Response to Reviewer 1: Thank you for the thoughtful and inspiring comments.

Q1. How much of an effect does low-level skill initialization scheme have on performance?

We test random skill initialization in Ant Maze task. Even random initialization is better than the non-hierarchical method TRPO, shown in Fig.5 below. And more reasonable initialization results in better performance. We will explore the effects of other initialization schemes in future works.

Response to Reviewer 2: Thank you for the detailed comments.

Q1. Consider potential based reward shaping in main evaluation.

We design a heuristic potential as the negative L2 distance between agent and goal. The curve of potential reward is higher than TRPO, but significantly worse than HAAR, shown in Fig.5. We note that the potential reward shaping method could not take advantage of hierarchical structure; it also heavily depends on the potential function design.

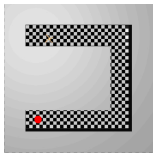

Figure 1: Task 1

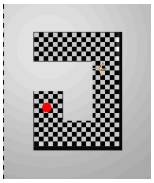

Figure 2: Task 2

Q2. More robust and systematic transfer experiments.

We design more new tasks in Fig.1(bigger maze), 2(sprial maze) to explore the effectiveness of low-level skill transfer. In the new tasks, the skill of turning right learned in Ant Maze (a) can be very useful, so low-level policy transfer shows much efficiency (shown in Fig.3, 4). New tasks and the old task share much information on the high level, so "transfer both" performs well, which partly comes by virtue of the transferable state representation (Konidaris, 2007).

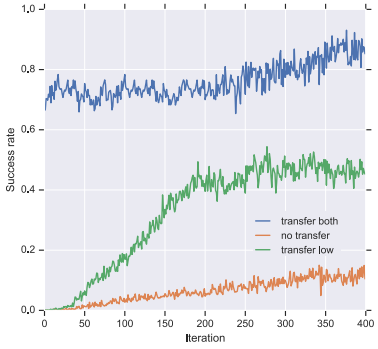

Figure 3: Transfer results for task 1.

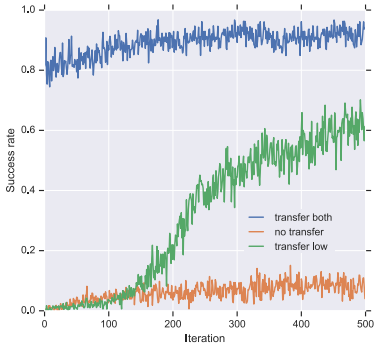

Figure 4: Transfer results for task 2.

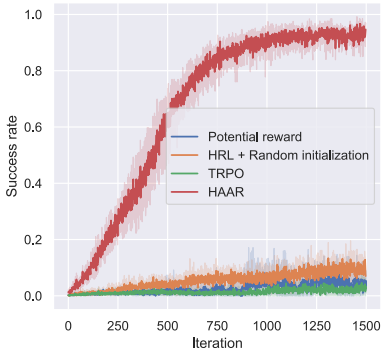

Figure 5: Potential based reward shaping and random low-level skill initialization in Ant Maze.

Q3. All assumptions for the proof of monotonic improvement.

(1) All assumptions for TRPO, including that the high-level and low-level states are Markovian; (2)The high level policy is fixed while optimizing the low level policy and vice versa [lines 116-7]; (3) Discount factors $\gamma_h \to 1, \gamma_l \to 1$ [lines 132-3].

Q4. Experiment setting seems to be non-Markovian [line 224-5], different states may have very similar representation.

In experiment, the agent uses a total of 20 rays to "see" the surroundings, and the goal can always be seen regardless of walls. We believe this is sufficient to distinguish between states, so it is approximately Markovian.

Q5. The benchmark, SNN4HRL, seems to run much faster in its original paper compared to in this paper.

The reviewer may have misread the experiment settings. Our result is actually consistent with the original SNN4HRL paper. In Swimmer Maze, the numbers of samples per iteration are different in SNN4HRL paper and our paper [line 405]. The performance in terms of samples is consistent. For Ant Maze, SNN4HRL paper does not provide results.

Q6. A precise description of the advantage function.

Our definition of advantage is consistent with the conventional definition, $Q(s,a) - V(s)$. Using a one-step expansion of Q, we can write it as $A_h(s^h) = E_{s^h_{t+k} \sim (\pi_h, \pi_l)}[r^h_t + \gamma_h V_h(s^h_{t+k}) | a^h_t = a^h, s^h_t = s^h] - V_h(s^h)$.

Q7. How to determine what to include in low-level state in experiments?

Our decision of what is included in low-level state is the same as SNN4HRL paper (such that the representation requires minimal domain knowledge in the pre-training phase), described in "Problem Statement" of their paper.

Response to Reviewer 3: Thank you for the thoughtful comments.

Q1. How will the algorithm perform when starting with random low-level policies?

We run experiments with random initial low-level policies in Ant Maze. Results in Fig.5 show that it performs better than TRPO. As expected, more meaningful low-level policies result in better performance.

Q2. Low level Markovness is not clear. Discuss the Markovness of states.

States for both high level and low level are Markovian. We concatenate the agent state and the high-level action a^h as the low-level state, so low-level policy is still running on an MDP.