[Reviewer 1] 1.1 Robust sequence submodular vs. robust set submodular. The main differences are two-fold. **(i)** From the algorithmic perspective, while there are some similarities in the designed algorithms, Algorithm 1 is designed specifically for the special case of the removal of contiguous elements and achieves a constant approximation ratio for any value of τ and k. This is different from the algorithms in [13][14], which achieve constant approximation ratios only when τ is very small compared to k. **(ii)** The analysis with respect to sequence functions is more challenging, which mainly stems from the new properties (that are irrelevant for set functions), such as multiple (nonequivalent) definitions of the diminishing returns property, two forms of monotonicity, and the impact of the ordering itself. Any attempt of adapting algorithms from robust set submodularity needs to carefully consider the aforementioned subtle yet critical differences while establishing approximation guarantees. The algorithms for robust set submodularity cannot be directly applied to sequence functions, as converting a set into a sequence could result in an arbitrarily bad performance. **1.2 Applications.** The order is important in many cases, e.g., when the recommended videos are part of a movie series or a TV show. In fact, movie recommendation and TV show recommendation have been modeled as sequence functions in [12] and [6], 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 objective functions in some applications may not always be sequence submodular, in Section 4, we introduce generalized formulations that account for *approximate sequence submodularity* and *approximate backward monotonicity* and leverage them to prove approximation results of the proposed algorithms under weaker assumptions, which we believe hold for a wider range of applications. **1.3 Empirical evaluation.** We agree that empirical evaluation is important. However, as a very first study on this new problem, we choose to focus on fundamentals and rigorous analyses, such as designing and theoretically analyzing algorithms, introducing and clarifying subtle yet critical properties of sequence functions, proving approximation guarantees for important variants of the problem, and providing useful insights for further studies. This is in line with other relevant work that investigates new problems and focuses on theoretical studies, such as robust set submodular maximization [13] and sequence submodular maximization [9]. We believe that this work serves as an important first step towards the design and analysis of efficient algorithms for robust sequence submodular maximization, which can be further explored through empirical evaluations for specific applications. **1.4 Simple algorithms.** While our greedy algorithms are simple, the theoretical analysis is more challenging, and the presented approximation guarantees are highly nontrivial. Please also see "1.1 Robust sequence submodular vs. robust set submodular."

[Reviewer 2] 2.1 Theoretical results. Algorithm 2 allows the removal of arbitrary τ elements, and thus, it is not surprising that achieving a stronger approximation guarantee (than that in Theorem 3) becomes more challenging if not impossible. On the other hand, Algorithm 1 achieves a constant approximation ratio for any value of τ in the special case of the removal of contiguous elements. Note that even for robust set submodularity, to the best of our knowledge, the developed algorithms achieve a constant approximation ratio only when τ is very small compared to k [13][14]. **2.2 Contiguous removal of 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 exploit the properties of such a special case to design Algorithm 1, which achieves a constant approximation ratio for any value of τ. Note that there is no such special case in robust (set) submodular maximization. **2.3 Extension of [13][14].** It is unclear whether the algorithms in [13][14] can be properly extended to our problem, and 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. We believe that our work serves as an important first step towards developing efficient algorithms for robust sequence submodular maximization. Note that the analysis of our simple greedy algorithms is already very sophisticated. Please also see "1.1 Robust sequence submodular vs. robust set submodular." **2.4 Evaluation.** Please see "1.3 Empirical evaluation." **2.5 References [6][12][MFKK].** While these references assume 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. We will elaborate on such discussions in the revised version.

[Reviewer 3] 3.1 Approximation ratios. The approximation ratio in Theorem 3 is pretty clean. The approximation ratio in Theorems 1 and 2 is the maximum of two terms (see supplementary files): the first is a constant; the second depends on τ and k. Thus, it is lower bounded by a (clean) constant *independent* of τ and k. We include the second term as it leads to a better overall approximation ratio for a larger k. The additional parameters (α and μ's) render the approximation ratios in Theorems 4-6 somewhat complex, but they are necessary for the generalized formulations.

[Reviewer 4] 4.1 Motivating examples. Please see "1.2 Applications." **4.2 Experiments.** Please see "1.3 Empirical evaluation." **4.3 Bounds.** Please see "2.1 Theoretical results." **4.4 Concrete functions.** Please refer to [5] for some concrete sequence submodular functions, such as the one in Eq. (10) of [5], which is the expected fraction of accomplished subtasks. **4.5 Bounds plot.** Thanks for the suggestion. We will add such figures in the revised version.