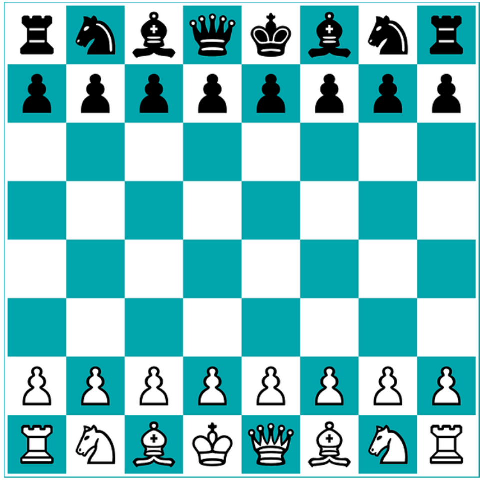
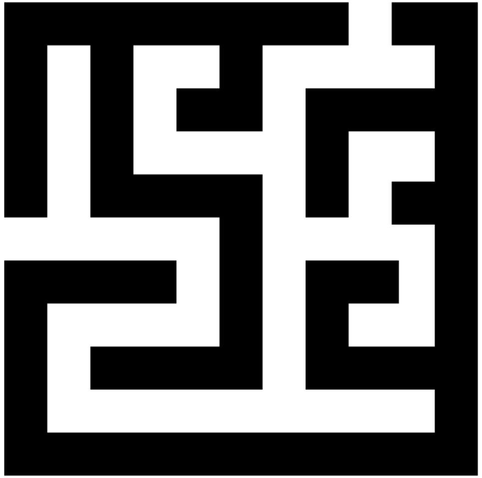
Learning *Abstract* World Models for Value-preserving Planning with Options

Rafael Rodriguez-Sanchez George Konidaris

What tasks in RL look like





- **Actions**: Move piece X to Y
- **State**: Discrete piece position

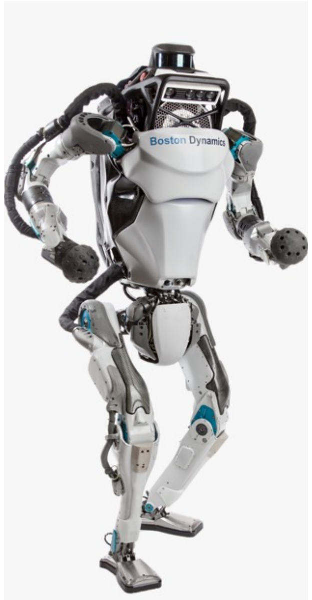
- **Actions**: Move agent in a direction
- **State**: (Continuous) Global position

But, embodied **general-purpose agents** must have fine control (action) space and rich observation spaces

Embodied General-purpose Agents

$$(\mathcal{S}, \mathcal{A}, T, R, \gamma, p_0)$$

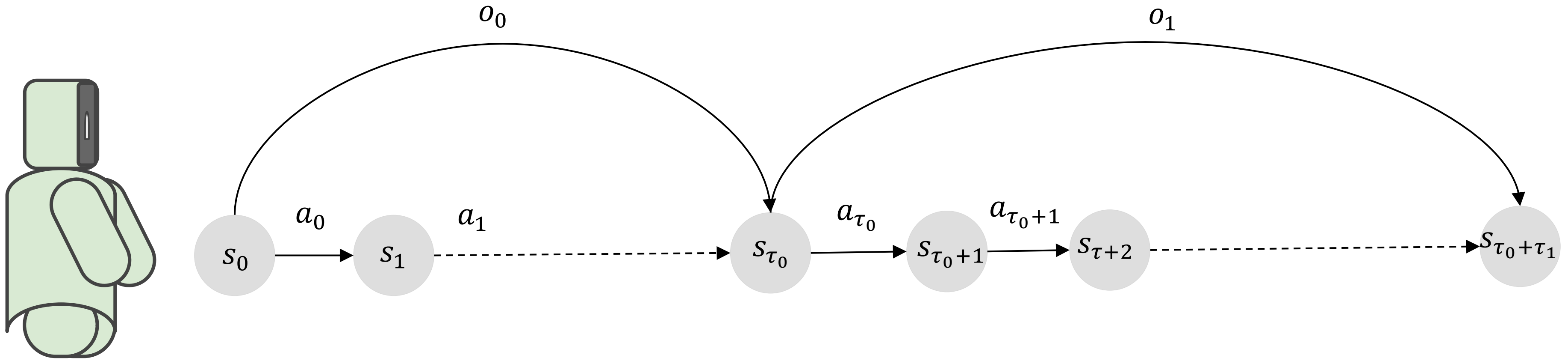
- Joint positions and Velocities
- Visual Inputs
- Force sensors
- …



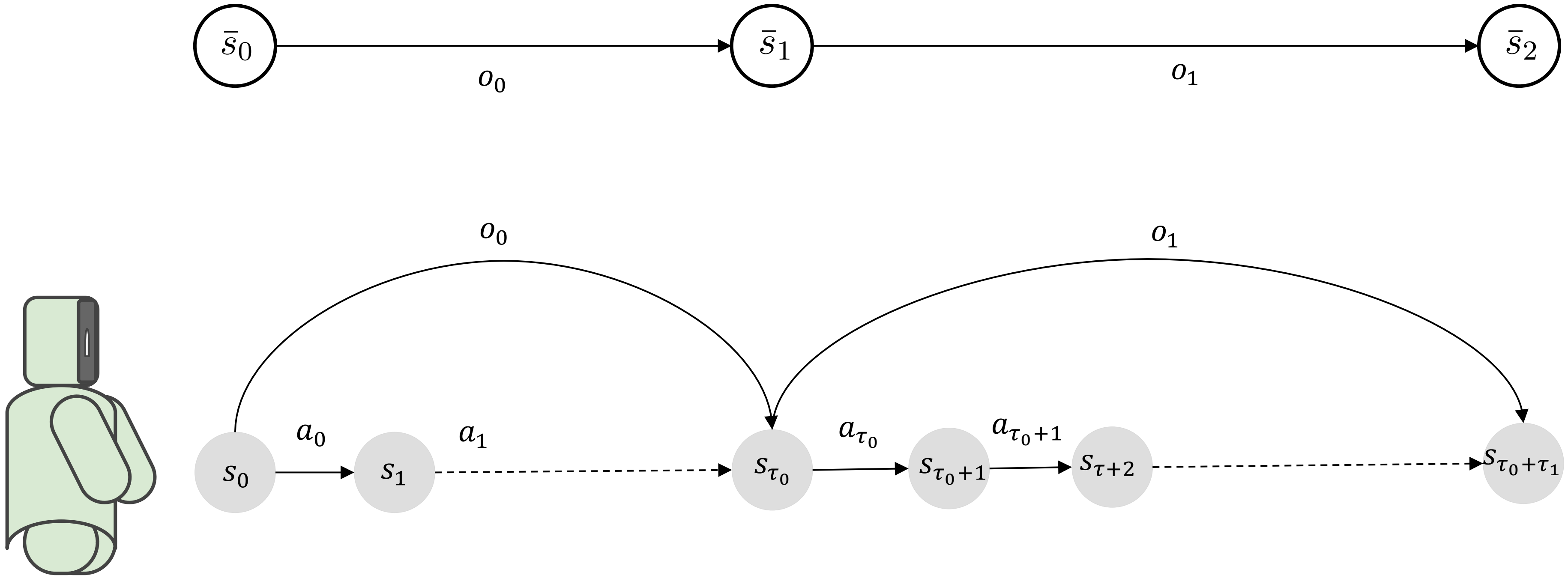
Solution: Abstractions!

Temporal Abstraction & Observed MDP

$$(\mathcal{S}, \mathcal{O}, T, R, \gamma, p_0)$$



Building an Abstract MDP



How do we build a **minimal abstract state** for planning?

- For each option $o \in \mathcal{O}$, $\phi : \mathcal{S} \rightarrow \mathcal{Z}$ is Dynamics-preserving iff

$$T(s' \mid s, o)\Pr(I_o = 1 \mid s) = T(s' \mid \phi(s), o)\Pr(I_o = 1 \mid \phi(s))$$

- We want the abstract state to be maximally predictive of the next state and option's initiation set.

Learning the Abstraction: Information Maximization!

$$\max_{\phi \in \Phi} MI(S'; \phi(S), O) + MI(I; \phi(S))$$

$MI(S'; \phi(S), O)$

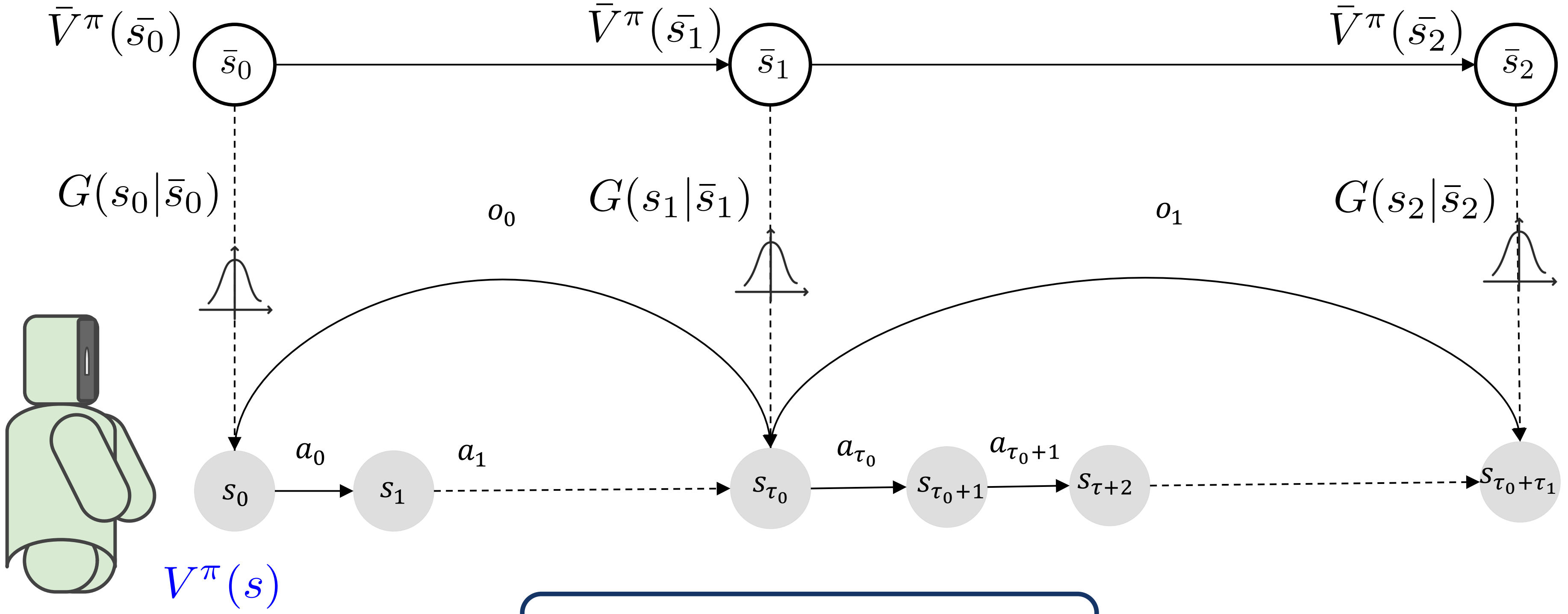
Use MI differentiable estimators **such as InfoNCE** [van der Oord, 2018] to learn contrastively the representation
Use NLL to learn the abstract transition function.

$MI(I; \phi(S))$

Learn the binary conditional distribution using NLL

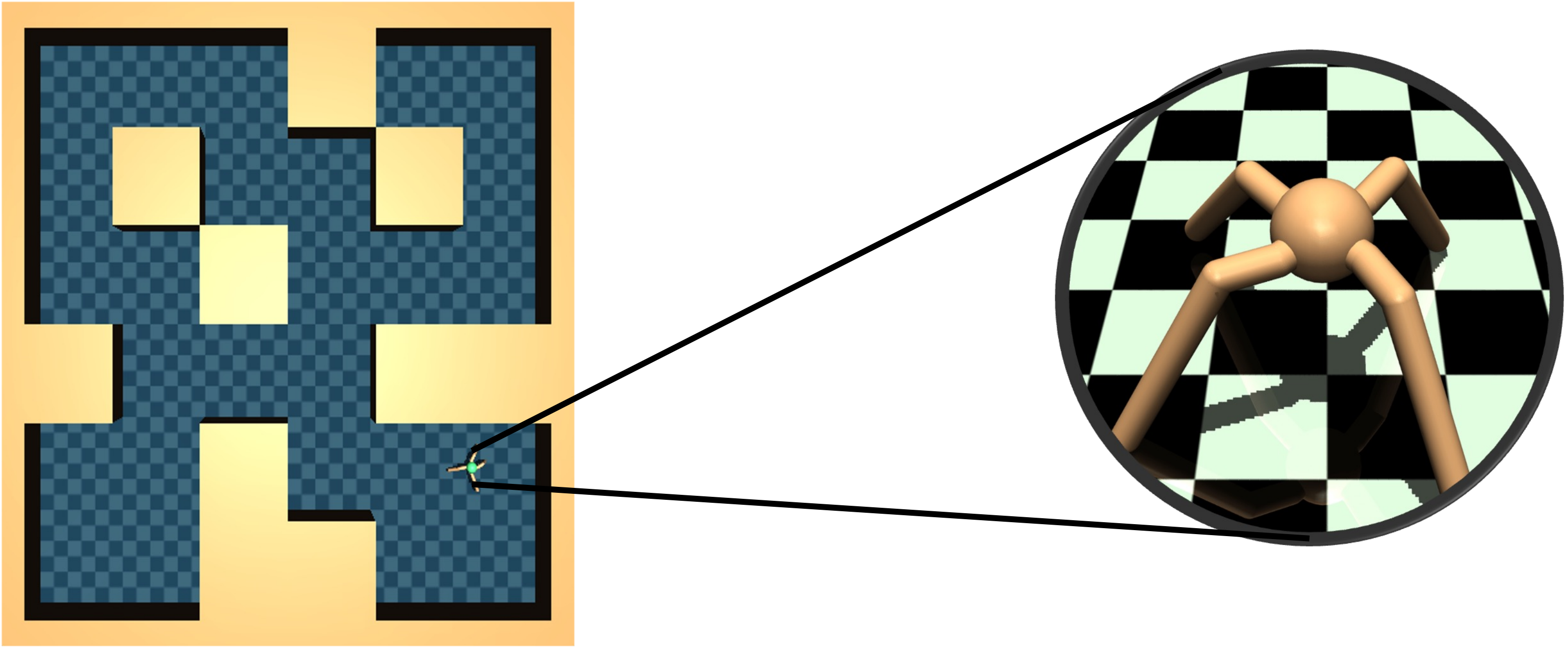
Learn the rest of the abstract MDP (reward function, abstract discount factor, etc. in the new latent space)

Does planning with an Abstract MDP make sense?

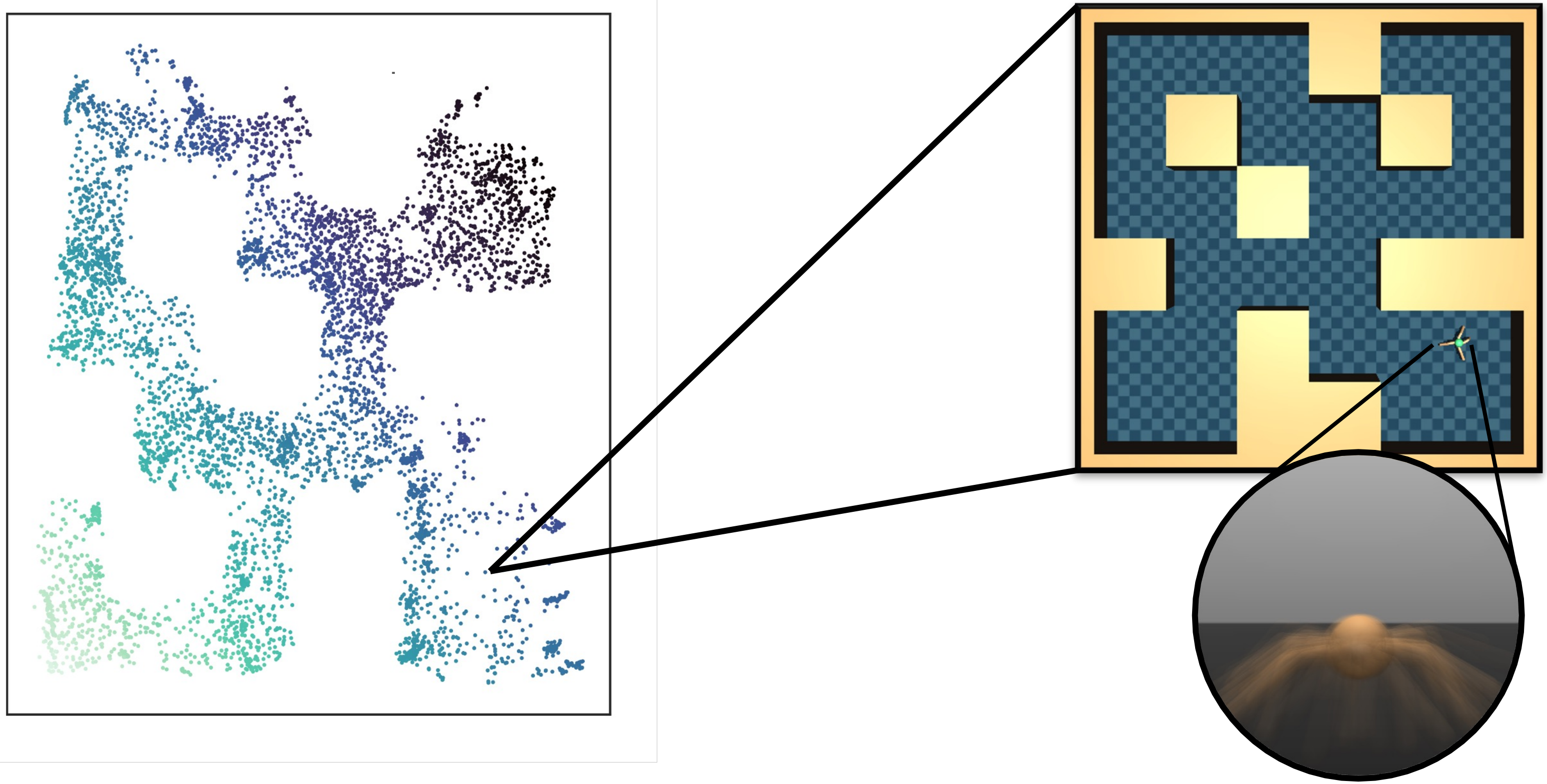


$$\bar{V}^\pi(\bar{s}) = \mathbb{E}_{s\sim G(\cdot|\bar{s})}[V^\pi(s)]$$

Ant in a Maze

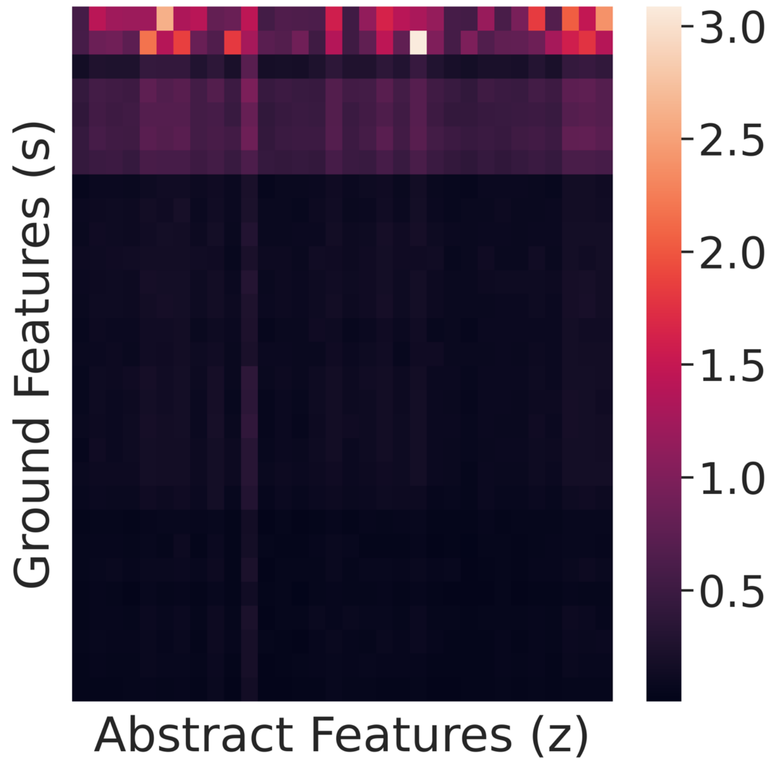


Learned Abstract State Representation



Mutual Information Matrix

- The most relevant features correspond to the global position in the maze and orientation.



Planning for abstract goals works!

- 9 goal positions
- Sparse task reward function