Robust Sequence Submodular Maximization

Gamal Sallam[1], Zizhan Zheng[2], Jie Wu[1], and Bo Ji [1, 3]

[1]Department of Computer and Information Sciences, Temple University
[2]Department of Computer Science, Tulane University
[3]Department of Computer Science, Virginia Tech

Abstract

Submodularity is an important property of set functions and has been extensively studied in the literature. It models set functions that exhibit a diminishing returns property, where the marginal value of adding an element to a set decreases as the set expands. This notion has been generalized to considering sequence functions, where the order of adding elements plays a crucial role and determines the function value; the generalized notion is called sequence (or string) submodularity. In this paper, we study a new problem of robust sequence submodular maximization with cardinality constraints. The robustness is against the removal of a subset of elements in the selected sequence (e.g., due to malfunctions or adversarial attacks). Compared to robust submodular maximization for set function, new challenges arise when sequence functions are concerned. Specifically, there are multiple definitions of submodularity for sequence functions, which exhibit subtle yet critical differences. Another challenge comes from two directions of monotonicity: forward monotonicity and backward monotonicity, both of which are important to proving performance guarantees. To address these unique challenges, we design two robust greedy algorithms: while one algorithm achieves a constant approximation ratio but is robust only against the removal of a subset of contiguous elements, the other is robust against the removal of an arbitrary subset of the selected elements but requires a stronger assumption and achieves an approximation ratio that depends on the number of the removed elements. Finally, we generalize the analyses to considering sequence functions under weaker assumptions based on approximate versions of sequence submodularity and backward monotonicity.

1 Introduction

Submodularity is an important property of set functions and has been extensively studied in the literature [1–4]. It models set functions that exhibit a diminishing returns property, where the marginal value of adding an element to a set decreases as the set expands (i.e., contains more elements). The notion of submodularity has been generalized to considering sequence functions, where the order of adding elements plays a crucial role and determines the function value; the generalized notion is called sequence (or string) submodularity [5–11]. Several real-world applications, including machine learning based recommendation systems, ads allocation, and automation and control, involve the selection of elements in sequence. In this paper, we study a new problem of robust sequence submodular maximization with cardinality constraints. The robustness is against the removal of a subset of elements in the selected sequence (e.g., due to malfunctions or adversarial attacks). To motivate the new problem studied in this paper, we begin with the discussions about two concrete applications (sensor activation and movie recommendation) and use them to illustrate the key differences between set functions and sequence functions.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Table 1: Representative work on submodular maximization

	(Set) Submodular Maximization	Sequence Submodular Maximization
Non-robust	[1–4]	[5–11]
Robust	[13–25]	**This paper**

Sensor Activation. Consider the problem of sensor activation for moving target detection [5], where the objective is to sequentially activate a certain number of sensors to maximize the probability of detecting a moving target. Suppose that each sensor covers a certain area. Without any prior knowledge of the target location or the probability of detection at each location, it is plausible to maximize the total area covered by the activated sensors. If the coverage area of each sensor remains constant over time, then it is sufficient to decide which subset of sensors to activate without concerning the order in which the sensors are activated. This scenario can typically be modeled as maximizing a (set) submodular function.

In practice, however, the coverage area of each sensor may decay over time for several reasons, such as battery decay and corrosive environment. Accounting for such factors, the coverage area of a sensor may be modeled as a decreasing function, e.g., in the form of $Ce^{-t/T}$, where C is the initial coverage area, T is the sensor's lifetime, and $t = 0, 1, \dots$ is the time index. In this scenario, the sequence in which the sensors are activated is of critical importance. The activation sequence determines the total coverage area and thus impacts the probability of successful detection.

The above problem becomes more challenging if some sensors may be malfunctioning after they are activated or, even worse, if there is an adversary that may attack some sensors and render them non-working. Depending on how critical the application scenario is, one must ensure resilience of the sensor activation plan such that a certain performance (i.e., probability of successful detection) can still be guaranteed even in the worst-case failure scenario.

Movie Recommendation. Consider that a service provider (e.g., Netflix) would like to recommend movies to a user [6, 12]. It is common that the service provider recommends movies of similar flavor to a user. To some user, however, the incremental level of entertainment of watching a movie may decrease if the user had watched more similar movies, which exhibits a diminishing returns property.

In reality, however, the order in which the movies are recommended to the user may also impact how the user perceives a specific movie. In fact, movie recommendation and TV show recommendation have been modeled as sequence functions in [6] and [12], respectively. As noted in the motivating example in [12], if the model determines that the user might be interested in *The Lord of the Rings* series, then recommending *The Return of the King* first and *The Fellowship of the Ring* last could make the user unsatisfied with an otherwise excellent recommendation. Moreover, the user may not watch all the recommended videos possibly because the user has already watched some of them or does not like them (e.g., due to low ratings and/or unfavorable reviews).

While the problem of submodular maximization is generally NP-hard due to its combinatorial nature, the property of submodularity has been exploited to design efficient approximation algorithms. Since the seminal work in [1], it has been well known that a simple greedy algorithm and its variants can achieve an optimal approximation ratio[1] of $(1 - 1/e)$ in various settings. Specifically, a variant of the greedy algorithm has also been shown to achieve the same approximation ratio for the problem of sequence submodular maximization [7]. Recently, robust versions of the submodular maximization problem have aroused a lot of research interests (e.g., [13, 15, 16]). The focus of these studies is on selecting a set of elements that is robust against the removal of a subset of them.

In this paper, we take one step further and consider a robust version of the sequence submodular maximization problem. The goal is to select a sequence of elements (i.e., elements in a specific order) with cardinality constraints such that the value of the sequence function is maximized when a certain number of the selected elements may be removed. In Table 1, we position our work along with the literature on submodular maximization. The generalization to robust sequence submodular maximization introduces new challenges. As far as sequence functions are concerned, not only do the notions of submodularity and monotonicity involve variants of subtle yet critical differences, but

[1] The approximation ratio is defined as the ratio of the objective value achieved by an algorithm over that achieved by an optimal algorithm.

the design and analysis of robust algorithms are also faced with novel technical difficulties, which render the proofs more challenging. In the sequel, we elaborate on these unique challenges.

First, there are two definitions of submodularity for set functions: (i) the marginal value of adding an element to a set decreases as the set expands; (ii) the marginal value of adding a set to another set decreases as the latter set expands. It is trivial to show that these two definitions are equivalent. Replacing "set" with "sequence" in the above definitions gives two similar definitions for sequence functions. We will show that while (ii) still trivially implies (i) for sequence functions, the opposite does not hold in general. This leads to the following important question: Which definition of submodularity should one consider for sequence functions? Interestingly, while the weaker form (i) is sufficient for establishing provable approximation guarantees for non-robust sequence submodular maximization, one needs the stronger form (ii) to obtain similar results for the robust counterpart.

Second, while monotonicity is a straightforward notion for set functions (i.e., the value of a set function does not decrease as the set expands), there are two directions of monotonicity for sequence functions: forward monotonicity and backward monotonicity. Forward (resp., backward) monotonicity means that adding a sequence to the end (resp., beginning) of another sequence does not decrease the overall value. Both monotonicity properties are important to proving performance guarantees.

Third, the impact of removing an element from a sequence depends both on the element itself and on its position in the sequence. This makes the robust algorithms designed for set functions inapplicable here and calls for new robust algorithms that are better suited for sequence functions. Besides, one needs stronger assumptions to proceed with performance analysis for sequence functions. Therefore, it is more important to prove performance guarantees under weaker assumptions based on approximate versions of submodularity and monotonicity, which are more likely to hold in practice.

Due to these unique challenges, it is unclear what conditions are sufficient for establishing provable approximation ratios for robust sequence submodular maximization, how to design efficient and robust algorithms, and how to prove performance guarantees for the designed algorithms. We aim to answer these questions in this paper. Our contributions are summarized as follows.

- To the best of our knowledge, this is the first work that considers the problem of robust sequence submodular maximization. It is well known that the traditional (set) submodular maximization problem is already NP-hard. Accounting for sequence functions and robustness guarantees adds extra layers of difficulty, as the submodular and monotone properties of sequence functions involve variants with subtle yet critical differences.

- To address these unique challenges, we design two robust greedy algorithms for maximizing a forward-monotone, backward-monotone, and sequence-submodular function with cardinality constraints. While one algorithm achieves a constant approximation ratio but is robust only against the removal of a subset of contiguous elements, the other is robust against the removal of an arbitrary subset of the selected elements but requires a stronger assumption and achieves an approximation ratio that depends on the number of the removed elements. Although our proposed greedy algorithms are quite intuitive, the theoretical analysis is more challenging, and the presented approximation guarantees are highly nontrivial.

- We consider different definitions of submodularity and monotonicity and investigate their impacts on the derived theoretical results. Our study reveals that compared to set functions, one needs more properties of sequence functions to establish similar approximation results. On the other hand, we introduce general versions of such properties, such as approximate sequence submodularity and approximate backward monotonicity, and leverage them to prove approximation results of the proposed algorithms under weaker assumptions, which are more likely to hold in practice. We hope that this work serves as an important first step towards the design and analysis of efficient algorithms for robust sequence submodular maximization, which is worth further investigation through empirical evaluations for specific applications.

Due to space limitations, we provide all the proofs in the supplementary document.

2 System Model and Problem Formulation

Consider a set of elements $\mathcal{V}$, with $V = |\mathcal{V}|$, where $|\cdot|$ denotes the cardinality of a set. Let $(v_1, \ldots, v_m)$ be a sequence of non-repeated[2] elements selected over m steps, where $v_i \in \mathcal{V}$ for $i = 1, \ldots, m$, $v_i \neq v_j$ for $i \neq j$, and $m = 0, \ldots, |\mathcal{V}|$. When $m = 0$, the sequence is empty and is denoted by $()$. We use $\mathcal{H}(\mathcal{V})$ to denote the set of all possible sequences of non-repeated elements in $\mathcal{V}$, and we use $\mathcal{V}(S)$ to denote the set of elements in sequence $S \in \mathcal{H}(\mathcal{V})$. By slightly abusing the notation, we use $|S|$ to denote the number of elements in sequence S, i.e., $|S| = |\mathcal{V}(S)|$. Consider a sequence $S \in \mathcal{H}(\mathcal{V})$ and a set $\mathcal{U} \subseteq \mathcal{V}$. We use $S - \mathcal{U}$ to denote the sequence that is constructed by removing all the elements in $\mathcal{U}$ from sequence S without changing the order of the remaining elements. For instance, suppose $S = (v_2, v_1, v_5, v_3)$ and $\mathcal{U} = \{v_2, v_4, v_5\}$. Then, we have $S - \mathcal{U} = (v_1, v_3)$. For two sequences $S_1, S_2 \in \mathcal{H}(\mathcal{V})$, sequence S_1 is said to be a subsequence of sequence S_2 if we can write S_1 as $S_2 - \mathcal{U}$ for some $\mathcal{U} \subseteq \mathcal{V}(S_2)$. Consider two sequences $S_1 = (v_1, \ldots, v_{m_1})$ and $S_2 = (u_1, \ldots, u_{m_2})$ in $\mathcal{H}(\mathcal{V})$, and let $S_2 - \mathcal{V}(S_1) = (w_1, \ldots, w_{m_3})$. We define a concatenation of S_1 and S_2 as

$$S_1 \oplus S_2 \triangleq (v_1, \ldots, v_{m_1}, w_1, \ldots, w_{m_3}). \tag{1}$$

Note that the concatenated sequence $S_1 \oplus S_2$ has no repeated elements. We write $S_1 \preceq S_2$ if we can write S_2 as $S_1 \oplus S_3$ for some $S_3 \in \mathcal{H}(\mathcal{V})$.

Before we define the problem of sequence submodular maximization, which was first considered in [7], we introduce some important definitions. Consider a sequence function $h : \mathcal{H}(\mathcal{V}) \to \mathbb{R}^+$, where $\mathbb{R}^+$ is the set of non-negative real numbers. Without loss of generality, we assume that the value of an empty sequence is zero, i.e., $h(()) = 0$. We define the marginal value of appending sequence S_2 to sequence S_1 as $h(S_2|S_1) \triangleq h(S_1 \oplus S_2) - h(S_1)$. Function h is said to be sequence-submodular if for all $S_3 \in \mathcal{H}(\mathcal{V})$, we have

$$h(S_3|S_1) \geq h(S_3|S_2), \ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \ \text{ such that } S_1 \preceq S_2. \tag{2}$$

The above inequality represents a diminishing returns property. Similarly, function h is said to be element-sequence-submodular if for all $v \in \mathcal{V}$, we have

$$h((v)|S_1) \geq h((v)|S_2), \ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \ \text{ such that } S_1 \preceq S_2. \tag{3}$$

From the above definitons, it is easy to see that a sequence-submodular function must also be element-sequence-submodular. However, an element-sequence-submodular function may not necessarily be sequence-submodular; we provide such an example in our supplementary material. This is in contrary to submodular (set) functions, for which one can easily verify that similar definitions of Eqs. (2) and (3) imply each other. Although it is noted (without a proof) in [5] that using an induction argument, one can show that an element-sequence-submodular function must also be sequence-submodular, we find this claim false due to the counterexample we find (see our supplementary material).

Also, function h is said to be forward-monotone if

$$h(S_1 \oplus S_2) \geq h(S_1), \ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}), \tag{4}$$

and is said to be backward-monotone if

$$h(S_1 \oplus S_2) \geq h(S_2), \ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}). \tag{5}$$

For the sensor activation example we discussed in the introduction, forward monotonicity (resp., backward monotonicity) means that adding a sequence of sensors to the end (resp., the beginning) of another sequence of sensors does not reduce the total coverage area. We will later introduce approximate versions of sequence submodularity and backward monotonicity and generalize the theoretical results under weaker assumptions based on such generalized properties (see Section 4).

The problem of selecting a sequence $S \in \mathcal{H}(\mathcal{V})$ with an objective of maximizing function h with cardinality constraints (i.e., selecting no more than k elements for $k > 0$) can be formulated as

$$\max_{S \in \mathcal{H}(\mathcal{V}),\ |S| \leq k} \quad h(S). \tag{P}$$

[2] This definition can be easily generalized to allow repetition by augmenting the ground set $\mathcal{V}$ as follows. Assume that each element $v_i \in \mathcal{V}$ can be repeated z_i times. Let v_i^j denote the j-th copy of element v_i. We use $\bar{\mathcal{V}}$ to denote the augmented ground set, which is defined as $\bar{\mathcal{V}} \triangleq \cup_{v_i \in \mathcal{V}} \{v_i^1, \ldots, v_i^{z_i}\}$. Therefore, we can replace $\mathcal{V}$ with the augmented ground set $\bar{\mathcal{V}}$, which essentially allows the repetition of elements in $\mathcal{V}$.

Next, we propose a robust version of Problem (P), which accounts for the removal of some of the selected elements. Consider $\tau \leq k$. The robust version of Problem (P) can be formulated as

$$\max_{S \in \mathcal{H}(\mathcal{V}),\ |S| \leq k} \ \min_{\mathcal{V}' \subseteq \mathcal{V}(S),\ |\mathcal{V}'| \leq \tau} \quad h(S - \mathcal{V}'). \tag{R}$$

Without loss of generality, we assume $k > 1$ for Problem (R). In the next section, we discuss the challenges of Problem (R) and present the proposed robust algorithms.

3 Proposed Robust Algorithms

We begin with a discussion about the non-robust sequence submodular maximization problem (Problem (P)), through which we provide useful insights into the understanding of the challenges of Problem (R). Although Problem (P) is NP-hard, it can be approximately solved using a simple *Sequence Submodular Greedy* (SSG) algorithm [7]. Under the SSG algorithm, we begin with an empty sequence S; in each iteration, we choose an element that leads to the largest marginal value with respect to S and append it to sequence S, i.e., $S = S \oplus \arg\max_{v \in \mathcal{V} \setminus \mathcal{V}(S)} h((v)|S)$. We repeat the above procedure until k elements have been selected. It has been shown in [7] that the SSG algorithm achieves an approximation ratio of $(1 - 1/e)$ for maximizing a forward-monotone, backward-monotone, and element-sequence-submodular function with cardinality constraints.

Although the SSG algorithm approximately solves Problem (P), it can perform very poorly if one directly applies it to solving its robust counterpart (Problem (R)). The intuition behind this is the following. The SSG algorithm tends to concentrate the value of the selected sequence on the first few elements. Selecting elements in this manner leaves the overall sequence vulnerable as removing some of these elements would have a high impact on the overall value. Consider the following example, where we assume $\tau = 1$ for simplicity. Let $\mathcal{V}_1 = \{v\}$, $\mathcal{V}_2 = \{u_1, \dots, u_n\}$, $\mathcal{V}_3 = \{w_1, \dots, w_n\}$, and $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3$. Assume $h((v)) = 1$, $h((u_i)) = 1/n$ for all $u_i \in \mathcal{V}_2$, and $h((w_i)) = \epsilon$ for all $w_i \in \mathcal{V}_3$, where ϵ is an arbitrarily small positive number. Also, assume that for any $S_1, S_2 \in \mathcal{H}(\mathcal{V})$ such that $v \in \mathcal{V}(S_1)$ and $v \notin \mathcal{V}(S_2)$, we have $h((u_i)|S_1) = 0$ and $h((u_i)|S_2) = 1/n$ for all $u_i \in \mathcal{V}_2$ and $h((w_i)|S_1) = h((w_i)|S_2) = \epsilon$ for all $w_i \in \mathcal{V}_3$. Suppose $k = n$. Then, the SSG algorithm will select v as the first element and select the subsequent $n - 1$ elements from $\mathcal{V}_3$. The value of the selected sequence will be $1 + (n-1)\epsilon$. If element v is removed, then the value of the remaining sequence will be $(n-1)\epsilon$, which can be arbitrarily small. In contrast, a sequence consisting of n elements from $\mathcal{V}_2$ will be robust against the removal of any element. This is because the overall value is equally distributed across all the elements in the sequence and the value of the sequence after removing any element is $(n-1)/n$.

The above example shows that the SSG algorithm may perform arbitrarily bad for Problem (R). To that end, we propose two greedy algorithms that can address this limitation and ensure robustness of the selected sequence for Problem (R). First, we propose an algorithm that achieves a constant approximation ratio but is robust only against the removal of τ contiguous elements (Section 3.1). Then, we further propose an algorithm that works in a general setting without the contiguous restriction and is robust against the removal of an arbitrary subset of τ selected elements, but it requires a stronger assumption and achieves an approximation ratio that depends on the number of removed elements (Section 3.2).

3.1 Robustness Against the Removal of Contiguous Elements

In this subsection, we wish to design an algorithm that is robust against the removal of τ contiguous elements. The assumption of the removal of contiguous elements can model a spatial relationship such as sensors in close proximity or a temporal relationship such as consecutive episodes of a TV show. We design a variant of the SSG algorithm that approximately solves Problem (R). The algorithm is presented in Algorithm 1. As we discussed earlier, the limitation of the SSG algorithm is that the selected sequence is vulnerable because the overall value might be concentrated in the first few elements. Algorithm 1 is motivated by this key observation and works in two steps. In Step 1, we select a sequence S_1 of τ elements from elements in $\mathcal{V}$ in a greedy manner as in SSG. In Step 2, we select another sequence S_2 of $k - \tau$ elements from elements in $\mathcal{V} \setminus \mathcal{V}(S_1)$, again in a greedy manner as in SSG. Note that when we select sequence S_2, we perform the greedy selection as if sequence S_1 does not exist at all. This ensures that the value of the final sequence $S = S_1 \oplus S_2$ is

Algorithm 1 Robust greedy algorithm against the removal of contiguous elements

```
Input: $\mathcal{V}, k, \tau$; Output: $S$
Initialization: $S = S_1 = S_2 = ()$
//**Step 1:**
**while** $|S_1| < \tau$ **do**
    $S_1 = S_1 \oplus \arg\max_{v \in \mathcal{V} \setminus \mathcal{V}(S_1)} h((v)|S_1)$
**end while**
//**Step 2:**
**while** $|S_2| < k - \tau$ **do**
    $S_2 = S_2 \oplus \arg\max_{v \in \mathcal{V} \setminus (\mathcal{V}(S_1) \cup \mathcal{V}(S_2))} h((v)|S_2)$
**end while**
$S = S_1 \oplus S_2$
```

Algorithm 2 Robust greedy algorithm against the removal of arbitrary elements

```
Input: $\mathcal{V}, k, \tau$; Output: $S$
Initialization: $S = S_1 = S_2 = ()$
//**Step 1:**
**while** $|S_1| < \tau$ **do**
    $S_1 = S_1 \oplus \arg\max_{v \in \mathcal{V} \setminus \mathcal{V}(S_1)} h((v))$
**end while**
//**Step 2:**
**while** $|S_2| < k - \tau$ **do**
    $S_2 = S_2 \oplus \arg\max_{v \in \mathcal{V} \setminus (\mathcal{V}(S_1) \cup \mathcal{V}(S_2))} h((v)|S_2)$
**end while**
$S = S_1 \oplus S_2$
```

not concentrated in either S_1 or S_2. The complexity of Algorithm 1 is $O(kV)$, which is in terms of the number of function evaluations used in the algorithm.

We first state the following assumption that is needed for deriving the main results in this subsection.

Assumption 1. *Function h is forward-monotone, backward-monotone, and sequence-submodular.*

In Theorem 1, we state the approximation result of Algorithm 1 in a special case of $\tau = 1$. We consider this special case for two reasons: (i) it is easier to explain the key ideas in the proof of this special case; (ii) we can prove better approximation ratios in this special case, which may not be obtained from the analysis in the case of $1 \leq \tau \leq k$.

Theorem 1. *Consider $\tau = 1$. Under Assumption 1, Algorithm 1 achieves an approximation ratio of $\max\left\{\frac{e-1}{2e}, \frac{e^{\frac{k-2}{k-1}}-1}{2e^{\frac{k-2}{k-1}}-1}\right\}$, which is lower bounded by a constant $\frac{e-1}{2e}$.*

Remark. *The two terms of the approximation ratio in Theorem 1 have different advantages. While the first term remains constant (i.e., $\frac{e-1}{2e} \approx 0.316$), the second term (i.e., $(e^{\frac{k-2}{k-1}} - 1)/(2e^{\frac{k-2}{k-1}} - 1)$) is a monotonically increasing function of k. The first term is larger for a small value of k (when $k < 4$); the second term becomes larger for a wide range of k (when $k \geq 4$).*

In Theorem 2, we state the approximation result of Algorithm 1 in the case of $1 \leq \tau \leq k$.

Theorem 2. *Consider $1 \leq \tau \leq k$. Under Assumption 1, Algorithm 1 achieves an approximation ratio of $\max\left\{\frac{(e-1)^2}{e(2e-1)}, \frac{(e-1)(e^{\frac{k-2\tau}{k-\tau}}-1)}{(2e-1)e^{\frac{k-2\tau}{k-\tau}}-(e-1)}\right\}$, which is lower bounded by a constant $\frac{(e-1)^2}{e(2e-1)}$.*

Remark. *The two terms of the approximation ratio in Theorem 2 have different advantages. While the first term remains constant (i.e., $\frac{(e-1)^2}{e(2e-1)} \approx 0.245$), the second term is a monotonically increasing (resp., decreasing) function of k (resp., τ). The first term is larger when $k < \frac{2-\ln(\frac{e^2+e-1}{2e-1})}{1-\ln(\frac{e^2+e-1}{2e-1})}\tau$; the second term becomes larger when $k \geq \frac{2-\ln(\frac{e^2+e-1}{2e-1})}{1-\ln(\frac{e^2+e-1}{2e-1})}\tau$. We provide the approximation ratio under different values of τ and k in Table 3 in the Appendix of the supplementary document.*

3.2 Robustness Against the Removal of Arbitrary Elements

In this subsection, we wish to design an algorithm that is robust against the removal of an arbitrary subset of τ selected elements, which are not necessarily contiguous. One weakness of Algorithm 1 is that the value of the selected sequence could be concentrated in the first few elements of subsequences S_1 and S_2. If we allow the removal of an arbitrary subset of τ selected elements, the removal of the first few elements of subsequences S_1 and S_2 could leave the remaining sequence with little or no value. By considering a special case in Section 3.1 where the removed elements are restricted to be contiguous, we have managed to prevent such worst case from happening. However, the problem becomes more challenging when we consider a more general case without such a restriction. In the following, we propose an algorithm that is robust against the removal of an arbitrary subset of τ selected elements, but it requires a stronger assumption and achieves an approximation ratio that depends on the value of τ. This new algorithm is presented in Algorithm 2.

Similar to Algorithm 1, Algorithm 2 works in two steps. However, there is a subtle yet critical difference in Step 1, which is the key to ensuring robustness in the general case. Specifically, in Step 1 of Algorithm 2, we select a sequence S_1 of τ elements from $\mathcal{V}$ by iteratively choosing an element v in a greedy manner, based on its absolute value $h((v))$ instead of its marginal value $h((v)|S_1)$ as in Algorithm 1. We then select a sequence S_2 in Step 2, which is the same as that of Algorithm 1. The final output is $S = S_1 \oplus S_2$. Algorithm 2 also has a complexity of $O(kV)$.

Before we state the approximation results of Algorithm 2, we introduce a generalized definition of sequence submodularity. Function h is said to be general-sequence-submodular if for all $S_3 \in \mathcal{H}(\mathcal{V})$, we have

$$h(S_3|S_1) \geq h(S_3|S_2),\ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \text{ such that } S_1 \text{ is a subsequence of } S_2. \tag{6}$$

Note that $S_1 \preceq S_2$ implies that S_1 is a subsequence of S_2, but not vice versa. Therefore, the general sequence submodularity defined above generalizes the sequence submodularity defined in Eq. (2) as a special case with $S_1 \preceq S_2$. Next, we state Assumption 2 and Theorem 3.

Assumption 2. *Function h is forward-monotone, backward-monotone, and general-sequence-submodular.*

Theorem 3. *Consider $1 \leq \tau \leq k$. Under Assumption 2, Algorithm 2 achieves an approximation ratio of $\frac{1-1/e}{1+\tau}$.*

Remark. *While we only need the simplest form of the diminishing returns definition (element-sequence-submodularity) to establish provable approximation guarantees for the non-robust sequence submodular maximization, for its robust counterpart, we require stronger assumptions (sequence-submodularity and general-sequence-submodularity vs. element-sequence-submodularity) to show provable performance guarantees. In addition, consider a set function r and ground set $\mathcal{V}$. While monotonicity of set function r implies monotonicity of the same function with respect to the marginal value of adding a set to another set (i.e., $r(\mathcal{V}_2|\mathcal{V}_1) \triangleq r(\mathcal{V}_1 \cup \mathcal{V}_2) - r(\mathcal{V}_1)$ for any $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}$), a similar property does not hold for backward monotonicity of sequence functions. This subtle difference results in a more involved analysis of showing similar results for sequence functions.*

4 Robust Approximate Sequence Submodular Maximization

In this section, we introduce generalized versions of sequence submodularity and backward monotonicity, which are called approximate sequence submodularity and approximate backward monotonicity. Then, we show that Algorithms 1 and 2 can also approximately solve Problem (R) under weaker assumptions based on such generalized properties.

We begin with some additional definitions. Consider $\mu_1 \in (0, 1]$. Function h is said to be μ_1-element-sequence-submodular if for all $v \in \mathcal{V}$, we have

$$h((v)|S_1) \geq \mu_1 h((v)|S_2), \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \text{ such that } S_1 \preceq S_2. \tag{7}$$

Also, consider $\mu_2 \in (0, 1]$. Function h is said to be μ_2-sequence-submodular if for all $S_3 \in \mathcal{H}(\mathcal{V})$, we have

$$h(S_3|S_1) \geq \mu_2 h(S_3|S_2),\ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \text{ such that } S_1 \preceq S_2. \tag{8}$$

Note that μ_1 could be greater than μ_2 for some function h. We distinguish μ_1 and μ_2 as some of our results depend on μ_1 only. Similarly, consider $\mu_3 \in (0, 1]$. Function h is said to be μ_3-general-sequence-submodular if for all $S_3 \in \mathcal{H}(\mathcal{V})$, we have

$$h(S_3|S_1) \geq \mu_3 h(S_3|S_2),\ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}) \text{ such that } S_1 \text{ is a subsequence of } S_2. \tag{9}$$

Consider $\alpha \in (0, 1]$. Function h is said to be α-backward-monotone if

$$h(S_1 \oplus S_2) \geq \alpha h(S_2),\ \forall S_1, S_2 \in \mathcal{H}(\mathcal{V}). \tag{10}$$

Next, we state several assumptions that will be needed for deriving the main results in this section.

Assumption 3. *Function h is forward-monotone, backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular.*

Assumption 4. *Function h is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular.*

Assumption 5. *Function h is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_3-general-sequence-submodular.*

We are now ready to state the generalized approximation results of Algorithm 1 under Assumptions 3 and 4, respectively.

Theorem 4. *Consider $\tau = 1$. Under Assumption 3, Algorithm 1 achieves an approximation ratio of $\frac{a(e^b-1)}{e^b-a}$, where $a = \frac{\mu_1\mu_2}{\mu_1+1}$ and $b = \mu_1 \cdot \frac{k-2}{k-1}$; under Assumption 4, Algorithm 1 achieves an approximation ratio of $\frac{\alpha^2\mu_1\mu_2(e^{\mu_1}-1)}{(\mu_1+\alpha)e^{\mu_1}}$.*

Theorem 5. *Consider $1 \leq \tau \leq k$. Under Assumption 3, Algorithm 1 achieves an approximation ratio of $\frac{a\mu_2(e^b-1)}{(a+1)e^b-a\mu_2}$, where $a = \mu_1 \cdot (1 - 1/e^{\mu_1})$ and $b = \mu_1 \cdot \frac{k-2\tau}{k-\tau}$; under Assumption 4, Algorithm 1 achieves an approximation ratio of $\frac{\alpha^2\mu_1\mu_2(e^{\mu_1}-1)^2}{\mu_1 e^{\mu_1}(e^{\mu_1}-1)+e^{2\mu_1}}$.*

Finally, we state the approximation result of Algorithms 2 under Assumption 5.

Theorem 6. *Consider $1 \leq \tau \leq k$. Under Assumption 5, Algorithm 2 achieves an approximation ratio of $\frac{\alpha^2\mu_1\mu_3(e^{\mu_1}-1)}{(\mu_1+\alpha\tau)e^{\mu_1}}$.*

5 Related Work

Since the seminal work in [1], submodular maximization has been extensively studied in the literature. Several efficient approximation algorithms have been developed for maximizing a submodular set function in various settings (e.g., [1–4]). The concept of sequence (or string) submodularity for sequence functions is a generalization of submodularity, which has been introduced recently in several studies (e.g., [5–11, 26]). In [7], it has been shown that a simple greedy algorithm can achieve an approximation ratio of $(1 - 1/e)$ for maximizing a forward-monotone, backward-monotone, and element-sequence-submodular function.

On the other hand, robust versions of submodular maximization has been considered in some recent studies (e.g., [13–16]), where the focus is on selecting a set of elements that is robust against the removal of a subset of them. In [13], the authors propose the first algorithm with a constant approximation ratio for the problem of robust submodular maximization with cardinality constraints, where the selected set is of size k and the robustness is against the removal of any τ elements of the selected set. The constant approximation ratio derived in [13] is valid as long as the number of removed elements is small compared to the selected set (i.e., $\tau = o(\sqrt{k})$). An extension that guarantees the same constant approximation ratio but allows the removal of a larger number of elements (i.e., $\tau = o(k)$) is presented in [14]. Another algorithm that allows the removal of an arbitrary number of elements under a mild assumption is presented in [15]. The work in [16] relaxes the restriction on τ, but the achieved approximation ratio depends on the value of τ. The work in [17] considers the same problem under different types of constraints, such as matroid and knapsack constraints. The work in [18, 19] extends the work in [16] to a multi-stage setting, where the decision at one stage takes into account the failures that happened in the previous stages. Other extensions that consider fairness and privacy issues are studied in [20, 21]. It is unclear whether all of these algorithms for robust set submodularity can be properly extended to our problem, as converting a set into a sequence could result in an arbitrarily bad performance. Even if so, it is more likely that establishing their approximation guarantees would require a more sophisticated analysis, which calls for more in-depth investigations. Note that the analysis of our simple greedy algorithms is already very sophisticated.

In [22], a different notion of robustness is considered, which is referred to as maximizing the minimum of multiple submodular functions. This work proposes a bicriterion approximation algorithm for the studied problem with cardinality constraints. Moreover, the work in [23–25] extends that of [22] to accommodate a wide variety of constraints, including matroid and knapsack constraints. The work in [27] develops an approximation algorithm for robust non-submodular maximization, using other characterizations such as the submodularity ratio and the inverse curvature. The work in [12] introduces the idea of adaptive sequence submodular maximization, which aims to utilize the feedback obtained in previous iterations to improve the current decision. Note that while the work in [6, 12, 26] assumes that the sequential relationship among elements is encoded as a directed acyclic graph, we consider a general setting without such structures. It would indeed be interesting to explore our algorithms when the sequential relationship is encoded in a specific graphical form.

6 Conclusion

In this paper, we investigated a new problem of robust sequence submodular maximization. We discussed the unique challenges introduced by considering sequence functions and ensuring robustness guarantees. To address these novel challenges, we proposed two robust greedy algorithms and proved that they can achieve certain approximation ratios for the considered problem, assuming forward-monotone, backward-monotone, and sequence-submodular functions. We further introduced approximate versions of sequence submodularity and backward monotonicity and showed that the proposed algorithms can also provide performance guarantees under a larger class of weaker assumptions based on such generalized properties. Our future work includes developing more efficient algorithms with better approximation ratios in the general settings and investigating the possibility of obtaining similar results under the assumption of generalized/approximate forward monotonicity.

7 Broader Impact

This work contributes to the state-of-the-art theory of submodular optimization. The proposed algorithms and the presented approximation results can be applied to real-world applications where the stated assumptions of sequence submodularity and monotonicity or their approximate versions are satisfied. Several real-world applications, including machine learning based recommendation systems, ads allocation, and automation and control, involve the selection of elements in sequence.

Acknowledgement

This work was supported in part by the NSF under Grants CNS-1651947, CNS-1824440, CNS-1828363, and CNS-1757533.

References

[1] G. L. Nemhauser and L. A. Wolsey, “Best algorithms for approximating the maximum of a submodular set function,” *Mathematics of operations research*, vol. 3, no. 3, pp. 177–188, 1978.

[2] S. Khuller, A. Moss, and J. S. Naor, “The budgeted maximum coverage problem,” *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[3] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák, “Maximizing a monotone submodular function subject to a matroid constraint,” *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[4] C. Chekuri, J. Vondrák, and R. Zenklusen, “Submodular function maximization via the multilinear relaxation and contention resolution schemes,” *SIAM Journal on Computing*, vol. 43, no. 6, pp. 1831–1879, 2014.

[5] Z. Zhang, E. K. Chong, A. Pezeshki, and W. Moran, “String submodular functions with curvature constraints,” *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 601–616, 2016.

[6] S. Tschiatschek, A. Singla, and A. Krause, “Selecting sequences of items via submodular maximization,” in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] M. Streeter and D. Golovin, “An online algorithm for maximizing submodular functions,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1577–1584.

[8] Z. Zhang, E. K. Chong, A. Pezeshki, W. Moran, and S. D. Howard, “Submodularity and optimality of fusion rules in balanced binary relay trees,” in *IEEE Conference on Decision and Control*, 2012, pp. 3802–3807.

[9] S. Alaei, A. Makhdoumi, and A. Malekian, “Maximizing sequence-submodular functions and its application to online advertising,” *arXiv preprint arXiv:1009.4153*, 2010.

[10] S. T. Jawaid and S. L. Smith, “Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems,” *Automatica*, vol. 61, pp. 282–288, 2015.

[11] Z. Zhang, Z. Wang, E. K. Chong, A. Pezeshki, and W. Moran, “Near optimality of greedy strategies for string submodular functions with forward and backward curvature constraints,” in *IEEE Conference on Decision and Control*, 2013, pp. 5156–5161.

[12] M. Mitrovic, E. Kazemi, M. Feldman, A. Krause, and A. Karbasi, “Adaptive sequence submodularity,” in *Advances in Neural Information Processing Systems*, 2019, pp. 5353–5364.

[13] J. B. Orlin, A. S. Schulz, and R. Udwani, “Robust monotone submodular function maximization,” *Mathematical Programming*, vol. 172, no. 1-2, pp. 505–537, 2018.

[14] I. Bogunovic, S. Mitrović, J. Scarlett, and V. Cevher, “Robust submodular maximization: A non-uniform partitioning approach,” in *International Conference on Machine Learning*, 2017, pp. 508–516.

[15] S. Mitrovic, I. Bogunovic, A. Norouzi-Fard, J. M. Tarnawski, and V. Cevher, “Streaming robust submodular maximization: A partitioned thresholding approach,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4557–4566.

[16] V. Tzoumas, K. Gatsis, A. Jadbabaie, and G. J. Pappas, “Resilient monotone submodular function maximization,” in *IEEE Conference on Decision and Control*, 2017, pp. 1362–1367.

[17] R. Iyer, “A unified framework of robust submodular optimization,” *arXiv preprint arXiv:1906.06393*, 2019.

[18] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, “Resilient monotone sequential maximization,” in *IEEE Conference on Decision and Control*, 2018, pp. 7261–7268.

[19] ——, “Robust and adaptive sequential submodular optimization,” *arXiv preprint arXiv:1909.11783*, 2019.

[20] B. Mirzasoleiman, A. Karbasi, and A. Krause, “Deletion-robust submodular maximization: Data summarization with the right to be forgotten,” in *International Conference on Machine Learning*, 2017, pp. 2449–2458.

[21] E. Kazemi, M. Zadimoghaddam, and A. Karbasi, “Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints,” in *International Conference on Machine Learning*, 2018, pp. 2544–2553.

[22] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, “Robust submodular observation selection,” *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.

[23] T. Powers, J. Bilmes, S. Wisdom, D. W. Krout, and L. Atlas, “Constrained robust submodular optimization,” in *NIPS OPT2016 workshop*, 2016.

[24] T. Powers, D. W. Krout, J. Bilmes, and L. Atlas, “Constrained robust submodular sensor selection with application to multistatic sonar arrays,” *IET Radar, Sonar & Navigation*, vol. 11, no. 12, pp. 1776–1781, 2017.

[25] N. Anari, N. Haghtalab, S. Naor, S. Pokutta, M. Singh, and A. Torrico, “Structured robust submodular maximization: Offline and online algorithms,” in *International Conference on Artificial Intelligence and Statistics*, 2019.

[26] M. Mitrovic, M. Feldman, A. Krause, and A. Karbasi, “Submodularity on hypergraphs: From sets to sequences,” *arXiv preprint arXiv:1802.09110*, 2018.

[27] I. Bogunovic, J. Zhao, and V. Cevher, “Robust maximization of non-submodular objectives,” in *International Conference on Artificial Intelligence and Statistics*, 2018.

[28] G. Sallam, Z. Zheng, and B. Ji, “Placement and allocation of virtual network functions: Multi-dimensional case,” *arXiv preprint arXiv:1910.06299*, 2019.

8 Appendix

8.1 Element-sequence-submodular vs. Sequence-submodular: A Counterexample

In [5], it is noted (without a proof) that an element-sequence-submodular function must also be sequence-submodular. However, we find this claim false and present a counterexample in Table 2 to show that an element-sequence-submodular function is not necessarily sequence-submodular. In the presented example, it is easy to verify that function h is element-sequence-submodular, i.e., Eq. (3) is satisfied; however, it is not sequence-submodular, i.e., Eq. (2) is not satisfied. Specifically, we have $h((v_2, v_3)|()) < h((v_2, v_3)|(v_1))$ due to $h((v_2, v_3)|()) = h((v_2, v_3)) - h(()) = 1.2 - 0 = 1.2$ and $h((v_2, v_3)|(v_1)) = h((v_1, v_2, v_3)) - h((v_1)) = 2.2 - 0.2 = 2$.

Table 2: A counterexample showing that an element-sequence-submodular function is not necessarily sequence-submodular, where $\mathcal{V} = \{v_1, v_2, v_3\}$ and the columns are input sequences, values of function h, and marginal values of v_1, v_2, and v_3 with respect to sequence S, respectively

Function h Sequence S	$h(S)$	$h((v_1)\|S)$	$h((v_2)\|S)$	$h((v_3)\|S)$
$()$	0	0.2	1.2	1
(v_1)	0.2	0	1	1
(v_2)	1.2	0	0	0
(v_3)	1	0.2	1	0
(v_1, v_2)	1.2	0	0	1
(v_2, v_1)	1.2	0	0	0
(v_1, v_3)	1.2	0	1	0
(v_3, v_1)	1.2	0	1	0
(v_2, v_3)	1.2	0	0	0
(v_3, v_2)	2	0.2	0	0
(v_1, v_2, v_3)	2.2	0	0	0
(v_1, v_3, v_2)	2.2	0	0	0
(v_2, v_1, v_3)	1.2	0	0	0
(v_2, v_3, v_1)	1.2	0	0	0
(v_3, v_1, v_2)	2.2	0	0	0
(v_3, v_2, v_1)	2.2	0	0	0

8.2 Approximation Ratio of Theorem 2 with Different τ and k

Table 3: Approximation ratio of Theorem 2 with different values of τ and k

k τ	50	52	54	56	58	60	62	64	66	68
2	0.28	0.281	0.281	0.281	0.281	0.281	0.281	0.282	0.282	0.282
4	0.275	0.275	0.275	0.276	0.276	0.277	0.277	0.277	0.277	0.278
6	0.268	0.269	0.269	0.27	0.271	0.271	0.272	0.272	0.273	0.273
8	0.26	0.261	0.262	0.263	0.264	0.265	0.266	0.267	0.267	0.268
10	0.25	0.252	0.254	0.255	0.257	0.258	0.259	0.261	0.262	0.262
12	0.245	0.245	0.245	0.246	0.248	0.25	0.252	0.253	0.255	0.256
14	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.247	0.249
16	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245
18	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245
20	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245

8.3 Preliminary Results: Sequence Submodular Maximization

In this section, we introduce some additional notations and present some important preliminary results that will be used in the proofs of Theorems 1, 2, and 3.

Algorithm 3 Sequence Submodular Greedy (SSG)

```
Input: elements $\mathcal{V}$, $k$; Output: $S$
Initialization: $S = ()$
**while** $|S| < k$ **do**
    $S = S \oplus \arg\max_{v \in \mathcal{V} \setminus \mathcal{V}(S)} h((v)|S)$
**end while**
```

We begin with the additional notations. We use sequence $S^*(\mathcal{V}, k, \tau)$ to denote an optimal solution to Problem (R). Note that Problem (P) is a special case of Problem (R) with $\tau = 0$. Therefore, sequence $S^*(\mathcal{V}, k, 0)$ denotes an optimal solution to Problem (P). For any given sequence $S \in \mathcal{H}(\mathcal{V})$, we use $\mathcal{Z}_\tau(S)$ to denote an optimal solution to the following minimization problem:

$$\min_{\mathcal{V}' \subseteq \mathcal{V}(S),\ |\mathcal{V}'| \leq \tau} \quad h(S - \mathcal{V}'). \tag{11}$$

Let $g_\tau(S)$ denote the value of function h with input S that has elements in $\mathcal{Z}_\tau(S)$ removed, i.e., $g_\tau(S) \triangleq h(S - \mathcal{Z}_\tau(S))$. Therefore, the optimal value of Problem (R) is $g_\tau(S^*(\mathcal{V}, k, \tau)) = h(S^*(\mathcal{V}, k, \tau) - \mathcal{Z}_\tau(S^*(\mathcal{V}, k, \tau)))$. Moreover, it is easy to see that we have

$$g_\tau(S^*(\mathcal{V}, k, \tau)) \leq h(S^*(\mathcal{V}, k - \tau, 0)). \tag{12}$$

This is because the left-hand side is the value of a feasible solution to Problem (P) while the right-hand side is the value of an optimal solution to Problem (P).

First, we restate the approximation performance of the SSG algorithm (presented in Algorithm 3) in [7]. Let sequence S with $|S| = k$ be the one selected by the SSG algorithm, and let S^i be the sequence consisting of the first i elements of sequence S with $1 \leq i \leq k$. The result in [7] is stated for the case of $i = k$ only, but it can be easily generalized for any i with $1 \leq i \leq k$.

Lemma 1 (Theorem 3 of [7])**.** *Consider $1 \leq i \leq k$. Under Assumption 1, we have $h(S^i) \geq (1 - 1/e^{\frac{i}{k}})h(S^*(\mathcal{V}, k, 0))$.*

Next, we state in Lemma 2 that the approximation performance of the SSG algorithm can be better characterized if the value of the selected sequence is concentrated in its first few elements. We present the proof of Lemma 2 in Appendix 8.4.

Lemma 2. *Consider $c \in (0, 1]$ and $1 \leq k' \leq k$. Suppose that the sequence selected by the SSG algorithm is S with $|S| = k$ and that there exists a sequence S_1 with $|S_1| = k - k'$ such that $S_1 \preceq S$ and $h(S_1) \geq c \cdot h(S)$. Then, under Assumption 1, we have $h(S) \geq \frac{e^{\frac{k'}{k}} - 1}{e^{\frac{k'}{k}} - c} h(S^*(\mathcal{V}, k, 0))$.*

Remark. *Lemma 2 implies that additional information about the sequence selected by the SSG algorithm can be exploited to prove a better approximation ratio. In the following, we provide a few examples to demonstrate the significance of Lemma 2. Assume $k = 20$ and $k' = 19$. Then, we have $|S_1| = k - k' = 1$. If the value of sequence S is evenly distributed over all of its twenty elements, then we have $c = 0.05$ and $h(S) \geq 0.62h(S^*(\mathcal{V}, k, 0))$. This approximation ratio is approximately equal to that in Lemma 1 (0.62 vs. $1 - 1/e \approx 0.63$). On the other hand, if the first element is worth a higher portion of the value of sequence S, then Lemma 2 leads to a better approximation ratio. For example, if $c = 0.5$ (resp., $c = 0.8$), then we have $h(S) \geq 0.76h(S^*(\mathcal{V}, k, 0))$ (resp., $h(S) \geq 0.88h(S^*(\mathcal{V}, k, 0))$). In particular, if $c = 1$, then we have $h(S) \geq h(S^*(\mathcal{V}, k, 0))$, which implies that the SSG algorithm yields an optimal solution to Problem (P). We state this special case in Corollary 1, which may be of independent interest.*

Corollary 1. *Suppose that the sequence selected by the SSG algorithm is S with $|S| = k$ and that there exists a sequence S_1 such that $S_1 \preceq S$ and $h(S_1) = h(S)$. Then, under Assumption 1, we have $h(S) = h(S^*(\mathcal{V}, k, 0))$, and thus, sequence S is an optimal solution to Problem (P).*

Remark. *Lemmas 1 and 2 and Corollary 1 can be proven in a similar manner under a weaker assumption that function h is forward-monotone, backward-monotone, and element-sequence-submodular.*

In the following, we compare the impact of removing a certain number of elements from a selected sequence with that of removing the same number of elements from the ground set $\mathcal{V}$ before the selection takes place. We state the result in Lemma 3.

Lemma 3. *Consider $1 \leq \tau \leq k$. Suppose that function h is forward-monotone. The following holds for any $\mathcal{V}' \subseteq \mathcal{V}$ with $|\mathcal{V}'| \leq \tau$:*

$$g_\tau(S^*(\mathcal{V}, k, \tau)) \leq h(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau, 0)). \tag{13}$$

Proof. Since function h is forward-monotone, we can assume $|\mathcal{Z}_\tau(S)| = \tau$ for any sequence S and $\tau \leq |S|$. The reason is the following. Suppose $|\mathcal{Z}_\tau(S)| < \tau$. Then, we can continue to remove more elements from the end of sequence S till $|\mathcal{Z}_\tau(S)| = \tau$, which does not increase the value of the remaining sequence due to the forward monotonicity of function h.

Let $\mathcal{U} \triangleq \mathcal{V}(S^*(\mathcal{V}, k, \tau)) \cap \mathcal{V}'$ and $\tau' = |\mathcal{U}|$. Then, we have $\mathcal{U} \subseteq \mathcal{V}(S^*(\mathcal{V}, k, \tau))$. A little thoughts give $\mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U}) \subseteq \mathcal{V}(S^*(\mathcal{V}, k, \tau))$ and $|\mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U})| = \tau$. This implies that set $\mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U})$ is a feasible solution to Problem (11) with respect to sequence $S^*(\mathcal{V}, k, \tau)$, which further implies the following:

$$h(S^*(\mathcal{V}, k, \tau) - \mathcal{Z}_\tau(S^*(\mathcal{V}, k, \tau))) \leq h(S^*(\mathcal{V}, k, \tau) - \mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U})). \tag{14}$$

Also, from the definition of $\mathcal{U}$, we have $S^*(\mathcal{V}, k, \tau) - \mathcal{U} = S^*(\mathcal{V}, k, \tau) - \mathcal{V}'$, and thus, $\mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U}) = \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{V}')$. Again, from the definition of $\mathcal{U}$, we have that the elements in $\mathcal{V}' \setminus \mathcal{U}$ are not in sequence $S^*(\mathcal{V}, k, \tau)$. Then, we have the following:

$$h(S^*(\mathcal{V}, k, \tau) - \mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U})) = h(S^*(\mathcal{V}, k, \tau) - \mathcal{V}' \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{V}')). \tag{15}$$

Note that sequence $S^*(\mathcal{V}, k, \tau) - \mathcal{V}'$ does not contain any elements in $\mathcal{V}'$ and has $k - \tau'$ elements. Hence, sequence $S^*(\mathcal{V}, k, \tau) - \mathcal{V}'$ is a feasible solution to Problem (R) (with respect to $\mathcal{V} \setminus \mathcal{V}'$, $k - \tau'$, and $\tau - \tau'$). This implies the following:

$$g_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{V}') \leq g_{\tau-\tau'}(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau', \tau - \tau')). \tag{16}$$

Also, by replacing $\mathcal{V}$, k, and τ in Eq. (12) with $\mathcal{V} \setminus \mathcal{V}'$, $k - \tau'$, and $\tau - \tau'$, respectively, we immediately obtain the following:

$$g_{\tau-\tau'}(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau', \tau - \tau')) \leq h(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau, 0)). \tag{17}$$

By combining Eqs. (14)-(17), we have the following:

$$\begin{aligned}
g_\tau(S^*(\mathcal{V}, k, \tau)) &\overset{\text{(a)}}{=} h(S^*(\mathcal{V}, k, \tau) - \mathcal{Z}_\tau(S^*(\mathcal{V}, k, \tau))) \\
&\overset{\text{(b)}}{\leq} h(S^*(\mathcal{V}, k, \tau) - \mathcal{U} \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{U})) \\
&\overset{\text{(c)}}{=} h(S^*(\mathcal{V}, k, \tau) - \mathcal{V}' \cup \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{V}')) \\
&\overset{\text{(d)}}{=} h((S^*(\mathcal{V}, k, \tau) - \mathcal{V}') - \mathcal{Z}_{\tau-\tau'}(S^*(\mathcal{V}, k, \tau) - \mathcal{V}')) \\
&\overset{\text{(e)}}{=} g_{\tau-\tau'}((S^*(\mathcal{V}, k, \tau) - \mathcal{V}')) \\
&\overset{\text{(f)}}{\leq} g_{\tau-\tau'}(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau', \tau - \tau')) \\
&\overset{\text{(g)}}{\leq} h(S^*(\mathcal{V} \setminus \mathcal{V}', k - \tau, 0)),
\end{aligned} \tag{18}$$

where (a) is from the definition of function g_τ, (b) is from Eq. (14), (c) is from Eq. (15), (d) is a rewriting, (e) is from the definition of function $g_{\tau-\tau'}$, (f) is from Eq. (16), and (g) is from Eq. (17). □

8.4 Proof of Lemma 2

Before we prove Lemma 2, we introduce two lemmas: Lemmas 4 and 5. Lemma 4 is borrowed from [9] and will be used in the proof of Lemma 5, which will be used in the proof of Lemma 2.

Lemma 4 (Lemma 10 of [9])**.** *Suppose that function h is sequence-submodular. For any sequences $S_1', S_2' \in \mathcal{H}$, there exists an element $v \in \mathcal{V}$ such that $h((v)|S_1') \geq \frac{1}{|S_2'|} h(S_2'|S_1')$.*

Lemma 5. *Consider $1 \leq k' \leq k$. Suppose that the sequence selected by the SSG algorithm is S with $|S| = k$ and that there exist sequences S_1 and S_2 such that sequence S can be written as $S = S_1 \oplus S_2$ with $|S_1| = k - k'$ and $|S_2| = k'$. Then, under Assumption 1, we have $h(S_2|S_1) \geq (1 - 1/e^{\frac{k'}{k}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1))$.*

Proof. Let v_2^i denote the i-th element of sequence S_2, and let $S_2^i \triangleq (v_2^1, \ldots, v_2^i)$ denote the sequence consisting of the first i elements of sequence S_2. Since function h is forward-monotone, we can assume that $|S^*(\mathcal{V}, k, 0)| = k$ as adding more elements to the end of a sequence does not reduce its overall value.

Due to Lemma 4, there exists some element $v' \in \mathcal{V}$ such that $h((v')|S_1 \oplus S_2^{i-1}) \geq \frac{1}{k} h(S^*(\mathcal{V}, k, 0)|S_1 \oplus S_2^{i-1})$. Then, we have the following:

$$\begin{aligned} h(S_1 \oplus S_2^i) - h(S_1 \oplus S_2^{i-1}) &= h((v_2^i)|S_1 \oplus S_2^{i-1}) \\ &\overset{\text{(a)}}{\geq} h((v')|S_1 \oplus S_2^{i-1}) \\ &\overset{\text{(b)}}{\geq} \frac{1}{k} h(S^*(\mathcal{V}, k, 0)|S_1 \oplus S_2^{i-1}) \\ &= \frac{1}{k}(h(S_1 \oplus S_2^{i-1} \oplus S^*(\mathcal{V}, k, 0)) - h(S_1 \oplus S_2^{i-1})) \\ &\overset{\text{(c)}}{\geq} \frac{1}{k}(h(S^*(\mathcal{V}, k, 0)) - h(S_1 \oplus S_2^{i-1})), \end{aligned} \tag{19}$$

where (a) is due to the greedy manner of the SSG algorithm (Line 4 of Algorithm 3), (b) is from the property of element v' (due to Lemma 4), and (c) is due to the backward monotonicity of function h. Rewriting Eq. (19) yields the following equivalent inequality:

$$h(S_1 \oplus S_2^i) \geq \frac{1}{k} h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{1}{k}) h(S_1 \oplus S_2^{i-1}). \tag{20}$$

By writing Eq. (20) for $i \in \{1, \ldots, k'\}$ and combining them, we obtain the following:

$$\begin{aligned} h(S_1 \oplus S_2^{k'}) &\geq \sum_{j=0}^{k'-1} \frac{1}{k}(1 - \frac{1}{k})^j \; h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{1}{k})^{k'} h(S_1) \\ &= (1 - (1 - \frac{1}{k})^{k'}) h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{1}{k})^{k'} h(S_1). \end{aligned} \tag{21}$$

Applying Eq. (21) and the fact that $S_2 = S_2^{k'}$ yields the following:

$$\begin{aligned} h(S_2|S_1) &= h(S_1 \oplus S_2^{k'}) - h(S_1) \\ &\geq (1 - (1 - \frac{1}{k})^{k'}) h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{1}{k})^{k'} h(S_1) - h(S_1) \\ &= (1 - (1 - \frac{1}{k})^{k'}) h(S^*(\mathcal{V}, k, 0)) - (1 - (1 - \frac{1}{k})^{k'}) h(S_1) \\ &= (1 - (1 - \frac{1}{k})^{k'})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)) \\ &\geq (1 - 1/e^{\frac{k'}{k}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)), \end{aligned} \tag{22}$$

where the last inequality holds because $(1 - \frac{1}{k}) \leq e^{-\frac{1}{k}}$ and $h(S^*(\mathcal{V}, k, 0)) - h(S_1)$ is nonnegative. This completes the proof. □

Remark. *We note a subtle yet critical difference between the monotonicity properties of set functions and sequence functions, which complicates the proof of Lemma 5. Consider a set function $r(\cdot)$. We define $r(\mathcal{V}_2|\mathcal{V}_1) \triangleq r(\mathcal{V}_1 \cup \mathcal{V}_2) - r(\mathcal{V}_1)$ as the marginal value of adding set $\mathcal{V}_2 \subseteq \mathcal{V}$ to another set $\mathcal{V}_1 \subseteq \mathcal{V}$. It is easy to see that the monotonicity of function $r(\cdot)$ implies the monotonicity of function $r(\cdot|\mathcal{V}_1)$ for any given $\mathcal{V}_1 \subseteq \mathcal{V}$. Such an analogy also exists for forward monotonicity of sequence functions, i.e., the forward monotonicity of a sequence function $h(\cdot)$ implies the forward monotonicity of function $h(\cdot|S_1)$ for any given $S_1 \in \mathcal{H}(\mathcal{V})$. However, a similar property does not hold for backward monotonicity. Due to lack of such a monotonicity property, the proof of Lemma 5 becomes more involved and requires more careful derivations.*

Having introduced Lemmas 4 and 5, we are now ready to prove Lemma 2.

Proof of Lemma 2. Suppose $h(S) = \delta \cdot h(S^*(\mathcal{V}, k, 0))$ for some $\delta \in (0, 1]$. Then, we have

$$\begin{aligned}
\delta \cdot h(S^*(\mathcal{V}, k, 0)) &= h(S) \\
&= h(S_1) + h(S_2|S_1) \\
&\overset{\text{(a)}}{\geq} h(S_1) + (1 - 1/e^{\frac{k'}{k}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)) \\
&= (1/e^{\frac{k'}{k}})h(S_1) + (1 - 1/e^{\frac{k'}{k}})h(S^*(\mathcal{V}, k, 0)) \\
&\overset{\text{(b)}}{\geq} (c \cdot \delta/e^{\frac{k'}{k}})h(S^*(\mathcal{V}, k, 0)) + (1 - 1/e^{\frac{k'}{k}})h(S^*(\mathcal{V}, k, 0)),
\end{aligned} \tag{23}$$

where (a) follows from Lemma 5 and (b) holds because $h(S_1) \geq c \cdot h(S) = c \cdot \delta \cdot h(S^*(\mathcal{V}, k, 0))$. Dividing both sides of Eq. (23) by $h(S^*(\mathcal{V}, k, 0))$ yields the following:

$$\delta \geq (c \cdot \delta/e^{\frac{k'}{k}}) + (1 - 1/e^{\frac{k'}{k}}), \tag{24}$$

which implies

$$\delta \geq \frac{e^{\frac{k'}{k}} - 1}{e^{\frac{k'}{k}} - c}. \tag{25}$$

The above equation, along with $h(S) = \delta \cdot h(S^*(\mathcal{V}, k, 0))$, implies $h(S) \geq \frac{e^{\frac{k'}{k}} - 1}{e^{\frac{k'}{k}} - c} h(S^*(\mathcal{V}, k, 0))$. This completes the proof. □

8.5 Proof of Theorem 1

Proof. Suppose that function h is forward-monotone, backward-monotone, and sequence-submodular (Assumption 1). We use Lemmas 1, 2, and 3 presented in Appendix 8.3 to prove that Algorithm 1 achieves an approximation ratio of $\max\left\{\frac{e-1}{2e}, \frac{e^{\frac{k-2}{k-1}} - 1}{2e^{\frac{k-2}{k-1}} - 1}\right\}$ in the case of $\tau = 1$.

Given $\tau = 1$, in Step 1 of Algorithm 1, the selected sequence S_1 consists of one element only; this element is denoted by v_1, i.e., $S_1 = (v_1)$. In Step 2 of Algorithm 1, it is equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \{v_1\}$, and we have $|S_2| = k - \tau = k - 1$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2 = (v_1) \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). Note that only one element will be removed from S, i.e., $|\mathcal{Z}_\tau(S)| = 1$. For ease of notation, we use z to denote the only element in $\mathcal{Z}_\tau(S)$, i.e., $\mathcal{Z}_\tau(S) = \{z\}$.

We want to show the following two bounds, which establish the approximation ratio of Algorithm 1:

$$h(S - \{z\}) \geq \frac{e - 1}{2e} g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{26a}$$

$$h(S - \{z\}) \geq \frac{e^{\frac{k-2}{k-1}} - 1}{2e^{\frac{k-2}{k-1}} - 1} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{26b}$$

To begin with, we present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned}
h(S_2) &\geq (1 - 1/e)h(S^*(\mathcal{V} \setminus \{v_1\}, k - \tau, 0)) \\
&\geq (1 - 1/e)g_\tau(S^*(\mathcal{V}, k, \tau)),
\end{aligned} \tag{27}$$

where the first inequality is from Lemma 1 (where we replace $\mathcal{V}$ with $\mathcal{V} \setminus \{v_1\}$ and both k and i with $k - \tau$) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\{v_1\}$).

The proof proceeds as follows. Element z is an element that will be removed, which can be either v_1 or an element in S_2. Therefore, we consider two cases: (I) $z = v_1$ and (II) $z \neq v_1$.

In Case I, we have $z = v_1$, which implies the following:

$$h(S - \{z\}) = h(S_2) \geq (1 - 1/e)g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{28}$$

where the inequality follows from Eq. (27).

In Case II, we have $z \neq v_1$ (or $z \in \mathcal{V}(S_2)$). Depending on the impact of removing element z, we consider two subcases: (II-a) $h(S_2) \leq h(S_2 - \{z\})$ and (II-b) $h(S_2) > h(S_2 - \{z\})$.

In Case II-a: we have $h(S_2) \leq h(S_2 - \{z\})$. In this case, the removal of element z does not reduce the overall value of the remaining sequence $S_2 - \{z\}$. Then, we have

$$h(S - \{z\}) = h((v_1) \oplus (S_2 - \{z\})) \overset{\text{(a)}}{\geq} h(S_2 - \{z\}) \overset{\text{(b)}}{\geq} h(S_2) \overset{\text{(c)}}{\geq} (1 - 1/e) g_\tau(S^*(\mathcal{V}, k, \tau)), \quad (29)$$

where (a) is due to the backward monotonicity of function h, (b) holds from the condition of this subcase, and (c) follows from Eq. (27).

In Case II-b: we have $h(S_2) > h(S_2 - \{z\})$. Suppose $k = 2$. Then, it is trivial that the sequence selected by Algorithm 1 (i.e., $S = (v_1) \oplus (z)$) yields an optimal solution. This is because removing element z from S gives (v_1), which has the largest individual value among all the elements. Therefore, we assume $k > 2$ throughout the rest of the proof. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \{z\})}{h(S_2)}$ denote the ratio of the loss of removing element z from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$ due to $h(S_2) > h(S_2 - \{z\})$. We first state the following:

$$h(S - \{z\}) \geq \max\{\eta \cdot h(S_2), (1 - \eta) \cdot h(S_2)\}, \quad (30a)$$

$$\max\{\eta, (1 - \eta)\} \geq \frac{1}{2}, \quad (30b)$$

$$h(S_2) \geq \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta} g_\tau(S^*(\mathcal{V}, k, \tau)), \quad (30c)$$

$$\max\left\{\eta \cdot \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta}, (1 - \eta) \cdot \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta}\right\} \geq \frac{e^{\frac{k-2}{k-1}} - 1}{2(e^{\frac{k-2}{k-1}} - \frac{1}{2})}. \quad (30d)$$

We will prove Eqs. (30a)-(30d) later; for now, we assume that they all hold. Then, we can obtain the following bound:

$$\begin{aligned} h(S - \{z\}) &\geq \max\{\eta \cdot h(S_2), (1 - \eta) \cdot h(S_2)\} \\ &\geq \max\{\eta, 1 - \eta\} \cdot (1 - 1/e) g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{e - 1}{2e} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \quad (31)$$

where the three inequalities are from Eqs. (30a), (27), and (30b), respectively.

Similarly, we can also obtain the following bound:

$$\begin{aligned} h(S - \{z\}) &\geq \max\{\eta \cdot h(S_2), (1 - \eta) \cdot h(S_2)\} \\ &\geq \max\left\{\frac{\eta(e^{\frac{k-2}{k-1}} - 1)}{e^{\frac{k-2}{k-1}} - \eta}, \frac{(1 - \eta)(e^{\frac{k-2}{k-1}} - 1)}{e^{\frac{k-2}{k-1}} - \eta}\right\} g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{e^{\frac{k-2}{k-1}} - 1}{2(e^{\frac{k-2}{k-1}} - \frac{1}{2})} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \quad (32)$$

where the three inequalities are from Eqs. (30a), (30c), and (30d), respectively.

Combining all the cases establishes an approximation ratio of Algorithm 1 and completes the proof. Specifically, combining the bounds in Eqs. (28), (29), and (31) (resp., (32)) yields the bound in Eq. (26a) (resp., (26b)).

Now, it remains to show that Eqs. (30a)-(30d) hold in Case II-b, where we have $z \in \mathcal{V}(S_2)$ and $h(S_2) > h(S_2 - \{z\})$. We first rewrite S_2 as $S_2 = S_2^1 \oplus (z) \oplus S_2^2$, where S_2^1 and S_2^2 denote the subsequences of S_2 before and after element z, respectively. Note that S_2^1 or S_2^2 could be an empty sequence, depending on the position of z in S_2. Then, we characterize $h((z))$ in terms of $h(S_2)$:

$$\begin{aligned}\eta \cdot h(S_2) &\overset{\text{(a)}}{=} h(S_2) - h(S_2 - \{z\}) \\ &= h(S_2^1 \oplus (z) \oplus S_2^2) - h(S_2^1 \oplus S_2^2) \\ &= h(S_2^1) + h((z)|S_2^1) + h(S_2^2|S_2^1 \oplus (z)) - h(S_2^1) - h(S_2^2|S_2^1) \\ &= h((z)|S_2^1) + h(S_2^2|S_2^1 \oplus (z)) - h(S_2^2|S_2^1) \\ &\leq h((z)|S_2^1) \\ &\leq h((z)),\end{aligned} \tag{33}$$

where (a) is from the definition of η and the two inequalities are due to the sequence submodularity of function h. We are now ready to prove Eqs. (30a)-(30d).

To prove Eq. (30a), we decompose it into two parts: (i) $h(S - \{z\}) \geq \eta \cdot h(S_2)$ and (ii) $h(S - \{z\}) \geq (1 - \eta) \cdot h(S_2)$.

Part (i) can be shown through the following:

$$h(S - \{z\}) \overset{\text{(a)}}{\geq} h((v_1)) \overset{\text{(b)}}{\geq} h((z)) \overset{\text{(c)}}{\geq} \eta \cdot h(S_2), \tag{34}$$

where (a) is form the forward monotonicity of function h, (b) is due to the greedy manner of Algorithm 1 (Lines 3-5), and (c) is from Eq. (33).

Part (ii) can be shown through the following:

$$h(S - \{z\}) = h((v_1) \oplus (S_2 - \{z\})) \overset{\text{(a)}}{\geq} h(S_2 - \{z\}) \overset{\text{(b)}}{=} (1 - \eta) \cdot h(S_2),$$

where (a) is from the backward monotonicity of function h and (b) is from the definition of η.

Eq. (30b) holds trivially for any $\eta \in (0, 1]$ by setting η and $1 - \eta$ to be equal and solving for η.

Next, we show that Eq. (30c) holds. Let v_2^1 denote the first element of sequence S_2. Then, we have the following:

$$h((v_2^1)) \overset{\text{(a)}}{\geq} h((z)) \overset{\text{(b)}}{\geq} \eta \cdot h(S_2),$$

where (a) holds because element v_2^1 has the largest individual value among all elements in S_2 and (b) follows from Eq. (33). Then, we can characterize the value of $h(S_2)$ as follows:

$$\begin{aligned}h(S_2) &\geq \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta} h(S^*(\mathcal{V} \setminus \{v_1\}, k - \tau, 0)) \\ &\geq \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta} g_\tau(S^*(\mathcal{V}, k, \tau)),\end{aligned}$$

where the first inequality is from Lemma 2 (where we replace $\mathcal{V}$, S, S_1, k, k', and c with $\mathcal{V} \setminus \{v_1\}$, S_2, (v_2^1), $k - 1$, $k - 2$, and η, respectively) and the second inequality is from Lemma 3.

Finally, we show that Eq. (30d) holds. We define two functions of η: $l_1(\eta) \triangleq \eta \cdot \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta}$ and $l_2(\eta) \triangleq (1 - \eta) \cdot \frac{e^{\frac{k-2}{k-1}} - 1}{e^{\frac{k-2}{k-1}} - \eta}$. It is easy to verify that for $k > 2$ and $\eta \in (0, 1]$, function $l_1(\eta)$ is monotonically increasing and function $l_2(\eta)$ is monotonically decreasing. Also, we have $l_1(\frac{1}{2}) = l_2(\frac{1}{2}) = \frac{e^{\frac{k-2}{k-1}} - 1}{2(e^{\frac{k-2}{k-1}} - \frac{1}{2})}$. We consider two cases for η: $\eta \in [\frac{1}{2}, 1]$ and $\eta \in (0, \frac{1}{2}]$. For $\eta \in [\frac{1}{2}, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta) \geq l_1(\frac{1}{2})$ as $l_1(\eta)$ is monotonically increasing; for $\eta \in (0, \frac{1}{2}]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_2(\eta) \geq l_2(\frac{1}{2}) = l_1(\frac{1}{2})$ as $l_2(\eta)$ is monotonically decreasing. Therefore, for $\eta \in (0, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\frac{1}{2}) = \frac{e^{\frac{k-2}{k-1}} - 1}{2(e^{\frac{k-2}{k-1}} - \frac{1}{2})}$. This gives Eq. (30d) and completes the proof. □

8.6 Proof of Theorem 2

Proof. Suppose that function h is forward-monotone, backward-monotone, and sequence-submodular (Assumption 1). We use Lemmas 1, 2, and 3 presented in Appendix 8.3 to prove that Algorithm 1 achieves an approximation ratio of $\max\left\{\frac{(e-1)^2}{e(2e-1)}, \frac{(e-1)(e^{\frac{k-2\tau}{k-\tau}}-1)}{(2e-1)e^{\frac{k-2\tau}{k-\tau}}-(e-1)}\right\}$ in the case of $1 \leq \tau \leq k$, assuming the removal of τ contiguous elements.

In Step 1 of Algorithm 1, it is equivalent that sequence S_1 is selected by the SSG algorithm from set $\mathcal{V}$, and we have $|S_1| = \tau$. Similarly, in Step 2 of Algorithm 1, it is also equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \mathcal{V}(S_1)$, and we have $|S_2| = k - \tau$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). We define $\mathcal{Z}_\tau^1(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_1)$ and $\mathcal{Z}_\tau^2(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_2)$ as the set of elements removed from subsequences S_1 and S_2, respectively.

The proof of Theorem 2 follows a similar line of analysis as in the proof of Theorem 1 for the case of $\tau = 1$. Specifically, we will also consider three cases: (I) $\mathcal{Z}_\tau^2(S) = \emptyset$, (II-a) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$, and (II-b) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. The proofs of Case I and Case II-a are almost the same as those in Theorem 1, except for some minor technical differences. However, the proof of Case II-b is substantially different. The reason is the following. In the proof of Theorem 1, only one element can be removed from S, and in Case II-b, this element has to be in $\mathcal{V}(S_2)$, which makes it easier to characterize the impact of removing such an element. In the case of $1 \leq \tau \leq k$, however, more than one element may be removed, which could be in either $\mathcal{V}(S_1)$ or $\mathcal{V}(S_2)$ or both. Therefore, we present a different approach to address this new technical challenge.

We want to show the following two bounds that establish the approximation ratio of Algorithm 1:

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{(e-1)^2}{e(2e-1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{35a}$$

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{(e-1)(e^{\frac{k-2\tau}{k-\tau}}-1)}{(2e-1)e^{\frac{k-2\tau}{k-\tau}}-(e-1)} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{35b}$$

To begin with, we present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned} h(S_2) &\geq (1-1/e)h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k-\tau, 0)) \\ &\geq (1-1/e)g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{36}$$

where the first inequality is from Lemma 1 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, $k - \tau$, and $k - \tau$, respectively) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\mathcal{V}(S_1)$).

The proof proceeds as follows. Elements in $\mathcal{Z}_\tau(S)$ will be removed from sequence S. These elements can be either fully or partially in $\mathcal{V}(S_1)$ (i.e., $\mathcal{Z}_\tau^2(S) = \emptyset$ or $\mathcal{Z}_\tau^2(S) \neq \emptyset$). Therefore, we consider two cases: (I) $\mathcal{Z}_\tau^2(S) = \emptyset$ and (II) $\mathcal{Z}_\tau^2(S) \neq \emptyset$.

In Case I, we have $\mathcal{Z}_\tau^2(S) = \emptyset$, i.e., $\mathcal{Z}_\tau(S) = \mathcal{V}(S_1)$. Then, we have the following:

$$h(S - \mathcal{Z}_\tau(S)) = h(S_2) \geq (1-1/e)g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{37}$$

where the inequality follows from Eq. (36).

In Case II, we have $\mathcal{Z}_\tau^2(S) \neq \emptyset$. Depending on the impact of removing elements in $\mathcal{Z}_\tau^2(S)$, we consider two subcases: (II-a) $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$ and (II-b) $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$.

In Case II-a: we have $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$. In this case, the removal of elements in $\mathcal{Z}_\tau^2(S)$ does not reduce the overall value of the remaining sequence $S_2 - \mathcal{Z}_\tau^2(S)$. Then, we have

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &= h((S_1 - \mathcal{Z}_\tau^1(S)) \oplus (S_2 - \mathcal{Z}_\tau^2(S))) \\ &\overset{\text{(a)}}{\geq} h(S_2 - \mathcal{Z}_\tau^2(S)) \\ &\overset{\text{(b)}}{\geq} h(S_2) \\ &\overset{\text{(c)}}{\geq} (1-1/e)g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{38}$$

where (a) is due to the backward monotonicity of function h, (b) holds from the condition of this subcase, and (c) follows from Eq. (36).

In Case II-b: we have $h(S_2) > h(S_2 - \mathcal{Z}^2_\tau(S))$. Let $\tau_1 \triangleq |\mathcal{Z}^1_\tau(S)|$ and $\tau_2 \triangleq |\mathcal{Z}^2_\tau(S)|$. Then, we have $\tau = \tau_1 + \tau_2$ and $k = |S| = |S_1| + |S_2| \geq \tau + \tau_2$. We consider two cases: $k = \tau + \tau_2$ and $k > \tau + \tau_2$.

Suppose $k = \tau + \tau_2$. Then, it implies $\mathcal{Z}^2_\tau(S) = \mathcal{V}(S_2)$, i.e., all the elements in S_2 are removed. This further implies that the elements in $\mathcal{Z}^1_\tau(S)$ are at the end of sequence S_1 (due to the contiguous assumption of elements in $\mathcal{Z}_\tau(S)$). Let $S_1^{\tau_2} \preceq S_1$ denote the subsequence consisting of the first τ_2 elements in S_1. It is easy to see $S_1 - \mathcal{Z}^1_\tau(S) = S_1^{\tau_2}$. Then, we have the following:

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &= h((S_1 - \mathcal{Z}^1_\tau(S)) \oplus (S_2 - \mathcal{Z}^2_\tau(S))) \\ &= h(S_1 - \mathcal{Z}^1_\tau(S)) \\ &= h(S_1^{\tau_2}) \\ &\geq (1 - 1/e) h(S^*(\mathcal{V}, \tau_2, 0)) \\ &\geq (1 - 1/e) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{39}$$

where the first inequality is from Lemma 1 (where we replace both k and i with τ_2) and the second inequality is due to $\tau_2 = k - \tau$ and Lemma 3 (where $\mathcal{V}'$ is an empty set).

Now, suppose $k > \tau + \tau_2$. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \mathcal{Z}^2_\tau(S))}{h(S_2)}$ denote the ratio of the loss of removing elements in $\mathcal{Z}^2_\tau(S)$ from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$ due to $h(S_2) > h(S_2 - \mathcal{Z}^2_\tau(S))$. We first state the following:

$$h(S - \mathcal{Z}_\tau(S)) \geq \max\{\eta(1 - 1/e)h(S_2), (1 - \eta)h(S_2)\}, \tag{40a}$$

$$\max\{\eta(1 - 1/e), (1 - \eta)\} \geq \frac{e - 1}{2e - 1}, \tag{40b}$$

$$h(S_2) \geq \frac{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1 - 1/e)} g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{40c}$$

$$\max\left\{ \frac{\eta(1 - 1/e)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1 - 1/e)}, \frac{(1 - \eta)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1 - 1/e)} \right\} \geq \frac{(e - 1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{(2e - 1)e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - (e - 1)}. \tag{40d}$$

We will prove Eqs. (40a)-(40d) later; for now, we assume that they all hold. Then, we can obtain the following bound:

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &\geq \max\{\eta(1 - 1/e)h(S_2), (1 - \eta)h(S_2)\} \\ &\geq \max\{\eta(1 - 1/e), (1 - \eta)\} \cdot (1 - 1/e) g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{(e - 1)^2}{e(2e - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{41}$$

where the three inequalities are from Eqs. (40a), (36), and (40b), respectively.

Similarly, we can also obtain the following bound:

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &\geq \max\{\eta(1 - 1/e)h(S_2), (1 - \eta)h(S_2)\} \\ &\geq \max\left\{ \frac{\eta(e - 1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{e(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1 - 1/e))}, \frac{(1 - \eta)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1 - 1/e)} \right\} g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{(e - 1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{(2e - 1)e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - (e - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{(e - 1)(e^{\frac{k-2\tau}{k-\tau}} - 1)}{(2e - 1)e^{\frac{k-2\tau}{k-\tau}} - (e - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{42}$$

where the first three inequalities are from Eqs. (40a), (40c), and (40d), respectively, and the last inequality is due to $\tau_2 \leq \tau$ and that $\frac{(e-1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-1)}{(2e-1)e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-(e-1)}$ is a decreasing function of τ_2.

Combining all the cases establishes the approximation ratios of Algorithm 1 and completes the proof. Specifically, combining the bounds in Eqs. (37), (38), (39), and (41) (resp., (42)) yields the bound in Eq. (35a) (resp., (35b)).

Now, it remains to show that Eqs. (40a)-(40d) hold in Case II-b, where we have $\mathcal{Z}^2_\tau(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}^2_\tau(S))$. Recall that the τ elements in $\mathcal{Z}_\tau(S)$ form a contiguous subsequence of S. Then, the elements in $\mathcal{Z}^1_\tau(S)$ and $\mathcal{Z}^2_\tau(S)$ also form a contiguous subsequence of S_1 and S_2, respectively. We use Z_1 and Z_2 to denote the contiguous subsequence of elements in $\mathcal{Z}^1_\tau(S)$ and $\mathcal{Z}^2_\tau(S)$, respectively. We first rewrite S_2 as $S_2 = S_2^1 \oplus Z_2 \oplus S_2^2$, where S_2^1 and S_2^1 denote the subsequences of S_2 before and after subsequence Z_2, respectively. Note that S_2^1 or S_2^2 could be an empty sequence, depending on the position of subsequence Z_2 in S_2. Then, we characterize $h(Z_2)$ in terms of $h(S_2)$:

$$\begin{aligned}
\eta \cdot h(S_2) &\overset{\text{(a)}}{=} h(S_2) - h(S_2 - \mathcal{Z}^2_\tau(S)) \\
&= h(S_2^1 \oplus Z_2 \oplus S_2^2) - h(S_2^1 \oplus S_2^2) \\
&= h(S_2^1) + h(Z_2|S_2^1) + h(S_2^2|S_2^1 \oplus Z_2) - h(S_2^1) - h(S_2^2|S_2^1) \\
&= h(Z_2|S_2^1) + h(S_2^2|S_2^1 \oplus Z_2) - h(S_2^2|S_2^1) \\
&\leq h(Z_2|S_2^1) \\
&\leq h(Z_2),
\end{aligned} \tag{43}$$

where (a) is from the definition of η and the two inequalities are due the sequence submodularity of function h. We are now ready to prove Eqs. (40a)-(40d).

To prove Eq. (40a), we decompose it into two parts: (i) $h(S - \mathcal{Z}_\tau(S)) \geq \eta(1-1/e)h(S_2)$ and (ii) $h(S - \mathcal{Z}_\tau(S)) \geq (1-\eta)h(S_2)$.

Recall that $S_1^{\tau_2}$ denote the subsequence consisting of the first τ_2 elements in S_1. Then, Part (i) can be shown through the following:

$$\begin{aligned}
h(S - \mathcal{Z}_\tau(S)) &= h((S_1 - \mathcal{Z}^1_\tau(S)) \oplus (S_2 - \mathcal{Z}^2_\tau(S))) \\
&\overset{\text{(a)}}{\geq} h(S_1 - \mathcal{Z}^1_\tau(S)) \\
&= h(S_1^{\tau_2}) \\
&\overset{\text{(b)}}{\geq} (1-1/e)h(S^*(\mathcal{V}, \tau_2, 0)) \\
&\overset{\text{(c)}}{\geq} (1-1/e)h(Z_2) \\
&\overset{\text{(d)}}{\geq} \eta(1-1/e)h(S_2),
\end{aligned} \tag{44}$$

where (a) is due to the forward monotonicity of function h, (b) is due to the greedy manner of selecting subsequence $S_1^{\tau_2}$ from set $\mathcal{V}$ and Lemma 1 (where we replace both k and i with τ_2), (c) holds because sequence Z_2 is a feasible solution to Problem (P) for selecting a sequence of τ_2 elements from $\mathcal{V}$, and (d) is from Eq. (43).

Part (ii) can be shown through the following:

$$\begin{aligned}
h(S - \mathcal{Z}_\tau(S)) &= h((S_1 - \mathcal{Z}^1_\tau(S)) \oplus (S_2 - \mathcal{Z}^2_\tau(S))) \\
&\overset{\text{(a)}}{\geq} h(S_2 - \mathcal{Z}^2_\tau(S)) \\
&\overset{\text{(b)}}{=} (1-\eta)h(S_2),
\end{aligned} \tag{45}$$

where (a) is from the backward monotonicity of function h and (b) is from the definition of η.

Eq. (40b) holds trivially for any $\eta \in (0,1]$ by setting $\eta(1-1/e)$ and $1-\eta$ to be equal, solving for η, and plugging it back.

Next, we show that Eq. (40c) holds. Let $S_2^{\tau_2} \preceq S_2$ denote the subsequence consisting of the first τ_2 elements of sequence S_2. Then, we have the following:

$$\begin{aligned} h(S_2^{\tau_2}) &\overset{\text{(a)}}{\geq} (1-1/e)h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), \tau_2, 0)) \\ &\overset{\text{(b)}}{\geq} (1-1/e)h(Z_2) \\ &\overset{\text{(c)}}{\geq} \eta(1-1/e)h(S_2), \end{aligned} \tag{46}$$

where (a) is due to the greedy manner of selecting subsequence $S_2^{\tau_2}$ from set $\mathcal{V} \setminus \mathcal{V}(S_1)$ and Lemma 1 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, τ_2, and τ_2, respectively), (b) holds because sequence Z_2 is a feasible solution to Problem (P) for selecting a sequence of τ_2 elements from $\mathcal{V} = \mathcal{V} \setminus \mathcal{V}(S_1)$, and (c) is from Eq. (43). Therefore, we can characterize the value of $h(S_2)$ as follows:

$$\begin{aligned} h(S_2) &\geq \frac{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1-1/e)} h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k-\tau, 0)) \\ &\geq \frac{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1-1/e)} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{47}$$

where the first inequality is from Lemma 2 (where we replace $\mathcal{V}$, S, S_1, k, k', and c with $\mathcal{V} \setminus \mathcal{V}(S_1)$, S_2, $S_2^{\tau_2}$, $k-\tau$, $k-\tau-\tau_2$, and $\eta(1-1/e)$, respectively), and the second inequality is from Lemma 3.

Finally, we show that Eq. (40d) holds. We define two functions of η:

$$l_1(\eta) = \eta(1-1/e) \cdot \frac{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1-1/e)} \text{ and } l_2(\eta) = (1-\eta) \cdot \frac{(e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - 1)}{e^{\frac{k-\tau-\tau_2}{k-\tau_2}} - \eta(1-1/e)}.$$

It is easy to verify that for $k > \tau + \tau_2$ and $\eta \in (0, 1]$, function $l_1(\eta)$ is monotonically increasing and function $l_2(\eta)$ is monotonically decreasing. Also, we have $l_1(\frac{e}{2e-1}) = l_2(\frac{e}{2e-1}) = \frac{(e-1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-1)}{(2e-1)e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-(e-1)}$. We consider two cases for η: $\eta \in [\frac{e}{2e-1}, 1]$ and $\eta \in (0, \frac{e}{2e-1}]$. For $\eta \in [\frac{e}{2e-1}, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta) \geq l_1(\frac{e}{2e-1})$ as $l_1(\eta)$ is monotonically increasing; for $\eta \in (0, \frac{e}{2e-1}]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_2(\eta) \geq l_2(\frac{e}{2e-1}) = l_1(\frac{e}{2e-1})$ as $l_2(\eta)$ is monotonically decreasing. Therefore, for $\eta \in (0, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\frac{e}{2e-1}) = \frac{(e-1)(e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-1)}{(2e-1)e^{\frac{k-\tau-\tau_2}{k-\tau_2}}-(e-1)}$. This gives Eq. (40d) and completes the proof. □

8.7 Proof of Theorem 3

Proof. Suppose that function h is forward-monotone, backward-monotone, and general-sequence-submodular (Assumption 2). We use Lemmas 1 and 3 presented in Appendix 8.3 to prove that Algorithm 2 achieves an approximation ratio of $\frac{1-1/e}{1+\tau}$ in the case of $1 \leq \tau \leq k$, assuming the removal of an arbitrary subset of τ selected elements, which are not necessarily contiguous.

In Step 1 of Algorithm 2, sequence S_1 is selected by choosing τ elements with the highest individual values in a greedy manner, and we have $|S_1| = \tau$. In Step 2 of Algorithm 1, it is equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \mathcal{V}(S_1)$, and we have $|S_2| = k - \tau$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). We define $\mathcal{Z}_\tau^1(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_1)$ and $\mathcal{Z}_\tau^2(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_2)$ as the set of elements removed from subsequences S_1 and S_2, respectively.

The proof of Theorem 3 follows a similar line of analysis as in the proof of Theorem 2. Specifically, we will also consider three cases: (I) $\mathcal{Z}_\tau^2(S) = \emptyset$, (II-a) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$, and (II-b) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. The proofs of Case I and Case II-a are identical to those in Theorem 2. Therefore, we focus on Case II-b, which requires a different proof strategy.

We want to show the following bound that establishes the approximation ratio of Algorithm 2:

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{1-1/e}{1+\tau} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{48}$$

To begin with, we present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned} h(S_2) &\geq (1-1/e) h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k-\tau, 0)) \\ &\geq (1-1/e) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{49}$$

where the first inequality is from Lemma 1 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, $k-\tau$, and $k-\tau$, respectively) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\mathcal{V}(S_1)$).

We now focus on Case II-b, where we have $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \mathcal{Z}_\tau^2(S))}{h(S_2)}$ denote the ratio of the loss of removing elements in $\mathcal{Z}_\tau^2(S)$ from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$ due to $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. We first state the following:

$$h(S - \mathcal{Z}_\tau(S)) \geq \max\{\frac{\eta}{\tau} \cdot h(S_2), (1-\eta) \cdot h(S_2)\}, \tag{50a}$$

$$\max\{\frac{\eta}{\tau}, (1-\eta)\} \geq \frac{1}{\tau+1}. \tag{50b}$$

We will prove Eqs. (50a) and (50b) later; for now, we assume that they both hold. Then, we can obtain the following:

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &\geq \max\{\frac{\eta}{\tau} \cdot h(S_2), (1-\eta) \cdot h(S_2)\} \\ &\geq \max\{\frac{\eta}{\tau}, (1-\eta)\} \cdot (1-1/e) g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{1-1/e}{\tau+1} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{51}$$

where the three inequalities are from Eqs. (50a), (49), and (50b), respectively.

Now, we show that Eqs. (50a) and (50b) hold. We start by characterizing the value of elements in $\mathcal{Z}_\tau^2(S)$. Let $\tau_2 \triangleq |\mathcal{Z}_\tau^2(S)|$, and let the elements in $\mathcal{Z}_\tau^2(S)$ be denoted by $z_1, z_2, \ldots, z_{\tau_2}$ according to their order in sequence S_2. Then, we can rewrite S_2 as $S_2 = S_2^1 \oplus (z_1) \oplus S_2^2 \oplus (z_2) \oplus \cdots \oplus S_2^{\tau_2+1}$, where S_2^i is the subsequence between elements z_{i-1} and z_i, for $i = 1, 2, \ldots, \tau_2 + 1$, and both z_0 and $z_{\tau+1}$ are an empty sequence. Note that subsequence S_2^i could be an empty sequence, for $i = 1, 2, \ldots, \tau_2 + 1$. We characterize the value of elements in $\mathcal{Z}_\tau^2(S)$ in the following:

$$\begin{aligned} \eta \cdot h(S_2) &\overset{\text{(a)}}{=} h(S_2) - h(S_2 - \mathcal{Z}_\tau^2(S)) \\ &= h(S_2^1) + h((z_1)|S_2^1) + h(S_2^2|S_2^1 \oplus (z_1)) + \ldots \\ &\quad + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) \\ &\quad - h(S_2^1) - h(S_2^2|S_2^1) - \cdots - h(S_2^{\tau_2+1}|S_2^1 \oplus \cdots \oplus S_2^{\tau_2}) \\ &= \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) + h(S_2^2|S_2^1 \oplus (z_1)) - h(S_2^2|S_2^1) \\ &\quad + \cdots + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) - h(S_2^{\tau_2+1}|S_2^1 \oplus \cdots \oplus S_2^{\tau_2}) \\ &\leq \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) \\ &\leq \sum_{i=1}^{\tau_2} h((z_i)), \end{aligned} \tag{52}$$

where (a) is from the definition of η and the two inequalities are due to the general sequence submodularity of function h.

To prove Eq. (50a), we decompose it into two parts: (i) $h(S - \mathcal{Z}_\tau(S)) \geq \frac{\eta}{\tau} \cdot h(S_2)$ and (ii) $h(S - \mathcal{Z}_\tau(S)) \geq (1-\eta) \cdot h(S_2)$.

Let v' denote the first element in $S_1 - \mathcal{Z}_\tau^1(S)$. Then, Part (i) can be shown through the following:

$$h(S - \mathcal{Z}_\tau(S)) \overset{\text{(a)}}{\geq} h((v')) \overset{\text{(b)}}{\geq} \frac{1}{\tau_2} \sum_{i=1}^{\tau_2} h((z_i)) \overset{\text{(c)}}{\geq} \frac{\eta}{\tau_2} \cdot h(S_2) \overset{\text{(d)}}{\geq} \frac{\eta}{\tau} \cdot h(S_2),$$

where (a) is due to the forward monotonicity of function h, (b) is due to $h(v') \geq h(z_i)$ for any $i = 1, 2, \ldots, \tau_2$, (c) is from Eq. (52), and (d) is due to $\tau_2 \leq \tau$.

Part (ii) can be shown through the following:

$$h(S - \mathcal{Z}_\tau(S)) = h((S_1 - \mathcal{Z}_\tau^1(S)) \oplus (S_2 - \mathcal{Z}_\tau^2(S))) \overset{\text{(a)}}{\geq} h(S_2 - \mathcal{Z}_\tau^2(S)) \overset{\text{(b)}}{=} (1-\eta) \cdot h(S_2),$$

where (a) is from the backward monotonicity of function h and (b) is from the definition of η.

Eq. (50b) holds trivially for any $\eta \in (0, 1]$ by setting $\frac{\eta}{\tau}$ and $1 - \eta$ to be equal, solving for η, and plugging it back. This completes the proof. □

8.8 Preliminary Results: Approximate Sequence Submodular Maximization

In this section, we generalize the results of Lemmas 1 and 2 under weaker assumptions based on approximate versions of sequence submodularity and backward monotonicity. Specifically, we introduce two weaker assumptions that will be used in this section.

Assumption 6. *Function h is forward-monotone, α-backward-monotone, and μ_1-element-sequence-submodular.*

Assumption 7. *Function h is forward-monotone, backward-monotone, and μ_1-element-sequence-submodular.*

We first generalize the approximation result of the SSG algorithm (Algorithm 3) under Assumption 6. Let sequence S with $|S| = k$ be the one selected by the SSG algorithm, let S^i be the sequence consisting of the first i elements of sequence S with $1 \leq i \leq k$. We borrow the result of [28] and restate a generalized version in Lemma 6. This is also a generalization of Lemma 1. While [28] assumes that function h is forward-monotone, α-backward-monotone, and element-sequence-submodular and considers the case of $i = k$ only, the generalization to the result in Lemma 6 is fairly straightforward. Therefore, we refer the interested reader to [28] for the proof.

Lemma 6 (Theorem 1 of [28])**.** *Consider $1 \leq i \leq k$. Under Assumption 6, we have $h(S^i) \geq \alpha(1 - \frac{1}{e^{\mu_1 \cdot \frac{i}{k}}})h(S^*(\mathcal{V}, k, 0))$.*

We also generalize the result of Lemma 2 and present the generalized result in Lemma 7. Note that the current proof techniques we use could only lead to the generalization under Assumption 7, which is slightly stronger than Assumption 6 (backward-monotone vs. α-backward-monotone).

Lemma 7. *Consider $c \in (0, 1]$ and $1 \leq k' \leq k$. Suppose that the sequence selected by the SSG algorithm is S with $|S| = k$ and that there exists a sequence S_1 with $|S_1| = k - k'$ such that $S_1 \preceq S$ and $h(S_1) \geq c \cdot h(S)$. Then, under Assumption 7, we have $h(S) \geq \frac{e^{\mu_1 \cdot \frac{k'}{k}} - 1}{e^{\mu_1 \cdot \frac{k'}{k}} - c} h(S^*(\mathcal{V}, k, 0))$.*

Before we prove Lemma 7, we introduce two lemmas: Lemmas 8 and 9. Lemma 8 is a generalization of Lemma 4 and will be used in the proof of Lemma 9, which will be used in the proof of Lemma 7.

Lemma 8. *Suppose that function h is μ_1-element-sequence-submodular. For any sequences $S_1', S_2' \in \mathcal{H}$, there exists an element $v \in \mathcal{V}$ such that $h((v)|S_1') \geq \frac{\mu_1}{|S_2'|} h(S_2'|S_1')$.*

Proof. Let $S_2' = (u_1, \ldots, u_{|S_2'|})$. We can rewrite $h(S_2'|S_1')$ as

$$h(S_2'|S_1') = \sum_{j=1}^{|S_2'|} h((u_j)|S_1' \oplus (u_1, \ldots, u_{j-1})), \tag{53}$$

where sequence $(u_1, \dots, u_{j-1})$ is an empty sequence when $j = 1$. Due to the Pigeonhole principle, there exists some $j' \in \{1, 2, \dots |S'|\}$ such that the following is satisfied:

$$\begin{aligned} h((u_{j'})|S_1' \oplus (u_1, \dots, u_{j'-1})) &\geq \frac{1}{|S_2'|} \sum_{j=1}^{|S_2'|} h((u_j)|S_1' \oplus (u_1, \dots, u_{j-1})) \\ &= \frac{1}{|S_2'|} h(S_2'|S_1'), \end{aligned} \tag{54}$$

where the equality is from Eq. (53). Furthermore, element $u_{j'}$ satisfies the following:

$$\begin{aligned} h((u_{j'})|S_1') &\overset{\text{(a)}}{\geq} \mu_1 \cdot h((u_{j'})|S_1' \oplus (u_1, \dots, u_{j'-1})) \\ &\overset{\text{(b)}}{\geq} \frac{\mu_1}{|S_2'|} h(S_2'|S_1'), \end{aligned} \tag{55}$$

where (a) is due to function h being μ_1-element-sequence-submodular and (b) is from Eq. (54). This completes the proof. □

Lemma 9. *Consider $1 \leq k' \leq k$. Suppose that the sequence selected by the SSG algorithm is S with $|S| = k$ and that there exist sequences S_1 and S_2 such that sequence S can be written as $S = S_1 \oplus S_2$ with $|S_1| = k - k'$ and $|S_2| = k'$. Then, under Assumption 7, we have $h(S_2|S_1) \geq (1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1))$.*

Proof. The proof follows a similar line of analysis as in Lemma 5. Let v_i denote the i-th element of sequence S_2, and let $S_2^i \triangleq (v_1, \dots, v_i)$ denote the sequence consisting of the first i elements of sequence S_2. Since function h is forward-monotone, we can assume that $|S^*(\mathcal{V}, k, 0)| = k$ as adding more elements to the end of a sequence does not reduce its overall value.

Due to Lemma 8, there exists some element $v' \in \mathcal{V}$ such that $h((v')|S_1 \oplus S_2^{i-1}) \geq \frac{\mu_1}{k} h(S^*(\mathcal{V}, k, 0)|S_1 \oplus S_2^{i-1})$. Then, we have the following:

$$\begin{aligned} h(S_1 \oplus S_2^i) - h(S_1 \oplus S_2^{i-1}) &= h((v_i)|S_1 \oplus S_2^{i-1}) \\ &\overset{\text{(a)}}{\geq} h((v')|S_1 \oplus S_2^{i-1}) \\ &\overset{\text{(b)}}{\geq} \frac{\mu_1}{k} h(S^*(\mathcal{V}, k, 0)|S_1 \oplus S_2^{i-1}) \\ &= \frac{\mu_1}{k} (h(S_1 \oplus S_2^{i-1} \oplus S^*(\mathcal{V}, k, 0)) - h(S_1 \oplus S_2^{i-1})) \\ &\overset{\text{(c)}}{\geq} \frac{\mu_1}{k} (h(S^*(\mathcal{V}, k, 0)) - h(S_1 \oplus S_2^{i-1})), \end{aligned} \tag{56}$$

where (a) is due to the greedy manner of the SSG algorithm (Line 4 of Algorithm 3), (b) is from the property of element v' (due to Lemma 8), and (c) is due to the backward monotonicity of function h. Rewriting Eq. (56) yields the following equivalent inequality:

$$h(S_1 \oplus S_2^i) \geq \frac{\mu_1}{k} h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{\mu_1}{k}) h(S_1 \oplus S_2^{i-1}). \tag{57}$$

By writing Eq. (57) for $i \in \{1, \dots, k'\}$ and combining them, we obtain the following:

$$\begin{aligned} h(S_1 \oplus S_2^{k'}) &\geq \sum_{j=0}^{k'-1} \frac{\mu_1}{k} (1 - \frac{\mu_1}{k})^j \, h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{\mu_1}{k})^{k'} h(S_1) \\ &= (1 - (1 - \frac{\mu_1}{k})^{k'}) h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{\mu_1}{k})^{k'} h(S_1). \end{aligned} \tag{58}$$

Applying Eq. (58) and the fact that $S_2 = S_2^{k'}$ yields the following:

$$\begin{aligned} h(S_2|S_1) &= h(S_1 \oplus S_2^{k'}) - h(S_1) \\ &\geq (1 - (1 - \frac{\mu_1}{k})^{k'})h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{\mu_1}{k})^{k'} h(S_1) - h(S_1) \\ &= (1 - (1 - \frac{\mu_1}{k})^{k'})h(S^*(\mathcal{V}, k, 0)) - (1 - (1 - \frac{\mu_1}{k})^{k'})h(S_1) \\ &= (1 - (1 - \frac{\mu_1}{k})^{k'})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)) \\ &\geq (1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)), \end{aligned} \tag{59}$$

where the last inequality holds because $(1 - \frac{\mu_1}{k}) \leq e^{-\frac{\mu_1}{k}}$ and $h(S^*(\mathcal{V}, k, 0)) - h(S_1)$ is nonnegative. This completes the proof. □

Having introduced Lemmas 8 and 9, we are now ready to prove Lemma 7.

Proof of Lemma 7. Suppose $h(S) = \delta \cdot h(S^*(\mathcal{V}, k, 0))$ for some $\delta \in (0, 1]$. Then, we have

$$\begin{aligned} \delta \cdot h(S^*(\mathcal{V}, k, 0)) &= h(S) \\ &= h(S_1) + h(S_2|S_1) \\ &\overset{\text{(a)}}{\geq} h(S_1) + (1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}})(h(S^*(\mathcal{V}, k, 0)) - h(S_1)) \\ &= \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}} h(S_1) + (1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}})h(S^*(\mathcal{V}, k, 0)) \\ &\overset{\text{(b)}}{\geq} \frac{c\delta}{e^{\mu_1 \cdot \frac{k'}{k}}} h(S^*(\mathcal{V}, k, 0)) + (1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}})h(S^*(\mathcal{V}, k, 0)), \end{aligned} \tag{60}$$

where (a) is from Lemma 9 and (b) is due to $h(S_1) \geq c \cdot h(S) = c \cdot \delta \cdot h(S^*(\mathcal{V}, k, 0))$. Dividing both sides of Eq. (60) by $h(S^*(\mathcal{V}, k, 0))$ yields the following:

$$\delta \geq \frac{c\delta}{e^{\mu_1 \cdot \frac{k'}{k}}} + 1 - \frac{1}{e^{\mu_1 \cdot \frac{k'}{k}}}, \tag{61}$$

which implies

$$\delta \geq \frac{e^{\mu_1 \cdot \frac{k'}{k}} - 1}{e^{\mu_1 \cdot \frac{k'}{k}} - c}. \tag{62}$$

Since $h(S) = \delta \cdot h(S^*(\mathcal{V}, k, 0))$, we have $h(S) \geq \frac{e^{\mu_1 \cdot \frac{k'}{k}} - 1}{e^{\mu_1 \cdot \frac{k'}{k}} - c} h(S^*(\mathcal{V}, k, 0))$. This completes the proof. □

8.9 Proof of Theorem 4

Proof. We use Lemma 3 presented in Appendix 8.3 and Lemmas 6 and 7 presented in Appendix 8.8 to prove the approximation ratios of Algorithm 1 in the case of $\tau = 1$. The proof follows a similar line of analysis as in the proof of Theorem 1.

Given $\tau = 1$, in Step 1 of Algorithm 1, the selected sequence S_1 consists of one element only; this element is denoted by v_1, i.e., $S_1 = (v_1)$. In Step 2 of Algorithm 1, it is equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \{v_1\}$, and we have $|S_2| = k - \tau = k - 1$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2 = (v_1) \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). Note that only one element will be removed from S, i.e., $|\mathcal{Z}_\tau(S)| = 1$. For ease of notation, we use z to denote the only element in $\mathcal{Z}_\tau(S)$, i.e., $\mathcal{Z}_\tau(S) = \{z\}$.

We want to show the following two bounds, which establish the approximation ratios of Algorithm 1: 1) Suppose that function h is forward-monotone, backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 3), we have

$$h(S - \{z\}) \geq \frac{a(e^b - 1)}{e^b - a} g_\tau(S^*(\mathcal{V}, k, \tau)), \text{ where } a = \frac{\mu_1 \mu_2}{\mu_1 + 1} \text{ and } b = \frac{\mu_1(k-2)}{k-1}; \tag{63}$$

2) Suppose that function h is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 4), we have

$$h(S - \{z\}) \geq \frac{\alpha^2 \mu_1 \mu_2 (e^{\mu_1} - 1)}{(\mu_1 + \alpha) e^{\mu_1}} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{64}$$

We begin with the proof of Eq. (64) as some of the intermediate results will be used in the proof of Eq. (63). We first present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned} h(S_2) &\geq \alpha(1 - 1/e^{\mu_1}) h(S^*(\mathcal{V} \setminus \{v_1\}, k - \tau, 0)) \\ &\geq \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{65}$$

where the first inequality is from Lemma 6 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \{v_1\}$, $k - \tau$, and $k - \tau$, respectively) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\{v_1\}$).

The proof proceeds as follows. Element z is an element that will be removed, which can be either v_1 or an element in S_2. Therefore, we consider two cases: (I) $z = v_1$ and (II) $z \neq v_1$.

In Case I, we have $z = v_1$, which implies the following:

$$h(S - \{z\}) = h(S_2) \geq \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{66}$$

where the inequality follows from Eq. (65).

In Case II, we have $z \neq v_1$ (or $z \in \mathcal{V}(S_2)$). Depending on the impact of removing element z, we consider two subcases: (II-a) $h(S_2) \leq h(S_2 - \{z\})$ and (II-b) $h(S_2) > h(S_2 - \{z\})$.

In Case II-a: we have $h(S_2) \leq h(S_2 - \{z\})$. In this case, the removal of element z does not reduce the overall value of the remaining sequence $S_2 - \{z\}$. Then, we have

$$\begin{aligned} h(S - \{z\}) &= h((v_1) \oplus (S_2 - \{z\})) \\ &\overset{\text{(a)}}{\geq} \alpha \cdot h(S_2 - \{z\}) \\ &\overset{\text{(b)}}{\geq} \alpha \cdot h(S_2) \\ &\overset{\text{(c)}}{\geq} \alpha^2 \cdot (1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{67}$$

where (a) is due to the α-backward monotonicity of function h, (b) holds from the condition of this subcase, and (c) follows from Eq. (65).

In Case II-b: we have $h(S_2) > h(S_2 - \{z\})$. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \{z\})}{h(S_2)}$ denote the ratio of the loss of removing element z from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$ due to $h(S_2) > h(S_2 - \{z\})$. We first state the following:

$$h(S - \{z\}) \geq \max\{\mu_1(\eta + \mu_2 - 1) \cdot h(S_2), \alpha(1 - \eta) \cdot h(S_2)\}, \tag{68a}$$

$$\max\{\mu_1(\eta + \mu_2 - 1), \alpha(1 - \eta)\} \geq \frac{\alpha \mu_1 \mu_2}{\mu_1 + \alpha}. \tag{68b}$$

We will prove Eqs. (68a) and (68b) later; for now, we assume that they both hold. Then, we can obtain the following bound:

$$\begin{aligned} h(S - \{z\}) &\geq \max\{\mu_1(\eta + \mu_2 - 1) \cdot h(S_2), \alpha(1 - \eta) \cdot h(S_2)\} \\ &\geq \max\{\mu_1(\eta + \mu_2 - 1), \alpha(1 - \eta)\} \cdot \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{\alpha^2 \mu_1 \mu_2 (e^{\mu_1} - 1)}{(\mu_1 + \alpha) e^{\mu_1}} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{69}$$

where the three inequalities are from Eqs. (68a), (65), and (68b), respectively.

By combining the bounds in Eqs. (66), (67), and (69), we obtain the bound in Eq. (64).

Next, we prove Eq. (68a). We first rewrite S_2 as $S_2 = S_2^1 \oplus (z) \oplus S_2^2$, where S_2^1 and S_2^2 denote the subsequences of S_2 before and after element z, respectively. Note that S_2^1 or S_2^2 could be an empty sequence, depending on the position of z in S_2. Then, we characterize $h((z))$ in terms of $h(S_2)$:

$$\begin{aligned}
\eta \cdot h(S_2) &\overset{\text{(a)}}{=} h(S_2) - h(S_2 - \{z\}) \\
&= h(S_2^1 \oplus (z) \oplus S_2^2) - h(S_2^1 \oplus S_2^2) \\
&= h(S_2^1) + h((z)|S_2^1) + h(S_2^2|S_2^1 \oplus (z)) - h(S_2^1) - h(S_2^2|S_2^1) \\
&= h((z)|S_2^1) + h(S_2^2|S_2^1 \oplus (z)) - h(S_2^2|S_2^1) \\
&\overset{\text{(b)}}{\leq} h((z)|S_2^1) + h(S_2^2|S_2^1 \oplus (z)) - \mu_2 h(S_2^2|S_2^1 \oplus (z)) \\
&= h((z)|S_2^1) + (1-\mu_2) h(S_2^2|S_2^1 \oplus (z)) \\
&\overset{\text{(c)}}{\leq} h((z)|S_2^1) + (1-\mu_2) h(S_2) \\
&\overset{\text{(d)}}{\leq} h((z))/\mu_1 + (1-\mu_2) h(S_2),
\end{aligned} \tag{70}$$

where (a) is from the definition of η, (b) is due to function h being μ_2-sequence-submodular, (c) holds because $\mu_2 \leq 1$ and $h(S_2^2|S_2^1 \oplus (z)) \leq h(S_2^2|S_2^1 \oplus (z)) + h(S_2^1 \oplus (z)) = h(S_2)$ (recall that $S_2 = S_2^1 \oplus (z) \oplus S_2^2$), and (d) is due to function h being μ_1-element-sequence-submodular. From Eq. (70), we have the following:

$$h((z)) \geq \mu_1(\eta + \mu_2 - 1) h(S_2). \tag{71}$$

To prove Eq. (68a), we decompose it into two parts: (i) $h(S - \{z\}) \geq \mu_1(\eta + \mu_2 - 1) \cdot h(S_2)$ and (ii) $h(S - \{z\}) \geq \alpha(1-\eta) \cdot h(S_2)$.

Part (i) can be shown through the following:

$$h(S - \{z\}) \overset{\text{(a)}}{\geq} h((v_1)) \overset{\text{(b)}}{\geq} h((z)) \overset{\text{(c)}}{\geq} \mu_1(\eta + \mu_2 - 1) \cdot h(S_2), \tag{72}$$

where (a) is from the forward monotonicity of function h, (b) is due to the greedy manner of Algorithm 1 (Lines 3-5), and (c) is from Eq. (71).

Part (ii) can be shown through the following:

$$h(S - \{z\}) = h((v_1) \oplus (S_2 - \{z\})) \overset{\text{(a)}}{\geq} \alpha \cdot h(S_2 - \{z\}) \overset{\text{(b)}}{=} \alpha(1-\eta) \cdot h(S_2),$$

where (a) is from the α-backward monotonicity of function h and (b) is from the definition of η.

Eq. (68b) holds trivially for any $\eta \in (0, 1]$ by setting $\mu_1(\eta + \mu_2 - 1)$ and $\alpha(1-\eta)$ to be equal, solving for η, and plugging it back. This completes the proof of the bound in Eq. (64).

Next, we prove the bound in Eq. (63). Note that the analysis so far applies to function h that is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 4), and hence, it also applies to Assumption 3, which is a special case of Assumption 4 with $\alpha = 1$. The bound in Eq. (63) requires backward monotonicity (i.e., $\alpha = 1$), but it becomes better than the bound in Eq. (64) when k is large. In the following analysis, we assume that function h is forward-monotone, backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 3).

The proof proceeds as follows. We borrow the analysis of Case-I, Case II-a, and Eq. (68a) from the previous analysis by setting $\alpha = 1$. For Case-II-b, we provide a different analysis. In Case II-b: we have $z \in \mathcal{V}(S_2)$ and $h(S_2) > h(S_2 - \{z\})$. Suppose $k = 2$. Then, it is trivial that the sequence selected by Algorithm 1 (i.e., $S = (v_1) \oplus (z)$) yields an optimal solution. This is because removing element z from S gives (v_1), which has the largest individual value among all the elements. Therefore, we assume $k > 2$ throughout the rest of the proof. Recall that $\eta \triangleq \frac{h(S_2) - h(S_2 - \{z\})}{h(S_2)}$. We first state the following:

$$h(S_2) \geq \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{73a}$$

$$\max\left\{ \frac{\mu_1(\eta + \mu_2 - 1)(e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)}, \frac{(1-\eta)(e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)} \right\}$$
$$\geq \frac{\mu_1 \mu_2}{1 + \mu_2} \cdot \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \frac{\mu_1 \mu_2}{1+\mu_2}}. \tag{73b}$$

We will prove Eqs. (73a) and (73b) later; for now, we assume that they both hold. Then, we can obtain the following bound:

$$\begin{aligned} h(S - \{z\}) &\geq \max\{\mu_1(\eta + \mu_2 - 1) \cdot h(S_2), (1 - \eta) \cdot h(S_2)\} \\ &\geq \max\left\{ \frac{\mu_1(\eta + \mu_2 - 1)(e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)}, \frac{(1-\eta)(e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)} \right\} g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{\mu_1 \mu_2}{1 + \mu_2} \cdot \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \frac{\mu_1 \mu_2}{1+\mu_2}} \cdot g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{74}$$

where the three inequalities are from Eq. (68a) (with $\alpha = 1$), (73a), and (73b), respectively.

By combining the bounds in Eqs. (66), (67) (with $\alpha = 1$), and (74), we obtain Eq. (63).

Next, we show that Eq. (73a) holds. Let v_2^1 denote the first element of sequence S_2. Then, we have the following:

$$h((v_2^1)) \overset{\text{(a)}}{\geq} h((z)) \overset{\text{(b)}}{\geq} \mu_1(\eta + \mu_2 - 1) \cdot h(S_2),$$

where (a) holds because element v_2^1 has the largest individual value among all elements in S_2 and (b) follows from Eq. (71).Then, we can characterize the value of $h(S_2)$ as follows:

$$\begin{aligned} h(S_2) &\geq \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)} h(S^*(\mathcal{V} \setminus \{v_1\}, k - \tau, 0)) \\ &\geq \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}} - 1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \mu_1(\eta + \mu_2 - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned}$$

where the inequalities are from Lemma 7 (where we replace $\mathcal{V}$, S, S_1, k, k', and c with $\mathcal{V} \setminus \{v_1\}$, S_2, (v_2^1), $k - 1$, $k - 2$, and $\mu_1(\eta + \mu_2 - 1)$, respectively) and Lemma 3, respectively.

Finally, we show that Eq. (73b) holds. We define two functions of η: $l_1(\eta) \triangleq \frac{\mu_1(\eta+\mu_2-1)(e^{\mu_1 \cdot \frac{k-2}{k-1}}-1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}}-\mu_1(\eta+\mu_2-1)}$ and $l_2(\eta) \triangleq \frac{(1-\eta)(e^{\mu_1 \cdot \frac{k-2}{k-1}}-1)}{e^{\mu_1 \cdot \frac{k-2}{k-1}}-\mu_1(\eta+\mu_2-1)}$. It is easy to verify that for $k > 2$ and $\eta \in (0, 1]$, function $l_1(\eta)$ is monotonically increasing and function $l_2(\eta)$ is monotonically decreasing. Let $\eta^* = \frac{1+\mu_1-\mu_1\mu_2}{1+\mu_1}$, and we have $l_1(\eta^*) = l_2(\eta^*) = \frac{\mu_1\mu_2}{1+\mu_2} \cdot \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}}-1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \frac{\mu_1\mu_2}{1+\mu_2}}$. We consider two cases for η: $\eta \in [\eta^*, 1]$ and $\eta \in (0, \eta^*]$. For $\eta \in [\eta^*, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta) \geq l_1(\eta^*)$ as $l_1(\eta)$ is monotonically increasing; for $\eta \in (0, \eta^*]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_2(\eta) \geq l_2(\eta^*) = l_1(\eta^*)$ as $l_2(\eta)$ is monotonically decreasing. Therefore, for $\eta \in (0, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta^*) = \frac{\mu_1\mu_2}{1+\mu_2} \cdot \frac{e^{\mu_1 \cdot \frac{k-2}{k-1}}-1}{e^{\mu_1 \cdot \frac{k-2}{k-1}} - \frac{\mu_1\mu_2}{1+\mu_2}}$. This gives Eq. (73b) and completes the proof. □

8.10 Proof of Theorem 5

Proof. We use Lemma 3 presented in Appendix 8.3 and Lemmas 6 and 7 presented in Appendix 8.8 to prove the approximation ratios of Algorithm 1 in the case of $1 \leq \tau \leq k$, assuming the removal of τ contiguous elements. The proof follows a similar line of analysis as in the proof of Theorem 2.

In Step 1 of Algorithm 1, it is equivalent that sequence S_1 is selected by the SSG algorithm from set $\mathcal{V}$, and we have $|S_1| = \tau$. Similarly, in Step 2 of Algorithm 1, it is also equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \mathcal{V}(S_1)$, and we have $|S_2| = k - \tau$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). We define $\mathcal{Z}_\tau^1(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_1)$ and $\mathcal{Z}_\tau^2(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_2)$ as the set of elements removed from subsequences S_1 and S_2, respectively.

We want to show the following two bounds, which establish the approximation ratios of Algorithm 1: 1) Suppose that function h is forward-monotone, backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 3), we have

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{a\mu_2(e^b - 1)}{(a+1)e^b - a\mu_2} g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{75}$$

where $a = \mu_1 \cdot (1 - 1/e^{\mu_1})$ and $b = \mu_1 \cdot \frac{k-2\tau}{k-\tau}$; 2) Suppose that function h is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 4), we have

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{\alpha^2\mu_1\mu_2(e^{\mu_1} - 1)^2}{\mu_1 e^{\mu_1}(e^{\mu_1} - 1) + e^{2\mu_1}} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{76}$$

We begin with the proof of Eq. (76) as some of the intermediate results will be used in the proof of Eq. (75). We first present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned} h(S_2) &\geq \alpha(1 - 1/e^{\mu_1}) h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k - \tau, 0)) \\ &\geq \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{77}$$

where the first inequality is from Lemma 6 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, $k - \tau$, and $k - \tau$, respectively) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\mathcal{V}(S_1)$).

The proof proceeds as follows. Elements in $\mathcal{Z}_\tau(S)$ will be removed from sequence S. These elements can be either fully or partially in $\mathcal{V}(S_1)$ (i.e., $\mathcal{Z}_\tau^2(S) = \emptyset$ or $\mathcal{Z}_\tau^2(S) \neq \emptyset$). Therefore, we consider two cases: (I) $\mathcal{Z}_\tau^2(S) = \emptyset$ and (II) $\mathcal{Z}_\tau^2(S) \neq \emptyset$.

In Case I, we have $\mathcal{Z}_\tau^2(S) = \emptyset$, i.e., $\mathcal{Z}_\tau(S) = \mathcal{V}(S_1)$. Then, we have the following:

$$h(S - \mathcal{Z}_\tau(S)) = h(S_2) \geq \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \tag{78}$$

where the inequality follows from Eq. (77).

In Case II, we have $\mathcal{Z}_\tau^2(S) \neq \emptyset$. Depending on the impact of removing elements in $\mathcal{Z}_\tau^2(S)$, we consider two subcases: (II-a) $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$ and (II-b) $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$.

In Case II-a: we have $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$. In this case, the removal of elements in $\mathcal{Z}_\tau^2(S)$ does not reduce the overall value of the remaining sequence $S_2 - \mathcal{Z}_\tau^2(S)$. Then, we have

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &= h((S_1 - \mathcal{Z}_\tau^1(S)) \oplus (S_2 - \mathcal{Z}_\tau^2(S))) \\ &\overset{\text{(a)}}{\geq} \alpha \cdot h(S_2 - \mathcal{Z}_\tau^2(S)) \\ &\overset{\text{(b)}}{\geq} \alpha \cdot h(S_2) \\ &\overset{\text{(c)}}{\geq} \alpha^2(1 - 1/e^{\mu_1}) \cdot g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{79}$$

where (a) is due to the α-backward monotonicity of function h, (b) holds from the condition of this subcase, and (c) follows from Eq. (77).

In Case II-b: we have $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \mathcal{Z}_\tau^2(S))}{h(S_2)}$ denote the ratio of the loss of removing elements in $\mathcal{Z}_\tau^2(S)$ from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$ due to $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. We first state the following:

$$h(S - \mathcal{Z}_\tau(S)) \geq \max\{\mu_1\alpha(\eta + \mu_2 - 1)(1 - 1/e^{\mu_1}) \cdot h(S_2), \alpha(1 - \eta) \cdot h(S_2)\}, \tag{80a}$$

$$\max\{\mu_1\alpha(\eta + \mu_2 - 1)(1 - 1/e^{\mu_1}), \alpha(1 - \eta)\} \geq \frac{\alpha\mu_1\mu_2(e^{\mu_1} - 1)}{\mu_1(e^{\mu_1} - 1) + e^{\mu_1}}. \tag{80b}$$

We will prove Eqs. (80a) and (80b) later; for now, we assume that they both hold. Then, we can obtain the following bound:

$$\begin{aligned}h(S-\mathcal{Z}_\tau(S)) &\geq \max\{\mu_1\alpha(\eta+\mu_2-1)(1-1/e^{\mu_1})\cdot h(S_2), \alpha(1-\eta)\cdot h(S_2)\}\\&\geq \max\{\mu_1\alpha(\eta+\mu_2-1)(1-1/e^{\mu_1}), \alpha(1-\eta)\}\cdot\alpha(1-1/e^{\mu_1})\cdot g_\tau(S^*(\mathcal{V},k,\tau))\\&\geq \frac{\alpha^2\mu_1\mu_2(e^{\mu_1}-1)^2}{\mu_1 e^{\mu_1}(e^{\mu_1}-1)+e^{2\mu_1}}\cdot g_\tau(S^*(\mathcal{V},k,\tau)),\end{aligned} \tag{81}$$

where the three inequalities are from Eqs. (80a), (77), and (80b), respectively.

By combining the bounds in Eqs. (78), (79), and (81), we obtain the second bound in Eq. (76).

Next, we show that Eq. (80a) holds. Recall that the τ elements in $\mathcal{Z}_\tau(S)$ form a contiguous subsequence of S. Then, the elements in $\mathcal{Z}^1_\tau(S)$ and $\mathcal{Z}^2_\tau(S)$ also form a contiguous subsequence of S_1 and S_2, respectively. We use Z_1 and Z_2 to denote the contiguous subsequence of elements in $\mathcal{Z}^1_\tau(S)$ and $\mathcal{Z}^2_\tau(S)$, respectively. We first rewrite S_2 as $S_2 = S_2^1\oplus Z_2\oplus S_2^2$, where S_2^1 and S_2^1 denote the subsequences of S_2 before and after subsequence Z_2, respectively. Note that S_2^1 or S_2^2 could be an empty sequence, depending on the position of subsequence Z_2 in S_2. Then, we characterize $h(Z_2)$ in terms of $h(S_2)$:

$$\begin{aligned}\eta\cdot h(S_2) &\overset{\text{(a)}}{=} h(S_2)-h(S_2-\mathcal{Z}^2_\tau(S))\\&= h(S_2^1\oplus Z_2\oplus S_2^2)-h(S_2^1\oplus S_2^2)\\&= h(S_2^1)+h(Z_2|S_2^1)+h(S_2^2|S_2^1\oplus Z_2)-h(S_2^1)-h(S_2^2|S_2^1)\\&= h(Z_2|S_2^1)+h(S_2^2|S_2^1\oplus Z_2)-h(S_2^2|S_2^1)\\&\overset{\text{(b)}}{\leq} h(Z_2|S_2^1)+h(S_2^2|S_2^1\oplus Z_2)-\mu_2 h(S_2^2|S_2^1\oplus Z_2)\\&= h(Z_2|S_2^1)+(1-\mu_2)h(S_2^2|S_2^1\oplus Z_2)\\&\overset{\text{(c)}}{\leq} h(Z_2|S_2^1)+(1-\mu_2)h(S_2)\\&\overset{\text{(d)}}{\leq} h(Z_2)/\mu_1+(1-\mu_2)h(S_2),\end{aligned} \tag{82}$$

where (a) is from the definition of η, (b) is due to function h being μ_2-sequence-submodular, (c) holds because $\mu_2\leq 1$ and $h(S_2^2|S_2^1\oplus Z_2)\leq h(S_2^2|S_2^1\oplus Z_2)+h(S_2^1\oplus Z_2)=h(S_2)$ (recall that $S_2=S_2^1\oplus Z_2\oplus S_2^2$), and (d) is due to function h being μ_1-element-sequence-submodular. From Eq. (82), we have the following:

$$h(Z_2)\geq\mu_1(\eta+\mu_2-1)h(S_2). \tag{83}$$

To prove Eq. (80a), we decompose it into two parts: (i) $h(S-\mathcal{Z}_\tau(S))\geq\mu_1\alpha(\eta+\mu_2-1)(1-1/e^{\mu_1})\cdot h(S_2)$ and (ii) $h(S-\mathcal{Z}_\tau(S))\geq\alpha(1-\eta)\cdot h(S_2)$.

Let $\tau_2\triangleq|\mathcal{Z}^2_\tau(S)|$. We use $S_1^{\tau_2}$ to denote the subsequence consisting of the first τ_2 elements in S_1. Then, Part (i) can be shown through the following:

$$\begin{aligned}h(S-\mathcal{Z}_\tau(S)) &= h((S_1-\mathcal{Z}^1_\tau(S))\oplus(S_2-\mathcal{Z}^2_\tau(S)))\\&\overset{\text{(a)}}{\geq} h(S_1-\mathcal{Z}^1_\tau(S))\\&= h(S_1^{\tau_2})\\&\overset{\text{(b)}}{\geq}\alpha(1-1/e^{\mu_1})\cdot h(S^*(\mathcal{V},\tau_2,0))\\&\overset{\text{(c)}}{\geq}\alpha(1-1/e^{\mu_1})\cdot h(Z_2)\\&\overset{\text{(d)}}{\geq}\mu_1\alpha(\eta+\mu_2-1)(1-1/e^{\mu_1})\cdot h(S_2),\end{aligned} \tag{84}$$

where (a) is due to the forward monotonicity of function h, (b) is due to the greedy manner of selecting subsequence $S_1^{\tau_2}$ from set $\mathcal{V}$ and Lemma 6 (where we replace both k and i with τ_2), (c) holds because sequence Z_2 is a feasible solution to Problem (P) for selecting a sequence of τ_2 elements from $\mathcal{V}$, and (d) is from Eq. (83).

Part (ii) can be shown through the following:

$$h(S-\mathcal{Z}_\tau(S)) = h((S_1-\mathcal{Z}^1_\tau(S))\oplus(S_2-\mathcal{Z}^2_\tau(S))) \overset{\text{(a)}}{\geq} \alpha\cdot h(S_2-\mathcal{Z}^2_\tau(S)) \overset{\text{(b)}}{=} \alpha(1-\eta)\cdot h(S_2), \quad (85)$$

where (a) is from the α-backward monotonicity of function h and (b) is from the definition of η.

Eq. (80b) holds trivially for any $\eta \in (0,1]$ by setting $\mu_1\alpha(\eta+\mu_2-1)(1-1/e^{\mu_1})$ and $\alpha(1-\eta)$ to be equal, solving for η, and plugging it back. This completes the proof of the bound in Eq. (76).

Now, we prove the bound in Eq. (75). Note that the analysis so far applies to any function h that is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 4), and hence, it also applies to Assumption 3, which is a special case of Assumption 4 with $\alpha = 1$. The bound in Eq. (75) requires backward monotonicity (i.e., $\alpha = 1$), but it becomes better than the bound in Eq. (76) when k is large. In the following analysis, we assume that function h is forward-monotone, backward-monotone, μ_1-element-sequence-submodular, and μ_2-sequence-submodular (Assumption 3).

The proof proceeds as follows. We borrow the analysis of Case-I, Case II-a, and Eq. (80a) from the previous analysis by setting $\alpha = 1$. Then, for Case-II-b, we provide a different analysis.

In Case II-b: we have $h(S_2) > h(S_2 - \mathcal{Z}^2_\tau(S))$. Let $\tau_1 \triangleq |\mathcal{Z}^1_\tau(S)|$ and recall that $\tau_2 \triangleq |\mathcal{Z}^2_\tau(S)|$. Then, we have $\tau = \tau_1+\tau_2$ and $k = |S| = |S_1|+|S_2| \geq \tau+\tau_2$. Therefore, we consider two cases: $k = \tau+\tau_2$ and $k > \tau+\tau_2$.

Suppose $k = \tau+\tau_2$. Then, it implies $\mathcal{Z}^2_\tau(S) = \mathcal{V}(S_2)$, i.e., all the elements in S_2 are removed. This further implies that the elements in $\mathcal{Z}^1_\tau(S)$ are at the end of sequence S_1 (due to the contiguous assumption of elements in $\mathcal{Z}_\tau(S)$). Let $S_1^{\tau_2} \preceq S_1$ denote the subsequence consisting of the first τ_2 elements in S_1. It is easy to see $S_1 - \mathcal{Z}^1_\tau(S) = S_1^{\tau_2}$. Then, we have the following:

$$\begin{aligned}
h(S-\mathcal{Z}_\tau(S)) &= h((S_1-\mathcal{Z}^1_\tau(S))\oplus(S_2-\mathcal{Z}^2_\tau(S)))\\
&= h(S_1-\mathcal{Z}^1_\tau(S))\\
&= h(S_1^{\tau_2})\\
&\geq (1-1/e^{\mu_1})h(S^*(\mathcal{V},\tau_2,0))\\
&\geq (1-1/e^{\mu_1})g_\tau(S^*(\mathcal{V},k,\tau)),
\end{aligned} \quad (86)$$

where the first inequality is from Lemma 6 (where we replace both k and i with τ_2 and set $\alpha = 1$) and the second inequality is due to $\tau_2 = k-\tau$ and Lemma 3 (where $\mathcal{V}'$ is an empty set).

Now, suppose $k > \tau+\tau_2$. Recall that $\eta \triangleq \frac{h(S_2)-h(S_2-\mathcal{Z}^2_\tau(S))}{h(S_2)}$. Let $a \triangleq \mu_1(1-1/e^{\mu_1})$, $b \triangleq \mu_1\cdot\frac{k-2\tau}{k-\tau}$, and $b' \triangleq \mu_1\cdot\frac{k-\tau-\tau_2}{k-\tau_2}$. We first state the following:

$$h(S_2) \geq \frac{e^{b'}-1}{e^{b'}-a(\eta+\mu_2-1)}g_\tau(S^*(\mathcal{V},k,\tau)), \quad (87a)$$

$$\max\left\{\frac{a(\eta+\mu_2-1)(e^{b'}-1)}{e^{b'}-a(\eta+\mu_2-1)}, \frac{(1-\eta)(e^{b'}-1)}{e^{b'}-a(\eta+\mu_2-1)}\right\} \geq \frac{a\mu_2(e^{b'}-1)}{(a+1)e^{b'}-a\mu_2}. \quad (87b)$$

We will prove Eqs. (87a) and (87b) later; for now, we assume that they both hold. Then, we can obtain the following bound:

$$\begin{aligned}
h(S-\mathcal{Z}_\tau(S)) &\geq \max\{\mu_1(\eta+\mu_2-1)(1-1/e^{\mu_1})\cdot h(S_2), (1-\eta)\cdot h(S_2)\}\\
&\geq \max\left\{\frac{a(\eta+\mu_2-1)(e^{b'}-1)}{e^{b'}-a(\eta+\mu_2-1)}, \frac{(1-\eta)(e^{b'}-1)}{e^{b'}-a(\eta+\mu_2-1)}\right\}g_\tau(S^*(\mathcal{V},k,\tau))\\
&\geq \frac{a\mu_2(e^{b'}-1)}{(a+1)e^{b'}-a\mu_2}g_\tau(S^*(\mathcal{V},k,\tau))\\
&\geq \frac{a\mu_2(e^{b}-1)}{(a+1)e^{b}-a\mu_2}g_\tau(S^*(\mathcal{V},k,\tau)),
\end{aligned} \quad (88)$$

where the first three inequalities are from Eq. (80a) ($\alpha = 1$) and Eqs. (87a) and (87b), respectively; for the last inequality, we replace τ_2 in b' with τ to obtain b, and the inequality holds due to $\tau_2 \leq \tau$ and that $\frac{a\mu_2(e^{b'}-1)}{(a+1)e^{b'}-a\mu_2}$ is a decreasing function of τ_2.

By combining the bounds in Eqs. (78), (79) (with $\alpha = 1$), (86), and (88), we obtain Eq. (75).

Next, we show that Eq. (87a) holds. Let $S_2^{\tau_2} \preceq S_2$ denote the subsequence consisting of the first τ_2 elements of sequence S_2. Then, we have the following:

$$\begin{aligned} h(S_2^{\tau_2}) &\overset{\text{(a)}}{\geq} (1 - 1/e^{\mu_1}) h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), \tau_2, 0)) \\ &\overset{\text{(b)}}{\geq} (1 - 1/e^{\mu_1}) h(Z_2) \\ &\overset{\text{(c)}}{\geq} \mu_1(\eta + \mu_2 - 1)(1 - 1/e^{\mu_1}) h(S_2), \end{aligned} \tag{89}$$

where (a) is due to the greedy manner of selecting subsequence $S_2^{\tau_2}$ from set $\mathcal{V} \setminus \mathcal{V}(S_1)$ and Lemma 6 (where we set $\alpha = 1$ and replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, τ_2, and τ_2, respectively), (b) holds because sequence Z_2 is a feasible solution to Problem (P) for selecting a sequence of τ_2 elements from $\mathcal{V} = \mathcal{V} \setminus \mathcal{V}(S_1)$, and (c) is from Eq. (83).

Therefore, we can characterize the value of $h(S_2)$ as follows (recall that $b' \triangleq \mu_1 \cdot \frac{k-\tau-\tau_2}{k-\tau_2}$):

$$\begin{aligned} h(S_2) &\geq \frac{e^{b'} - 1}{e^{b'} - a(\eta + \mu_2 - 1)} h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k - \tau, 0)) \\ &\geq \frac{e^{b'} - 1}{e^{b'} - a(\eta + \mu_2 - 1)} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{90}$$

where the inequalities are from Lemma 7 (where we replace $\mathcal{V}$, S, S_1, k, k', and c with $\mathcal{V} \setminus \mathcal{V}(S_1)$, S_2, $S_2^{\tau_2}$, $k - \tau$, $k - \tau - \tau_2$, and $a(\eta + \mu_2 - 1)$, respectively) and Lemma 3, respectively.

Finally, we show that Eq. (87b) holds. We define two functions of η:

$$l_1(\eta) = \frac{a \cdot (\eta + \mu_2 - 1)(e^{b'} - 1)}{e^{b'} - a \cdot (\eta + \mu_2 - 1)} \text{ and } l_2(\eta) = \frac{(1 - \eta)(e^{b'} - 1)}{e^{b'} - a \cdot (\eta + \mu_2 - 1)}.$$

It can be verified that for $k > \tau + \tau_2$ and $\eta \in (0, 1]$, function $l_1(\eta)$ is monotonically increasing and function $l_2(\eta)$ is monotonically decreasing. Also, consider $\eta^* \triangleq \frac{1+a-a\cdot\mu_2}{a+1}$. We have $l_1(\eta^*) = l_2(\eta^*) = \frac{a\mu_2(e^{b'}-1)}{(a+1)e^{b'}-a\mu_2}$. We consider two cases for η: $\eta \in [\eta^*, 1]$ and $\eta \in (0, \eta^*]$. For $\eta \in [\eta^*, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta) \geq l_1(\eta^*)$ as $l_1(\eta)$ is monotonically increasing; for $\eta \in (0, \eta^*]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_2(\eta) \geq l_2(\eta^*) = l_1(\eta^*)$ as $l_2(\eta)$ is monotonically decreasing. Therefore, for $\eta \in (0, 1]$, we have $\max\{l_1(\eta), l_2(\eta)\} \geq l_1(\eta^*) = \frac{a\mu_2(e^{b'}-1)}{(a+1)e^{b'}-a\mu_2}$. This gives Eq. (87b) and completes the proof of Eq. (75). □

8.11 Proof of Theorem 6

Proof. Suppose that function h is forward-monotone, α-backward-monotone, μ_1-element-sequence-submodular, and μ_3-general-sequence-submodular (Assumption 5). We use Lemma 3 presented in Appendix 8.3 and Lemma 6 presented in Appendix 8.8 to prove that Algorithm 2 achieves an approximation ratio of $\frac{\alpha^2\mu_1\mu_3(e^{\mu_1}-1)}{(\mu_1+\alpha\tau)e^{\mu_1}}$ in the case of $1 \leq \tau \leq k$, assuming the removal of an arbitrary subset of τ selected elements, which are not necessarily contiguous.

In Step 1 of Algorithm 2, sequence S_1 is selected by choosing τ elements with the highest individual values in a greedy manner, and we have $|S_1| = \tau$. In Step 2 of Algorithm 1, it is equivalent that sequence S_2 is selected by the SSG algorithm from set $\mathcal{V} \setminus \mathcal{V}(S_1)$, and we have $|S_2| = k - \tau$. Hence, the sequence selected by Algorithm 1 can be written as $S = S_1 \oplus S_2$. Recall that for any given sequence S, set $\mathcal{Z}_\tau(S)$ denotes the set of elements removed from sequence S in the worst case (i.e., $\mathcal{Z}_\tau(S)$ is an optimal solution to Problem (11)). We define $\mathcal{Z}_\tau^1(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_1)$ and $\mathcal{Z}_\tau^2(S) \triangleq \mathcal{Z}_\tau(S) \cap \mathcal{V}(S_2)$ as the set of elements removed from subsequences S_1 and S_2, respectively.

The proof of Theorem 6 follows a similar line of analysis as in the proof of Theorem 5. Specifically, we will also consider three cases: (I) $\mathcal{Z}_\tau^2(S) = \emptyset$, (II-a) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) \leq h(S_2 - \mathcal{Z}_\tau^2(S))$, and (II-b) $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. The proofs of Case I and Case II-a are identical to those in Theorem 5. Therefore, we focus on Case II-b, which requires a different proof strategy.

We want to show the following bound that establishes the approximation ratio of Algorithm 2:

$$h(S - \mathcal{Z}_\tau(S)) \geq \frac{\alpha^2 \mu_1 \mu_3 (e^{\mu_1} - 1)}{(\mu_1 + \alpha\tau) e^{\mu_1}} g_\tau(S^*(\mathcal{V}, k, \tau)). \tag{91}$$

To begin with, we present a lower bound on $h(S_2)$, which will be used throughout the proof:

$$\begin{aligned} h(S_2) &\geq \alpha(1 - 1/e^{\mu_1}) h(S^*(\mathcal{V} \setminus \mathcal{V}(S_1), k - \tau, 0)) \\ &\geq \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{92}$$

where the first inequality is from Lemma 6 (where we replace $\mathcal{V}$, k, and i with $\mathcal{V} \setminus \mathcal{V}(S_1)$, $k - \tau$, and $k - \tau$, respectively) and the second inequality is from Lemma 3 (where we replace $\mathcal{V}'$ with $\mathcal{V}(S_1)$).

We now focus on Case II-b, where we have $\mathcal{Z}_\tau^2(S) \neq \emptyset$ and $h(S_2) > h(S_2 - \mathcal{Z}_\tau^2(S))$. Let $\eta \triangleq \frac{h(S_2) - h(S_2 - \mathcal{Z}_\tau^2(S))}{h(S_2)}$ denote the ratio of the loss of removing elements in $\mathcal{Z}_\tau^2(S)$ from sequence S_2 to the value of sequence S_2, and we have $\eta \in (0, 1]$. We first state the following:

$$h(S - \mathcal{Z}_\tau(S)) \geq \max\{\frac{\mu_1(\eta + \mu_3 - 1)}{\tau} \cdot h(S_2), \alpha(1 - \eta) \cdot h(S_2)\}, \tag{93a}$$

$$\max\left\{\frac{\mu_1(\eta + \mu_3 - 1)}{\tau}, \alpha(1 - \eta)\right\} \geq \frac{\alpha \mu_1 \mu_3}{\mu_1 + \tau\alpha}. \tag{93b}$$

We will prove Eqs. (93a) and (93b) later; for now, we assume that they both hold. Then, we can obtain the following bound:

$$\begin{aligned} h(S - \mathcal{Z}_\tau(S)) &\geq \max\left\{\frac{\mu_1(\eta + \mu_3 - 1)}{\tau} \cdot h(S_2), \alpha(1 - \eta) \cdot h(S_2)\right\} \\ &\geq \max\left\{\frac{\mu_1(\eta + \mu_3 - 1)}{\tau}, \alpha(1 - \eta)\right\} \cdot \alpha(1 - 1/e^{\mu_1}) g_\tau(S^*(\mathcal{V}, k, \tau)) \\ &\geq \frac{\alpha^2 \mu_1 \mu_3 (e^{\mu_1} - 1)}{(\mu_1 + \alpha\tau) e^{\mu_1}} g_\tau(S^*(\mathcal{V}, k, \tau)), \end{aligned} \tag{94}$$

where the three inequalities are from Eqs. (93a), (92), and (93b), respectively.

Now, we show that Eqs. (93a) and (93b) hold. We start by characterizing the value of elements in $\mathcal{Z}_\tau^2(S)$. Let $\tau_2 \triangleq |\mathcal{Z}_\tau^2(S)|$, and let the elements in $\mathcal{Z}_\tau^2(S)$ be denoted by $z_1, z_2, \ldots, z_{\tau_2}$ according to their order in sequence S_2. Then, we can rewrite S_2 as $S_2 = S_2^1 \oplus (z_1) \oplus S_2^2 \oplus (z_2) \oplus \cdots \oplus S_2^{\tau_2 + 1}$, where S_2^i is the subsequence between elements z_{i-1} and z_i, for $i = 1, 2, \ldots, \tau_2 + 1$, and both z_0 and $z_{\tau+1}$ are an empty sequence. Note that subsequence S_2^i could be an empty sequence, for

$i = 1, 2, \dots, \tau_2 + 1$. We characterize the value of elements in $\mathcal{Z}_\tau^2(S)$ in the following:

$$
\begin{aligned}
\eta \cdot h(S_2) \overset{\text{(a)}}{=} & h(S_2) - h(S_2 - \mathcal{Z}_\tau^2(S)) \\
= & h(S_2^1) + h((z_1)|S_2^1) + h(S_2^2|S_2^1 \oplus (z_1)) \\
& + \cdots + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) \\
& - h(S_2^1) - h(S_2^2|S_2^1) - \cdots - h(S_2^{\tau_2+1}|S_2^1 \oplus \cdots \oplus S_2^{\tau_2}) \\
= & \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) + h(S_2^2|S_2^1 \oplus (z_1)) - h(S_2^2|S_2^1) \\
& + \cdots + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) - h(S_2^{\tau_2+1}|S_2^1 \oplus \cdots \oplus S_2^{\tau_2}) \\
\overset{\text{(b)}}{\leq} & \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) \\
& + h(S_2^2|S_2^1 \oplus (z_1)) - \mu_3 \cdot h(S_2^2|S_2^1 \oplus (z_1)) \\
& + \cdots + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) \\
& - \mu_3 \cdot h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) \\
= & \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) + (1 - \mu_3) \cdot h(S_2^2|S_2^1 \oplus (z_1)) \\
& + \cdots + (1 - \mu_3) \cdot h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2})) \\
= & \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) \\
& + (1 - \mu_3)(h(S_2^2|S_2^1 \oplus (z_1)) + \cdots + h(S_2^{\tau_2+1}|S_2^1 \oplus (z_1) \oplus \cdots \oplus S_2^{\tau_2} \oplus (z_{\tau_2}))) \\
\overset{\text{(c)}}{\leq} & \sum_{i=1}^{\tau_2} h((z_i)|S_2^1 \oplus (z_1) \oplus \cdots \oplus (z_{i-1}) \oplus S_2^i) + (1 - \mu_3) \cdot h(S_2) \\
\overset{\text{(d)}}{\leq} & \sum_{i=1}^{\tau_2} h((z_i))/\mu_1 + (1 - \mu_3) \cdot h(S_2),
\end{aligned}
\tag{95}
$$

where (a) is from the definition of η, (b) is due to function h being μ_3-general-sequence-submodular, (c) holds because $\mu_3 \leq 1$ and $S_2 = S_2^1 \oplus (z_1) \oplus S_2^2 \oplus (z_2) \oplus \cdots \oplus S_2^{\tau_2+1}$, and (d) is due to function h being μ_1-element-sequence-submodular. From Eq. (95), we have the following:

$$
\sum_{i=1}^{\tau_2} h((z_i)) \geq \mu_1(\eta + \mu_3 - 1)h(S_2). \tag{96}
$$

To prove Eq. (93a), we decompose it into two parts: (i) $h(S - \mathcal{Z}_\tau(S)) \geq \frac{\mu_1(\eta+\mu_3-1)}{\tau} \cdot h(S_2)$ and (ii) $h(S - \mathcal{Z}_\tau(S)) \geq \alpha(1 - \eta) \cdot h(S_2)$.

Let v' denote the first element in $S_1 - \mathcal{Z}_\tau^1(S)$. Then, Part (i) can be shown through the following:

$$
h(S - \mathcal{Z}_\tau(S)) \overset{\text{(a)}}{\geq} h((v')) \overset{\text{(b)}}{\geq} \frac{1}{\tau_2} \sum_{i=1}^{\tau_2} h((z_i)) \overset{\text{(c)}}{\geq} \frac{\mu_1(\eta + \mu_3 - 1)}{\tau_2} \cdot h(S_2) \overset{\text{(d)}}{\geq} \frac{\mu_1(\eta + \mu_3 - 1)}{\tau} \cdot h(S_2),
$$

where (a) is due to the forward monotonicity of function h, (b) is due to $h(v') \geq h(z_i)$ for any $i = 1, 2, \dots, \tau_2$, (c) is from Eq. (96), and (d) is due to $\tau_2 \leq \tau$.

Part (ii) can be shown through the following:

$$
h(S - \mathcal{Z}_\tau(S)) = h((S_1 - \mathcal{Z}_\tau^1(S)) \oplus (S_2 - \mathcal{Z}_\tau^2(S))) \overset{\text{(a)}}{\geq} \alpha \cdot h(S_2 - \mathcal{Z}_\tau^2(S)) \overset{\text{(b)}}{=} \alpha(1 - \eta) \cdot h(S_2),
$$

where (a) is from the α-backward monotonicity of function h and (b) is from the definition of η.

Eq. (93b) holds trivially for any $\eta \in (0, 1]$ by setting $\frac{\mu_1(\eta+\mu_3-1)}{\tau}$ and $\alpha(1 - \eta)$ to be equal, solving for η, and plugging it back. This completes the proof. □