A Framework for Adapting Offline Algorithms to Solve Combinatorial Multi-Armed Bandit Problems with Bandit Feedback

Guanyu Nie [1] Yididiya Y Nadew [1] Yanhui Zhu [1] Vaneet Aggarwal [2 3] Christopher John Quinn [1]

Abstract

We investigate the problem of stochastic, combinatorial multi-armed bandits where the learner only has access to bandit feedback and the reward function can be non-linear. We provide a general framework for adapting discrete offline approximation algorithms into sublinear α-regret methods that only require bandit feedback, achieving $\mathcal{O}\left(T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ expected cumulative α-regret dependence on the horizon T. The framework only requires the offline algorithms to be robust to small errors in function evaluation. The adaptation procedure does not even require explicit knowledge of the offline approximation algorithm — the offline algorithm can be used as black box subroutine. To demonstrate the utility of the proposed framework, the proposed framework is applied to multiple problems in submodular maximization, adapting approximation algorithms for cardinality and for knapsack constraints. The new CMAB algorithms for knapsack constraints outperform a full-bandit method developed for the adversarial setting in experiments with real-world data.

1. Introduction

Many real world sequential decision problems can be modeled using the framework of stochastic multi-armed bandits (MAB), such as scheduling, assignment problems, ad-campaigns, and product recommendations, among others. The decision maker sequentially selects actions and receives stochastic rewards from an unknown distribution. The goal of the decision maker is to maximize the expected cumulative reward over a (possibly unknown) time horizon. Actions result both in the immediate reward and, more importantly, information about that action's reward distribution. Such problems result in a trade-off between trying actions the agent is uncertain of (*exploring*) or only taking the action that is empirically the best seen so far (*exploiting*).

In the classic MAB setting, the number of possible actions is small relative to the time horizon, meaning each action can be taken at least once, and there is no assumed relationship between the reward distributions of different arms. The *combinatorial* multi-armed bandit (CMAB) setting involves a large but structured action space. For example, in product recommendation problems, the decision maker may select a subset of products (base arms) from among a large set. There are several aspects that can affect the difficulty of these problems. First, MAB methods are typically compared against a learner with access to a value oracle of the reward function (an offline problem). For some problems, it is NP-hard for the baseline learner with value oracle access to optimize. An example is if the expected/averaged reward function is submodular and actions are subsets constrained by cardinality. At best, for these problems, approximation algorithms may exist. Thus, unless the time horizon is large (exponentially long in the number of base arms, for instance), it would be more reasonable to compare the CMAB agent against the performance of the approximation algorithm for the related offline problem. Likewise, one could apply state of the art methods for (unstructured) MAB problems treating each subset as a separate arm, and obtain $\tilde{\mathcal{O}}(T^{\frac{1}{2}})$ dependence on the horizon T for the subsequent regret bound. However, that dependence would only apply for exponentially large T.

Feedback plays an important role in how challenging the problem is. When the decision maker only observes a (numerical) reward for the action taken, that is known as bandit or full-bandit feedback. When the decision maker observes additional information, such as contributions of each base arm in the action, that is semi-bandit feedback. Semi-bandit feedback greatly facilitates learning. Suppose for instance that the reward function (on average) was monotone increasing over the inclusion lattice and there was a cardinality constraint of size k. The agent would know from the start that no set of size smaller than k could be optimal (or could

[1]Computer Science Department, Iowa State University, Ames, IA, USA [2]School of IE and School of ECE, Purdue University, West Lafayette, IN, USA [3]Computer Science Program, KAUST, Saudi Arabia. Correspondence to: Christopher John Quinn <cjquinn@iastate.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

even be the near-optimal solution the baseline learning using a value oracle would find). However, there would be $\binom{n}{k}$ sets of size k. For $n = 100$ and $k = 10$, the agent would need a horizon $T > 10^{12}$ to try each cardinality k set even just once. If the reward function belongs to a certain class, such as the class of submodular functions, then one approach would be to use a greedy procedure based on base arm values. With semi-bandit feedback, the agent could on the one hand only take actions of cardinality k (putatively optimal actions), gain the subsequent rewards, and yet also observe samples of the base arms' values to improve future actions.

Bandit feedback is much more challenging, as only the joint reward is observed. In general, for non-linear reward functions, the individual values or marginal gains of base arms can only be loosely bounded if actions only consist of maximal subsets. Thus, to estimate values or marginal gains of base arms, the agent would need to deliberately spend time sampling actions (such as smaller sets) that are known to be sub-optimal in order to estimate their values to later better select actions of cardinality k. Standard MAB methods like UCB or TS based methods by design do not take actions known to be sub-optimal. Thus, while such strategies could be used when semi-bandit feedback is available, it is less clear whether they can be effectively used when only bandit feedback is available.

There are important applications where semi-bandit feedback may not be available, such as in influence maximization and recommender systems. Influence maximization models the problem of identifying a low-cost subset (seed set) of nodes in a (known) social network that can influence the maximum number of nodes in a network (Nguyen and Zheng, 2013; Leskovec et al., 2007; Bian et al., 2020). Recent research has generalized the problem to online settings where the knowledge of the network and diffusion model is not required (Wang et al., 2020; Perrault et al., 2020) but extra feedback is assumed. However, for many networks the user interactions and user accounts are private; only aggregate feedback (such as the count of individuals using a coupon code or going to a website) might be visible to the decision maker.

In this work, we seek to address these challenges by proposing a *general framework* for adapting offline approximation algorithms into algorithms for stochastic CMAB problems when only bandit feedback is available. We identify that a single condition related to the robustness of the approximation algorithm to erroneous function evaluations is sufficient to guarantee that a simple explore-then-commit (ETC) procedure accessing the approximation algorithm as a black box results in a sublinear α-regret CMAB algorithm despite having only bandit feedback available. The approximation algorithm does not need to have any special structure (such as an iterative greedy design). Importantly, no effort is needed on behalf of the user in mapping steps in the offline method into steps of the CMAB method.

We demonstrate the utility of this framework by assessing the robustness of several approximation algorithms in the submodular optimization literature (three approximation algorithms designed for knapsack constraints and one designed for cardinality constraints) which immediately result in sublinear α-regret CMAB algorithms that only rely on bandit-feedback, the first such algorithms for CMAB problems with submodular rewards and knapsack constraints. We also show that despite the simplicity and universal design of the adaptation, the resulting CMAB algorithms work well on budgeted influence maximization and song recommendation problems using real world data.

The main contributions of this paper can be summarized as:

1. We provide a general framework for adapting discrete offline approximation algorithms into sublinear α-regret methods for stochastic CMAB problems where only bandit feedback is available. The framework only requires the offline algorithms to be robust to small errors in function evaluation, a property important in its own right for offline problems. The algorithms are not required to have a special structure — instead they are used as black boxes. Our procedure has minimal storage and time-complexity overhead, and achieves a regret bound with $\tilde{\mathcal{O}}(T^{\frac{2}{3}})$ dependence on the horizon T.

2. We illustrate the utility of the proposed framework by assessing the robustness of several approximation algorithms for (offline) constrained submodular optimization, a class of reward functions lacking simplifying properties of linear or Lipschitz reward functions. Specifically, we prove the robustness of approximation algorithms given in (Nemhauser et al., 1978; Badanidiyuru and Vondrák, 2014; Sviridenko, 2004; Khuller et al., 1999; Yaroslavtsev et al., 2020) with cardinality or knapsack constraints, and use the general framework to give regret bounds for the stochastic CMAB. In particular, we note that this paper gives the first regret bounds for stochastic submodular CMAB with knapsack constraints under bandit feedback.

3. We evaluate the performance of proposed framework through the stochastic submodular CMAB with knapsack constraints problem for two applications: Budgeted Influence Maximization, and Song Recommendation. The evaluation results demonstrate that the proposed approach significantly outperforms a full-bandit method for a related problem in the adversarial setting.

2. Related Work

We now briefly discuss only the most closely related works. See the supplementary material for more discussion.

Adversarial CMAB The closest related works are on adversarial CMAB. In (Niazadeh et al., 2021), the authors propose a framework for transforming greedy α-approximation algorithms for offline problems to online methods in an adversarial bandit setting, for both semi-bandit (achieving $\widetilde{O}(T^{1/2})$ $\alpha-$regret) and full-bandit feedback (achieving $\widetilde{O}(T^{2/3})$ $\alpha-$regret). Their framework requires the offline approximation algorithm to have an iterative greedy structure (unlike ours), satisfy a robustness property (like ours), and satisfy a property referred to as Blackwell reducibility (unlike ours). In addition to these conditions, the adaptation depends on the number of subproblems (greedy iterations) which for some algorithms can be known ahead of time (such as with cardinality constraints) but for other algorithms can only be upper-bounded. (Our adaptation uses the offline algorithm as a black box.) The authors check those conditions and explicitly adapt several offline approximation algorithms. In this paper, we consider an approach for converting offline approximation algorithm to online for stochastic CMAB, while requiring less assumptions.

We also note that (Niazadeh et al., 2021) do not consider submodular CMAB with knapsack constraints, and thus do not verify whether any approximation algorithms for the offline problem satisfy the required properties (of sub-problem structure or robustness or Blackwell reducibility) to be transformed, and this is an example we consider for our general framework. Consequently, in our experiments for submodular CMAB with knapsack constraints in Section 7, we use the algorithm in (Streeter and Golovin, 2008) designed for a knapsack constraint (in expectation) as representative of methods for the adversarial setting. Other related works for adversarial stochastic CMAB are described in Appendix H.

Stochastic Submodular CMAB with Full Bandit Feedback Recently, (Nie et al., 2022) propose an algorithm for stochastic MAB with submodular rewards, when there is a cardinality constraint. Their algorithm is a specific adaptation of an offline greedy method. In our work, we propose a general framework that employs the offline algorithm as a black box (and this result becomes a special case of our approach). While there are multiple results for semi-bandit feedback (see Appendix H.4), this paper considers full bandit feedback.

3. Problem Statement

We consider sequential, combinatorial decision-making problems over a finite time horizon T. Let Ω denote the ground set of base elements (arms). Let $n = |\Omega|$ denote the number of arms. Let $D \subseteq 2^{\Omega}$ denote the subset of feasible actions (subsets), for which we presume membership can be efficiently evaluated. We will later consider applications with cardinality and knapsack constraints, though our methods are not limited to those. We will use the terminologies *subset* and *action* interchangeably throughout the paper.

At each time step t, the learner selects a feasible action $A_t \in D$. After the subset A_t is selected, the learner receives reward $f_t(A_t)$. We assume the reward f_t is stochastic, bounded in $[0, 1]$, and i.i.d. conditioned on a given subset. Define the expected reward function as $f(A) = \mathbb{E}[f_t(A)]$.

The goal of the learner is to maximize the cumulative reward $\sum_{t=1}^{T} f_t(A_t)$. To measure the performance of the algorithm, one common metric is to compare the learner to an agent with access to a value oracle for f. However, if optimizing f over D is NP-hard, such a comparison would not be meaningful unless the horizon is exponentially large in the problem parameters.

If there is a known approximation algorithm $\mathcal{A}$ with approximation ratio $\alpha \in (0, 1]$ for optimizing f over D, a more natural alternative is to evaluate the performance of a CMAB algorithm against what $\mathcal{A}$ could achieve. Thus, we consider the the expected cumulative α-regret $\mathcal{R}_{\alpha,T}$, which is the difference between α times the cumulative reward of the optimal subset's expected value and the average received reward, (we write $\mathcal{R}_T$ when α is understood from context)

$$\mathbb{E}[\mathcal{R}_T] = \alpha T f(\mathrm{OPT}) - \mathbb{E}\left[\sum_{t=1}^{T} f_t(A_t)\right], \qquad (1)$$

where OPT is the optimal solution, i.e., $\mathrm{OPT} \in \arg\max_{A \in D} f(A)$ and the expectations are over both the random rewards and the sequence of actions.

4. Robustness of Offline Algorithms

In this section, we introduce a criterion for an offline approximation algorithm's sensitivity to (bounded) additive perturbations to function evaluations. Investigating robustness of approximation algorithms in offline settings is valuable in its own right. Importantly, we will show that this property alone is sufficient to guarantee that the offline algorithm can be adapted to solve analogous combinatorial multi-armed bandit (CMAB) problems with just bandit feedback and yet achieve sub-linear regret. Furthermore, the CMAB adaptation will not rely on any special structure of the algorithm design, instead employing it as a black box.

Definition 4.1 ((α, δ)-Robust Approximation)**.** An algorithm $\mathcal{A}$ is an (α, δ)-robust approximation algorithm for the combinatorial optimization problem of maximizing a function $f : D \to \mathbb{R}$ over a finite domain $D \subseteq 2^{\Omega}$ if its output S^* using a value oracle for $\hat{f}$ satisfies the relation below with the optimal solution OPT under f, provided that for any $\epsilon > 0$ that $|f(S) - \hat{f}(S)| < \epsilon$ for all $S \in D$,

$$f(S^*) \geq \alpha f(\mathrm{OPT}) - \delta\epsilon.$$

Note that the perturbed $\hat{f}$ is not required to be in the same class as f (linear, quadratic, submodular, etc.). Thus, this definition is a stronger notion of robustness than one limited to $\hat{f}$ in the same class that have bounded L_∞ distance from f.

For (unstructured) k armed bandit problems, one can view the analogous offline algorithm with access to a value oracle for the elements as first evaluating each arm ($D = \{\{1\}, \{2\}, \ldots, \{k\}\}$), so $N = k$ queries total, and then evaluating $\arg\max$ over the k values. That algorithm trivially is a $(1, 2)$-robust approximation algorithm.

Remark 4.2. In (Niazadeh et al., 2021), there is a related definition of robustness for offline approximation algorithms. That definition and the subsequent offline-to-online adaptation procedure is restricted to approximation algorithms with an iterative greedy structure. The criterion Definition 4.1 we consider does not require the approximation algorithm to have an iterative greedy structure.

To illustrate the utility of our proposed framework, in Section 6 we will show that several approximation algorithms from the constrained submodular maximization literature are (α, δ)-robust, leading to new sublinear α-regret algorithms for related stochastic CMAB problems with submodular rewards.

5. C-ETC Algorithm: Offline to Stochastic

In this section, we present our proposed algorithm for adapting offline approximation to algorithms for stochastic CMAB, *Combinatorial Explore-Then-Commit* (C-ETC). The pseudo-code is shown in Algorithm 1. The algorithm takes an offline (α, δ) robust algorithm $\mathcal{A}$ with an upper bound N on the number of oracle queries by $\mathcal{A}$. In the exploration phase, when the offline algorithm queries the value oracle for action A, C-ETC will play action A for m times, where m is a constant chosen to minimizing regret. C-ETC then computes the empirical mean $\bar{f}$ of rewards for A and feeds $\bar{f}$ back to the offline algorithm $\mathcal{A}$. In the exploitation phase, C-ETC keeps playing the solution S output from algorithm $\mathcal{A}$. Thus, the CMAB procedure does not need $\mathcal{A}$ to have any special structure. No careful construction is needed for the CMAB procedure beyond running $\mathcal{A}$. All that is needed is checking robustness (Definition 4.1). Also, there is no over-heard in terms of storage and per-round time complexities— C-ETC is as efficient as the offline algorithm $\mathcal{A}$ itself.

Now we analyze the α-regret for C-ETC (Algorithm 1).

Theorem 5.1. *For the sequential decision making problem defined in Section 2 and $T \geq \frac{2\sqrt{2}N}{\delta}$, the expected cumulative α-regret of C-ETC using an (α, δ)-robust approximation algorithm as subroutine is at most $\mathcal{O}\left(\delta^{\frac{2}{3}} N^{\frac{1}{3}} T^{\frac{2}{3}} \log(T)^{\frac{1}{3}}\right)$, where N upper-bounds the number of value oracle queries made by the offline algorithm $\mathcal{A}$.*

Algorithm 1 Combinatorial Explore-then-Commit

Input: horizon T, set of base elements Ω, an offline (α, δ)-robust algorithm $\mathcal{A}$, and an upper-bound N on the number of $\mathcal{A}$'s queries to the value oracle

Initialize $m \leftarrow \left\lceil \frac{\delta^{2/3} T^{2/3} \log(T)^{1/3}}{2N^{2/3}} \right\rceil$

// Exploration Phase //
while $\mathcal{A}$ queries the value of some $A \subseteq \Omega$ **do**
  For m times, play action A
  Calculate the empirical mean $\bar{f}$
  Return $\bar{f}$ to $\mathcal{A}$
end while

// Exploitation Phase //
for *remaining time* **do**
  Play action S output by algorithm $\mathcal{A}$.
end for

The detailed proof is in the supplementary material. We highlight some key steps.

We show that with high probability, the empirical means of all actions taken during exploration phase will be within $\mathrm{rad} = \sqrt{\frac{\log T}{2m}}$ of their corresponding statistical means. As is common in proofs for ETC methods, we refer to this occurrence as the *clean event* $\mathcal{E}$. Then, using an (α, δ)-robust approximation algorithm as subroutine will guarantee the quality of of the set S used in the exploitation phase of Algorithm 1:

$$f(S) \geq \alpha f(\mathrm{OPT}) - \delta \cdot \mathrm{rad}. \tag{2}$$

We then break up the expected cumulative α-regret conditioned on the clean event $\mathcal{E}$,

$$\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] = \underbrace{\sum_{i=1}^{N} m\left(\alpha f(S^*) - \mathbb{E}[f(S_t)]\right)}_{\text{exploration phase}} + \underbrace{\sum_{t=T_N+1}^{T} \left(\alpha f(S^*) - \mathbb{E}[f(S)]\right)}_{\text{exploitation phase}}. \tag{3}$$

Using the fact that the reward is bounded between $[0, 1]$, we have

$$\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] \leq Nm + T\delta \mathrm{rad}.$$

Optimizing over m then results in

$$\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] = \mathcal{O}\left(\delta^{\frac{2}{3}} N^{\frac{1}{3}} T^{\frac{2}{3}} \log(T)^{\frac{1}{3}}\right).$$

We then show that because the clean event $\mathcal{E}$ happens with high probability, the expected cumulative regret $\mathbb{E}[\mathcal{R}(T)]$ is dominated by $\mathbb{E}[\mathcal{R}(T)|\mathcal{E}]$, which concludes the proof.

Lower bounds: For the general setting we explore in this paper, with stochastic (or even adversarial) combinatorial MAB and only bandit feedback, it is unknown whether $\tilde{\mathcal{O}}(T^{1/2})$ expected cumulative α-regret is possible (ignoring problem parameters like n). For special cases, such as linear reward functions, $\tilde{\mathcal{O}}(T^{1/2})$ is known to be achievable even with bandit feedback. Even for the special case of submodular reward functions and a cardinality constraint, it remains an open question. (Niazadeh et al., 2021) obtain $\tilde{\Omega}(T^{2/3})$ lower bounds for the harder setting where feedback is only available during "exploration" rounds chosen by the agent, who incurs an associated penalty.

Remark 5.2. C-ETC uses knowledge of the horizon T to optimize the number m of samples per action. When the time horizon T is not known, we can use geometric doubling trick to extend our result to an anytime algorithm. We refer to the general detailed procedure in (Besson and Kaufmann, 2018). From Theorem 4 in (Besson and Kaufmann, 2018), we can show that the regret bound conserves the original $T^{2/3}\log(T)^{1/3}$ dependence with only changes in constant factors.

6. Applications on Submodular Maximization

In this section, we apply our general framework to stochastic CMAB problems with monotone submodular rewards where only bandit feedback is available. This application results in the first sublinear α-regret CMAB algorithms for knapsack constraints under bandit feedback. We begin with a brief background, and analyze the robustness of offline approximation algorithms, and then obtain problem independent regret bounds.

6.1. Background and Definitions

Denote the *marginal gain* $f(e|A) = f(A \cup e) - f(A)$ and the *marginal density* $\rho(e|A) = \frac{f(A\cup e)-f(A)}{c(e)}$ for any subset $A \subseteq \Omega$ and element $e \in \Omega \setminus A$. A set function $f : 2^\Omega \to \mathbb{R}$ defined on a finite ground set Ω is said to be *submodular* if it satisfies the diminishing return property: for all $A \subseteq B \subseteq \Omega$, and $e \in \Omega \setminus B$, it holds that $f(e|A) \geq f(e|B)$. A set function is said to be monotonically non-decreasing if $f(A) \leq f(B)$ for all $A \subseteq B \subseteq \Omega$. Our aim is to find a set S such that $f(S)$ is maximized subject to some constraints.

For knapsack constraints, we assume that the cost function $c : \Omega \to R_{>0}$ is known and linear, so the cost of a subset is be the sum of the costs of individual items: $c(A) = \sum_{v\in A} c(v)$. To simplify the presentation, we avoid the cases of trivially large budgets $B > \sum_{v\in\Omega} c(v)$ and assume all items have non-trivial costs $0 < c(v) \leq B$. A cardinality constraint is a special case with unit costs.

In the following, we consider both types of those constraints: cardinality and knapsack. Maximizing a monotone submodular set function under a k-cardinality constraint is NP-hard even with a value oracle (Nemhauser et al., 1978). The best achievable approximation ratio with a polynomial time algorithm is $1-1/e$ (Nemhauser et al., 1978) using $\mathcal{O}(nk)$ oracle calls. In Badanidiyuru and Vondrák (2014), $1 - 1/e - \epsilon'$ is achieved within $\mathcal{O}(\frac{n}{\epsilon'} \log \frac{n}{\epsilon'})$ time, where ϵ' is a user selected parameter to balance accuracy and time complexity.

Maximizing a monotone submodular set function under a knapsack constraint is consequently also NP-hard (Khuller et al., 1999). The best achievable approximation ratio with a polynomial time algorithm is $1 - 1/e$ (Sviridenko, 2004; Khuller et al., 1999), but that requires $\mathcal{O}(n^5)$ function evaluations, making it prohibitive for many applications. There are other offline algorithms that achieve worse approximation ratios but are much more efficient. We adapt a $\frac{1}{2}$ approximation algorithm (Yaroslavtsev et al., 2020) and a $\frac{1}{2}(1 - 1/e)$ approximation algorithm (Khuller et al., 1999), both of which use $\mathcal{O}(n^2)$ function evaluations. There is another algorithm proposed recently in (Li et al., 2022), but since it queries infeasible sets (i.e., it evaluates for some subsets whose cost is above budget B), we do not consider it (see Appendix H for more details).

6.2. Offline Approximation Algorithms – Robustness

For an overview of offline approximation algorithms for submodular optimization, please refer Appendix B. We next state our results on (α, δ)-robustness of the offline algorithms considered. The assumption of complete/noiseless access to a value oracle is often a strong assumption for real world applications. Thus, even for offline applications, it is worthwhile knowing how robust an algorithm is. So the following results are relevant even in the offline setting. For the CMAB setting we consider, robustness is also a sufficient property to guarantee a no-regret adaptation of the offline algorithm. Detailed proofs are included in Appendix C in the supplementary material.

Proposition 6.1 (Corollary 4.3 of Nie et al. (2022))**.** GREEDY *in (Nemhauser et al., 1978) is a $(1 - \frac{1}{e}, 2k)$-robust approximation algorithm for submodular maximization under a k-cardinality constraint.*

Proposition 6.2. THRESHOLDGREEDY *(Badanidiyuru and Vondrák, 2014) is a $(1 - \frac{1}{e} - \epsilon', 2(2 - \epsilon')k)$-robust approximation algorithm for submodular maximization under a k-cardinality constraint.*

Proposition 6.3. PARTIALENUMERATION *(Sviridenko,* *2004; Khuller et al., 1999) is a $(1-\frac{1}{e}, 4+2\tilde{K}+2\beta)$-robust approximation algorithm for submodular maximization under a knapsack constraint.*

Proposition 6.4. GREEDY+MAX *(Yaroslavtsev et al., 2020) is a $(\frac{1}{2}, \frac{1}{2}+\tilde{K}+2\beta)$-robust approximation algorithm for submodular maximization problem under a knapsack constraint.*

Proposition 6.5. GREEDY+ *(Khuller et al., 1999) is a $(\frac{1}{2}(1-\frac{1}{e}), 2+\tilde{K}+\beta)$-robust approximation algorithm for submodular maximization problem under a knapsack constraint.*

Remark 6.6. For the offline setting, GREEDY+MAX is superior to GREEDY+, as it achieves a better α approximation ratio with the same calls to the value oracle. However, their (α, δ) pairs are incomparable, as for $\beta > 1.5$ (with $\beta = 1$ corresponding to a cardinality constraint), GREEDY+ has a smaller δ (thus more robust) which affects exploration time in their adaptations and in turn affects their regret.

To illustrate the robustness analysis, we highlight some key steps for the proof of Proposition 6.4 for GREEDY+MAX. Let $o_1 \in \arg\max_{e:e\in\text{OPT}} c(e)$ denote the most expensive element in OPT. Inspired by the proof techniques in (Yaroslavtsev et al., 2020), we consider the last item added by the greedy solution (based on noisy evaluation) before the cost of this solution exceeds $B - c(o_1)$. Let G_i denote the set selected by GREEDY that has cardinality i and denote the constituent elements as $G_i = \{g_1, \cdots, g_i\}$. Denote G_ℓ as the largest greedy sequence that consumes less than $B - c(o_1)$ of the budget B, so $c(G_\ell) \le B - c(o_1) < c(G_{\ell+1})$. Let S_i denote the augmented set at i-th iteration and S denote the final output of the algorithm. Denote $\hat{f}(e|S) := \hat{f}(S \cup e) - \hat{f}(S)$ and $\hat{\rho}(e|S) := \frac{\hat{f}(S\cup e)-\hat{f}(S)}{c(e)}$. We prove the following lemma.

Lemma 6.7 (GREEDY+MAX inequality)**.** *For $i \in \{0, 1, \cdots, \ell\}$, the following inequality holds:*

$$\begin{aligned}\hat{f}(G_i \cup o_1) + &\max\{0, \hat{\rho}(g_{i+1}|G_i)\}(B - c(o_1)) \\ &\ge f(\text{OPT}) - (2\tilde{K}-1)\epsilon.\end{aligned}$$

For $i = \ell$, Lemma 6.7 tells us that there can be two cases:

$$\hat{f}(G_\ell \cup o_1) \ge \frac{1}{2} f(\text{OPT}) - \left(\tilde{K} - \frac{1}{2} + \gamma\right)\epsilon, \text{ or}$$

$$\hat{\rho}(g_{\ell+1}|G_\ell)(B - c(o_1)) \ge \frac{1}{2} f(\text{OPT}) - \left(\tilde{K} - \frac{1}{2} - \gamma\right)\epsilon,$$

where γ will be selected later to minimize the additive error δ coefficient.

If $\hat{f}(G_\ell \cup o_1) \ge \frac{1}{2} f(\text{OPT}) - \left(\tilde{K} - \frac{1}{2} + \gamma\right)\epsilon$, then denote $a_\ell = \arg\max_{e\in\Omega\setminus G_\ell} \hat{f}(e|G_\ell)$, which is the element selected to augment G_ℓ. We have

$$\begin{aligned}\hat{f}(G_\ell \cup a_\ell) &\ge \hat{f}(G_\ell \cup o_1) \\ &\ge \frac{1}{2} f(\text{OPT}) - \left(\tilde{K} - \frac{1}{2} + \gamma\right)\epsilon. \end{aligned} \tag{4}$$

Then the final output of the algorithm S will satisfy

$$\begin{aligned}f(S) &\ge \hat{f}(S) - \epsilon \\ &\ge \hat{f}(G_\ell \cup a_\ell) - \epsilon \\ &\ge \frac{1}{2} f(\text{OPT}) - \left(\tilde{K} + \frac{1}{2} + \gamma\right)\epsilon. \qquad \text{(using (4))}\end{aligned}$$

If $\hat{\rho}(g_{\ell+1}|G_\ell)(B - c(o_1)) \ge \frac{1}{2} f(\text{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon$, rearranging we have

$$\hat{\rho}(g_{\ell+1}|G_\ell) \ge \frac{f(\text{OPT})}{2(B - c(o_1))} - \frac{(\tilde{K} - \frac{1}{2} - \gamma)\epsilon}{B - c(o_1)}. \tag{5}$$

Moreover,

$$\begin{aligned}\hat{f}(G_\ell) &= \sum_{j=0}^{l-1} \hat{\rho}(g_{j+1}|G_j) c(g_{j+1}) \\ &\ge \sum_{j=0}^{l-1} \hat{\rho}(g_{\ell+1}|G_j) c(g_{j+1}) \qquad (6)\\ &\ge \sum_{j=0}^{l-1} \left(\rho(g_{\ell+1}|G_j) - \frac{2\epsilon}{c(g_{\ell+1})}\right) c(g_{j+1}) \\ &\ge \sum_{j=0}^{l-1} \left(\rho(g_{\ell+1}|G_\ell) - \frac{2\epsilon}{c(g_{\ell+1})}\right) c(g_{j+1}) \qquad (7)\\ &= \left(\rho(g_{\ell+1}|G_\ell) - \frac{2\epsilon}{c(g_{\ell+1})}\right) c(G_\ell) \\ &\ge \left(\hat{\rho}(g_{\ell+1}|G_\ell) - \frac{4\epsilon}{c(g_{\ell+1})}\right) c(G_\ell) \\ &\ge \hat{\rho}(g_{\ell+1}|G_\ell) c(G_\ell) - 4\beta\epsilon, \qquad (8)\end{aligned}$$

where (6) follows from the greedy selection rule, the (7) follows from submodularity of f, and (8) follows from the definition of β. We then have

$$\begin{aligned}&\hat{f}(G_{\ell+1}) \\ &= \hat{f}(G_\ell) + c(g_{\ell+1})\hat{\rho}(g_{\ell+1}|G_\ell) \\ &\ge \left(\hat{\rho}(g_{\ell+1}|G_\ell) c(G_\ell) - 4\beta\epsilon\right) + c(g_{\ell+1})\hat{\rho}(g_{\ell+1}|G_\ell) \qquad (9)\\ &= \hat{\rho}(g_{\ell+1}|G_\ell) c(G_{\ell+1}) - 4\beta\epsilon \\ &\ge \frac{\frac{1}{2} f(\text{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon}{B - c(o_1)} c(G_{\ell+1}) - 4\beta\epsilon \qquad (10)\\ &\ge \frac{1}{2} f(\text{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon - 4\beta\epsilon \qquad (11)\end{aligned}$$

$$= \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} - \frac{1}{2} - \gamma + 4\beta\right)\epsilon, \tag{12}$$

where (9) follows from (8), (10) follows from (5), and (11) follows from the chosen ℓ satisfies $c(G_{\ell+1}) > B - c(o_1)$. Thus, the final output of the algorithm S will satisfy

$$\begin{aligned} f(S) &\geq \hat{f}(S) - \epsilon \\ &\geq \hat{f}(G_{\ell+1}) - \epsilon \\ &\geq \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} + \frac{1}{2} - \gamma + 4\beta\right)\epsilon. \end{aligned}$$

Finally, combining both cases and selecting $\gamma = 2\beta$ completes the proof.

6.3. CMAB algorithms for Submodular Rewards with Knapsack Constraints

Now that we have analyzed the robustness of several offline algorithms, we can invoke Theorem 5.1 to bound the expected cumulative α regret for stochastic CMAB adaptations that rely only on bandit feedback. We name the adapted algorithms as C-ETC-N, C-ETC-B for cardinality constraint, C-ETC-S C-ETC-K and C-ETC-Y for knapsack constraint, respectively, based on which offline algorithm it is adapted from (using the first author's last name); which are in order (Nemhauser et al., 1978; Badanidiyuru and Vondrák, 2014; Sviridenko, 2004; Khuller et al., 1999; Yaroslavtsev et al., 2020). PARTIALENUMERATION was first proposed and analyzed by Khuller et al. (1999) for maximum coverage problems and then analyzed by Sviridenko (2004) for monotone submodular functions. To distinguish CMAB adaptations of GREEDY+ and C-ETC-K, both proposed in Khuller et al. (1999), we use C-ETC-S for the adaption of PARTIALENUMERATION. The following corollaries hold immediately from Propositions 6.1 to 6.5:

Corollary 6.8. *For an online submodular maximization under a cardinality constraint, the expected cumulative* $(1-1/e)$*-regret of C-ETC-N is at most* $\mathcal{O}\left(kn^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ *for* $T \geq \sqrt{2}n$.

Remark 6.9. This result improves upon the result from Nie et al. (2022) by a factor of $k^{\frac{1}{3}}$ despite our use of a generic framework.

Corollary 6.10. *For an online submodular maximization under a cardinality constraint, the expected cumulative* $(1-1/e-\epsilon')$*-regret of C-ETC-B is at most* $\mathcal{O}\left(k^{\frac{2}{3}}n^{\frac{1}{3}}(\epsilon')^{\frac{1}{3}}(\log\frac{n}{\epsilon'})^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ *for* $T \geq \frac{\sqrt{2}n}{(2-\epsilon')\epsilon' k}\log\frac{n}{\epsilon'}$.

Corollary 6.11. *For an online submodular maximization under a knapsack constraint, the expected cumulative* $(1-1/e)$*-regret of C-ETC-S is at most* $\mathcal{O}\left(\beta^{\frac{2}{3}}\tilde{K}^{\frac{1}{3}}n^{\frac{4}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ *for* $T \geq \frac{\sqrt{2}\tilde{K}n^4}{2+\tilde{K}+\beta}$.

Corollary 6.12. *For an online submodular maximization under a knapsack constraint, the expected cumulative* $\frac{1}{2}$*-regret of C-ETC-Y is at most* $\mathcal{O}\left(\beta^{\frac{2}{3}}\tilde{K}^{\frac{1}{3}}n^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ *for* $T \geq \frac{2\sqrt{2}\tilde{K}n}{\frac{1}{2}+\tilde{K}+2\beta}$.

Corollary 6.13. *For an online submodular maximization under a knapsack constraint, the expected cumulative* $\frac{1}{2}(1-\frac{1}{e})$*-regret of C-ETC-K is at most* $\mathcal{O}\left(\beta^{\frac{2}{3}}\tilde{K}^{\frac{1}{3}}n^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ *for* $T \geq \frac{2\sqrt{2}\tilde{K}n}{2+\tilde{K}+\beta}$.

Comparison with OG°: Streeter and Golovin (2008) proposed and analyzed an algorithm for adversarial CMAB with submodular rewards, full-bandit feedback, and under a knapsack constraint (only the expected cost of the (randomly selected) action was required to be under budget). We discuss this in more detail in Appendix H, here only highlighting a few key points. We also use this as a baseline in our experiments in Section 7. The authors adapted a simpler greedy algorithm than the one we adapt (Khuller et al., 1999), using an ϵ-greedy exploration type framework. We provide evidence in our experiments that their algorithm requires large horizons to learn. The offline algorithm they adapted achieves an approximation ratio $(1-1/e)$ for budgets that exactly match the cost used up by the greedy solution, but otherwise does not achieve a constant approximation (Khuller et al., 1999).

Storage and Per-Round Time Complexities: C-ETC-Y and C-ETC-K have low storage complexity and per-round time-complexity. During exploitation, only the indices of at most $\tilde{K}$ base arms are needed in memory and does not need any computation. During exploration, they just need to update the empirical mean for the current action at time t, which can be done in $\mathcal{O}(1)$ time. It additionally stores the highest empirical density so far in the current iteration of the greedy routine and its associated base arm (C-ETC-K needs to store one more arm and C-ETC-Y an additional $\mathcal{O}(\tilde{K})$ storage is needed to store the augmented set). Thus, C-ETC-Y and C-ETC-K have $\mathcal{O}(\tilde{K})$ storage complexity and $\mathcal{O}(1)$ per-round time complexity. For comparison, the algorithm proposed by (Streeter and Golovin, 2008) for an averaged knapsack constraint in the adversarial setting uses $\mathcal{O}(n\tilde{K})$ storage complexity and $\mathcal{O}(n)$ per-round time complexity. Some comments on lower bound are given in Appendix E.

7. Experiments

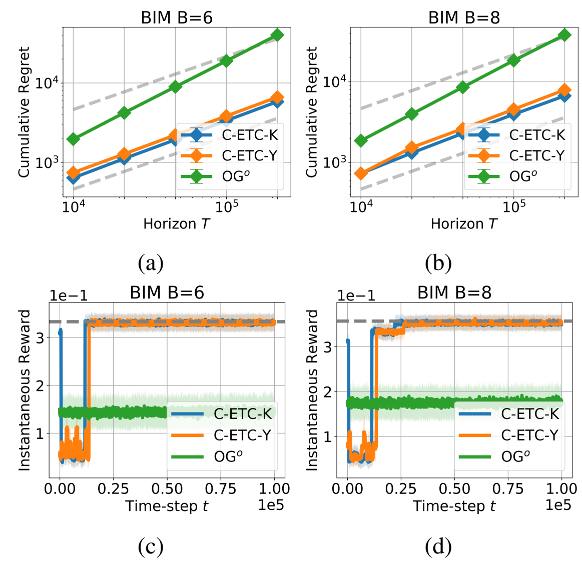
In this section, we conduct experiments on real world data with a Budgeted Influence Maximization (BIM). We also conduct experiments on Song Recommendation (SR) in Appendix I. Both of these are applications of stochastic CMAB with submodular rewards under a knapsack constraint. There are three adaptions we considered in Section 6 for knapsack constraint. Since the time complexity for PARTIALENUMERATION is much larger than the other two offline algorithms we consider, it will use at least $T \approx 10^8$ for C-ETC-S to finish exploration. For this reason, we do not consider C-ETC-S in the experiments. To our knowledge, our work is the first to consider these applications with only bandit feedback available.

Baseline: The only other algorithm designed for combinatorial MAB with general submodular rewards, under a knapsack constraint, and using full-bandit feedback is **Online Greedy with opaque feedback model (OG^o)** proposed by Streeter and Golovin (2008) for the adversarial setting. However, OG^o only satisfies the knapsack constraint in expectation, while our algorithms C-ETC-K ands C-ETC-Y satisfies a strict constraint (i.e. every action A_t must be under budget). See Appendix D for more details about OG^o and its implementation.

In Section 6, we used $N = \tilde{K}n$ as an upper bound on the number of function evaluations for both C-ETC-K and C-ETC-Y, where n is the number of base arms and $\tilde{K}$ is an upper bound of the cardinality of any feasible set. When the time horizon T is small, it is possible that the exploration phase will not finish due to the formula being optimized for m (the number of plays for each action queried by $\mathcal{A}$) uses a loose bound on the exploitation time. When this is the case, we select the largest m (closest to the formula) for which we can guarantee that exploration will finish. For details, see Appendix F.

We first conduct experiments for the application of budgeted influence maximization (BIM) on a portion of the Facebook network graph. BIM models the problem of identifying a low-cost subset (seed set) of nodes in a (known) social network that can influence the maximum number of nodes in a network. While there are prior works proposing algorithms for budgeted online influence maximization problems, the state of the art (e.g., (Perrault et al., 2020)) presumes knowledge of the diffusion model (such as independent cascade) and, more importantly, extensive semi-bandit feedback on individual diffusions, such as which specific nodes became active or along which edges successful infections occurred, in order to estimate diffusion parameters. For social networks with user privacy, this information is not available.

Data Set Description and Experiment Details: The Facebook network dataset was introduced in (Leskovec and Mcauley, 2012). To facilitate running multiple experiments for different horizons, we used the community detection method proposed by (Blondel et al., 2008) to detect a community with 354 nodes and 2853 edges. We further changed the network to be directed by replacing every undirected edge by two directed edge with opposite directions, yielding a directed network with 5706 edges. The diffusion process



(a) (b)

(c) (d)

Figure 1: Plots for budgeted influence maximization (BIM) example. (a) and (b) are comparison results for cumulative regret as a function of time horizon T. (c) and (d) are the moving average plot with window size 100 of instantaneous reward as a function of t. The gray dashed lines in (a) and (b) represent $y = aT^{2/3}$ for various values of a for visual reference. The gray dashed lines in (c) and (d) represent expected rewards for the action chosen by an offline greedy algorithm.

is simulated using the independent cascade model (Kempe et al., 2003), where in each discrete step, an active node (that was inactive at the previous time step) independently attempts to infect each of its inactive neighbors. Following existing work of Tang et al. (2015; 2018); Bian et al. (2020), we set the probability of each edge (u, v) as $1/d_{\text{in}}(v)$, where $d_{\text{in}}(v)$ is the in-degree of node v. Moreover, we consider a user u is more influential if the user has more out-degrees, $d_{\text{out}}(u)$. In our experiment, we only consider influential users to spend our budget more efficiently. We pick the users with out-degrees that are above 95^{th} percentile (18 users). Denote this set as $\mathcal{I}$, then for a user $u \in \mathcal{I}$, the cost is defined as $c(u) = 0.01 d_{\text{out}}(u) + 1$, similar to (Wu et al., 2022). For each time horizon that was used, we ran each method ten times.

For this set of experiments, instead of cumulative $\frac{1}{2}$-regret, which requires knowing OPT, we compare the cumulative rewards achieved by C-ETC and OG^o against $Tf(S^{\text{grd}})$, where S^{grd} is the solution returned by the offline $\frac{1}{2}$-approximation algorithm proposed by (Yaroslavtsev et al., 2020). $Tf(S^{\text{grd}}) \geq \frac{1}{2}Tf(\text{OPT})$, so $Tf(S^{\text{grd}})$ is a more challenging reference value.

Results and Discussion: Figures 1a and 1b show average cumulative regret curves for C-ETC-K (in blue), C-ETC-Y (in orange) and OG^o (in green) for different horizon T values when the budget constraint B is 6 and 8, respectively.

For $B = 8$, the turning point is $T = 21544$. Standard errors of means are presented as error bars, but might be too small to be noticed. Figures 1c and 1d are the instantaneous reward plots. In these plots, standard errors of means are presented as shaded areas. The peaks at the very beginning of exploration phase correspond to the time step that the single person with highest influence is sampled.

C-ETC significantly outperforms OG^{o} for all time horizons and budget considered. To evaluate the gap between the empirical performance and the theoretical guarantee, we estimated the slope for both methods on log-log scale plots. Over the horizons tested, OG^{o}'s cumulative regret (averaged over ten runs) has a growth rate of 0.98. The growth rates of C-ETC-K for budgets 6 and 8 are 0.76 and 0.68, respectively. The growth rates of C-ETC-Y for budgets 6 and 8 are 0.75 and 0.69, respectively. The slopes are close to the $2/3 \approx 0.67$ theoretical guarantee, and notably, the performance for larger B is better.

8. Conclusions and Future Directions

In this paper, we provide a general framework for adapting discrete offline approximation algorithms for combinatorial optimization problems into sublinear α-regret methods that only require bandit feedback. Through our proposed framework, we achieve $\mathcal{O}\left(T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right)$ expected cumulative α-regret dependence on the horizon T. Importantly, our approach only relies on the offline algorithms being robust to small errors in function evaluation. The offline algorithm can be treated as a black box subroutine, making our framework easily applicable in practice. The results are demonstrated on multiple problems in constrained submodular optimization.

Our findings pave the way for further exploration and development of algorithms for similar problems, opening up new avenues for research in this area. Recently, (Fourati et al., 2023) considered submodular maximization with bandit feedback and non-monotone rewards. Exploring this as a special case of framework will be considered as a future work. Further, finding regret guarantees for non-monotone submodular maximization subject to a knapsack constraint with bandit feedback is open, where the result with semi-bandit feedback has been studied in (Amanatidis et al., 2020). Finally, exploring product ranking optimization in online platforms and reserve price optimization in auctions (considered in (Niazadeh et al., 2021) for adversarial CMAB) as a special case of our framework will be considered as future work.

9. Acknowledgements

This work was supported in part by the National Science Foundation under grants 2149588 and 2149617.

References

Amanatidis, G., Fusco, F., Lazos, P., Leonardi, S., and Reiffenhäuser, R. (2020). Fast adaptive non-monotone submodular maximization subject to a knapsack constraint. *Advances in neural information processing systems*, 33:16903–16915.

Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164.

Badanidiyuru, A. and Vondrák, J. (2014). Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1497–1514. SIAM.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Besson, L. and Kaufmann, E. (2018). What doubling tricks can and can't do for multi-armed bandits. *ArXiv*, abs/1803.06971.

Bian, S., Guo, Q., Wang, S., and Yu, J. X. (2020). Efficient algorithms for budgeted influence maximization on massive social networks. *Proceedings of the VLDB Endowment*, 13(9):1498–1510.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:10008.

Fourati, F., Aggarwal, V., Quinn, C., and Alouini, M.-S. (2023). Randomized greedy learning for non-monotone stochastic submodular maximization under full-bandit feedback. In *International Conference on Artificial Intelligence and Statistics*, pages 7455–7471. PMLR.

Golovin, D., Krause, A., and Streeter, M. (2014). Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*.

Hiranandani, G., Singh, H., Gupta, P., Burhanuddin, I. A., Wen, Z., and Kveton, B. (2020). Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In Adams, R. P. and Gogate, V., editors, *Proceedings of The 35th Uncertainty* *in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 722–732. PMLR.

Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146.

Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.

Krause, A. and Guestrin, C. (2005). A note on the budgeted maximization on submodular functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 420–429. Association for Computing Machinery.

Leskovec, J. and Mcauley, J. (2012). Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Li, W., Feldman, M., Kazemi, E., and Karbasi, A. (2022). Submodular maximization in clean linear time. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Lin, T., Li, J., and Chen, W. (2015). Stochastic online greedy learning with semi-bandit feedbacks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*, pages 352–360.

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

Nguyen, H. and Zheng, R. (2013). On budgeted influence maximization in social networks. *IEEE Journal on Selected Areas in Communications*, 31:1084–1094.

Niazadeh, R., Golrezaei, N., Wang, J. R., Susan, F., and Badanidiyuru, A. (2021). Online learning via offline greedy algorithms: Applications in market design and optimization. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 737–738.

Nie, G., Agarwal, M., Umrawal, A. K., Aggarwal, V., and Quinn, C. J. (2022). An explore-then-commit algorithm for submodular maximization under full-bandit feedback. In *Uncertainty in Artificial Intelligence*, pages 1541–1551. PMLR.

Perrault, P., Healey, J., Wen, Z., and Valko, M. (2020). Budgeted online influence maximization. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7620–7631. PMLR.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

Streeter, M. and Golovin, D. (2008). An online algorithm for maximizing submodular functions. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, page 1577–1584.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43.

Takemori, S., Sato, M., Sonoda, T., Singh, J., and Ohkuma, T. (2020a). Submodular bandit problem under multiple constraints. In *Conference on Uncertainty in Artificial Intelligence*, pages 191–200. PMLR.

Takemori, S., Sato, M., Sonoda, T., Singh, J., and Ohkuma, T. (2020b). Submodular bandit problem under multiple constraints. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 191–200. PMLR.

Tang, J., Tang, X., Xiao, X., and Yuan, J. (2018). Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*, pages 991–1005.

Tang, Y., Shi, Y., and Xiao, X. (2015). Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554.

Wang, S., Yang, S., Xu, Z., and Truong, V. (2020). Fast Thompson sampling algorithm with cumulative oversampling: Application to budgeted influence maximization. *CoRR*, abs/2004.11963.

Wu, J., Gao, J., Zhu, H., and Zhang, Z. (2022). Budgeted influence maximization via boost simulated annealing in social networks. *arXiv preprint arXiv:2203.11594*.

Yaroslavtsev, G., Zhou, S., and Avdiukhin, D. (2020). "Bring your own greedy"+max: Near-optimal 1/2-approximations for submodular knapsack. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty*

Third International Conference on Artificial Intelligence and Statistics, volume 108 of *Proceedings of Machine Learning Research*, pages 3263–3274. PMLR.

Yu, B., Fang, M., and Tao, D. (2016). Linear submodular bandits with a knapsack constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Yue, Y. and Guestrin, C. (2011). Linear submodular bandits and their application to diversified retrieval. *Advances in Neural Information Processing Systems*, 24.

A. Proof for Regret of C-ETC

In this section, we prove Theorem 5.1 in Section 4 of the main paper. We restate the theorem: For the sequential decision making problem defined in Section 2 and $T \geq \frac{2\sqrt{2}N}{\delta}$, the expected cumulative α-regret of C-ETC using an (α, δ)-robust approximation algorithm as subroutine is at most $\mathcal{O}\left(\delta^{\frac{2}{3}} N^{\frac{1}{3}} T^{\frac{2}{3}} \log(T)^{\frac{1}{3}}\right)$, where N upper-bounds the number of value oracle queries made by the offline algorithm $\mathcal{A}$.

A.1. Overview and Notations

We will separate the proof into two cases. The first case is for when the clean event $\mathcal{E}$ happens, which we will show in Lemma A.3 happens with high probability. Under the clean event, using the fact that the offline algorithm is an (α, δ)-robust approximation, C-ETC's chosen set S for the exploitation phase will nonetheless be near-optimal. The second case is when the complementary event happens, which occurs with low probability.

The proof structure analyzing a high-probability "clean event" where empirical estimates are sufficiently concentrated around their means is analogous to that for the unstructured non-combinatorial setting (see for instance, Section 1.2 in (Slivkins, 2019)). However, unlike the ETC procedure for non-combinatorial MAB problems, C-ETC makes sequences of decisions during exploration. Furthermore, the combinatorial action space, non-linearity of the reward function, and lack of extra feedback (like marginal gains) make the problem challenging. Even in the special setting of deterministic rewards, the standard MAB problem becomes trivial (finding the largest of n base arms) while the problem we considered are NP-hard.

Recap that for any (feasible) action A, $f_t(A)$ denotes a (random) reward at time t for the agent taking that action, $f(A)$ denotes the expected value for action A. Let $\bar{f}_t(A)$ denote the empirical mean of rewards received from playing action A up to and including time t. In the following, we will drop the subscript t from the empirical mean, writing $\bar{f}(A)$ when it is clear from context that action A has been played m times. Also, we use A_i, $i \in \{1, \cdots, N\}$ denotes the i-th action the algorithm samples. We further denote $T_i, i \in \{1, \ldots, N\}$ as the time step when the sampling of the i-th action has been determined, or A_i has been played m times. For notation consistency, we also denote $T_0 = 0$ and $T_{N+1} = T$.

A.2. Probability of the Clean Event

Now we define events that are important in our analysis. Recall that for each action A being explored, the m rewards are i.i.d. with mean $f(A)$ and bounded in $[0, 1]$. Thus, we can bound the deviation of the (unbiased) empirical mean $\bar{f}(A_i)$ from the expected value $f(A_i)$ for each action played. Specifically, we can use a two-sided Hoeffding bound for bounded variables.

Remark A.1. For convenience, we assume the reward function bounded in $[0, 1]$, but the result can be generalized to the case where the deviation of the true reward and the expected reward has a light tailed distribution (e.g., sub-Gaussian).

Lemma A.2 (Hoeffding's inequality)**.** *Let $X_1, \cdots, X_n$ be independent random variables bounded in the interval $[0, 1]$, and let $\bar{X}$ denote their empirical mean. Then we have for any $\epsilon > 0$,*

$$\mathbb{P}\left(\left|\bar{X} - \mathbb{E}[\bar{X}]\right| \geq \epsilon\right) \leq 2\text{exp}\left(-2n\epsilon^2\right). \tag{13}$$

By C-ETC, each sampled action will be played the same number of times, denoted by m, so we consider bounding the probabilities of equal-sized confidence radii $\mathrm{rad} := \sqrt{\log(T)/2m}$ for all the actions played during exploration.

We next analyze the probability of the event that the empirical means of all actions played during exploration are concentrated around their statistical means within a radius rad. Denote the corresponding events for each action played having empirical means concentrated around their respective statistical means as $\mathcal{E}_i$,

$$\mathcal{E}_i := \bigcap \{\left|\bar{f}(A_i) - f(A_i)\right| < \mathrm{rad}\}, \quad i \in \{1, \cdots, N\}. \tag{14}$$

Define the *clean event* $\mathcal{E}$ to be the event that the empirical means of all actions played in the exploration phase are within rad of their corresponding statistical means:

$$\mathcal{E} := \mathcal{E}_1 \cap \cdots \cap \mathcal{E}_N. \tag{15}$$

Lemma A.3. *The probability of the clean event $\mathcal{E}$ (15) satisfies:*

$$\mathbb{P}(\mathcal{E}) \geq 1 - \frac{2N}{T}.$$

Proof. Applying the Hoeffding bound Lemma A.2 to the empirical mean $\bar{f}(A_i)$ of m rewards for action A_i and choosing $\epsilon = \mathrm{rad} = \sqrt{\log(T)/2m}$ gives

$$\begin{aligned}
\mathbb{P}(\bar{\mathcal{E}}_i) &= \mathbb{P}\left[\left|\bar{f}(A_i) - f(A_i)\right| \geq \mathrm{rad}\right] \\
&\leq 2\exp\left(-2m\mathrm{rad}^2\right) \\
&= 2\exp\left(-2m(\log(T)/2m)\right) \\
&= 2\exp\left(-\log(T)\right) \\
&= \frac{2}{T}.
\end{aligned} \tag{16}$$

Then, we can bound the probability of clean events

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \mathbb{P}(\mathcal{E}_1 \cap \cdots \cap \mathcal{E}_N) \\
&= 1 - \mathbb{P}(\bar{\mathcal{E}}_1 \cup \cdots \cup \bar{\mathcal{E}}_N) && \text{(De Morgan's Law)} \\
&\geq 1 - \sum_{i=1}^{N} \mathbb{P}(\bar{\mathcal{E}}_i) && \text{(union bounds)} \\
&\geq 1 - \frac{2N}{T}. && \text{(using (16))}
\end{aligned}$$

□

A.3. Near Optimality of the final S (Exploitation Phase Action)

In Lemma A.3, we showed that the clean event $\mathcal{E}$ will happen with high probability. When the clean event $\mathcal{E}$ happens, we have $|\bar{f}(A) - f(A)| \leq \mathrm{rad}$ for all evaluated action A. For an online algorithm (with output S) using an (α, δ)-robust approximation as subroutine, we have

$$f(S) \geq \alpha f(\mathrm{OPT}) - \delta \cdot \mathrm{rad}. \tag{17}$$

A.4. Final Regret

Now we are ready to show the regret of C-ETC (Theorem 5.1 in Section 4 of the main paper).

CASE 1: CLEAN EVENT $\mathcal{E}$ HAPPENS

In the first case we analyse the expected regret under the condition that the clean event $\mathcal{E}$ happens. In this section, all expectations will be conditioned on $\mathcal{E}$, but to simplify notation we will write $\mathbb{E}[\cdot]$ instead of $\mathbb{E}[\cdot|\mathcal{E}]$ in some cases.

First we can break up the expected α-regret conditioned on $\mathcal{E}$ into two parts, one for the first L exploration iterations, and the second for the exploitation iteration. Although the number of actions taken per iteration and the number of iterations of the greedy is not known a priori, we can upper bound the duration. Also recall that $f_t(A_t)$ is the random reward for taking action A_t, which itself is random, depending on empirical means of actions in earlier iterations.

$$\begin{aligned}
\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] &= \alpha T f(\mathrm{OPT}) - \sum_{t=1}^{T} \mathbb{E}[f_t(A_t)] \\
&= \alpha T f(\mathrm{OPT}) - \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[f_t(A_t)|A_t]] && \text{(law of total expectation)} \\
&= \alpha T f(\mathrm{OPT}) - \sum_{t=1}^{T} \mathbb{E}[f(A_t)] && \text{($f(\cdot)$ defined as expected reward)} \\
&= \sum_{t=1}^{T} (\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_t)]) && \text{(rearranging)} \\
&= \underbrace{\sum_{i=1}^{N} m\left(\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_i)]\right)}_{\text{Exploration phase}} + \underbrace{\sum_{t=T_N+1}^{T} (\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_t)])}_{\text{Exploitation phase}} \\
&= \sum_{i=1}^{N} m\left(\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_i)]\right) + \sum_{t=T_N+1}^{T} (\alpha f(\mathrm{OPT}) - \mathbb{E}[f(S)]) . && (18)
\end{aligned}$$

Case 1 (clean event): Bounding exploration regret: We will separately bound the regret incurred from the exploration and exploitation. We begin with bounding regret from exploration,

$$\begin{aligned}
&\sum_{i=1}^{N} m\left(\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_i)]\right) \\
&\qquad \le \sum_{i=1}^{N} m\left(\alpha - 0\right) && \text{(rewards are bounded in $[0,1]$)} \\
&\qquad \le Nm. && (19)
\end{aligned}$$

Case 1 (clean event): Bounding exploitation regret: We next bound the regret incurred during the exploitation iteration. Since the set S used during exploitation is a random variable, we can take the expectation of (17) (conditioned on event $\mathcal{E}$), to bound the expected instantaneous regret for each time step of the exploitation iteration,

$$\alpha f(\mathrm{OPT}) - \mathbb{E}[f(S)] \le \delta \mathrm{rad}. \tag{20}$$

Using a loose bound for the duration of the exploitation iteration, $T - T_L + 1 < T$,

$$\begin{aligned}
\sum_{t=T_N+1}^{T} (\alpha f(\mathrm{OPT}) - \mathbb{E}[f(S)]) &\le \sum_{t=T_N+1}^{T} \delta \mathrm{rad} && \text{(using (20))} \\
&\le T\delta \mathrm{rad}. && (21)
\end{aligned}$$

Case 1 (clean event): Bounding total regret: Then the expected cumulative regret (18) can be bounded as

$$\begin{aligned}
\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] &= \sum_{i=1}^{N} m\left(\alpha f(\mathrm{OPT}) - \mathbb{E}[f(A_i)]\right) + \sum_{t=T_N+1}^{T} (\alpha f(\mathrm{OPT}) - \mathbb{E}[f(S)]) && \text{(copying (18))} \\
&\le Nm + T\delta \mathrm{rad} && \text{(using (19), (21))}
\end{aligned}$$

Plugging in the formula for the confidence radius $\mathrm{rad} = \sqrt{\log(T)/2m}$, we have

$$\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] \leq Nm + T\delta\sqrt{\log(T)/2m}$$

We want to optimize m, the number of times each action is played. Denoting the regret bound (22) as a function of m

$$g(m) = Nm + T\delta\sqrt{\log(T)/2m}, \tag{22}$$

then

$$g'(m) = N - \frac{1}{2}T\delta\sqrt{\log(T)/2}m^{-3/2}. \tag{23}$$

Setting $g'(m) = 0$ and solving for m,

$$m^* = \frac{\delta^{2/3}T^{2/3}\log(T)^{1/3}}{2N^{2/3}}. \tag{24}$$

We next check the second derivative,

$$g''(m) = \frac{3}{4}\delta T\sqrt{\log(T)/2}m^{-5/2}. \tag{25}$$

For positive values of m, $g''(m) > 0$, thus $g(m)$ reaches a minimum at (24).

Since m is the number of times actions are played, we (trivially) need $m \geq 1$ and m to be an integer. We choose

$$m^\dagger = \left\lceil \frac{\delta^{2/3}T^{2/3}\log(T)^{1/3}}{2N^{2/3}} \right\rceil. \tag{26}$$

Since from (25) we have that $g''(m) > 0$ for positive m, $g(m^*) \leq g(m^\dagger)$. For $T \geq \frac{2\sqrt{2}N}{\delta}$, we have $m^* \geq 1$.

Plugging (26) back in to (22),

$$\begin{aligned}
\mathbb{E}[\mathcal{R}(T)|\mathcal{E}] &\leq m^\dagger N + T\delta\sqrt{\log(T)/2m^\dagger} && \text{((22) with } m^\dagger \text{ samples for each action)} \\
&= \lceil m^* \rceil N + T\delta\sqrt{\log(T)/2\lceil m^* \rceil} \\
&\leq \lceil m^* \rceil N + T\delta\sqrt{\log(T)/2m^*} && \text{(Since } \lceil m^* \rceil \geq m^*) \\
&\leq 2m^* N + T\delta\sqrt{\log(T)/2m^*} && \text{(Since } m^* \geq 1, \lceil m^* \rceil \leq 2m^*) \\
&= 2\frac{\delta^{2/3}T^{2/3}\log(T)^{1/3}}{2N^{2/3}}N \\
&\quad + T\delta\sqrt{\log(T)/2}\left(\frac{\delta^{2/3}T^{2/3}\log(T)^{1/3}}{2N^{2/3}}\right)^{-1/2} && \text{(using (24))} \\
&= 3\delta^{2/3}N^{1/3}T^{2/3}\log(T)^{1/3} && (27) \\
&= \mathcal{O}\left(\delta^{\frac{2}{3}}N^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right).
\end{aligned}$$

In conclusion, the expected α-regret of C-ETC using an (α, δ)-robust approximation as subroutine is upper bounded by (27) if the clean event $\mathcal{E}$ happens.

CASE 2: CLEAN EVENT $\mathcal{E}$ DOES NOT HAPPEN

We next derive an upper bound for the expected α-regret for case that the event $\mathcal{E}$ does not happen. By Lemma A.3,

$$\mathbb{P}(\bar{\mathcal{E}}) = 1 - \mathbb{P}(\mathcal{E}) \leq \frac{2N}{T}.$$

Since the reward function $f_t(\cdot)$ is upper bounded by 1, the expected α-regret incurred under $\bar{\mathcal{E}}$ for a horizon of T is at most T,

$$\mathbb{E}[\mathcal{R}(T)|\bar{\mathcal{E}}] \leq T. \tag{28}$$

PUTTING IT ALL TOGETHER

Combining Cases 1 and 2 we have,

$$\begin{aligned}
\mathbb{E}[\mathcal{R}(T)] &= \mathbb{E}[\mathcal{R}(T)|\mathcal{E}]\cdot\mathbb{P}(\mathcal{E}) + \mathbb{E}[\mathcal{R}(T)|\bar{\mathcal{E}}]\cdot\mathbb{P}(\bar{\mathcal{E}}) && \text{(Law of total expectation)}\\
&\le 3\delta^{2/3}N^{1/3}T^{2/3}\log(T)^{1/3}\cdot 1 + T\cdot\frac{2N}{T} && \text{(using (27), Lemma A.3, and (28))}\\
&= \mathcal{O}\left(\delta^{\frac{2}{3}}N^{\frac{1}{3}}T^{\frac{2}{3}}\log(T)^{\frac{1}{3}}\right).
\end{aligned}$$

This concludes the proof.

B. Offline Approximation Algorithms – Overview

We give a brief overview of the offline approximation algorithms which we will analyze (α, δ) robustness for.

For a k-cardinality constraint, the greedy algorithm GREEDY proposed in Nemhauser et al. (1978) starts from an empty set $G \leftarrow \emptyset$. Then it repeatedly add the element with highest marginal gain $f(e|G)$ until the cardinality $|G|$ reaches k. THRESHOLDGREEDY, proposed in Badanidiyuru and Vondrák (2014), considers a sequence of decreasing thresholds: $\{\tau = d; \tau \ge \frac{\epsilon'}{n}d; \tau \leftarrow (1-\epsilon')\tau\}$ where $d = \max_{e\in\Omega} f(e)$. Then starting from empty set $G = \emptyset$, the algorithm includes any element $e \notin G$ such that $f(e|G) \ge \tau$ whenever the cardinality is smaller than k. The algorithm then repeats using a lower threshold. Badanidiyuru and Vondrák (2014) showed that THRESHOLDGREEDY can achieve $1 - 1/e - \epsilon'$ approximation.

For a knapsack constraint, several algorithms run the following greedy subroutine, which we refer to as GREEDY (cardinality is a special case of this routine with budget k and unit cost, so we keep the same name without confusion). Start with empty set $G \leftarrow \emptyset$. Repeatedly add the element e with the highest marginal density $\rho(e|G)$ that fits into the budget. Let G_i denote the set selected by GREEDY that has cardinality i and denote the constituent elements as $G_i = \{g_1, \cdots, g_i\}$. Let L denote the cardinality of the final greedy set (i.e. when no more elements remain that are under budget), so G_L is output by GREEDY. Note that L can only be bounded ahead of time—there could be maximal subsets (to which no other elements could be added without violating the budget) of different cardinalities.

GREEDY can have an unbounded approximation ratio Khuller et al. (1999) for knapsack constraint. Khuller et al. (1999) proposed GREEDY+, which outputs the better of the best individual element $a^* \in \arg\max_{e\in\Omega} f(e)$ and the output of GREEDY, $\arg\max_{S\in\{G_L, a^*\}} f(S)$. Khuller et al. (1999) proved that GREEDY+ achieves a $\frac{1}{2}(1-\frac{1}{e})$ approximation ratio. Then, Sviridenko (2004); Khuller et al. (1999) proposed PARTIALENUMERATION. It first enumerate all sets with cardinality up to three. For each enumerated triplets, it build the rest of the solution set greedily. Then it outputs the set with largest value among all evaluated sets. They showed that PARTIALENUMERATION can achieve $1 - 1/e$ approximation ratio.

Greedy+Max generalizes GREEDY+ by augmenting each set $\{G_i\}_{i=1}^{L}$ in the nested sequence produced by GREEDY with another element. For $0 \le i \le L-1$, define $G'_i \leftarrow G_i \cup \arg\max_{e\in\Omega: c(G_i)+c(e)\le B} f(G_i \cup e)$. By construction, $G'_0 = \{a^*\}$, the best individual element. For $i = L$, $G'_L \leftarrow G_L$. GREEDY+MAX then outputs the best set in the augmented sequence, $\arg\max_{S\in\{G'_0,\ldots,G'_L\}} f(S)$. (Yaroslavtsev et al., 2020) proposed GREEDY+MAX and proved it achieves an approximation ratio of $\frac{1}{2}$.

A bound on the number of value oracle calls will be important in adapting offline methods. Denote $\beta := B/c_{\min}$ and $\tilde{K} := \min\{n, \beta\}$ as an upper bound of the number of items in any feasible set. We note here that while PARTIALENUMERATION uses $\mathcal{O}(\tilde{K}n^4)$ function evaluations, both GREEDY+MAX and GREEDY+ use $\mathcal{O}(\tilde{K}n)$ oracle calls, same as GREEDY. We use $N = \tilde{K}n$ in the analysis for GREEDY+MAX and GREEDY+.

C. Proof for Robustness of Offline Algorithms

In this section, we prove the (α, δ) robustness of algorithms considered in Section 6 of the main paper.

C.1. Notation

We first review notations used in the analysis. Recall that we are only able to evaluate the surrogate function $\hat{f}$ such that $|\hat{f}(S) - f(S)| \le \epsilon$ for any feasible set S and some $\epsilon > 0$, we further denote $\hat{f}(e|S) = \hat{f}(S\cup e) - \hat{f}(S)$ and $\hat{\rho}(e|S) = \frac{\hat{f}(S\cup e)-\hat{f}(S)}{c(e)}$. Let G_i denote the set selected by basic GREEDY (based on surrogate function $\hat{f}$) as described in Section 3 up until ith item and $G_i = \{g_1, \cdots, g_i\}$ in the order of each item is selected. Without loss of generality, define $G_0 = \emptyset$ and $f(G_0) = \hat{f}(G_0) = 0$. Denote $c_{\min} = \min_{e\in\Omega} c(e)$ be the item with lowest individual cost. Let $\beta = B/c_{\min}$ and $\tilde{K} = \min\{n, \beta\}$ being an upper bound of the number of items in any feasible set. Since all selected actions should be feasible, for ease of notation, we omit denoting that condition throughout the proof. For example, we write $\arg\max_{e\in\Omega\setminus A} f(e|A)$ to simplify the notation of $\arg\max_{e:e\in\Omega\setminus A \text{ and } A\cup e\in D} f(e|A)$. Let S be the set returned by modified algorithms in corresponding context.

C.2. Robustness of Offline Methods for Submodular Maximization under Cardinality Constraint

C.2.1. GREEDY

We consider the original greedy algorithm GREEDY proposed in Nemhauser et al. (1978), which gives a $(1-\frac{1}{e})$-approximation guarantee for submodular maximization under a k-cardinality constraint. To restate Proposition 6.1 in the main paper, GREEDY is a $(1-\frac{1}{e}, 2k)$-robust approximation algorithm for submodular maximization under a k-cardinality constraint. The result follows from Corollary 4.3 of Nie et al. (2022), part of the regret analysis for a CMAB adaptation of GREEDY.

C.2.2. THRESHOLDGREEDY

We then consider the threshold greedy algorithm THRESHOLDGREEDY proposed in Badanidiyuru and Vondrák (2014), which gives a $(1-\frac{1}{e}-\epsilon')$-approximation guarantee for submodular maximization under a k-cardinality constraint, where ϵ' is a user specified parameter to balance accuracy and run time. Restating Proposition 6.2 in the main paper, THRESHOLDGREEDY is a $(1-\frac{1}{e}-\epsilon', 2(2-\epsilon')k)$-robust approximation algorithm for submodular maximization under a k-cardinality constraint.

Proof. From the assumption of the surrogate function $\hat{f}$ we know

$$f(e|S) - 2\epsilon \le \hat{f}(e|S) \le f(e|S) + 2\epsilon$$

for any $e \in \Omega \setminus S$ and $S \subseteq \Omega$. Now assume the the next chosen element is a and the current partial solution is S. On one hand, we have

$$\hat{f}(a|S) \ge w \Longrightarrow f(a|S) \ge w - 2\epsilon, \tag{29}$$

on the other hand, for every $e \in \mathrm{OPT} \setminus S$,

$$\hat{f}(e|S) \le \frac{w}{1-\epsilon'} \Longrightarrow f(e|S) \le \frac{w}{1-\epsilon'} + 2\epsilon. \tag{30}$$

Combining and manipulating (29) and (30) we have for any $e \in \mathrm{OPT} \setminus S$:

$$f(a|S) + 2\epsilon \ge (f(e|S) - 2\epsilon)(1-\epsilon') \Longrightarrow f(a|S) \ge (1-\epsilon')f(e|S) - 2(2-\epsilon')\epsilon. \tag{31}$$

Taking an average over all $e \in \mathrm{OPT} \setminus S$,

$$\begin{aligned} f(a|S) &\ge \frac{1-\epsilon'}{|\mathrm{OPT}\setminus S|} \sum_{e\in \mathrm{OPT}\setminus S} f(e|S) - 2(2-\epsilon')\epsilon \\ &\ge \frac{1-\epsilon'}{k} \sum_{e\in \mathrm{OPT}\setminus S} f(e|S) - 2(2-\epsilon')\epsilon. \end{aligned} \tag{32}$$

Now consider after $i \in [k-1]$ steps, we get a partial solution $S_i = \{a_1, \cdots, a_i\}$. By (32), we have

$$\begin{aligned} f(a_{i+1}|S_i) &\ge \frac{1-\epsilon'}{k} \sum_{e\in \mathrm{OPT}\setminus S} f(e|S_i) - 2(2-\epsilon')\epsilon \\ &\ge \frac{1-\epsilon'}{k} f(\mathrm{OPT}|S_i) - 2(2-\epsilon')\epsilon && \text{(submodularity)} \\ &\ge \frac{1-\epsilon'}{k} \left(f(\mathrm{OPT}) - f(S_i)\right) - 2(2-\epsilon')\epsilon, && \text{(monotonicity)} \end{aligned}$$

and hence for $i \in [k-1]$,

$$f(S_{i+1}) - f(S_i) = f(a_{i+1}|S_i) \geq \frac{1-\epsilon'}{k}\left(f(\mathrm{OPT}) - f(S_i)\right) - 2(2-\epsilon')\epsilon. \tag{33}$$

Using (33) as induction hypothesis, we then prove by induction (omitted) that for $i \in [k-1]$,

$$f(S_{i+1}) \geq \left[1 - \left(1 - \frac{1-\epsilon'}{k}\right)^{i+1}\right] f(\mathrm{OPT}) - 2(i+1)(2-\epsilon')\epsilon,$$

and plugging in $i = k-1$ we get

$$\begin{aligned} f(S_k) &\geq \left[1 - \left(1 - \frac{1-\epsilon'}{k}\right)^{k}\right] f(\mathrm{OPT}) - 2k(2-\epsilon')\epsilon \\ &\geq (1 - e^{-(1-\epsilon')}) f(\mathrm{OPT}) - 2k(2-\epsilon')\epsilon \\ &\geq (1 - 1/e - \epsilon') f(\mathrm{OPT}) - 2k(2-\epsilon')\epsilon. \end{aligned}$$

We finish the proof by observing that S_k is the output. □

C.3. Proof for Robustness of GREEDY+MAX

In this section, we give a detailed proof for Proposition 6.4 in Section 6 of the main paper. Recall the statement is that GREEDY+MAX is a $(\frac{1}{2}, \frac{1}{2} + \tilde{K} + 2\beta)$-robust approximation algorithm for submodular maximization problem under a knapsack constraint.

Let $o_1 \in \arg\max_{e:e\in\mathrm{OPT}} c(e)$ denote the most expensive element in OPT. During the ith iteration of the greedy process, having previously selected the set G_{i-1} with $i-1$ elements, it will select the element g_i with highest marginal density (based on surrogate function $\hat{f}$) among feasible elements,

$$g_i = \underset{e:\, e\in\Omega\setminus G_{i-1}}{\arg\max}\ \hat{\rho}(e|G_{i-1}). \tag{34}$$

Inspired by the proof techniques in Yaroslavtsev et al. (2020), we consider the last item added by the greedy solution (based the surrogate function $\hat{f}$) before the cost of this solution exceeds $B - c(o_1)$. Denote G_ℓ as the largest greedy sequence that consumes less than $B - c(o_1)$ budgets, $c(G_\ell) \leq B - c(o_1) < c(G_{\ell+1})$. Let a_i denote the element selected to augment with the greedy solution G_i, i.e., $a_i = \arg\max_{e\in\Omega\setminus G_i} \hat{f}(e|G_i)$, and S_i denote the augmented set at i-th iteration. Before proving the theorem, we show Lemma 6.7 in Section 6 of the main paper, that for $i \in \{0, 1, \cdots, \ell\}$, the following inequality holds:

$$\hat{f}(G_i \cup o_1) + \max\{0, \hat{\rho}(g_{i+1}|G_i)\}(B - c(o_1)) \geq f(\mathrm{OPT}) - (2\tilde{K} - 1)\epsilon.$$

Proof. Recall that from the definition of $\hat{f}$, we have $|\hat{f}(S) - f(S)| \leq \epsilon$ for any evaluated set S and some $\epsilon > 0$. Consequently, we have for any $i \in \{0, 1, \cdots, \ell\}$,

$$|\hat{f}(G_i) - f(G_i)| \leq \epsilon. \tag{35}$$

Now we evaluate the set $G_i \cup o_1$.

- **Case 1:** If o_1 has already been added, $o_1 \in G_i$, then

$$|\hat{f}(G_i \cup o_1) - f(G_i \cup o_1)| = |\hat{f}(G_i) - f(G_i)| \leq \epsilon.$$

- **Case 2**: If $o_1 \notin G_i$, then $\hat{f}(G_i \cup o_1)$ is evaluated in iteration $i+1$. This iteration $i+1$ does exist[1] because for any $i \in \{0, 1, \cdots, \ell\}$, we only used less than $B - c(o_1)$ budget. For the remaining budget, at least o_1 can still fit into the budget so $G_i \cup o_1$ will be evaluated in iteration $i+1$. In this case, we still have

$$|\hat{f}(G_i \cup o_1) - f(G_i \cup o_1)| \leq \epsilon.$$

[1] For (α, δ) robustness alone, this point is not necessary due to the assumption of $|f(S) - \hat{f}(S)| \leq \epsilon$ for all $S \subseteq \Omega$. For the regret bound proof of our proposed C-ETC method in Appendix A.4, the "clean event" (corresponding to concentration of empirical mean of set values around their statistical means) will only imply concentration for those actions taken and thus for which empirical estimates exist.

Combining these two cases, we have

$$|\hat{f}(G_i \cup o_1) - f(G_i \cup o_1)| \leq \epsilon. \tag{36}$$

Also, for any evaluated action in iteration $i+1$, namely the actions $\{G_i \cup e | e \in \Omega \setminus G_i \text{ and } c(e) + c(G_i) \leq B\}$, we have

$$\begin{aligned}\rho(e|G_i) &= \frac{f(G_i \cup e) - f(G_i)}{c(e)} \\ &\leq \frac{\hat{f}(G_i \cup e) - \hat{f}(G_i)}{c(e)} + \frac{2\epsilon}{c(e)} \\ &= \hat{\rho}(e|G_i) + \frac{2\epsilon}{c(e)}. \end{aligned} \tag{37}$$

Then we have

$$\begin{aligned} f(\mathrm{OPT}) &\leq f(G_i \cup \mathrm{OPT}) && \text{(Monotonicity of } f) \\ &\leq f(G_i \cup o_1) + f(\mathrm{OPT} \setminus (G_i \cup o_1) | G_i \cup o_1) \\ &\leq f(G_i \cup o_1) + \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} f(e | G_i \cup o_1) && \text{(Submodularity of } f) \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} c(e) \rho(e | G_i \cup o_1). && (38) \end{aligned}$$

where (38) uses (36).

Since we picked iteration i such that $c(G_i) \leq B - c(o_1)$, then all items in $\mathrm{OPT} \setminus (G_i \cup o_1)$ still fit, as o_1 is the largest item in OPT. Since the greedy algorithm always selects the item with the largest marginal density with respect to the surrogate function $\hat{f}$, $g_i = \arg\max_{e \in \Omega \setminus G_i} \hat{\rho}(e|G_i)$, thus we have

$$\hat{\rho}(g_{i+1}|G_i) = \max_{e \in \Omega \setminus G_i} \hat{\rho}(e|G_i) \geq \max_{e \in \Omega \setminus (G_i \cup o_1)} \hat{\rho}(e|G_i). \tag{39}$$

Hence, continuing with (38),

$$\begin{aligned} f(\mathrm{OPT}) &\leq \hat{f}(G_i \cup o_1) + \epsilon + \left(\sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} c(e) \rho(e | G_i \cup o_1) \right) \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} c(e) \rho(e | G_i) && \text{(Submodularity)} \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} c(e) \left(\hat{\rho}(e | G_i) + \frac{2\epsilon}{c(e)} \right) && \text{(using (37))} \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} \left(c(e) \hat{\rho}(e | G_i) \right) + 2\epsilon |\mathrm{OPT} \setminus (G_i \cup o_1)| \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \hat{\rho}(g_{i+1}|G_i) \sum_{e \in \mathrm{OPT} \setminus (G_i \cup o_1)} \left(c(e) \right) + 2\epsilon |\mathrm{OPT} \setminus (G_i \cup o_1)| && \text{(Using (39))} \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \hat{\rho}(g_{i+1}|G_i) c(\mathrm{OPT} \setminus (G_i \cup o_1)) + 2\epsilon |\mathrm{OPT} \setminus (G_i \cup o_1)| \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \max\{0, \hat{\rho}(g_{i+1}|G_i)\} c(\mathrm{OPT} \setminus (G_i \cup o_1)) + 2\epsilon |\mathrm{OPT} \setminus (G_i \cup o_1)| \\ &\leq \hat{f}(G_i \cup o_1) + \epsilon + \max\{0, \hat{\rho}(g_{i+1}|G_i)\} (g_{i+1}|G_i)(B - c(o_1)) + 2\epsilon |\mathrm{OPT} \setminus (G_i \cup o_1)| \\ &\leq \hat{f}(G_i \cup o_1) + \max\{0, \hat{\rho}(g_{i+1}|G_i)\} (g_{i+1}|G_i)(B - c(o_1)) + (2\tilde{K} - 1)\epsilon. \end{aligned}$$

Rearranging terms gives the desired result.

□

Now we are ready to prove Proposition 6.4 (robustness of GREEDY+MAX algorithm). Applying Lemma 6.7 (GREEDY+MAX inequality) for $i = \ell$, and recalling that ℓ is chosen as the index of the last greedy set such that $c(G_\ell) \leq B - c(o_1) < c(G_{\ell+1})$,

$$\hat{f}(G_\ell \cup o_1) + \max\{0, \hat{\rho}(g_{\ell+1}|G_\ell)\}(B - c(o_1)) \geq f(\mathrm{OPT}) - (2\tilde{K} - 1)\epsilon. \tag{40}$$

From (40), we will next argue at least one of the terms in the left hand side must be large. We will consider cases for the two terms being large. To minimize the worst-case additive error term from the cases, we will split the cases into whether $\hat{f}(G_\ell \cup o_1)$ is larger than or equal to $\frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} + \gamma)\epsilon$, or $\max\{0, \hat{\rho}(g_{\ell+1}|G_\ell\}(B - c(o_1))$ is larger than or equal to $\frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon$, where γ will be selected later to minimize the additive error δ coefficient.

Case 1: If $\hat{f}(G_\ell \cup o_1) \geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} + \gamma)\epsilon$, recall that a_ℓ as the element selected to augment with the greedy solution G_ℓ, $a_\ell = \arg\max_{e \in \Omega \setminus G_\ell} \hat{f}(e|G_\ell)$, then

$$\begin{aligned}\hat{f}(G_\ell \cup a_\ell) &\geq \hat{f}(G_\ell \cup o_1) \\ &\geq \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} - \frac{1}{2} + \gamma\right)\epsilon.\end{aligned} \tag{41}$$

The set S that the algorithm selects in the end will be the set with the highest mean (based on surrogate function $\hat{f}$) among all those evaluated (both sets in the greedy process and their augmentations). Also, its observed value $\hat{f}(S_\ell)$ is at most ϵ above $f(S)$. Thus

$$\begin{aligned}f(S) &\geq \hat{f}(S) - \epsilon \\ &\geq \hat{f}(G_\ell \cup a_\ell) - \epsilon \\ &\geq \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} + \frac{1}{2} + \gamma\right)\epsilon. && \text{(using (41))}\end{aligned}$$

Case 2(a): If $\max\{0, \hat{\rho}(g_{\ell+1}|G_\ell)\}(B - c(o_1)) \geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon$ and $\hat{\rho}(g_{\ell+1}|G_\ell) > 0$, rearranging we have

$$\hat{\rho}(g_{\ell+1}|G_\ell) \geq \frac{f(\mathrm{OPT})}{2(B - c(o_1))} - \frac{(\tilde{K} - \frac{1}{2} - \gamma)\epsilon}{B - c(o_1)}. \tag{42}$$

Then,

$$\begin{aligned}\hat{f}(G_\ell) &= \hat{f}(G_\ell) - \hat{f}(G_{\ell-1}) + \hat{f}(G_{\ell-1}) + \cdots - \hat{f}(G_1) + \hat{f}(G_1) - \hat{f}(G_0) && \text{(telescoping sum; } G_0 = \emptyset,\ \hat{f}(G_0) := 0) \\ &= \sum_{j=1}^{l-1} \hat{f}(g_{j+1}|G_j) && \text{(Definition of } \hat{f}(\cdot|\cdot)) \\ &= \sum_{j=0}^{l-1} \hat{\rho}(g_{j+1}|G_j)c(g_{j+1}) && \text{(Definition of } \hat{\rho}(\cdot|\cdot)) \\ &\geq \sum_{j=0}^{l-1} \hat{\rho}(g_{\ell+1}|G_j)c(g_{j+1}) && \text{(greedy choice of } g_{j+1}) \\ &\geq \sum_{j=0}^{l-1} \left(\rho(g_{\ell+1}|G_j) - \frac{2\epsilon}{c(g_{\ell+1})}\right)c(g_{j+1}) \\ &\geq \sum_{j=0}^{l-1} \left(\rho(g_{\ell+1}|G_\ell) - \frac{2\epsilon}{c(g_{\ell+1})}\right)c(g_{j+1}) && \text{(submodularity of } f) \\ &= \left(\rho(g_{\ell+1}|G_\ell) - \frac{2\epsilon}{c(g_{\ell+1})}\right)c(G_\ell) && \text{(simplifying)} \\ &\geq \left(\hat{\rho}(g_{\ell+1}|G_\ell) - \frac{4\epsilon}{c(g_{\ell+1})}\right)c(G_\ell) \\ &\geq \hat{\rho}(g_{\ell+1}|G_\ell)c(G_\ell) - 4\beta\epsilon.\end{aligned} \tag{43}$$

Recalling that ℓ is chosen as the index of the last greedy set that has a remaining budget as big as the cost of the heaviest element in OPT, $c(G_\ell) \leq B - c(o_1) < c(G_{\ell+1})$,

$$\begin{aligned}
\hat{f}(G_{\ell+1}) &= \hat{f}(G_\ell \cup g_{\ell+1}) \\
&= \hat{f}(G_\ell) + c(g_{\ell+1})\hat{\rho}(g_{\ell+1}|G_\ell) \\
&\geq \left(\hat{\rho}(g_{\ell+1}|G_\ell)c(G_\ell) - 4\beta\epsilon\right) + c(g_{\ell+1})\hat{\rho}(g_{\ell+1}|G_\ell) && \text{(from (43))} \\
&= \hat{\rho}(g_{\ell+1}|G_\ell)c(G_{\ell+1}) - 4\beta\epsilon && \text{(simplifying)} \\
&\geq \frac{\frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon}{B - c(o_1)} c(G_{\ell+1}) - 4\beta\epsilon && \text{(case 2 condition)} \\
&\geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon - 4\beta\epsilon && (\ell \text{ chosen so that } c(G_{\ell+1}) > B - c(o_1)) \\
&= \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} - \frac{1}{2} - \gamma + 4\beta\right)\epsilon. && (44)
\end{aligned}$$

The set S that the algorithm selects at the end of the exploitation phase will be the set with the highest empirical mean among all those explored (both sets in the greedy process and augmented sets). Thus its empirical mean is at most ϵ above $f(S)$.

$$\begin{aligned}
f(S) &\geq \hat{f}(S) - \epsilon \\
&\geq \hat{f}(G_{\ell+1}) - \epsilon \\
&\geq \frac{1}{2}f(\mathrm{OPT}) - \left(\tilde{K} + \frac{1}{2} - \gamma + 4\beta\right)\epsilon. && \text{(using (44))}
\end{aligned}$$

Case 2(b): If $\max\{0, \hat{\rho}(g_{\ell+1}|G_\ell)\}(B - c(o_1)) \geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon$ and $\hat{\rho}(g_{\ell+1}|G_\ell) \leq 0$, then the set S that the algorithm selects at the end satisfies

$$\begin{aligned}
f(S) &\geq 0 \\
&\geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma)\epsilon && \text{(Case 2(b) condition)} \\
&\geq \frac{1}{2}f(\mathrm{OPT}) - (\tilde{K} - \frac{1}{2} - \gamma + 4\beta)\epsilon.
\end{aligned}$$

Thus, combining cases 1 and 2, and selecting $\gamma = 2\beta$, the additive $\frac{1}{2}$-approximation error we get by the modified Greedy+Max algorithm is at most $(\frac{1}{2} + \tilde{K} + 2\beta)\epsilon$, which concludes the proof.

C.4. Proof for Robustness of GREEDY+

In this section, we prove Proposition 6.5 in Section 6 of the main paper. The following statements, Lemmas C.1,C.2 and C.4, and their proofs are adapted from the proof of $\frac{1}{2}(1 - \frac{1}{e})$ approximation ratio in the offline setting (Khuller et al., 1999) using a value oracle. (Krause and Guestrin, 2005) adapted the proof of (Khuller et al., 1999) to an offline setting where the greedy process relies on an exact oracle to evaluate individual element values and to compare the best individual element to the set output by the greedy process, but use an inexact value oracle (within ϵ of the correct value) to evaluate marginal densities.

The main differences arise from (i) the algorithms of (Khuller et al., 1999; Krause and Guestrin, 2005) evaluate densities before checking for feasibility,[2] leading to different definitions of the augmented greedy sequence, necessitating us to use more care to show analogous properties, (ii) exact value oracles for best individual elements and for selecting OPT are used in (Khuller et al., 1999; Krause and Guestrin, 2005), simplifying work to conclude the final bound for the approximation ratio $\alpha = \frac{1}{2}(1 - \frac{1}{e})$ and leading to a different δ.

[2]As noted in Footnote 1, concentration of estimates (i.e. the surrogate $\hat{f}$) used by C-ETC in the bandit setting will only be for evaluated subsets, which by restriction will all be feasible.

Recall that Proposition 6.5 in Section 6 of the main paper states that GREEDY+ is a $(\frac{1}{2}(1-\frac{1}{e}), 2+\tilde{K}+\beta)$-robust approximation algorithm for submodular maximization problem under a knapsack constraint.

We define G_i and g_i the same as previous section. Recall that the greedy process (using a surrogate $\hat{f}$) produces a nested sequence of subsets $\emptyset = G_0 \subset G_1 \subset \cdots \subset G_L$, where L denotes the cardinality of the set final output of the greedy process. For the proof, we describe the greedy process as running for $L+1$ iterations, though on the final iteration no elements are added.

For any action $G_{i-1} \cup a$ evaluated in iteration i of the greedy process, its marginal gains are upper bounded by that of the best subset based on surrogate function $\hat{f}$,

$$\begin{aligned}
\frac{f(G_{i-1}\cup a) - f(G_{i-1}) - 2\epsilon}{c(a)} &\leq \frac{\hat{f}(G_{i-1}\cup a) - \hat{f}(G_{i-1})}{c(a)} \\
&\leq \frac{\hat{f}(G_{i-1}\cup g_i) - \hat{f}(G_{i-1})}{c(g_i)} && \text{(g_i selected by greedy rule based on $\hat{f}$)} \\
&\leq \frac{f(G_{i-1}\cup g_i) - f(G_{i-1}) + 2\epsilon}{c(g_i)} \\
&= \frac{f(G_i) - f(G_{i-1}) + 2\epsilon}{c(g_i)}, && (45)
\end{aligned}$$

where (45) just uses the definition of $G_i \leftarrow G_{i-1} \cup g_i$. We will use (45) to lower bound the true marginal gains (i.e. in terms of f) achieved for each iteration of the greedy process.

Let $\ell \in \{1, \ldots, L+1\}$ denote the first iteration for which there was an element $a' \in \Omega \backslash G_{\ell-1}$ whose cost exceeds the remaining budget ($c(a') + c(G_{\ell-1}) > B$) (thus subset $G_{\ell-1} \cup a'$ was not sampled), yet whose marginal density was higher than the marginal density of the chosen element g_ℓ up to $\pm 2\epsilon$ normalized by the cost, specifically, for $\ell \leq L$,

$$\frac{f(G_{\ell-1}\cup a') - f(G_{\ell-1}) - 2\epsilon}{c(a')} > \frac{f(G_{\ell-1}\cup a_\ell) - f(G_{\ell-1}) + 2\epsilon}{c(a_r)}. \tag{46}$$

If there is no such iteration $\ell < L+1$, then for $\ell = L+1$, we take the element a' maximizing the term on the left hand side of (46),

$$a' = \underset{a \in \Omega \backslash G_{\ell-1}}{\arg\max} \frac{f(G_{\ell-1}\cup a) - f(G_{\ell-1}) - 2\epsilon}{c(a)}. \tag{47}$$

Likewise, if there is more than one element satisfying (46) for some (earliest) iteration r, then we also take the maximizer (47).

We define an "augmented" greedy sequence of length ℓ which matches the greedy sequence up to the set of cardinality ℓ, where the element a' is selected despite violating the budget,

$$\{\widetilde{G}_0 = G_0 = \emptyset, \widetilde{G}_1 = G_1, \ldots, \widetilde{G}_{\ell-1} = G_{\ell-1}, \widetilde{G}_\ell = G_{\ell-1} \cup \{a'\}\} \tag{48}$$

and correspondingly enumerate the elements of $\widetilde{G}_\ell$ in the order they were selected,

$$\{\widetilde{g}_1 = g_1, \ldots, \widetilde{g}_{\ell-1} = g_{\ell-1}, \widetilde{g}_\ell = g'\}. \tag{49}$$

We first prove the following lemma, bounding the marginal gains of the augmented greedy sequence $\{\widetilde{G}_0, \ldots, \widetilde{G}_\ell\}$.

Lemma C.1. *For all $i \in \{1, 2, \cdots, \ell\}$, the following inequality holds:*

$$f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) \geq \frac{c(\widetilde{g}_i)}{B}\left[f(\mathrm{OPT}) - f(\widetilde{G}_{i-1})\right] - 2\left(1 + \frac{\tilde{K}c(\widetilde{g}_i)}{B}\right)\epsilon.$$

Proof. Set any $i \in \{1, 2, \cdots, \ell\}$. Let $\{v_1, v_2, \cdots, v_k\} = \mathrm{OPT} \setminus \widetilde{G}_{i-1}$. Note that by construction (48), we have $\widetilde{G}_{i-1} = G_{i-1}$.

The difference $f(\mathrm{OPT}) - f(\widetilde{G}_{i-1})$ can be bounded by the marginal gains of elements in the set difference,

$$\begin{aligned}
f(\mathrm{OPT}) - f(\widetilde{G}_{i-1}) &\leq \sum_{j=1}^{k} \left[f(\widetilde{G}_{i-1} \cup v_j) - f(\widetilde{G}_{i-1}) \right] && \text{(Fact 1)} \\
&= \sum_{j=1}^{k} \left[f(\widetilde{G}_{i-1} \cup v_j) - f(\widetilde{G}_{i-1}) - 2\epsilon + 2\epsilon \right] \\
&= \sum_{j=1}^{k} c(v_j) \frac{f(\widetilde{G}_{i-1} \cup v_j) - f(\widetilde{G}_{i-1}) - 2\epsilon}{c(v_j)} + 2k\epsilon \\
&\leq \sum_{j=1}^{k} c(v_j) \frac{f(\widetilde{G}_{i-1} \cup \widetilde{g}_i) - f(\widetilde{G}_{i-1}) + 2\epsilon}{c(\widetilde{g}_i)} + 2k\epsilon && (50) \\
&= \sum_{j=1}^{k} c(v_j) \frac{f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) + 2\epsilon}{c(\widetilde{g}_i)} + 2k\epsilon && (51)
\end{aligned}$$

where (50) holds by following. We consider four cases, depending on whether or not $\hat{f}(G_{i-1} \cup v_j)$ was evaluated during the iteration i.

- **Case 1 ($\hat{f}(G_{i-1} \cup v_j)$ was evaluated and $i < \ell$):** At iteration i (necessarily $i \leq L$ since no subsets were evaluated in iteration $L+1$) with current greedy set G_{i-1}, adding the element v_j to the current greedy set was feasible, $c(v_j) \leq B - c(G_{i-1})$. Then GREEDY+ would have evaluated $\hat{f}(G_{i-1} \cup v_j)$. Since v_j was not selected, the chosen element $g_i = G_i \backslash G_{i-1}$ must have had a higher surrogate density $\hat{f}(G_{i-1} \cup v_j) > \hat{f}(G_{i-1} \cup g_i)$, so for $i < \ell$, for which $\widetilde{g}_i = g_i$ by construction (49), (45) implies (50).
- **Case 2 ($\hat{f}(G_{i-1} \cup v_j)$ was evaluated and $i = \ell$):** By the reasoning in the previous case, for the item a_ℓ chosen at iteration ℓ by the greedy process (due to feasibility and having the highest surrogate density), we still have the bound (45) on true values, which coupled with our specific construction of $\widetilde{g}_\ell$ (46) means

$$\begin{aligned}
\frac{f(\widetilde{G}_{\ell-1} \cup v_j) - f(\widetilde{G}_{\ell-1}) - 2\epsilon}{c(v_j)} &\leq \frac{f(\widetilde{G}_{\ell-1} \cup a_r) - f(\widetilde{G}_{\ell-1}) + 2\epsilon}{c(a_r)} && \text{(by (45))} \\
&< \frac{f(\widetilde{G}_{\ell-1} \cup \widetilde{g}_r) - f(\widetilde{G}_{\ell-1}) - 2\epsilon}{c(\widetilde{g}_r)} && \text{(by construction (46))} \\
&< \frac{f(\widetilde{G}_{\ell-1} \cup \widetilde{g}_r) - f(\widetilde{G}_{\ell-1}) + 2\epsilon}{c(\widetilde{g}_r)}.
\end{aligned}$$

- **Case 3 ($\hat{f}(G_{i-1} \cup v_j)$ was not evaluated and $i < \ell$):** At iteration $i < \ell \leq L+1$ with the current greedy set G_{i-1}, adding the element v_j to the current greedy set was not feasible, $c(v_j) > B - c(G_{i-1})$. By construction of the augmented greedy sequence, only at iteration ℓ was there an infeasible element whose surrogate marginal density satisfied the inequality (46). Thus, for iterations $i < \ell$, $G_{i-1} = \widetilde{G}_{i-1}$ and $G_i = \widetilde{G}_i$, so (50) holds.
- **Case 4 ($\hat{f}(G_{i-1} \cup v_j)$ was not evaluated and $i = \ell$):** For iteration $i = \ell$, with current greedy set G_{i-1}, the augmented greedy sequence construction implies (50). Namely, with $i = \ell$,

$$\begin{aligned}
\frac{f(\widetilde{G}_{\ell-1} \cup v_j) - f(\widetilde{G}_{\ell-1}) - 2\epsilon}{c(v_j)} &< \frac{f(\widetilde{G}_{\ell-1} \cup \widetilde{g}_r) - f(\widetilde{G}_{\ell-1}) - 2\epsilon}{c(\widetilde{g}_r)} && \text{(by (47))} \\
&< \frac{f(\widetilde{G}_{\ell-1} \cup \widetilde{g}_r) - f(\widetilde{G}_{\ell-1}) + 2\epsilon}{c(\widetilde{g}_r)}.
\end{aligned}$$

menaing (50) holds.

We now continue lower bounding $f(\mathrm{OPT}) - f(\widetilde{G}_{i-1})$,

$$
\begin{aligned}
f(\mathrm{OPT}) - f(\widetilde{G}_{i-1}) &\leq \left[\sum_{j=1}^{k} c(v_j) \frac{f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) + 2\epsilon}{c(\widetilde{g}_i)}\right] + 2k\epsilon && \text{(copying (51))} \\
&= \left[\sum_{j=1}^{k} c(v_j)\right] \frac{f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) + 2\epsilon}{c(\widetilde{g}_i)} + 2k\epsilon \\
&\leq B \frac{f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) + 2\epsilon}{c(\widetilde{g}_i)} + 2k\epsilon && \text{(OPT is feasible, so } \textstyle\sum_{j=1}^{k} c(v_j) \leq B) \\
&\leq \frac{B}{c(\widetilde{g}_i)} \left[f(\widetilde{G}_i) - f(\widetilde{G}_{i-1})\right] + 2\left[\frac{B}{c(\widetilde{g}_i)} + \tilde{K}\right]\epsilon. && \text{(rearranging; } k \leq \tilde{K})
\end{aligned}
$$

Multiplying both sides by $\frac{c(\widetilde{g}_i)}{B}$ and rearranging finishes the proof. ☐

We unravel the recurrence in Lemma C.1 to lower bound $f(\widetilde{G}_i)$.

Lemma C.2. *For all $i \in \{1, 2, \cdots, \ell\}$,*

$$
f(\widetilde{G}_i) \geq \left[1 - \prod_{j=1}^{i}(1 - \frac{c(\widetilde{g}_j)}{B})\right] f(\mathrm{OPT}) - 2(\beta + \tilde{K})\epsilon.
$$

Remark C.3. The steps to unravel the recurrence to obtain the first term (coefficient of $f(\mathrm{OPT})$) is the same as the proof for the analogous result in the offline setting (Khuller et al., 1999). The second term (with ϵ) is due to working with marginal densities of a surrogate function $\hat{f}$. The basic steps for working with that second term is the same as (Krause and Guestrin, 2005), though we use a looser bound β; in (Krause and Guestrin, 2005) we think there may be a mistake in applying the induction step (with "$c(X_i)$" fixed for different i in the proof), though they were loosely bounded with β later on.

Proof. The proof will follow by induction. We first show the base case $i = 1$ using Lemma C.1.

$$
\begin{aligned}
f(\widetilde{G}_1) &= f(\widetilde{G}_1) - f(\widetilde{G}_0) && (f \text{ is normalized; } \widetilde{G}_0 = \emptyset) \\
&\geq \frac{c(\widetilde{g}_1)}{B}\left[f(\mathrm{OPT}) - f(\widetilde{G}_0)\right] - 2\left(1 + \frac{\tilde{K} c(\widetilde{g}_1)}{B}\right)\epsilon && \text{(using Lemma C.1)} \\
&= \left[1 - \left(1 - \frac{c(\widetilde{g}_1)}{B}\right)\right] f(\mathrm{OPT}) - 2\left(1 + \frac{\tilde{K} c(\widetilde{g}_1)}{B}\right)\epsilon && (52)
\end{aligned}
$$

where (52) follows from rearranging. For the second term in (52), using that

$$
\begin{aligned}
1 + \frac{\tilde{K} c(\widetilde{g}_1)}{B} &\leq \frac{B}{c(\widetilde{g}_1)}\left(1 + \frac{\tilde{K} c(\widetilde{g}_1)}{B}\right) && \text{(since } \tfrac{B}{c(\widetilde{g}_1)} \geq 1) \\
&= \frac{B}{c(\widetilde{g}_1)} + \tilde{K} \\
&\leq \frac{B}{c_{\min}} + \tilde{K} \\
&= \beta + \tilde{K}, && (53)
\end{aligned}
$$

then

$$
\begin{aligned}
f(\widetilde{G}_1) &\geq \left[1-\left(1-\frac{c(\widetilde{g}_1)}{B}\right)\right] f(\mathrm{OPT}) - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_1)}{B}\right)\epsilon && \text{(copying (52))}\\
&\geq \left[1-\left(1-\frac{c(\widetilde{g}_1)}{B}\right)\right] f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon. && \text{(using (53))}
\end{aligned}
$$

This completes the base case of $i = 1$.

We next consider $i > 1$. Unraveling the recurrence shown in Lemma C.1,

$$
\begin{aligned}
f(\widetilde{G}_i) &= f(\widetilde{G}_i) - f(\widetilde{G}_{i-1}) + f(\widetilde{G}_{i-1})\\
&\geq \left[\frac{c(\widetilde{g}_i)}{B}\left(f(\mathrm{OPT}) - f(\widetilde{G}_{i-1})\right) - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_i)}{B}\right)\epsilon\right] + f(\widetilde{G}_{i-1}) && \text{(using Lemma C.1)}\\
&= \left[\frac{c(\widetilde{g}_i)}{B}\right] f(\mathrm{OPT}) - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_i)}{B}\right)\epsilon + \left[1-\frac{c(\widetilde{g}_i)}{B}\right] f(\widetilde{G}_{i-1}) && \text{(rearranging)}\\
&= \left[1-(1-\frac{c(\widetilde{g}_i)}{B})\right] f(\mathrm{OPT}) - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_i)}{B}\right)\epsilon\\
&\quad + \left[1-\frac{c(\widetilde{g}_i)}{B}\right] f(\widetilde{G}_{i-1}) && \text{(rearranging)}\\
&\geq \left[1-(1-\frac{c(\widetilde{g}_i)}{B})\right] f(\mathrm{OPT}) - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_i)}{B}\right)\epsilon\\
&\quad + \left(1-\frac{c(\widetilde{g}_i)}{B}\right)\left[\left(1-\prod_{j=1}^{i-1}(1-\frac{c(\widetilde{g}_j)}{B})\right) f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon\right] && \text{(induction step)}\\
&= \left[1-(1-\frac{c(\widetilde{g}_i)}{B}) + \left(1-\frac{c(\widetilde{g}_i)}{B}\right)\left(1-\prod_{j=1}^{i-1}(1-\frac{c(\widetilde{g}_j)}{B})\right)\right] f(\mathrm{OPT})\\
&\quad - 2\left(1+\frac{\tilde{K}c(\widetilde{g}_i)}{B} + \left(1-\frac{c(\widetilde{g}_i)}{B}\right)(\beta+\tilde{K})\right)\epsilon && \text{(rearranging)}\\
&= \left[1-\prod_{j=1}^{i}(1-\frac{c(\widetilde{g}_j)}{B})\right] f(\mathrm{OPT})\\
&\quad - 2\left(1+\beta-\beta\frac{c(\widetilde{g}_i)}{B}+\tilde{K}\right)\epsilon. && (54)
\end{aligned}
$$

For the second term in (54), using that

$$
\begin{aligned}
\beta\frac{c(\widetilde{g}_i)}{B} &= \frac{B}{c_{\min}}\frac{c(\widetilde{g}_i)}{B} && \text{(def. of } \beta)\\
&= \frac{c(\widetilde{g}_i)}{c_{\min}}\\
&\geq 1, && (55)
\end{aligned}
$$

then

$$
\begin{aligned}
-2\left(1+\beta-\beta\frac{c(\widetilde{g}_i)}{B}+\tilde{K}\right)\epsilon &= -2\left(\beta+\tilde{K}\right)\epsilon + 2\left(\beta\frac{c(\widetilde{g}_i)}{B}-1\right)\epsilon && \text{(rearranging)}\\
&\geq -2\left(\beta+\tilde{K}\right)\epsilon. && \text{(using (55))}
\end{aligned}
$$

Applying this to (54) completes the proof. □

The inequality in Lemma C.2 for the augmented greedy set of cardinality ℓ can be further simplified. We will use the following observations.

Lemma C.4. *The following inequality holds:*

$$f(\widetilde{G}_\ell) \geq (1-\frac{1}{e})f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon.$$

Proof. Applying $i=\ell$ to Lemma C.2 and bounding the coefficient for $f(\mathrm{OPT})$,

$$\begin{aligned}
f(\widetilde{G}_\ell) &\geq \left[1-\prod_{j=1}^{\ell}(1-\frac{c(\widetilde{g}_j)}{B})\right] f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon \\
&\geq \left[1-\prod_{j=1}^{\ell}(1-\frac{c(\widetilde{g}_j)}{c(\widetilde{G}_\ell)})\right] f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon && \text{(by construction, } c(\widetilde{G}_\ell) > B) \\
&\geq \left[1-\prod_{j=1}^{\ell}(1-\frac{c(\widetilde{G}_\ell)/\ell}{c(\widetilde{G}_\ell)})\right] f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon && \text{(using Fact 2)} \\
&= \left[1-(1-\frac{1}{\ell})^\ell\right] f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon && \text{(simplifying)} \\
&\geq \left(1-\frac{1}{e}\right) f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon. && \text{(using Fact 3)}
\end{aligned}$$

□

Using the aforementioned lemmas, we are now ready to complete the proof for Theorem 3 (robustness of GREEDY+ algorithm). We will bound the value of set G_L using the results on the augmented greedy set (48) of cardinality ℓ, and in turn bound the value of the set S, the final output of GREEDY+.

Recall that GREEDY+ chooses the set S to be either the best individual element (based on $\hat{f}$) $a^* \leftarrow \arg\max_{e\in\Omega} \hat{f}(e)$ or the output of the greedy process G_L. Let $a^{\mathrm{OPT}} = \arg\max_{e\in\Omega} f(e)$ denote the element with the highest value under f. Then

$$\begin{aligned}
f(a^*) &\geq \hat{f}(a^*) - \epsilon \\
&\geq \hat{f}(a^{\mathrm{OPT}}) - \epsilon && \text{(by definition of } a^*) \\
&\geq f(a^{\mathrm{OPT}}) - 2\epsilon. && (56)
\end{aligned}$$

By construction (48), $\widetilde{G}_\ell$ includes one more element a' than $\widetilde{G}_{\ell-1}$ (and a' maximizes (47)). By submodularity, the marginal gain of a' is bounded by $f(a')$ and in turn by the best individual element based on surrogate function $\hat{f}$,

$$\begin{aligned}
f(\widetilde{G}_{\ell-1}) + f(a^{\mathrm{OPT}}) &\geq f(\widetilde{G}_{\ell-1}) + f(a') && \text{(by definition of } a^{\mathrm{OPT}}) \\
&\geq f(\widetilde{G}_{\ell-1}) + \left[f(\widetilde{G}_{\ell-1}\cup a') - f(\widetilde{G}_{\ell-1})\right] && \text{(by submodularity)} \\
&= f(\widetilde{G}_{\ell-1}\cup a') \\
&= f(\widetilde{G}_\ell) && \text{(by construction (48))} \\
&\geq (1-\frac{1}{e})f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon, && (57)
\end{aligned}$$

where (57) follows from Lemma C.4.

Also by construction (48), the greedy and augmented greedy processes match up to and including the set of cardinality $\ell-1$, so

$$\begin{aligned}
f(G_L) &\geq f(G_{\ell-1}) && \text{(monotonicity)} \\
&= f(\widetilde{G}_{\ell-1}). && \text{(By construction (48))}
\end{aligned}$$

Thus,

$$\begin{aligned} f(G_L) + f(a^{\mathrm{OPT}}) &\geq f(\widetilde{G}_{\ell-1}) + f(a^{\mathrm{OPT}}) \\ &\geq (1 - \frac{1}{e}) f(\mathrm{OPT}) - 2(\beta + \tilde{K})\epsilon. && \text{(using (57))} \end{aligned}$$

At least one of $f(G_L)$ and $f(a^{\mathrm{OPT}})$ is at least half of the value of the right hand side,

$$\max\{f(G_L), f(a^{\mathrm{OPT}})\} \geq \frac{1}{2}(1 - \frac{1}{e}) f(\mathrm{OPT}) - (\beta + \tilde{K})\epsilon \tag{58}$$

Thus, for the chosen set S

$$\begin{aligned} f(S) &\geq \hat{f}(S) - \epsilon \\ &= \max\{\hat{f}(G_L), \hat{f}(a^*)\} - \epsilon \\ &\geq \max\{\hat{f}(G_L), \hat{f}(a^{\mathrm{OPT}})\} - \epsilon && \text{(a^* is the element with largest $\hat{f}$ value)} \\ &\geq \max\{f(G_L) - \epsilon, f(a^{\mathrm{OPT}}) - \epsilon\} - \epsilon && \text{(element-wise dominance)} \\ &= \max\{f(G_L), f(a^{\mathrm{OPT}})\} - 2\epsilon \\ &\geq \frac{1}{2}(1 - \frac{1}{e}) f(\mathrm{OPT}) - (\beta + \tilde{K})\epsilon - 2\epsilon && \text{(from (58))} \\ &= \frac{1}{2}(1 - \frac{1}{e}) f(\mathrm{OPT}) - (2 + \beta + \tilde{K})\epsilon. \end{aligned}$$

which completes the proof.

C.5. Proof for Robustness of PARTIALENUMERATION

Now we analyze the PARTIALENUMERATION algorithm for submodular maximization under a knapsack constraint proposed in Sviridenko (2004); Khuller et al. (1999). Recall that Proposition 6.3 in Section 6 of the main paper states PARTIALENUMERATION is a $(1 - \frac{1}{e}, 4 + 2\tilde{K} + 2\beta)$-robust approximation algorithm for submodular maximization under a knapsack constraint.

Proof. Assume $|\mathrm{OPT}| > 3$, otherwise the algorithm finds a $(1, 2)$-robust approximation, so it is also a $(1 - \frac{1}{e}, 2(\tilde{K} + \beta))$-robust approximation for non-trivial cases where $\tilde{K} \geq 1$ and $\beta \geq 1$. Enumerate the elements of the optimal solution as $\mathrm{OPT} = \{Y_1, \cdots, Y_m\}$, corresponding to the order they would be selected by the simple greedy algorithm (iteratively selecting the element with the largest marginal gain, not the largest marginal density)

$$Y_{i+1} = \underset{Y \in \mathrm{OPT}}{\arg\max} f(\{Y_1, \cdots, Y_i, Y\}) - f(\{Y_1, \cdots, Y_i\}), \tag{59}$$

and let $R = \{Y_1, Y_2, Y_3\}$. Consider the iteration where the algorithm considers R. Define the function

$$f'(A) = f(A \cup R) - f(R). \tag{60}$$

f' is a non-decreasing submodular set function with $f'(\emptyset) = 0$, and the optimal solution (with budget $B - c(R)$) is $\mathrm{OPT} \setminus R$ since for any set S with cost $c(S) \leq B - c(R)$,

$$\begin{aligned} f'(\mathrm{OPT} \setminus R) &= f(\mathrm{OPT} \cup R) - f(R) && \text{(def of f')} \\ &= f(\mathrm{OPT}) - f(R) && \text{($R \subseteq \mathrm{OPT}$ by construction)} \\ &\geq f(S \cup R) - f(R) \\ &= f'(S). \end{aligned}$$

Hence we can apply GREEDY+ algorithm to f' (based on noisy evaluations). Let g_ℓ be the first element from $\mathrm{OPT} \setminus R$ which could not be added due to budget constraints, and let $A = \{g_1, \cdots, g_{\ell-1}\}$ be first $\ell - 1$ elements selected by GREEDY+ algorithm. Let $G = A \cup R$. Using Lemma C.4, we get

$$f'(A \cup g_\ell) \geq (1 - \frac{1}{e}) f'(\mathrm{OPT} \setminus R) - 2(\beta' + \tilde{K}')\epsilon,$$

where $\beta' = \frac{B-c(R)}{c'_{\min}}$, $\tilde{K}' = \min\{n-3, \beta'\}$ and $c'_{\min} = \min_{e\in\Omega\setminus R} c(e)$. Simple calculation can show that $\beta' \leq \beta$ and $\tilde{K}' \leq \tilde{K}$. Thus,

$$f'(A \cup g_\ell) \geq (1-\frac{1}{e})f'(\mathrm{OPT}\setminus R) - 2(\beta+\tilde{K})\epsilon,$$

From the definition of f', we have $f(G) = f'(A) + f(R)$. Let $\Delta = f'(A\cup g_\ell) - f'(A)$. We have

$$f'(A) + \Delta \geq (1-\frac{1}{e})f'(\mathrm{OPT}\setminus R) - 2(\beta+\tilde{K})\epsilon. \tag{61}$$

Further observe that elements in OPT are ordered that for all $1 \leq i \leq 3$,

$$\begin{aligned}
&f(\{Y_1,\cdots,Y_i\}) - f(\{Y_1,\cdots,Y_{i-1}\}) \\
&\quad\geq f(\{Y_1,\cdots,Y_{i-1},g_\ell\}) - f(\{Y_1,\cdots,Y_{i-1}\}) && \text{(ordering rule)} \\
&\quad\geq f(R\cup A\cup g_\ell) - f(R\cup A) && (\{Y_1,\cdots,Y_{i-1}\}\subseteq R \text{ when } 1\leq i\leq 3 \text{ and submodularity}) \\
&\quad= f(R\cup A\cup g_\ell) - f(R) - (f(R\cup A) - f(R)) \\
&\quad= f'(A\cup g_\ell) - f'(A) \\
&\quad= \Delta.
\end{aligned}$$

By telescoping sum, $f(R) \geq 3\Delta$. Now we get

$$\begin{aligned}
f(G) &= f(R) + f'(A) \\
&\geq f(R) + (1-\frac{1}{e})f'(\mathrm{OPT}\setminus R) - 2(\beta+\tilde{K})\epsilon - \Delta \\
&\geq f(R) + (1-\frac{1}{e})f'(\mathrm{OPT}\setminus R) - 2(\beta+\tilde{K})\epsilon - f(R)/3 \\
&\geq (1-\frac{1}{3})f(R) + (1-\frac{1}{e})f'(\mathrm{OPT}\setminus R) - 2(\beta+\tilde{K})\epsilon \\
&\geq (1-\frac{1}{e})\left[f'(\mathrm{OPT}\setminus R) + f(R)\right] - 2(\beta+\tilde{K})\epsilon && (e\leq 3) \\
&= (1-\frac{1}{e})f(\mathrm{OPT}) - 2(\beta+\tilde{K})\epsilon. && \text{(definition of } f')
\end{aligned}$$

The output of the algorithm is not necessarily G because the values of the evaluated triplets are based on surrogate function $\hat{f}$. Denote $\mathcal{O}$ as the output of the algorithm and denote G' as the best evaluated set (with respect to $\hat{f}$) with size $\ell+2$ (same as G). We must have that $\hat{f}(G') \geq \hat{f}(G)$. Also denote the final set (until violating budget) continuing G' as G''. We have,

$$\begin{aligned}
f(\mathcal{O}) &\geq \hat{f}(\mathcal{O}) - \epsilon \\
&\geq \hat{f}(G'') - \epsilon && \text{(selection rule of the algorithm)} \\
&\geq f(G'') - 2\epsilon \\
&\geq f(G') - 2\epsilon && (G'\subseteq G'' \text{ and monotonicity of } f) \\
&\geq \hat{f}(G') - 3\epsilon \\
&\geq \hat{f}(G) - 3\epsilon \\
&\geq f(G) - 4\epsilon \\
&\geq (1-\frac{1}{e})f(\mathrm{OPT}) - (4+2\beta+2\tilde{K})\epsilon,
\end{aligned}$$

finishing the proof. □

D. Implementation of Algorithm OG°

In this section we describe implementation details and parameter selection for OG° algorithm (Streeter and Golovin, 2008). The choice of exploration probability is given by the original paper:$\gamma = n^{1/3}\beta\left(\frac{\log(n)}{T}\right)^{1/3}$, where $\beta = B/c_{\min}$. Note that

Algorithm 2 Online Greedy for Opaque Feedback Model (OG$^\text{o}$)

Input: set of base arms Ω, horizon T, cost for each arm $c(a)$, budget B
Initialize $n \leftarrow |\Omega|$, $c_{\min} \leftarrow \min_{a\in\Omega}\{c(a)\}$, $\beta \leftarrow \frac{B}{c_{\min}}$, $\gamma \leftarrow n^{1/3}\beta\left(\frac{\log(n)}{T}\right)^{1/3}$, $\epsilon \leftarrow \sqrt{\frac{\beta\log(n)}{\gamma T}}$
Initialize $\boldsymbol{\omega}_1 \leftarrow \text{ones}(\beta, n)$
for $t \in [1, \cdots, T]$ **do**
  $S_t \leftarrow \emptyset$
  $l \leftarrow \text{zeros}(\beta, n)$ *// loss*
  Randomly sample a value $\xi \sim \text{Uniform}([0,1])$
  if $\xi \le \gamma$ **then**
    $e \sim \text{Uniform}(\{1, \cdots, \beta\})$
    for $i \in [1, \cdots, e-1]$ **do**
      // For experts before e, exploit
      Select an arm a with probability $\frac{\boldsymbol{\omega}_t[i,a]}{\sum \boldsymbol{\omega}_t[i,:]}$, re-sample if $a \in S_t$
      $S_t \leftarrow S_t \cup \{a\}$ with probability $\frac{c_{\min}}{c(a)}$; $S_t \leftarrow S_{t-1}$ otherwise
    end for
    $a \sim \text{Uniform}(\{1, \cdots, n\}\backslash S_t)$ *// For expert e, explore*
    $S_t \leftarrow S_t \cup \{a\}$
    Play action S_t, observe $f_t(S_t)$
    Update $l[i,j] \leftarrow \frac{c_{\min} f_t(S_t)}{c(a)}$ for all $i = e$ and $j \neq a$ *// Feed $\frac{c_{\min} f_t(S_t)}{c(a)}$ back to expert e associated with action a*
    Update $\boldsymbol{\omega}_{t+1}[i,j] \leftarrow \boldsymbol{\omega}_t[i,j]\exp(-\epsilon l[i,j])$ for all pairs of i and j
  else
    // Exploitation with probability $1-\gamma$
    for $i \in [1, \cdots, \beta]$ **do**
      // For experts before e, exploit
      Select arm a with probability $\frac{\boldsymbol{\omega}_t[i,a]}{\sum \boldsymbol{\omega}_t[i,:]}$, re-sample if $a \in S_t$
      $S_t \leftarrow S_t \cup \{a\}$ with probability $\frac{c_{\min}}{c(a)}$; $S_t \leftarrow S_{t-1}$ otherwise
    end for
    Play action S_t, observe $f_t(S_t)$
    $\boldsymbol{\omega}_{t+1}[i,j] \leftarrow \boldsymbol{\omega}_t[i,j]$ *// Since feeding back 0 to all expert-action payoffs, loss is 0, no update*
  end if
end for

in the original paper, B is used instead of β, because they assume the minimum cost is 1. Here we generalize it to arbitrary non-negative costs. ϵ is the learning rate for Randomized Weighted Majority (WMR) expert algorithm (Arora et al., 2012). It is chosen by setting the derivative of regret upper bound to zero, which is $\epsilon = \sqrt{\frac{\log(n)}{T_e}}$, where T_e is the time spent on updating expert e. Since it explores with probability γ, and there are β expert algorithms, we have $T_e \approx \frac{\gamma T}{\beta}$. Thus we pick $\epsilon = \sqrt{\frac{\beta\log(n)}{\gamma T}}$. In experiments, there are many cases the chosen γ is large or even larger than 1, so we cap the probability of exploring γ by 1/2 to avoid exploring too much. Note that unlike a hard budget in our setting, for OG$^\text{o}$, it only requires the budget to be satisfied in expectation, so in general we might choose sets over budget. Algorithm 2 is the pseudo code for implementation details of OG$^\text{o}$.

E. Comments on Lower bounds of Submodular CMAB

For the setting we explore in this paper, with stochastic (or even adversarial) knapsack-constrained combinatorial MAB with submodular expected rewards and just bandit feedback, it remains an open question if $\tilde{\mathcal{O}}(T^{1/2})$ expected cumulative α-regret is possible (ignoring n and β). Both (Streeter and Golovin, 2008) and (Niazadeh et al., 2021) analyze lower bounds for the adversarial setting. However, (Streeter and Golovin, 2008) obtain bounds for 1-regret (it is NP-hard in offline setting to obtain an approximation ratio better than $1-1/e$). (Niazadeh et al., 2021) obtain $\tilde{\Omega}(T^{2/3})$ lower bounds for the harder setting where feedback is only available during "exploration" rounds chosen by the agent, who incurs an associated penalty.

F. Dealing with Small Time Horizons in Experiments

In Section 6, we used $N = \tilde{K}n$ as an upper bound on the number of function evaluations for both C-ETC-K and C-ETC-Y, where n is the number of base arms and $\tilde{K}$ is an upper bound of the cardinality of any feasible sets. When the time horizon T is small, it is possible that the exploration phase will not finish due to the formula being optimized for m (the number of plays for each action queried by $\mathcal{A}$) uses a loose bound on the exploitation time. When this is the case, we select the largest m (closest to the formula) for which we can guarantee that exploration will finish. Recall that for C-ETC-Y and C-ETC-K, the number of oracle calls can only be upper bounded in advance.

We first calculate $m^\dagger$ using (26):

$$m^\dagger = \left\lceil \frac{\delta^{2/3} T^{2/3} \log(T)^{1/3}}{2\tilde{K}^{2/3} n^{2/3}} \right\rceil.$$

Note that a (slightly tighter) upper bound on the number of subsets evaluated during the exploration phase (with $\tilde{K}$ bounding the number of iterations of the greedy process) is

$$\begin{aligned} N &\leq n + (n-1) + \cdots + (n - \tilde{K} + 1) \\ &= \left(n - \frac{\tilde{K}}{2} + \frac{1}{2}\right)\tilde{K}. \end{aligned}$$

We compare $\left(n - \frac{\tilde{K}}{2} + \frac{1}{2}\right)\tilde{K}m^\dagger$ with T.

- **Case 1.** If $\left(n - \frac{\tilde{K}}{2} + \frac{1}{2}\right)\tilde{K}m^\dagger < T$, C-ETC can finish exploring. We select $m = m^\dagger$.
- **Case 2.** If $\left(n - \frac{\tilde{K}}{2} + \frac{1}{2}\right)\tilde{K}m^\dagger \geq T$, it is possible that the algorithm cannot finish exploring. In this case, we will find a new m, so that the exploration can be guaranteed to finish. We select the largest m (closest to $m^\dagger$) so that the exploration time is upper bounded by T,

$$m = \frac{T}{\left(n - \frac{\tilde{K}}{2} + \frac{1}{2}\right)\tilde{K}}.$$

G. Basic Facts

Fact 1. For a monotonically non-decreasing submodular set function f defined over subsets of Ω, we have for arbitrary subsets $A, B \subseteq \Omega$,

$$f(B) - f(A) \leq \sum_{j \in B \setminus A} \left[f(A \cup \{j\}) - f(A)\right].$$

Fact 2. (Khuller et al., 1999) For $x_1, \cdots, x_n \in \mathbb{R}^+$ such that $\sum x_i = A$, the function $[1 - \prod_{i=1}^n (1 - \frac{x_i}{A})]$ achieves its minimum at $x_1 = x_2 = \cdots = x_n = A/n$.
Fact 3. For $k \geq 1$,

$$1 - \left(1 - \frac{1}{k}\right)^k \geq 1 - \frac{1}{e}.$$

H. Expanded Discussions of Other Related Works

H.1. Relation to (Streeter and Golovin, 2008)

In (Streeter and Golovin, 2008), an offline iterative greedy algorithm was adapted for the knapsack constraint. In addition to the differences between adversarial and stochastic CMAB problem formulations and regret definitions discussed in Section 5, there are two key differences between the regret bounds of OG$^{\text{o}}$ in (Streeter and Golovin, 2008) and the regret bounds for the proposed adaptations C-ETC-B, C-ETC-Y, C-ETC-K, C-ETC-S making them incomparable.

The first key difference is that Streeter and Golovin (2008) only adapted an offline iterative greedy algorithm that in general does not achieve a constant approximation. The algorithm is a natural extension of the offline algorithm proposed by Nemhauser et al. (1978) for cardinality constraints. It iteratively adds elements based on density (marginal gain divided by cost). However, Khuller et al. (1999) implicitly showed that unless the greedy procedure happens to use up the whole budget exactly, which in general does not happen, the procedure will not achieve a constant approximation ratio. The offline algorithms we adapted, (Sviridenko, 2004; Yaroslavtsev et al., 2020), all use that iterative greedy procedure as a sub-routine, but then augment the output with elements based on values (instead of densities). While the iterative greedy subroutine is straightforward to adapt like OG$^{\text{o}}$ was (with a caveat described below as the second difference), it is unclear how the additional augmentation should be implemented. One possibility would be to add additional expert algorithms but they would not be sequential (the augmentations are distinct/independent of each other).

The second key difference is that while we considered "hard" knapsack constraints (i.e., every action/subset must be within budget), OG$^{\text{o}}$ was only designed to handle knapsack constraints *in expectation*, where the expectation is over the algorithm's randomness. More specifically, for each round, the algorithm constructs the actions by sampling base arms one by one based on some probabilities, so that overall the expected budget used is B. That means that OG$^{\text{o}}$ is allowed to select actions whose cost is larger than the budget B.

In (Golovin et al., 2014), the authors propose an algorithm for adversarial setting with submodular rewards when there is a matroid constraint (neither knapsack nor matroid constraints are special cases of the other).

H.2. Relation to (Niazadeh et al., 2021)

For the particular problem of submodular CMAB with knapsack constraints, we do not believe Niazadeh et al. (2021)'s results hold directly because of the required sub-problem structure:

- First, Niazadeh et al. (2021)'s framework requires a known number of sub-problems.
- Second, efficient offline approximation algorithms for knapsack-constrained submodular maximization do not have the "purely" iterative-greedy sub-problem structure required by Niazadeh et al. (2021)'s framework.

In the following, we explain these two points in detail. Before we discuss those points, we would also like to point out that Niazadeh et al. (2021)'s framework is for adversarial problems while ours is for stochastic problems; the two settings (and subsequently frameworks) are different from one another and cannot be specialized to the other. Thus, our novelty is not just in proposing a framework through which example offline algorithms can be adapted which were not adaptable under theirs, but in proposing a framework for stochastic problems where only bandit feedback (the reward) is available. Finally, we also note that the general regret guarantees of our framework are different than those of Niazadeh et al. (2021)'s; that may in part be due to different problem setups.

1. Niazadeh et al. (2021)'s framework requires the offline algorithm to have a known number of sub-problems (iterations). For offline approximation algorithms for submodular maximization with knapsack constraints, the number of iterations (corresponding to the number of elements added to the greedy set) is not known ahead of time. It varies with different problem instances. It can be upper-bounded (by $B/c_{\min}$), but for certain problem instances the greedy set chosen could be a single element and for other instances it could be a large cardinality set. A potential, albeit partial, "fix" that we believe could be done for iterative greedy offline approximation algorithms where the number of iterations is upper-bounded but not known a priori is to extend Niazadeh et al. (2021)'s framework for weakened constraints, namely that the constraint is only required to be met in expectation. We believe this could be done since Streeter and Golovin (2008)'s adaptation of the standard greedy algorithm for cardinality-constrained submodular maximization was similar to (Niazadeh et al., 2021)'s adaptation of the same algorithm. For knapsack problems, Streeter and Golovin (2008) adapted an offline iterative greedy algorithm but only considered satisfying the knapsack constraint in expectation. They take an upper bound on the number of iterations and converted each (potential) iteration into an experts algorithm. For each experts algorithm, the adaptation included an element with some probability so that the expected size was the budget. We anticipate a similar modification could be done for Niazadeh et al. (2021)'s framework as well.

2. Niazadeh et al. (2021)'s framework requires the offline algorithm to have an iterative greedy structure, where the output of the i-th subproblem is a feasible solution that is fed into the $(i+1)$-th subproblem. However, unlike the case for the cardinality constraint, the structures of the offline algorithms for knapsack-constrained submodular maximization are not "pure" iterative greedy. They do all employ an iterative greedy sub-routine that adds elements based on density (marginal value divided by cost), but that iterative greedy sub-routine alone does not achieve a constant approximation (Khuller et al., 1999). Offline approximation algorithms for this problem that achieve constant approximations all employ additional steps.
   - For example, Yaroslavtsev et al. (2020)'s procedure takes the output of the iterative greedy procedure and then augments it with additional elements. It may be possible to implement this second sub-routine as separate sub-problems (each adapted using expert algorithms), but these new subproblems would not be part of a chain of subproblems, the output of one feeding into the next. And these extra sub-problems would have different characteristics than those in the main iterative-greedy chain.
   - For Sviridenko (2004)'s partial enumeration procedure, which is known to achieve the best approximation ratio of $1-1/e$ but with high complexity, the solution is selected by re-running an iterative greedy sub-routine on every feasible subset of size at most three. This algorithm might be revised as a purely iterative greedy procedure, such as the first sub-problem selecting the subset of size as most three (so over $\binom{n}{3}$ possible choices), but this will lead to extremely slow update for the first sub-problem because of its dimension and bandit feedback, rendering it impractical for the online setting.

H.3. Relation to (Li et al., 2022)

The offline algorithm proposed in (Li et al., 2022) outputs a feasible solution, but to select that solution, it queries the value oracle for some subsets whose cost is above the budget. Specifically, in (Li et al., 2022), the first subroutine (used to bound $f(\mathrm{OPT})$) in the algorithm is an iterative greedy approximation algorithm, with selected set S' yielding an upper and lower bound on the optimal value $f(\mathrm{OPT})$, namely $\frac{1}{4}f(S') \leq f(\mathrm{OPT}) \leq 2f(S')$. Those bounds on the optimal value are then used in later sub-routines.

That first sub-routine iterates over all elements once (in an arbitrary order) and adds element u if $\frac{f(u|S')}{c(u)} \geq \frac{f(S')}{B}$, where S' is the currently selected set. There is no enforcement that S' constructed in this sub-routine remains feasible (i.e., its cost is under budget, $c(S') \leq B$). For example, consider f being linear (thus submodular), $B=2$, $f(1)=1$, $f(2)=2$, $f(3)=3$, and $c(u)=1$ for all $u \in \{1,2,3\}$, where the value u corresponds to the order the elements would be evaluated. That subroutine will select $\{1,2,3\}$ as the final set, with a total cost of $3>B$. Even if that sub-routine is modified to have two passes, such as one pass to evaluate marginal values, then a second pass over them in order of decreasing marginal values, that counter-example could be expanded by having all marginal values the same and then linear conditional values with similar construction as the example above.

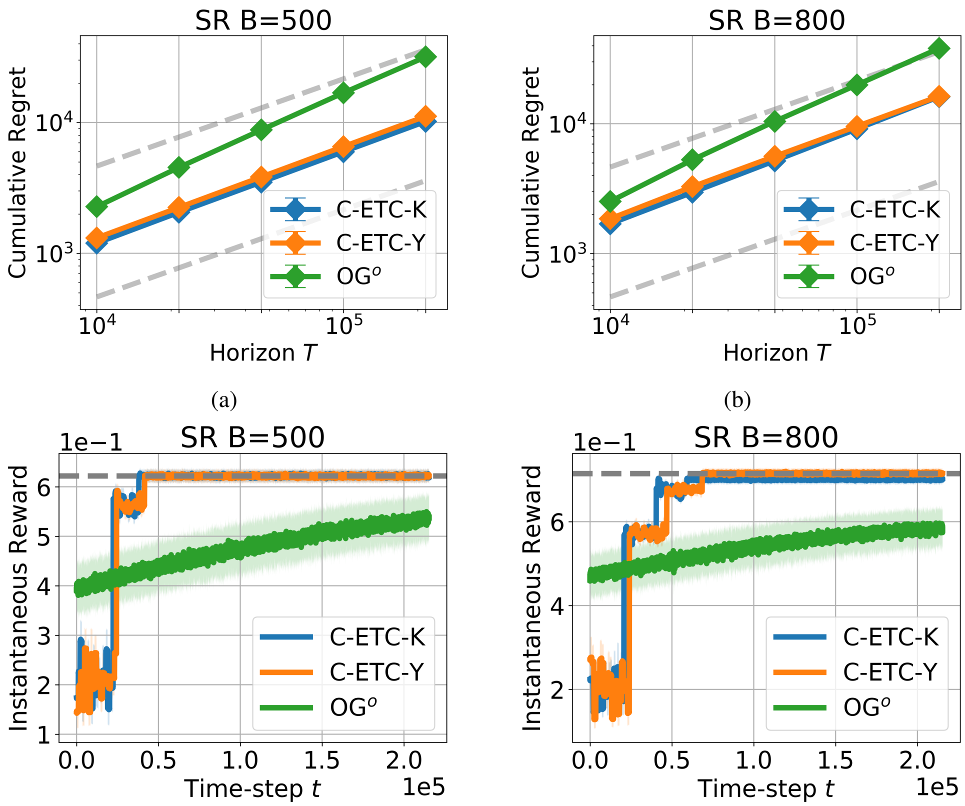
As mentioned, the sub-routine in question is only used for upper and lower bounding $f(\mathrm{OPT})$. Looser upper and lower bounds could be used so that only feasible sets are evaluated, namely $\max_u f(u) \leq f(\mathrm{OPT}) \leq \frac{B}{c_{\min}} \max_u f(u)$, but these would not provide upper and lower bound approximations that are a constant fraction within $f(\mathrm{OPT})$, which we expect would lead to the computational complexity depending on the budget B and thus no longer being a "clean linear time" algorithm.

H.4. Related work on Stochastic Submodular CMAB with Semi-Bandit Feedback

There are also a number of works that require additional "semi-bandit" feedback. For combinatorial MAB with submodular rewards, a common type of semi-bandit feedback are marginal gains (Lin et al., 2015; Yue and Guestrin, 2011; Yu et al., 2016; Takemori et al., 2020a), which enable the learner to take actions of maximal cardinality or budget, receive a corresponding reward, and gain information not just on the set but individual elements. For the full-bandit setting we consider, to greedily build a solution, we need to spend time taking small cardinality actions to estimate their quality, incurring regret.

I. Experiments with Song Recommendation

We test our methods on the application of song recommendation on the Million Song Dataset (Bertin-Mahieux et al., 2011). In this problem, the agents aims to recommend a bundle of songs to users such that they are liked by as many users as possible.



(a) (b) (c) (d)

Figure 2: Plots for song recommendation example. (a) and (b) are comparison results for cumulative regret as a function of time horizon T. (c) and (d) are the moving average plot with window size 100 of instantaneous reward as a function of t. The gray dashed lines in (a) and (b) represent $y = aT^{2/3}$ for various values of a for visual reference. The gray dashed lines in (c) and (d) represent expected rewards for the action chosen by an offline greedy algorithm.

Data Set Description and Experiment Details

From the Million Song Dataset, we extract most popular 20 songs and 100 most active users. As in Yue and Guestrin (2011), we model the system as having a set of topics (or genres) $\mathcal{G}$ with $|\mathcal{G}| = d$ and for each item $e \in \Omega$, there is a feature vector $x(e) := (P_g(e))_{g \in \mathcal{G}} \in \mathbb{R}^d$ that represents the information coverage on different genres. For each genre g, we define the probabilistic coverage function $f_g(S)$ by $1 - \prod_{e \in S}(1 - P_g(e))$ and define the reward function $f(S) = \sum_i w_i f_i(S)$ with linear coefficients w_i. The vector $w := [w_1, \ldots, w_d]$ represents user preference on genres. In calculating $P_g(e)$ and w, we use the same formula for calculating $\bar{w}(e, g)$ and θ^* in Hiranandani et al. (2020). Like Takemori et al. (2020b), we define the cost of a song by its length (in seconds). For each user, the stochastic rewards of set S are sampled from a Bernoulli distribution with parameter $f(S)$. For the total reward, we take the average over all users. When making the plots, we use statistics taken from 10 runs.

Results and Discussion

Figures 2a and 2b show average cumulative regret curves for C-ETC-K (in blue), C-ETC-Y (in orange) and OG^o (in green) for different horizon T values when the budget constraint B is 500 and 800, respectively. Figures 2c and 2d are the instantaneous reward plots over a single horizon $T = 215,443$. Again, C-ETC significantly outperforms OG^o for all time horizons and budget considered. We again estimated the slopes for both methods on log-log scale plots. Over the horizons tested, OG^o's cumulative regret (averaged over ten runs) has a growth rate above 0.85. The growth rates of C-ETC-K for budgets 500 and 800 are 0.70 and 0.73, respectively. The growth rates of C-ETC-Y for budgets 500 and 800 are 0.70 and 0.71, respectively.