We want to thank all reviewers for their time and feedback.

To Reviewer #1:

* "Do you believe that your method could nonetheless be applied to more realistic settings?" **A1**: Yes. Our method is inspired by the influential empirical paper by [Tang et al. 2017] as we mentioned in Line 188-190.

* "Is it fair/accurate to say that you are efficiently exploring?..." **A2**: Our method does guide exploration. Our framework relies on a provably efficient tabular RL algorithm which has to have an efficient exploration component, such as UCB or Thompson sampling. We want to emphasize that a tabular algorithm that uses random policy, e.g., ϵ-greedy, *does not* have the desired properties we need because it requires an exponential number of samples. See the lower bound in Section A of [Jin et al. 2018]. We will add more clarifications on this point.

* "Why the reward function depends on the time step?" **A3**: This setting is more general than the case where the reward function is independent of the time step. This setting has also been studied in many previous works. We will add more clarifications.

* "Does your method support learning if H is unknown?" **A4**: Yes, our algorithm still works as long as we know an upper bound of H. We will add discussions about this. Thanks for pointing it out.

* "Are levels the same as time steps within the episode?" **A5**: Yes.

* "...one embedding for each level?" **A6**: In our description of the framework, we need one decoding function per level. But this is mainly for the convenience of analysis and is not necessary. Observation data can be mixed up among levels and used to train ONE decoding function for all levels. This is what we implemented in the numerical test. We will add more clarification about this part in the next version. Thank you for asking.

* "Why run N times?" **A7**: If we directly require every output policy (Line 11, Algorithm 1) to be near-optimal with probability $1-\delta$, the sample complexity will be $\mathcal{O}(1/\delta)$ instead of $\mathcal{O}(\log(1/\delta))$. To avoid this, we only require each output policy to be near-optimal with a parameter-free probability e.g., 0.99 and then we run $N = \mathcal{O}(\log(1/\delta))$ times to select the best policy out of N and achieve the final goal.

* "...continuous low-dimensional representations..." **A8**: We are using clustering methods on the low-dimensional embedding space to reduce it to a finite-state space. This is also the implementation in [Tang et at. 2017].

To Reviewer #3: Thank you for the positive review.

To Reviewer #4:

* "...the framework doesn't consider the probability that the ULO encodes inappropriately the observations." **A1**: In our definition of ULO (Definition 1), we do consider the probability that ULO encodes wrongly, which is described by the term $g(n, \delta)$. Besides, in the definition, the correctness of a function $\hat{f}$ is on-average, i.e., if for some state s, $\mu(s)$ is close to 0, $\hat{f}$ can be very inaccurate on s.

* "...the training of ULO is not explained in detail." **A2**: The training of ULO is described in Line 3-12 Algorithm 2. We also explained in Line 218-232. Due to the page limit, we defer some parts in the appendix. We use ULO as a black-box oracle for abstraction and generality. We discuss some specific examples in Section B. We will add some details to the main text in the final version. Thanks for asking.

* "...examine and/or discuss the applicability of the framework in more challenging environments." **A3**: We have more discussion in the appendix about, for example, the choice of ULO one can consider solving real-world problems. We will consider this in the next version.

* "...the label matching process and parts of Alg.2 should be explained in a more clear way. For instance, what is the purpose of the training data?" **A4**: Thanks for the advice. Due to the page limit, we defer some descriptions to the appendix. We will improve this part in the next version. For your question, the training data is collected to feed into the ULO and generate a more refined decoding function for each iteration.

To Reviewer #5:

* "...some curves lack error bars and clarification in the caption." **A1**: Thanks for mentioning this. We double-checked the graphs and we are sure that no curve lacks an error bar in the current version. They might be too tiny to observe. Yes, it is 1 standard deviation. We will add more clarification in the caption in the next version.

* "There are some small typos." **A2**: Thank you. We will double-check and revise them in the next version.

* "More can be added such as the difference between their setting and real-world settings." **A3**: Very good advice. We will consider this in the next version.