{R}| < \alpha$. It is equivalent with asking $|X \cap R| > \gamma$ and $|X \cap \bar{R}| < \alpha - \gamma$. Moreover, $\Pr(g(X) \neq g'(X))$, where randomness is over random subsets $R \subseteq V$ of size α, is maximized when $|X| = \alpha$ (Svitkina & Fleischer, 2011). In this case, the two conditions become identical, and since $|X| = |X \cap \bar{R}| + |X \cap R|$, the condition $g(X) > g'(X)$ happens when only $|X \cap R| > \gamma$. Intuitively, $E|X \cap R| = \frac{\alpha^2}{n} = \frac{\gamma}{5}$ where R is a random set (of arbitrary size) and X is an arbitrary but fixed set of size α. So $|X \cap R|$ is located in small interval around $\frac{\gamma}{5}$ and is hardly ever be larger than γ for large n according to the law of large numbers. While this is only the intuition, a similar reasoning in (Svitkina & Fleischer, 2011) offers more details.

Therefore, the output $\hat{X}$ of any polynomial algorithm must satisfies $g(\hat{X}) \leq \max_{X \leq k} g'(X)$ since, otherwise the algorithm actually distinguishes the two function at $\hat{X}$, $g(\hat{X}) > \max_{X \leq k} g'(X) \geq g'(\hat{X})$. The approximate ratio $\frac{g(\hat{X})}{\text{OPT}} \leq \frac{\text{OPT}'}{\text{OPT}} = \frac{\alpha - \kappa^g \alpha}{\alpha - \kappa^g \gamma} = (1 - \kappa^g) \frac{1}{1 - \kappa^g \sqrt{\frac{\omega(\log n)}{n}}} \leq 1 - \kappa^g + \epsilon$. Therefore, the guarantee of any polynomial algorithm, that, by definition, the best case approximate ratio, is no greater than $1 - \kappa^g + \epsilon$ for any $\epsilon > 0$ since, otherwise contradicts the information theoretic hardness.

□

[6] Indistinguishable means for all sets X that the algorithm evaluates, $g(X) = g'(X)$.

[7] These examples and the specific parameters like 5 are adopted from (Svitkina & Fleischer, 2011).

[8] Note that $g(X) \geq g'(X)$ for all $X \subseteq V$ for any α and γ.

J. Proof of Theorem 4.2

Theorem J.1. Hardness for p matroids constraint case. *For all $0 \leq \beta \leq 1$, there exists an instance of a BP function $h = f + g$ with supermodular curvature $\kappa^g = \beta$ such that no poly-time algorithm can achieve an approximation factor better than $(1 - \kappa^g)O(\frac{\ln p}{p})$ unless P=NP.*

Proof. Consider the p-set problem (Hazan et al., 2006), let R be the maximum disjoint sets of these p sets. No polynomial algorithm can find a larger number of disjoint sets than $O(\frac{\ln p}{p})|R|$ (Hazan et al., 2006). Let $k = O(\frac{\ln p}{p})|R|$. So no polynomial algorithm can find a feasible set with size larger than k unless P=NP.

Let $h(X) = (1 - \beta)|X| + \beta \max\{|X| - k, 0\}$. It is easy to check that h is a BP function with $f = 0$ and $g = h$ with $\kappa^g = \beta$.

Therefore, the output $\hat{X}$ of any polynomial algorithm that maximizes h under the p-set constraint (expressible via the intersection of p matroids) must satisfy that $|X| \leq k$ and, therefore, $h(\hat{X}) \leq (1 - \beta)k$ unless P=NP. But $h(X^*) \geq h(R) = (1 - \beta)|R| + \beta(|R| - k) = |R| - \beta k$.

Thus, the approximate ratio

$$\frac{h(\hat{X})}{h(X^*)} \leq \frac{(1-\beta)k}{|R| - \beta k} \leq \frac{(1-\beta)O(\frac{\ln p}{p})}{1 - \beta O(\frac{\ln p}{p})} \leq \frac{(1-\beta)O(\frac{\ln p}{p})}{\frac{1}{2}} = (1 - \kappa^g)O(\frac{\ln p}{p}). \quad (57)$$

since the denominator $1 - \beta O(\frac{\ln p}{p}) \geq \frac{1}{2}$ asymptotically and $2O(\frac{\ln p}{p}) = O(\frac{\ln p}{p})$. $\square$

K. Submodularity Ratio and Generalized Curvature

In this section, we compare the pair κ_f, κ^g of curvatures with the submodularity ratio (Das & Kempe, 2011; Bian et al., 2017). We also show that both the generalized curvature introduced in (Bian et al., 2017) and the submodularity ratio (Das & Kempe, 2011) appear to be hard to compute in general under the oracle model. Lastly, we compare the pair κ_f, κ^g with another notion of curvature introduced in (Sviridenko et al., 2013), showing a simple inequality relationship in general and a correspondence when $h = g$.

K.1. Submodularity ratio

The submodularity ratio is defined as

$$\gamma_{U,k}(h) = \min_{L \subseteq U, S: |S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} h(x|L)}{h(S|L)} \quad (58)$$

with $U \subseteq V$ and $1 \leq k \leq |V| = n$, and typically we consider $\gamma_{V,n}$. We can establish a simple lower bound of the submodularity ratio based on the supermodular curvature as follows.

Lemma K.1. *$\gamma_{V,n}(h) \geq 1 - \kappa^g$ when $h = f + g$.*

Proof. For all $L \subseteq V$ and $S \cap L = \emptyset$, we have $\frac{\sum_{x \in S} h(x|L)}{h(S|L)} \geq 1 - \kappa^g$ which follows from Lemma C.2(iv) Thus, $\gamma_{V,n}(h) \geq 1 - \kappa^g$. $\square$

The function h is submodular if and only if $\gamma_{V,n} = 1$ so one might hope that given a BP function $h = f + g$, that as $\gamma_{V,n}(h) \to 1$, correspondingly $\kappa^g \to 0$. This is not the case, however, as can be seen by considering the following example.

Let a be an element of V and define the function $g(A) = |A \cap (V \setminus \{a\})| + \epsilon |A \cap (V \setminus \{a\})| |A \cap \{a\}|$, where $\epsilon > 0$ is a very small number. Immediately, we have that g being supermodular and monotone. Also note, if $a \notin A$ then $g(A) = |A|$; if $a \in A$ then $g(A) = (|A| - 1)(1 + \epsilon)$.

First, we calculate the supermodular curvature κ^g. We have that $g(a) = 0$ and also $g(a|V \setminus \{a\}) = \epsilon(n - 1)$. Therefore, the function is fully curved, $\kappa^g = 1$.

Next, we calculate the submodularity ratio $\gamma_{V,n} = \min_{L,S \subset V, S \cap L = \emptyset} \frac{\sum_{v \in S} g(v|L)}{g(S|L)}$. When $|S| = 1$, $\frac{\sum_{v \in S} g(v|L)}{g(S|L)} = 1$. When $|S| \geq 2$, we have the following 3 cases (recall that $S \cap L = \emptyset$ so there is no forth case):

- $a \in S$. $g(S|L) = g(S \cup L) - g(L) = (|S| + |L| - 1)(1 + \epsilon) - |L|$ is very close to $|S| - 1$ for very small ϵ. $\sum_{v \in S} g(v|L) = \epsilon |L| + |S| - 1$, which is also very close to $|S| - 1$ for small ϵ. So $\frac{\sum_{v \in S} g(v|L)}{g(S|L)} \approx 1$ for small ϵ.
- $a \in L$. $g(S|L) = g(S \cup L) - g(L) = |S|(1 + \epsilon)$. $\sum_{v \in S} g(v|L) = |S|(1 + \epsilon)$. So $\frac{\sum_{v \in S} g(v|L)}{g(S|L)} = 1$
- $a \notin S \cup L$. $g(S|L) = |S|$ and $\sum_{v \in S} g(v|L) = |S|$. Therefore, $\frac{\sum_{v \in S} g(v|L)}{g(S|L)} = 1$.

In all cases, $\frac{\sum_{v \in S} g(v|L)}{g(S|L)}$ is either 1 or very close to 1 for small ϵ, so $\gamma_{V,n}$ has only 1 as an upper bound. That is, we have an example function that is purely supermodular and fully curved ($\kappa^g = 1$) for all non-zero values of ϵ, but the submodularity ratio can be arbitrarily close to 1. If we consider a weighted sum of a submodular function and this supermodular function, the submodularity ratio is again arbitrarily close to 1. Therefore, there does not seem to be an immediately accessible strong relationship between the supermodular curvature and the submodularity ratio.

K.2. Hardness of Generalized Curvature and Submodularity Ratio

The generalized curvature Bian et al. (2017) of a non-negative function h is the smallest scalar α s.t.

$$h(v|S \setminus \{v\} \cup \Omega) \geq (1-\alpha)h(v|S \setminus \{v\}) \quad (59)$$

for all $S, \Omega \subseteq V$ and $v \in S \setminus \Omega$ and this is used, in concert with the submodularity ratio, to produce bounds such as $\frac{1}{\alpha}(1-e^{-\alpha\gamma})$ for the greedy algorithm. Unfortunately, the generalized curvature is hard to compute under the oracle model. We have the following.

Lemma K.2. *There exists an instance of a non-negative function h whose generalized curvature can not be calculated in polynomial time, when we have only oracle access to the function.*

Proof. We consider a non-negative function $h' : 2^V \to R$ with ground set size equals n (n is even number). Let $h'(X) = |X|$ for all $X \subseteq V$. Let $R \subseteq V$ be an arbitrary set with $|R| = \frac{n}{2}$. Define another set function $h : 2^V \to R$, $h(X) = h'(X)$ for all $X \subseteq V$ and $X \neq R$; $h(R) = \frac{n}{2} - 1$.

First, we can easily calculate the generalized curvature of h' and h. We have that $\alpha_{h'} = 0$ since h' is a non-decreasing modular function. For h, let $S \cup \Omega = R$, $S \cap \Omega = \emptyset$, $|S|, |\Omega| \geq 1$ and $v \in S$, we have $h(v|S \setminus \{v\} \cup \Omega) = 0$ and $h(v|S \setminus \{v\}) = 1$. Therefore $\alpha = 1$ is the smallest scalar s.t. $h(v|S \setminus \{v\} \cup \Omega) \geq (1-\alpha)h(v|S \setminus \{v\})$. So, as a conclusion of this part, the generalized curvature of the two functions are not the same.

Next we use a proof technique similar to (Svitkina & Fleischer, 2011). Note that $h'(X) = h(X)$ if and only if $X \neq R$. So for any algorithm trying to calculate α_h, before it evaluates $h(R)$, all function evaluations are the same with calculating $\alpha_{h'}$. Additionally, since $h(X) = |X|$, it is permutation symmetric. Therefore, the algorithm can only do random search to find R. If the algorithm acquires a polynomial number $O(n^m)$ of sets of size $\frac{n}{2}$, the probability of finding R is $\frac{O(n^m)}{\binom{n}{\frac{n}{2}}} \leq \frac{O(n^m)}{(n/\frac{n}{2})^{\frac{n}{2}}} = \frac{O(n^m)}{2^{n/2}} \leq O(2^{-n/2+\epsilon n})$ for all $\epsilon > 0$.

Therefore, no algorithm can be guaranteed to distinguish h and h' in polynomial time. Since the generalized curvature of h and h' are different, neither of them can be calculated in polynomial time.

□

Likewise, the submodularity ratio is unfortunately also hard to compute exactly, in the oracle model.

Lemma K.3. *There exists an instance of a non-negative function h whose submodularity ratio (Equation (58)) can not be calculated in polynomial time under only oracle access to that function.*

Proof. We consider a non-negative function $h' : 2^V \to R$ with ground set size n (where n is an even number). Let $h'(X) = |X|$ for all $X \subseteq V$. Let $R \subseteq V$ be an arbitrary set with $|R| = \frac{n}{2}$. Define another set function $h : 2^V \to R$, $h(X) = h'(X)$ for all $X \subseteq V$ and $X \neq R$ and $h(R) = \frac{n}{2} - 1$.

We can easily calculate the submodularity ratio of both h' and h as follows. We have that $\gamma_{V,n}(h') = 1$ since h' is a non-decreasing modular (and thus submodular) function. For h, choose an element $v_1 \in R$ and another element $v_2 \in V \setminus R$, and let $L = R \setminus \{v_1\}$ and $S = \{v_1, v_2\}$. We have $\frac{\sum_{v \in S} h(v|L)}{h(S|L)} = \frac{h(R)+h(R\setminus\{v_1\}\cup\{v_2\})-2h(R\setminus\{v_1\})}{h(R\cup\{v_2\})-h(R\setminus\{v_1\})} = \frac{1}{2}$ and thus $\gamma_{V,n}(h) = \min_{L,S \subseteq V, S \cap L = \emptyset} \frac{\sum_{v \in S} h(v|L)}{h(S|L)} \leq \frac{1}{2}$. Therefore, the submodularity ratio of the two functions are not the same. Given the submodularity ratio of the two functions, we would be able to tell them apart.

Next we use a proof technique similar to (Svitkina & Fleischer, 2011). We have that $h'(X) = h(X)$ if and only if $X \neq R$. So for any algorithm trying to calculate $\gamma_{V,n}(h)$, before it evaluates $h(R)$, all function evaluations are the same with calculating $\gamma_{V,n}(h')$. Additionally, since $h(X) = |X|$ is permutation symmetric, the algorithm can only do a random search to find R. If the algorithm queries a polynomial number $O(n^m)$ of sets of size $\frac{n}{2}$, the probability of finding R is $\frac{O(n^m)}{\binom{n}{\frac{n}{2}}} \leq \frac{O(n^m)}{(n/\frac{n}{2})^{\frac{n}{2}}} = \frac{O(n^m)}{2^{n/2}} \leq O(2^{-n/2+\epsilon n})$ for all $\epsilon > 0$.

Therefore, no algorithm can guarantee to distinguish h and h' in polynomial time. Since the submodularity ratio of h and h' are different, this means that neither of them can be calculated in polynomial time.

□

K.3. Comparison to Sviridenko et al. (2013)'s curvature

Sviridenko et al. (2013) (in their Section 8) define a notion of curvature as follows:

$$1 - c = \min_j \min_{A,B \subseteq V \setminus j} \frac{h(j|A)}{h(j|B)} \quad (60)$$

We can establish a simple upper bound on c based on submodular and supermodular curvature. We calculate $\frac{h(j|A)}{h(j|B)}$ given $h = f + g$ and κ_f and κ^g. First, $f(j|B) \leq f(j) \leq \frac{1}{1-\kappa_f} f(j|A)$ which follows from Lemma C.2 (i). Thus $\frac{f(j|A)}{f(j|B)} \geq 1 - \kappa_f$. Next, $g(j|A) \geq g(j) \geq (1-\kappa^g)g(j|B)$

which follows from Lemma C.2 (ii). Thus, $\frac{g(j|A)}{g(j|B)} \geq 1 - \kappa^g$. Therefore,

$$\frac{h(j|A)}{h(j|B)} = \frac{f(j|A) + g(j|A)}{f(j|B) + g(j|B)} \tag{61}$$

$$=\geq \frac{(1-\kappa_f)f(j|B) + (1-\kappa^g)g(j|B)}{f(j|B) + g(j|B)} \tag{62}$$

$$\geq \frac{\min(1-\kappa_f, 1-\kappa^g)(f(j|B) + g(j|B))}{f(j|B) + g(j|B)} \tag{63}$$

$$\geq \min(1-\kappa_f, 1-\kappa^g) \tag{64}$$

Thus we have $1 - c \geq \min(1 - \kappa_f, 1 - \kappa^g)$, or $c \leq \max(\kappa_f, \kappa^g)$.

Note that for purely supermodular functions, $\kappa_f = 0$ and, considering Equation (60), we have $c = \kappa^g$. This coincides with the $1 - \kappa^g$ bound and hardness for monotone supermodular functions — compare Theorem 8.1 of Sviridenko et al. (2013) with the present paper's item 3 in Section 3.2.1 and Theorem 4.1.