We thank all the reviewers for their valuable feedback. We first address some common concerns.

Deterministic systems. Although deterministic systems seem restrictive in theory, in practice, lots of RL problems are indeed deterministic. Also, this assumption makes the problem more tractable. We will emphasize that we focus on deterministic systems in the next version, and will also leave extending our result to stochastic environments as an open problem.

Practicality. We stress that our goal is to design provably efficient algorithms for RL with general reward functions. In this paper, we focus on giving sufficient and necessary conditions that admit efficient algorithms, and we believe our algorithmic insights (discretization, augmenting state space) can be applied in practice, which we are currently exploring. However, this is not the focus of the current work.

—— To Reviewer #1 ——

Section 5. The main goal of the algorithm in Section 5 is to motivate the discretization procedure used in the more complicated algorithm in Section 6, and thus we do not focus on optimizing the approximation guarantee. In the next version we will provide pseudocode in Section 5 to make the description of the algorithm formal.

Correctness of the complexity result / The algorithm in Section 6 is also rather brute force. The number of multisets of cardinality k, with elements taken from a finite set of cardinality n, is $\binom{n+k-1}{k} \leq (k+1)^n$. This bound can be found on the wikipedia page of multiset. For our case, $k = H$ and $n = \Theta(\log(1/\hat{\delta})/\bar{\delta})$, and thus the number of possible multisets is at most $H^{\Theta(\log(1/\hat{\delta})/\bar{\delta})}$. Notice that "multisets" are different from "sequences", i.e., for multisets we do not care about orders of elements. Indeed, the number of sequences of length k, with elements taken from a finite set of cardinality n, could be as large as n^k. This also explains why our algorithm is not brute force: we carefully discretize reward values to make the number of possible elements small, and we exploit the symmetric of the objective function so that we only need to deal with multisets instead of sequences to avoid an exponential dependency on H.

—— To Reviewer #2 ——

We are grateful to the reviewer for providing detailed comments on the writing of our paper, and we will revise the paper according to the reviewer's comment in the next version.

What prevents these techniques from being applied in stochastic environments? Consider the following case in the symmetric norm setting (Section 5 in our paper), which suggests that the stochastic case is fundamentally more difficulty: there are two actions at the initial state, and all further actions do not affect the rewards. If the first action is chosen, then half of the reward values will be 1 and half of the reward values will be 0. If the second action is chosen, then all reward values will be a fair coin (0 or 1 with equal probability). To find a near-optimal policy, the agent must carefully compare the expected objective values of both choices and thus cannot be handled by our algorithm. However, if rewards are deterministic, one can simply return the action with more reward values of 1 which is optimal. Thus, even for the setting that rewards are stochastic and transitions are deterministic, the problem becomes much harder.

—— To Reviewer #3 ——

Finite-horizon problems / exponential in the horizon length. We would like to remind the reviewer that the running time of our algorithm is polynomial in the planning horizon H instead of being exponential in H. See Theorem 4.1 for the precise statement. One can reduce discounted MDPs to finite-horizon MDPs by considering the first $\widetilde{O}(1/(1-\gamma))$ levels, and the H dependency in the complexity of our algorithm will be replaced by $\widetilde{O}(1/(1-\gamma))$.

Improve the presentation. We will revise the paper, make the proofs more readable and improve the presentation in general according to the reviewer's comments in the next version. In particular, we will explain the high-level ideas at the beginning of Section 4 and 5 instead of at the end.

—— To Reviewer #4 ——

Improvements to the presentation / care around the use of language / Broader Impact. We are grateful to the reviewer for providing detailed comments on the writing of our paper, and we will revise the paper according to the reviewer's comments in the next version.

"query" / "exponential number of values". We will make it clear in the next version that we are proving lower bounds on the number of times that the agent evaluates the objective function f. Here by a "query", we mean that the agent evaluates the objective function f on some specific input.

Literature on non-Markovian reward. We are grateful to the reviewer for providing a comprehensive list of papers on non-Markovian reward, and we are planning to add them into the related work section in the next version.