Composing Skill Primitives for Long Horizon Activities

Stanley Cao, Michael Lingelbach, Jiajun Wu

Stanford Vision and Learning Lab, Computer Science, Stanford University

Background

Training agents from end-to-end with reinforcement learning requires good demonstrations of action, state pairs that solve a task starting from scratch.

This gets especially difficult when our action space (e.g., controlling a robot in 3D continuous space) and/or observation space (e.g., RGB, large scenes) is complex.

Imitation learning (IL) is a set of techniques for training agents from human demonstrations, but on long horizons, IL breaks down as the policy fails to recapitulate human behavior, ending up in states that have no corresponding demonstration in the training set.

It's difficult to train a "do-all" imitation learning policy, especially for tasks that have many steps (e.g., making coffee involves heating coffee, pouring, etc).

- Collecting demonstrations for these tasks is time consuming; we need many people to go through an entire task like making coffee.
- Moreover, if we desired to leverage those demonstrations for another task, we would have to re-train the entire policy; we cannot extract individual primitives (e.g., boiling water) out of an IL model trained end-to-end on complex tasks (e.g., making coffee).

Can we decompose a task into sub-trajectories corresponding to individual subgoals in order to make a composable set of skill primitives trained with IL?

Methodology/Proposed Experiment

Environment: A 10 × 10 grid world task where the agent is supposed to navigate to 5 sequential goals in a given order. The agent receives as input the target goals, the grid observations, a reward of −1 for every timestep that it does not reach a goal, and a reward of +10 when it reaches a goal.

We trained a behavioral cloning agent by selecting datasets of the agent navigating from a random point in a randomly initialized scene to target A, B, C, D, and E respectively, and fitting specific subpolicies to accomplish each goal.

After training the 5 subpolicies, we compose them using a high level planner, which continually uses policy 1 to navigate to target A, then uses policy 2 to navigate to target B, and so on.

In parallel, we train an IL agent that attempts to navigate to all 5 targets directly from the demonstrations.

We compare both of these agents against the A* search algorithm, which acts as an optimal planner.

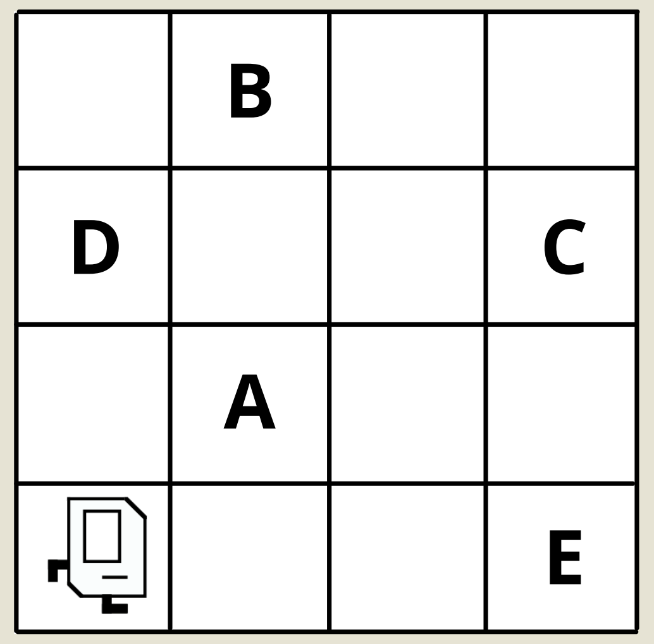


Figure 1: Example grid world with 5 goals. Agent needs to reach points A, B, C, D, and E, in that order.

Results

Figure 2: 200 epochs of training one policy over a long horizon. Same set of evaluation episodes are run through A*.

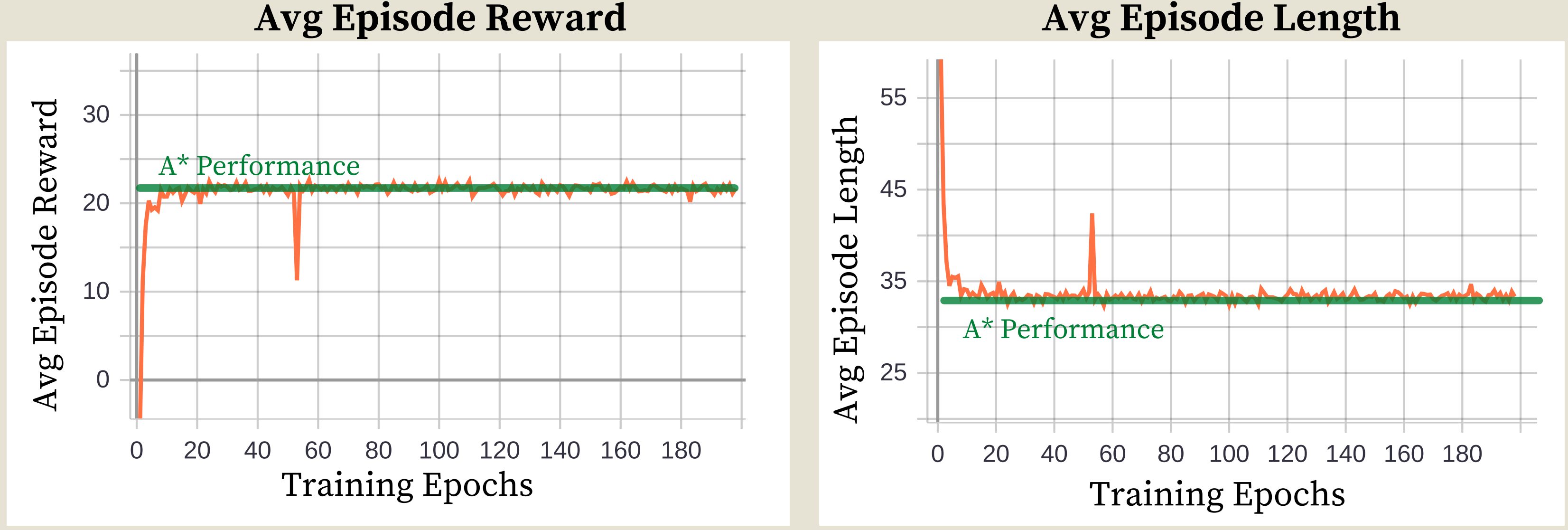


Figure 2a: Monolithic Policy Approach, Avg Episode Reward

Figure 2b: Monolithic Policy Approach, Avg Episode Length

Figure 3: 20 epochs of training each of the 5 subpolicies. Same set of evaluation episodes are run through A*.

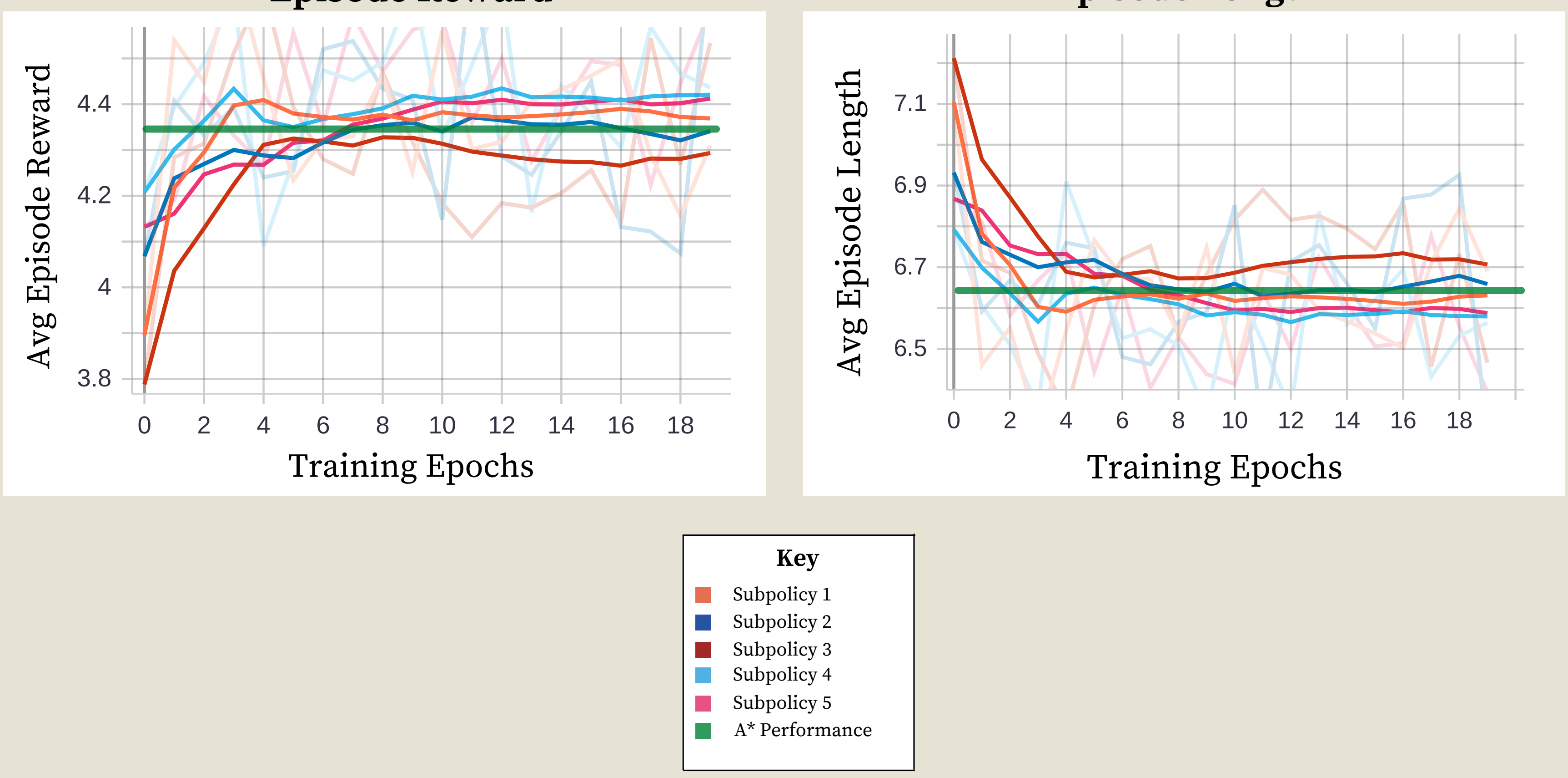


Figure 3a: Skill Primitive Approach, Avg Episode Reward

Figure 3b: Skill Primitive Approach, Avg Episode Length

Figure 4: Average episode reward on 1000 episodes after training.

	Mean Episode Reward
Monolithic Policy Approach	21.665 ± 10.216
Skill Primitive Approach	21.843 ± 8.020
A* search algorithm	21.868 ± 8.021

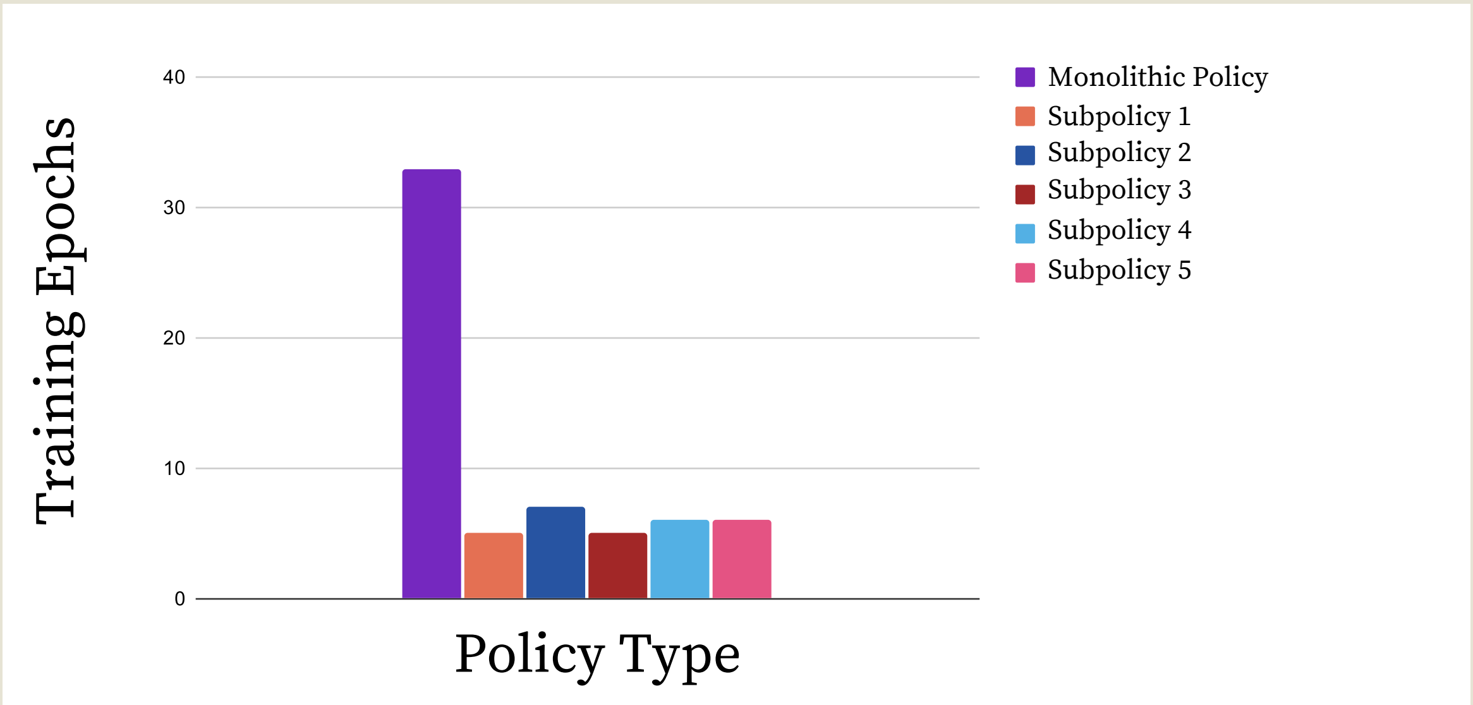
Discussion

In order to track training, after every training epoch, 500 evaluation episodes are given to the IL agents (both the long horizon policy and the 5 subpolicies) to navigate, and the average episode reward and lengths are graphed. These same 500 episodes are given to A* search algorithm to compare performance.

From figure 5, we observe that when we try to train one policy on a long horizon, it generally converges to the optimal policy by epoch 33, and that when we try to train 5 subpolicies on shorter horizons, they all converge to their respective optimal policies within 5-7 epochs.

Despite the significantly shorter training time, the high level planner using 5 subpolicies achieves comparable performance to the end-to-end trained agent.

Figure 5: Number of Training Epochs before Converging to Optimal Policy



Future Work

- See if this result holds for larger, multidimensional grid sizes and more target goals
- Apply the same approach to more complex environments

References

1. Kipf, T., Li, Y., Dai, H., Zambaldi, V., Sanchez-Gonzalez, A., Grefenstette, E., ... & Battaglia, P. (2019, May). Compile: Compositional imitation learning and execution. In International Conference on Machine Learning (pp. 3418-3428). PMLR.
2. Paul, S., van Baar, J., & Roy-Chowdhury, A. K. (2019). Learning from trajectories via subgoal discovery. ArXiv:1911.07224 [Cs, Stat]. http://arxiv.org/abs/1911.07224
3. Tanwani, A. K., Yan, A., Lee, J., Calinon, S., & Goldberg, K. (2021). Sequential robot imitation learning from observations. The International Journal of Robotics Research, 027836492110327. https://doi.org/10.1177/02783649211032721