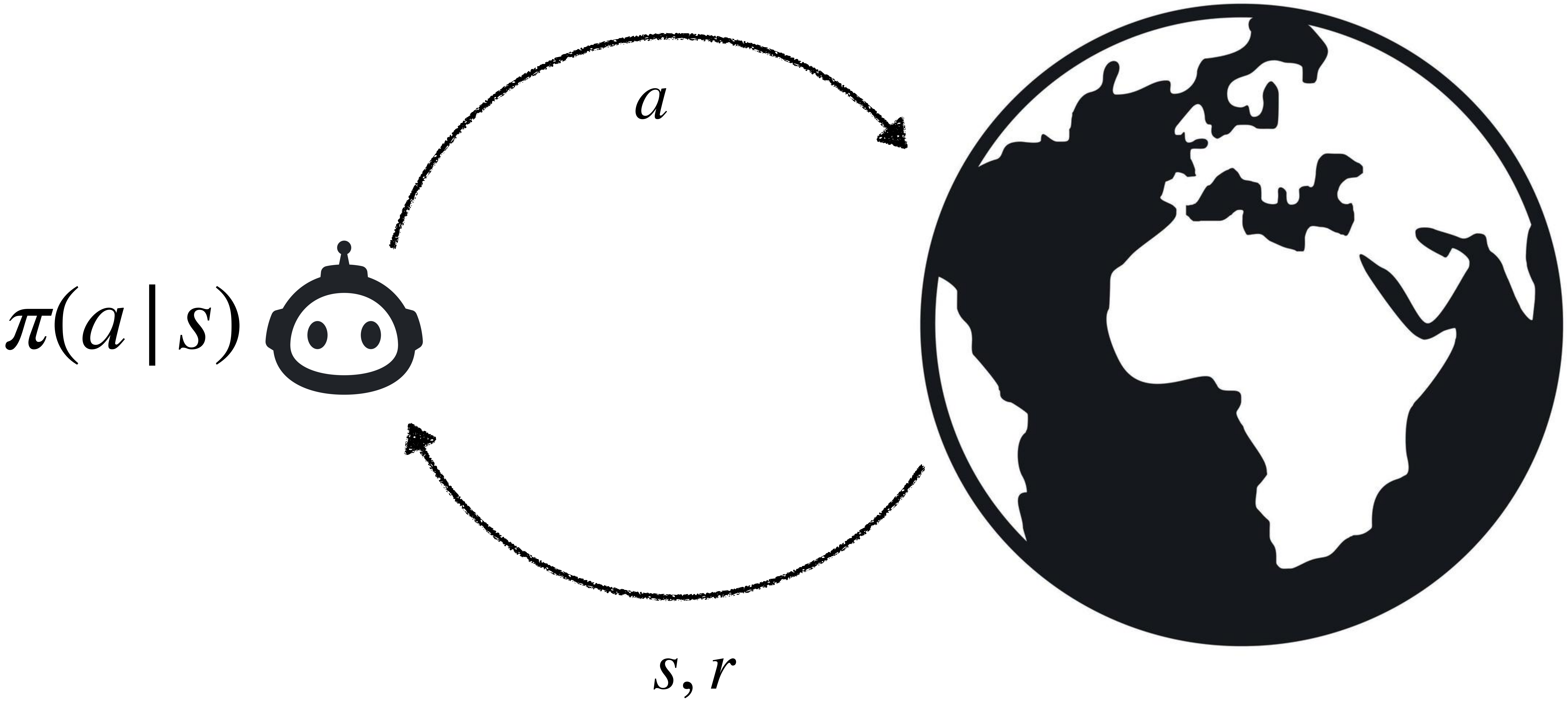
Goal-Space Planning with Subgoal Models

The Problem Setting



The Problem Setting

$$\left.\begin{array}{c} \langle S_t, A_t, R_{t+1}, S_{t+1} \rangle \\ \\ \pi_t(a \mid s) \end{array}\right\} \quad \pi_{t+1}(a \mid s)$$

How can we make the most of our experiences?

Background Planning

Background Planning

- “Planning is any computational process that uses a model to create or improve a policy.” - Sutton & Barto, 2018

- “A model is anything the agent can use to predict how the environment will respond to its actions.” - Sutton & Barto, 2018

Dyna

$$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle \xrightarrow{\text{RL Update}} \pi(a \mid s)$$

Model Learning

$$\langle s, a, r, s' \rangle$$

Planning Update $\times\, n$

(Sutton, 1991)

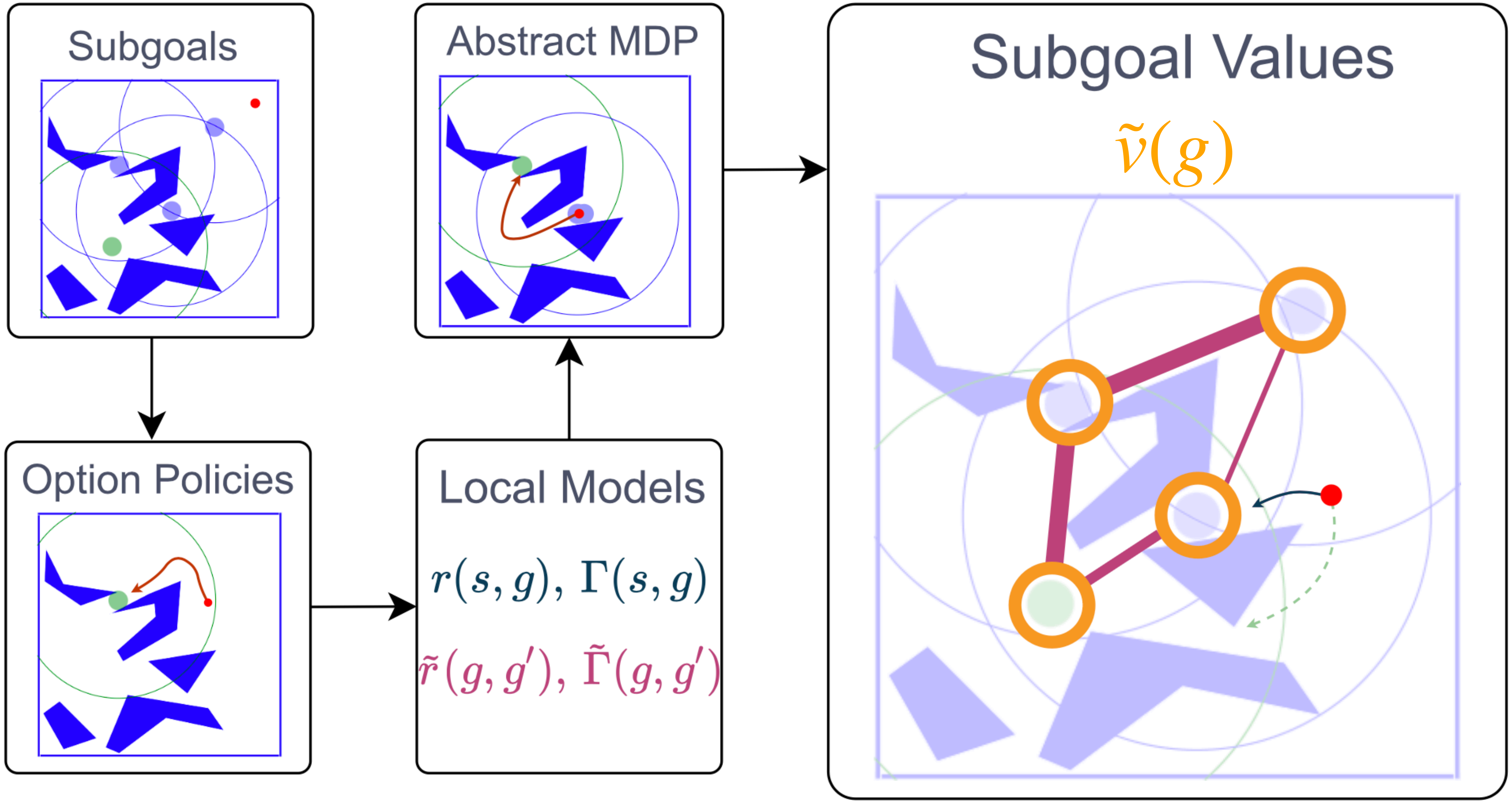
Challenges with Learned Models

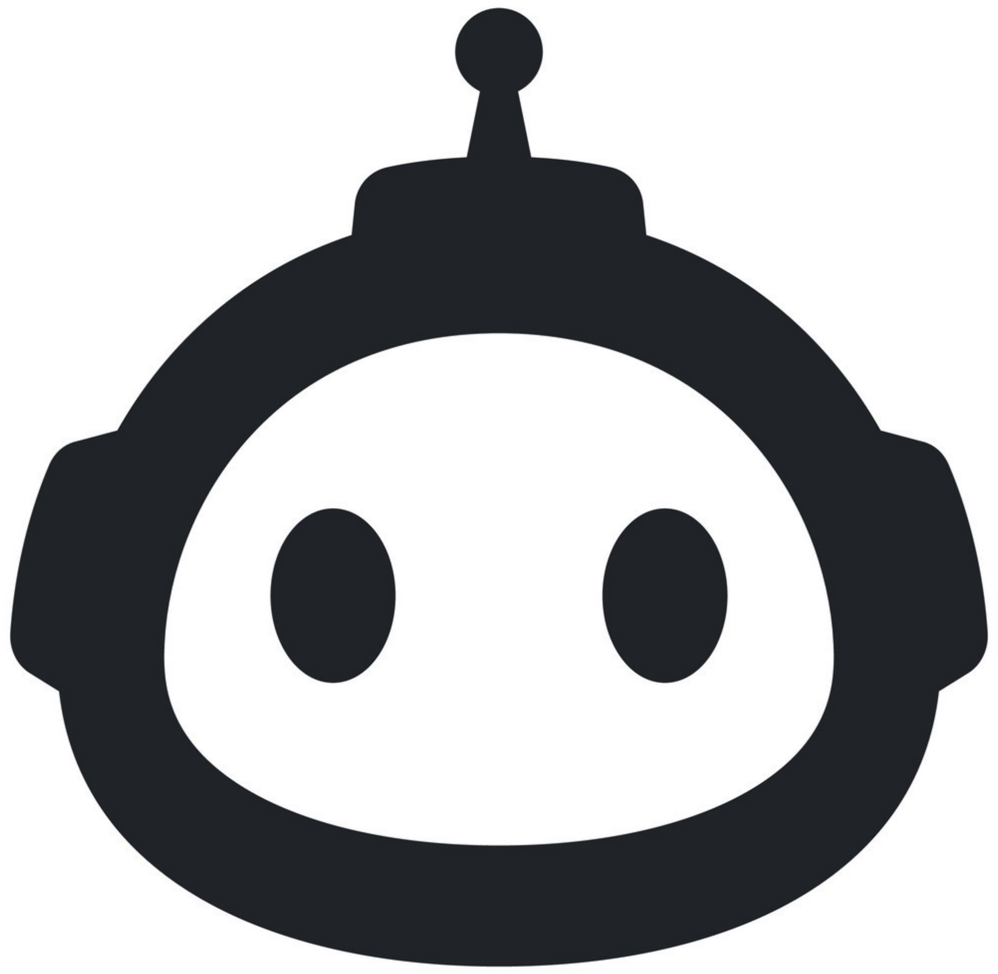
- Exhaustive planning updates are expensive in huge state spaces
- Compounding error and invalid states when planning over longer horizons
- Doesn't directly prioritize what transitions to use
- Expensive to learn a world model, especially in high dimensional tasks
- Vanilla Dyna models a deterministic environment: $s, a \rightarrow r, s'$

Can we avoid some of these problems with background planning?

Abstract Models

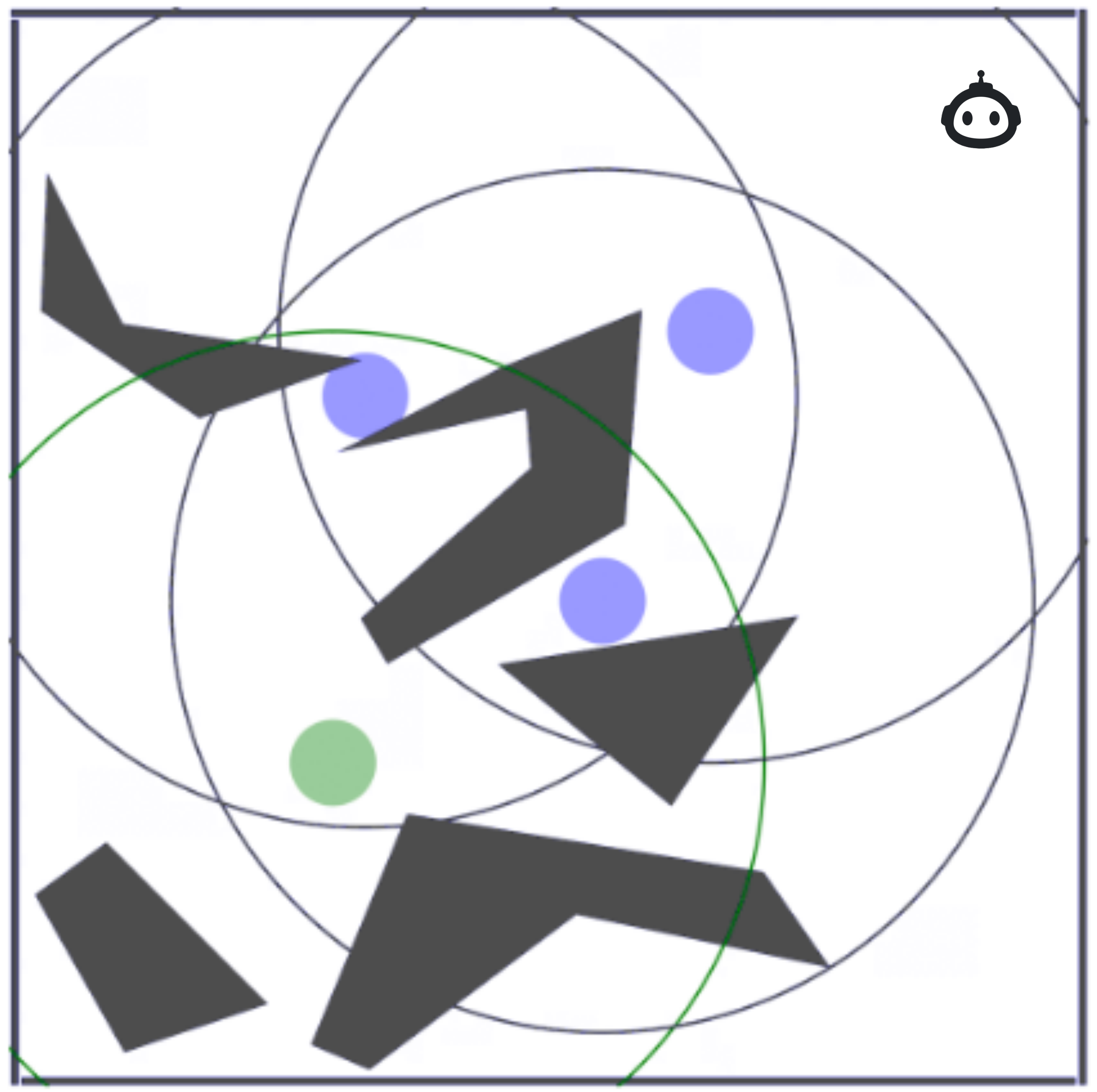
Our Approach: Goal Space Planning

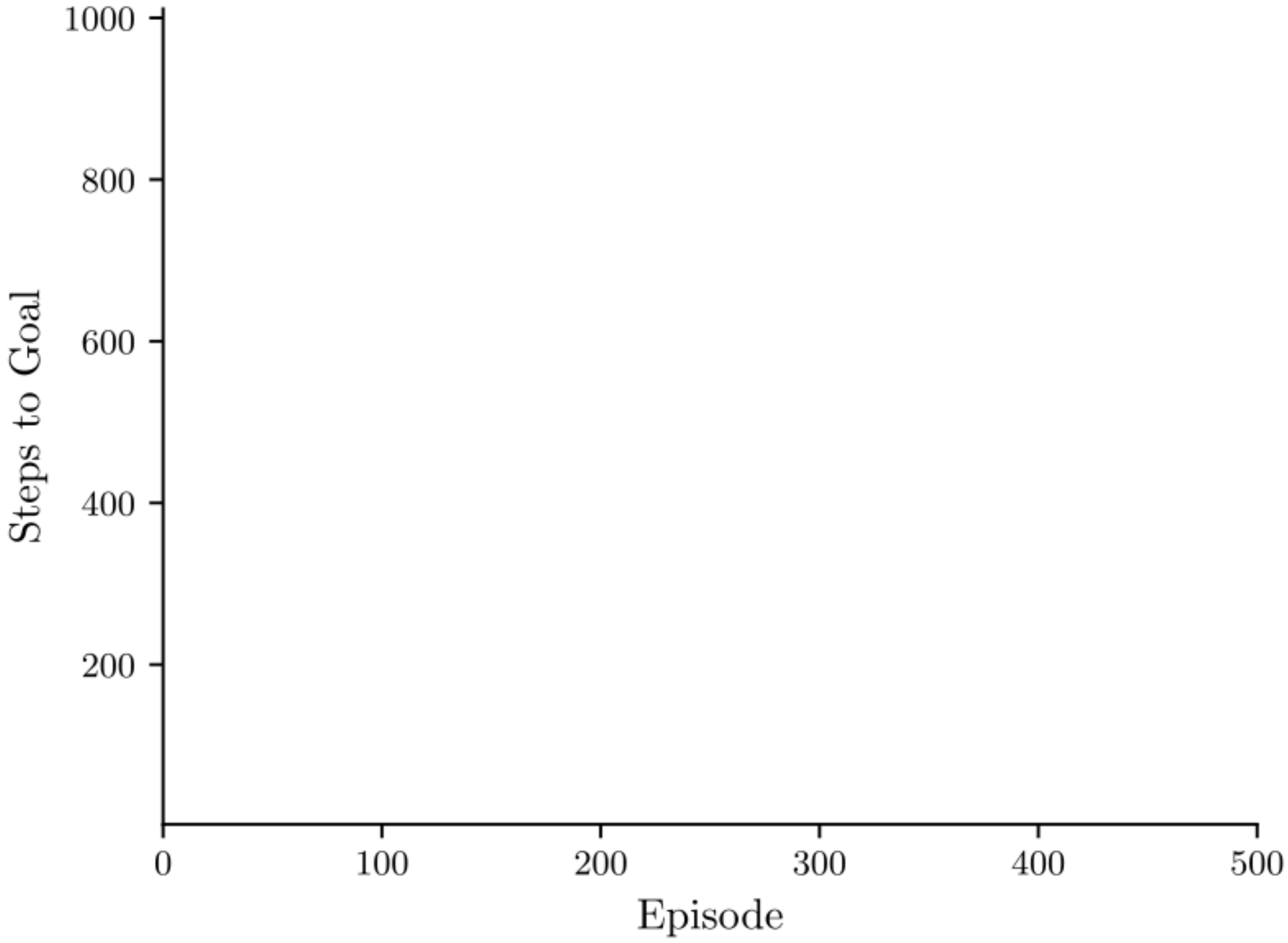




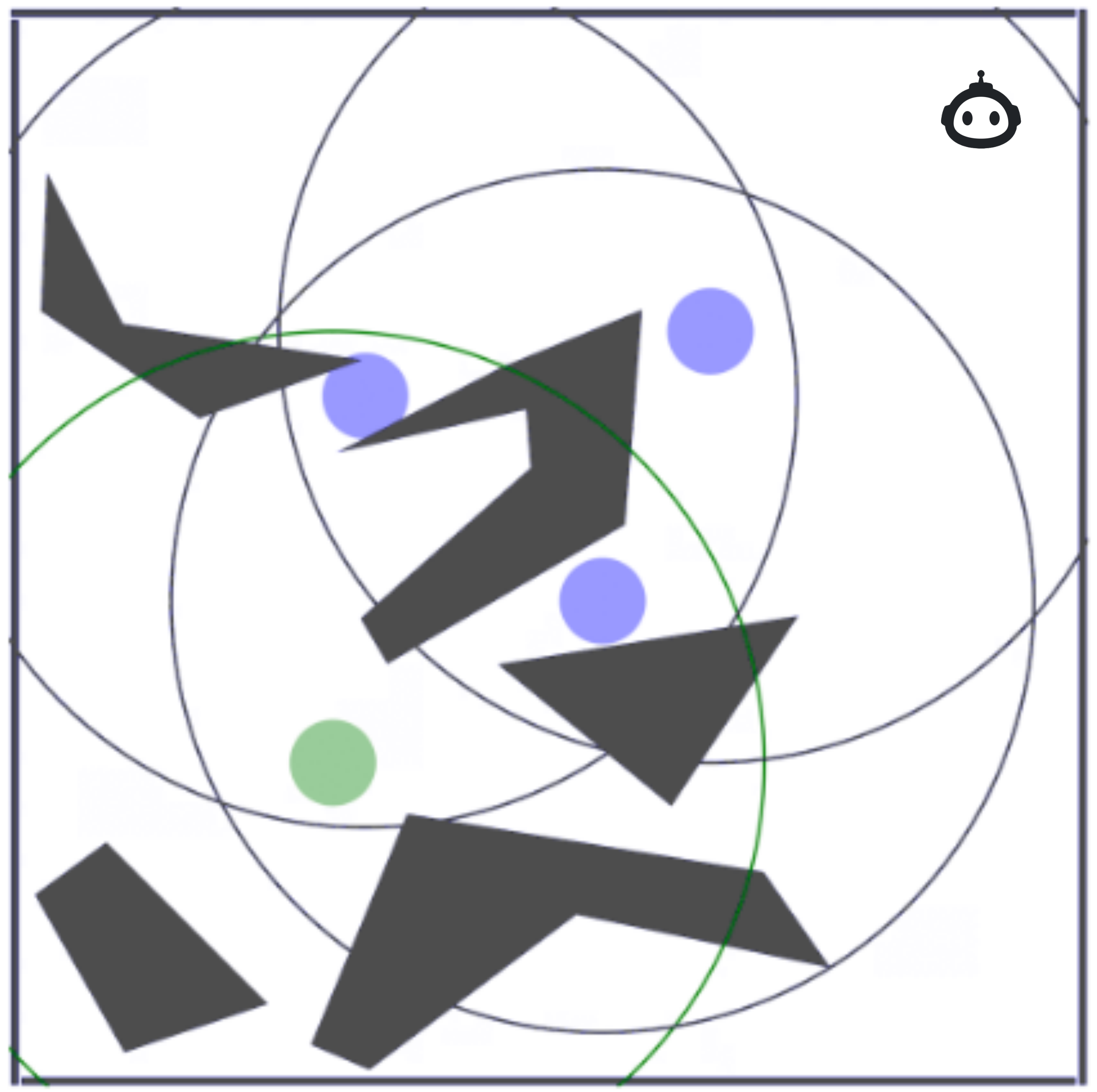
Results

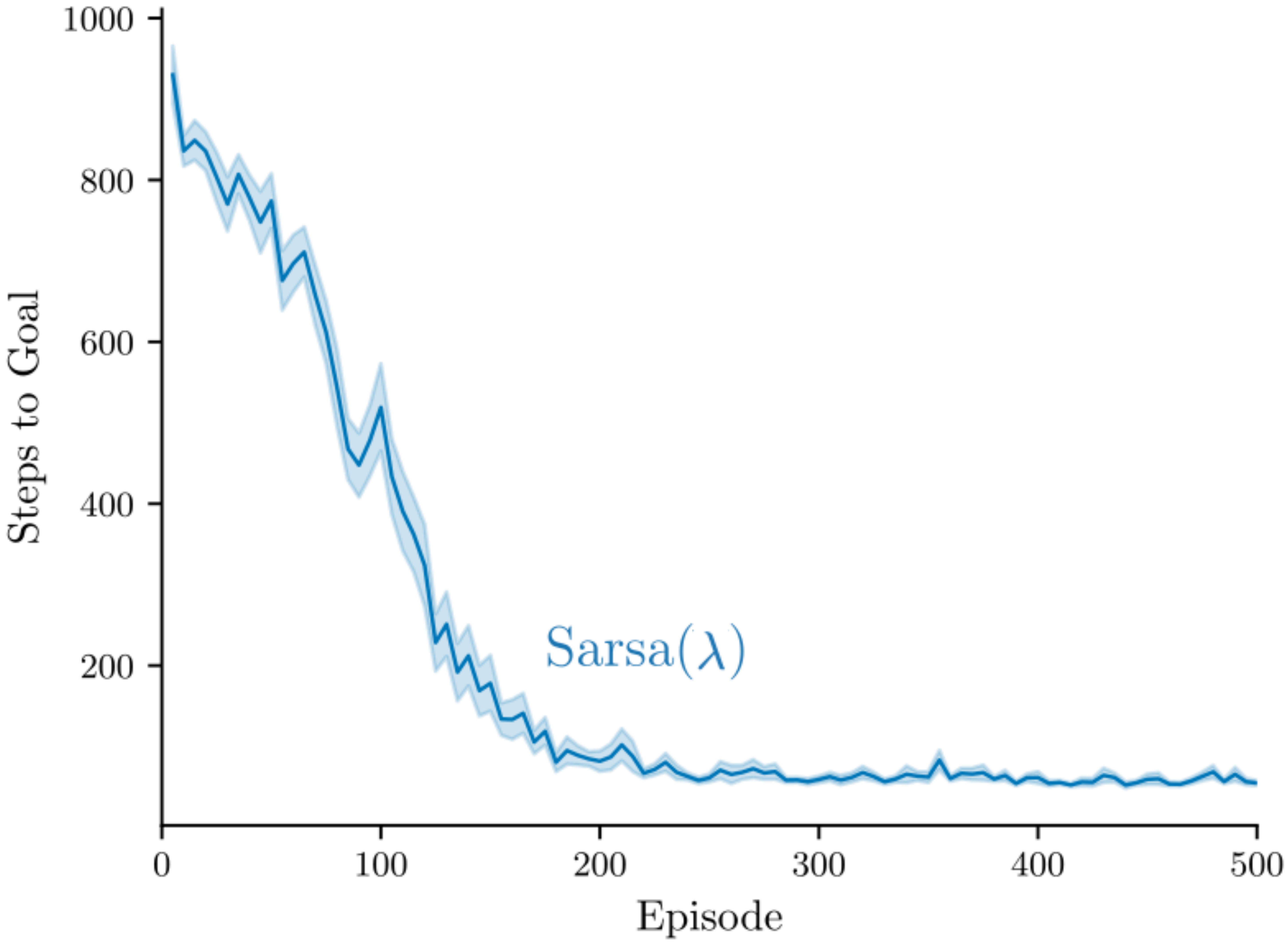
Results: PinBall



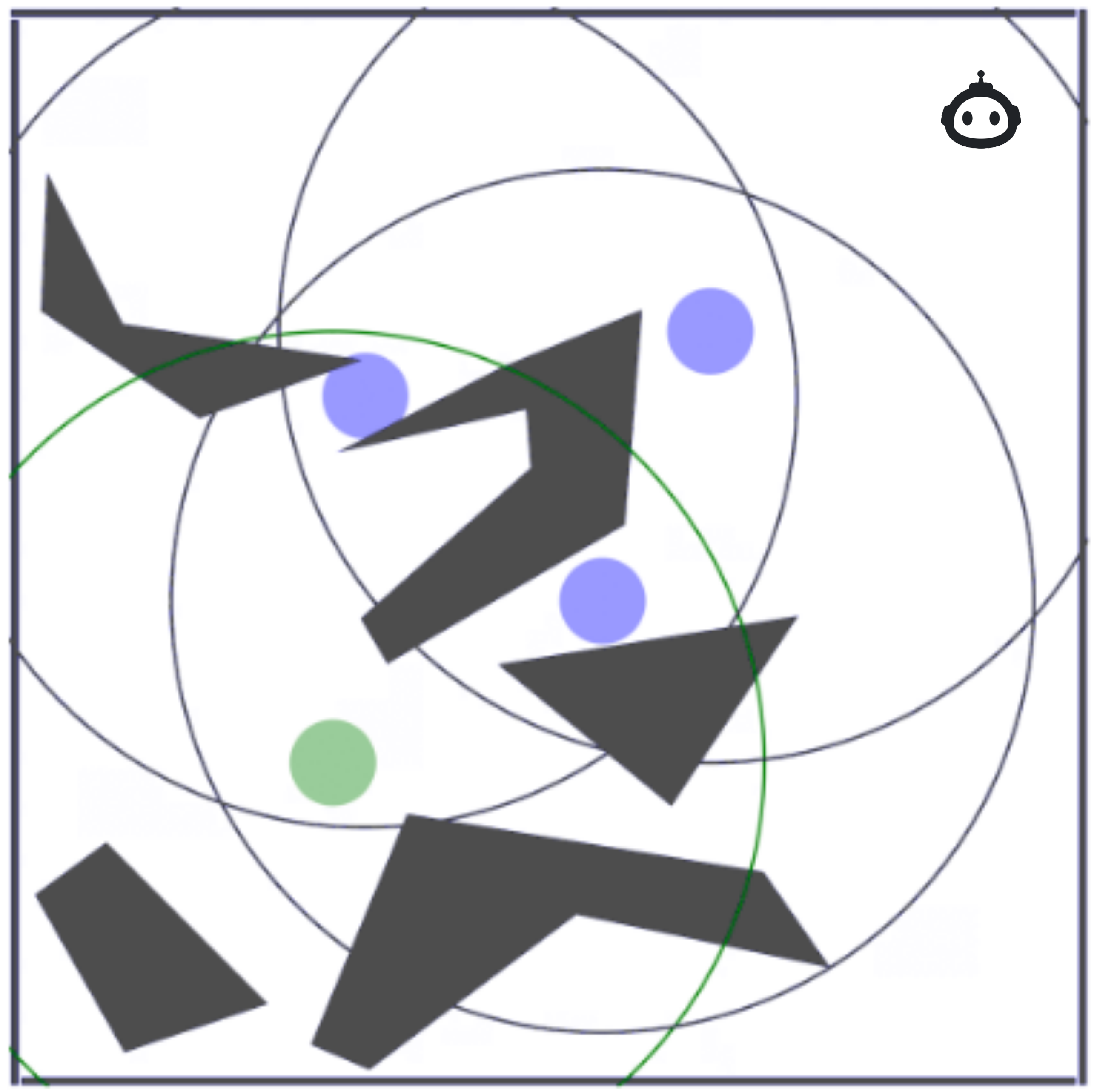


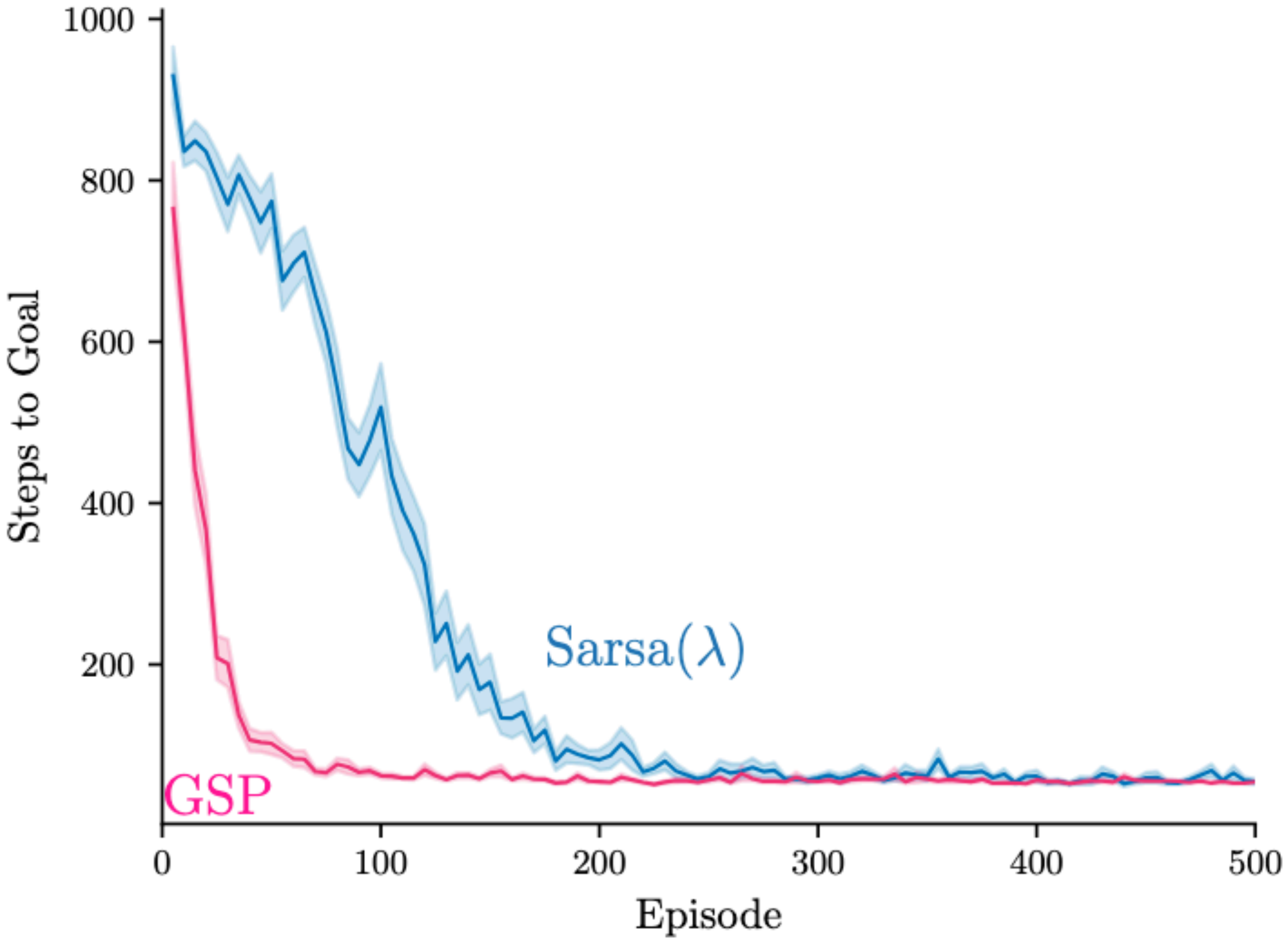
Results: PinBall





Results: PinBall





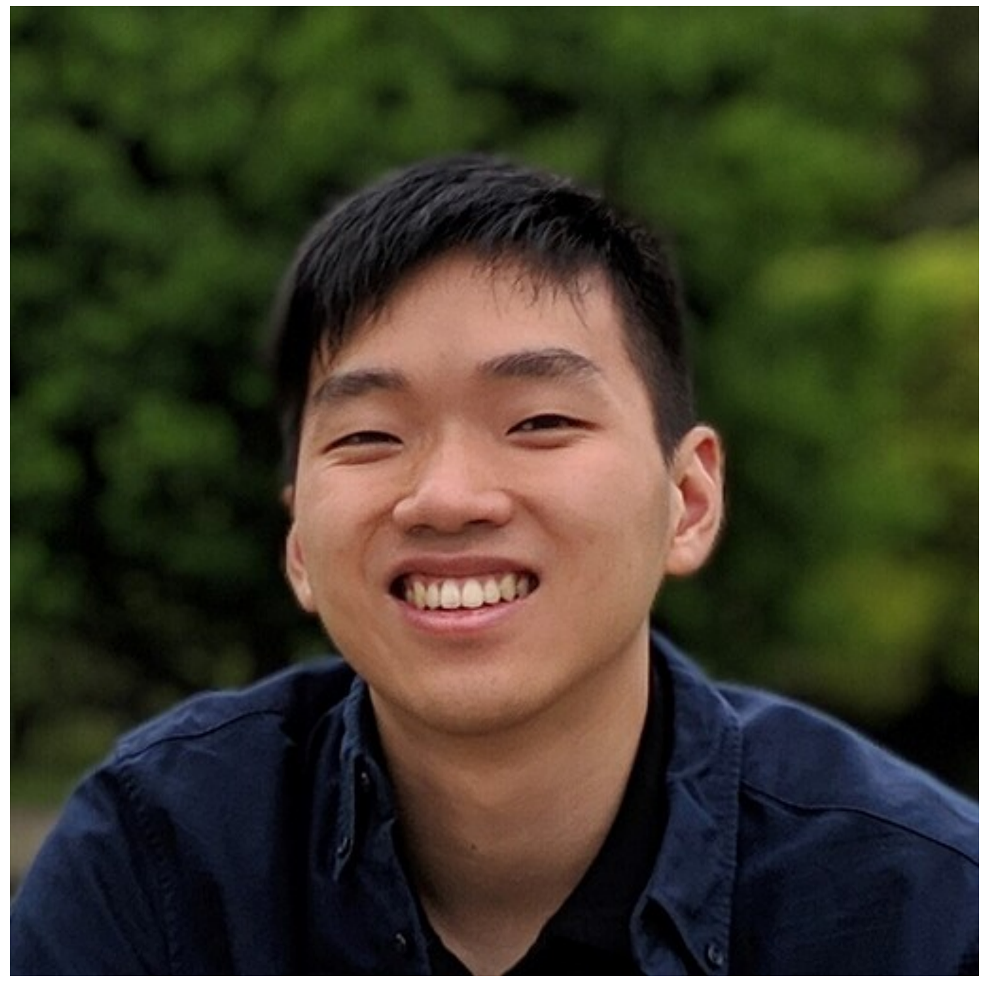
Summary

- We need abstract planning for more sample efficient learning
- We present a technique to learn and use local, temporally abstract models.
- We propose reward shaping as a way to use such models of the environment.

See the Poster for:

- A new way to incorporate an abstract model into TD updates.
- An analysis of this by varying the:
  - Value-Based Learner
  - State space
  - Goal space
- Cases where such planning **does** and **does not** aid learning and adaptation to the world

The Team








Thank You!