Generalization to Real Robots

We were unable to validate our method on real robots due to the current COVID-19 pandemic. We believe our proposed method, LOOP, will also generalize to real-world robots for the following reasons.

LOOP achieves better simulation results than prior work which has been demonstrated to work on real robots.

More sample efficient than SAC: SAC [5] has been demonstrated to work well on real robots and is widely used in a number of robotics application [2, 60, 61]. We build upon SAC without making any algorithmic assumptions or having additional requirements (such as using a ground truth dynamics model). Also, Figure 3 shows that LOOP is consistently more sample efficient than SAC. This implies that it requires less deployment time of robot in real world and thus decreases wear-and-tear to the hardware. Hence we believe LOOP should generalize to real robots with the added improvement in sample efficiency.

Less runtime than PDDM: PDDM [21] was demonstrated to work on physical robots. In terms of trajectory optimization, LOOP optimizes for a shorter horizon utilizing its terminal value function and can achieve a better runtime performance on the real robot. In addition, PDDM also requires more hyperparameter tuning than LOOP as it sets different hyperparameters for different environments.

LOOP is evaluated over a wide range of simulated tasks and RL domains.

LOOP outperform the baselines across a range of simulated tasks consisting of Locomotion, Manipulation and Navigation experiments. Section 6 shows that LOOP shows improved sample efficiency in both the Online RL and Safe RL domain, making it a preferable choice to deploy with physical robots. Offline RL is an important problem in robotics as it allows for learning behaviors from static, previously collected datasets. LOOP shows versatility by improving over the performance of the parameterized actor in Offline RL as well.

LOOP uses more interpretable policies that are more amenable to safety increasing its utility in real world applications

LOOP uses semiparameteric policies in the form of H-step lookahead. This way of online planning allows to interpret and reason about the outcome of the policy, while giving us flexibility to incorporate constraints during deployment. Incorporating constraints during deployment is important in physical robots where constraints could be safety considerations (possibly non-stationary). Online planning also allow LOOP to adapt faster to changing dynamics due to wear and tear of robot or any other phenomena.

LOOP requires less hyperparameter-tuning for each task compared to model-based alternatives.

Tuning hyperparameters in real world requires multiple environment runs for the experiments and is a non-trivial task. LOOP for Online RL works with a single set of hyperparameters across all environments and did not require much hyperparameter tuning. In contrast, other ways of utilizing model in off-policy methods like MBPO relies on hyperparameters specifically tuned for each environment as can be seen in Appendix C from [13]. The ease of hyperparameter tuning in LOOP will likely make it easier to be applied to real-robots compared to the previous works.