Skill Preferences: Learning to Extract and Execute Robotic Skills from Human Feedback

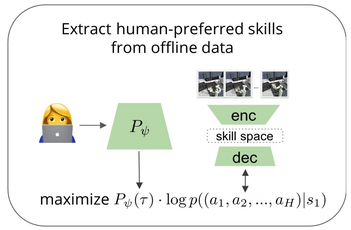
Xiaofei Wang, Kimin Lee, Kourosh Hakhamaneshi, Pieter Abbeel & Michael Laskin
University of California, Berkeley

1 Problem: Solving Long-Horizon Robotics Tasks

➢ We are interested in the following research question - **how can we learn robotic control policies that are aligned with human intent and capable of solving complex real-world tasks?** Some challenges includes the tasks being long-horizon and have only sparse reward. Human-in-the-loop RL has emerged as a promising approach to better align RL with human intent. To scale to more complex long-horizon tasks, a number of recent works have proposed data-driven extraction of behavioral priors, which we refer to as skills.

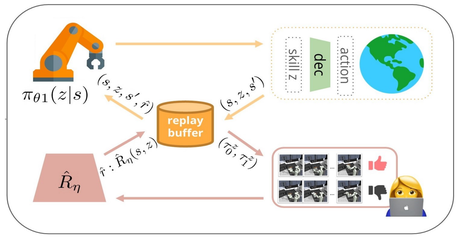
➢ In this work, we introduce **Ski**ll **P**references (SkiP), an algorithm that integrates human-in-the-loop RL with data-driven skill extraction

2 Method: Extract and Execute Skill with Human



Skill Extraction with Human Feedback

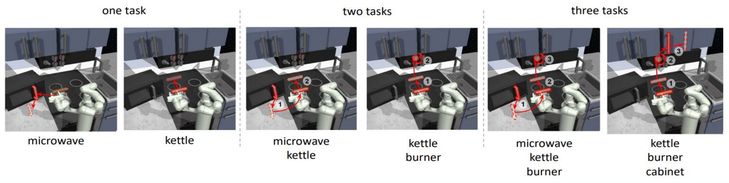
➢ In this offline phase, SkiP learns a VAE that encodes action sequences into skill latent conditioned on the starting state with the ELBO loss weighed by human feedback



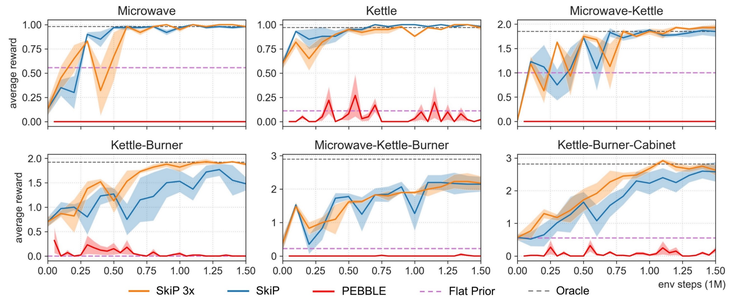
Skill Execution with Human Preference

➢ The orange loop is the RL learning loop over skill latents, and the pink loop is the "human loop". In the pink loop, it learns a reward model from human preference and use it as the reward function

3 Environments: manipulation tasks of varying difficulties



4 Experiments: Compare SkiP (orange&blue) with baselines



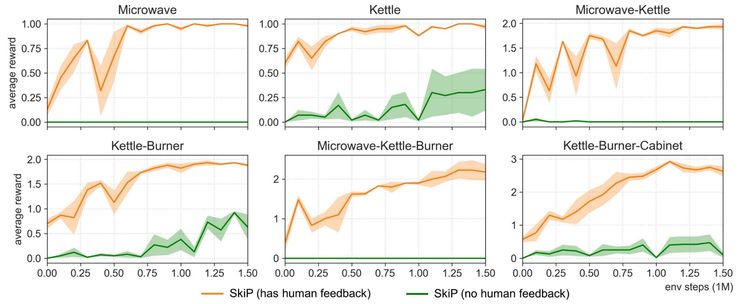
➢ SkiP3x: SkiP with 3 times more human labels

➢ PEBBLE: state-of-the-art human preference RL method

➢ Flat Prior: learns an action prior on the atomic action space over the optimal dataset and trains an SAC regularized with that action prior over ground-truth reward.

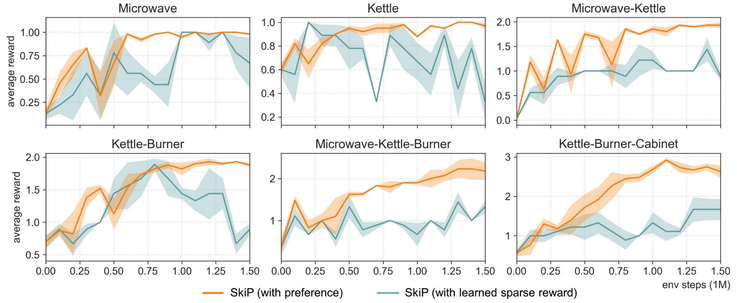
➢ Oracle: SPiRL

5 Is it necessary to provide human feedback during skill extraction?



➢ The method that extracts skills without human feedback is unable to solve any of the tasks, suggesting that human feedback is essential for skill extraction from suboptimal offline data

How should we incorporate human feedback during the skill execution phase?



➢ RL with a reward classifier for sub-task completion is able to solve some tasks but generally performs much worse than RL with human preferences