We thank all reviewers for detailed and valuable comments, and will revise the paper accordingly as described below.

Minor changes and typos. We thank all reviewers for pointing those out, and will do corrections in the revision.

R1 & R2: Discussions of deterministic MDPs in theoretical results. Although our theoretical results are derived in the context of deterministic MDPs, they are instructive for the practical algorithm design in general cases, and this assumption is also exploited by some prior work that investigate distance metrics in MDPs (e.g. reference [11] in the paper and the work of Castro mentioned by R1). Empirically, we have also showed that our method is robust to certain kinds of stochasticity and outperforms the best baseline in stochastic environments (see line 40 – 48 and Figure 1).

R1: Subgoals as short-term changes v.s. k-step adjacency constraint. Interpreting the subgoals as short-term changes amounts to setting a general constraint in the raw state space (e.g. HIRO) or in an embedding space (e.g. FeUdal networks). However, the Euclidean distance between states in these spaces instead of the adjacency space may not indicate the real adjacency relation: e.g. consider a grid-world environment where two states are separated by a wall. Also, these general constraints control the maximum magnitude of the changes by a human-specified hyperparameter, which is hard to choose in a principled way, while our method can learn the constraint automatically.

R1: Claim of the empirical results. We agree with the reviewer and will change the wording in the revision.

R1: Comparison with two recent works. (1) Castro proposes an algorithm for computing bisimulation metrics, which reflect behavioral equivalence between states, using sampled transitions. Compared to our work, bisimulation metrics depend on both the dynamics and the rewards, while the shortest transition distance depends only on the dynamics and therefore can be easily applied to multi-task settings. (2) Khetarpal et al. present a theory and an algorithm of affordances in RL, formulating the fact that certain states only enable certain actions. They construct the affordances based on indents, i.e. desired state distributions, which are specified by humans a priori. In contrast, our method learns subgoals, which can be interpreted as a kind of temporally extended indents, by RL rather than human prior.

R2: Comparison with Options and EigenOptions with successor representation (SR). (1) The Option framework maintains a finite set of low-level Options (macro-actions), while our method (which falls into goal-conditioned HRL) maintains an universal goal-reaching low-level policy whose behavior is modulated by subgoals. As shown in the HIRO paper, goal-conditioned HRL often yields better performance than HRL with Options. (2) SR can be used to measure the temporal distance between states and discover eigenoptions. Compared to our work, SR depends on both environmental dynamics and a specific policy, while the shortest transition distance relies only on the dynamics.

R2: Experimental settings. We followed HIRO to use off-policy TD3 in continuous control tasks; in discrete control tasks we found that whether using on-policy or off-policy methods (e.g. double DQN) does not make much differece.

R3: Comparison with graph-based methods (missing baselines). We believe that our method has essential difference with graph-based methods: our method models high-level learning as a RL process and thus can tackle more general problems, while current graph-based methods derive high-level policy *without* a training process, exploiting specific problem structure. E.g. all graph-based works cited in the review obtain the subgoal sequence by solving a shortest-path problem to a *known* goal node in a high-level graph (e.g. using Dijkstra's algorithm), which cannot be applied to more general problems where there does not exist a single "goal" state (e.g. PointGather) or the goal state needs to be explored by the agent instead of being given in advance (e.g. KeyChest), as in many of our experiments. To the best of our knowledge, none of these graph-based works has reported results on these general problems without additional task-specific prior. Therefore, we found it hard to fairly compare our method with graph-based methods in experiments and thus did not add them to baselines. In the revision, we will add these discussions to the related work section.

R3: Empirical study in stochastic MDPs. To empirically verify the stochasticity robustness of our method, we have applied HRAC to a set of stochastic AntMaze tasks, which have relatively larger state (30-d continuous) and action space (8-d continuous) than the Maze task. We added Gaussian noise with different STDs (σ) to the (x, y) position of the ant robot at every step. As shown in Figure 1, HRAC achieves similar asymptotic success rates with different noise magnitudes. Due to the time limit, we only compare HRAC with the best baseline HIRO when $\sigma = 0.1$, representing the noise magnitude that approximately equals to 20% of the maximum step size of the ant robot on average, where HRAC achieves better performance than HIRO. We will add more experimental results in stochastic environments to the revised paper.

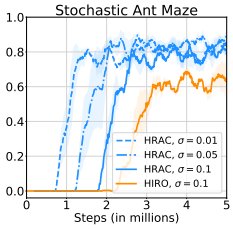

Figure 1: Learning curves.

R3: Difference between HRAC and NegReward. HRAC plugs the adjacency loss as an extra term into the loss of a specific RL algorithm, rather than treat it as a negative reward.

R4: Explanation of the approximation in Eq. (9). In Eq. (9), we apply a minimizing operation instead of an expectation operation over a finite policy set to approximate the original minimizing operation over the whole (indefinite) policy set. Therefore, we do not require that these policies are drawn i.i.d..