We thank the reviewers for the thorough feedbacks. Based on those, we have made numerous improvements.

Implement a new IM baseline: ICM (Pathak 2017 [23]. Original code is for decrete actions.) As suggested by *reviewer #1, #3*, we have implemented ICM for the synthetic environment (Sec.4, Fig. 3 of the manuscript). The ICM baseline uses SAC with an augmented reward: $r_t = r_t^{\text{ex}} + \alpha r^{\text{in}}$, where r_t^{ex} is the extrinsic reward (negative distance to goal) and r_t^{in} is an intrinsic reward.

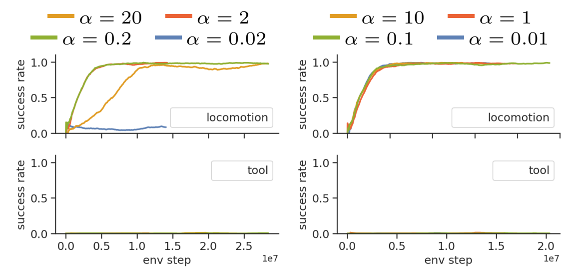

Fig. 1: Synthetic environment in Sec. 4. Left: prediction error; right: surprise. α is a hyperparameter we scanned for.

The first experiment (Fig. 1 Left) follows the original ICM, where the intrinsic reward signal is given by the total prediction error: $r^{\text{in}} = \sum_i e_i(t)$, where the sum is over all goal spaces/coordinates. Furthermore, we adapted ICM to make use of the surprise signals that have shown to be important in the manuscript. Thus, in a second experiment (Fig. 1 Right), the intrinsic reward is given by the surprise signal: $r^{\text{in}} = \max_i \text{surprise}_i(t)$, where max is over goal spaces. Despite scanning the hyperparameter α, both IM baselines perform poorly and only solve the locomotion task, see Fig. 1. Despite the seemingly simple environment, a random encounter of objects in continuous control is rare, given an agent with heavy mass and a large arena.

To address *rev. #2*'s concern over "object can't be moved, a model-error driven IM will stop", we first clarify that the issue, in fact, lies with the "random object" (in Sec. 4), not an unmovable object. We further tested the above-mentioned IM baseline with the random object. The plot is similar to "tool" in Fig. 1 and we omit it due to space constraints.

Clarify novelty and main contributions We agree that each individual component is not original, as we have clearly indicated they are from task-motion planning, IM, RL communities. We have already given references in the manuscript (including Klyubin and Battaglia's work(s) mentioned by *rev. #1*). But combining them to successfully solve the continuous control and robot trajectory optimization problem is novel (cf. *rev. #3*, originality).

Rev. #1 suggested that the environments could be solved by classic planning methods. If one has an environment model with an analytically (or accurate numerical) gradient, iLQR(G) may (without guarantee) solve the nonlinear program (NLP). We have discussed this and other planning ideas (e.g. PRM) in the related work section. However, this paper is based on model-free RL to solve the robot trajectory optimization through contact. We demonstrated IM/RL can solve this as an alternative to NLP/sampling-based planning. This is beyond the scope of existing works such as Klyubin et al.

It is true that our method shares certain points with the concept of empowerment. We would like to emphasize that the structure that we proposed leads to more efficient learning while maintaining the idea of maximizing controllability.

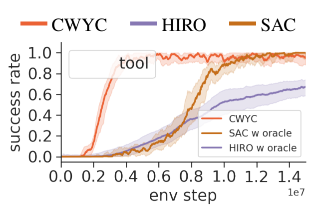

Fig. 2: Baselines with reward-shaping

Concerning the complexity of our method We acknowledge that the original Fig. 1 suggests an overwhelming complexity due to the detailed break-down (we will simplify this). In fact, our *inductive bias (c.f. [Tenenbaum (2011) "How to grow..."]) has only 3 modules (not 8)*: the task selector, planner, and subgoal generator. All other modules are common among RL algorithms. In the ablation studies, we demonstrated that every component is required to solve the task/maintain data efficiency. To further validate this claim, we report additional results in Fig. 2, where the baselines are able to learn the tool task with a hand-engineered reward: $r_t = r_t^{\text{ex}} - \text{dist}(\text{agent-pos}_t, \text{tool-pos}_t)$. Therefore, our method in fact removes this additional layer of supervision.

Further improvements. *Code is uploaded* to the website as given in the paper. Concerning our argument for playfulness, see [Smith (2005) "The dev. of embodied..."; Ryan (2000) "Intrinsic and extrinsic..."]. Regarding prediction error vs. learning progress: prediction error fails in stochastic environments, see [Oudeyer (2007) "Intrinsic motiv. systems..."; Burda (2018) "Large-scale..."].

Q: subgoal attention requires attending over all possible goals...? **A:** Our specific form of the goal generation network allows for a closed-form solution to compute the argmax of the function. **Q:** The task graph is not a function of the particular goal in the final task...? **A:** True. A limitation of our current architecture. **Q:** Goals within one task have different difficulty. **A:** True. Interesting future direction. **Q:** When is the transition between sub-tasks happening...? **A:** Your understanding is correct. If the goal can not be reached, the rollout is terminated after the maximum timesteps per rollout is reached. We clarify this.

All text errors or vague language will be fixed. We have addressed other review comments but omit reporting them here due to the space constraint. We gratefully acknowledge your help in improving the work.