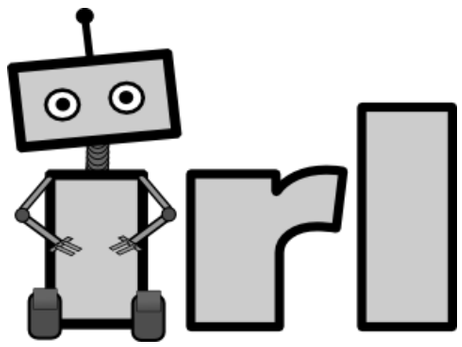
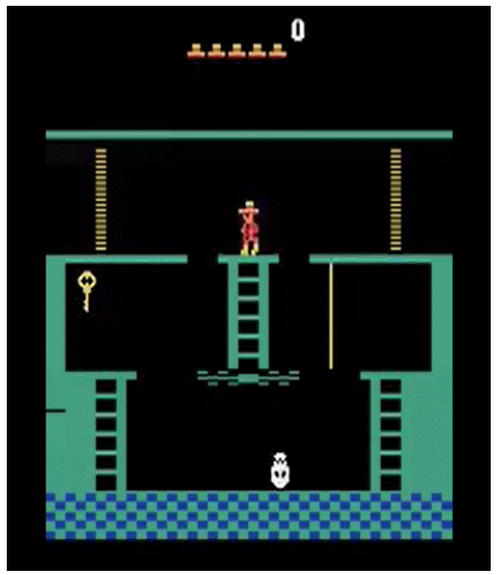
Skill Discovery for Exploration and Planning using Deep Skill Graphs

Akhil Bagaria, Jason Senthil, George Konidaris
Brown University

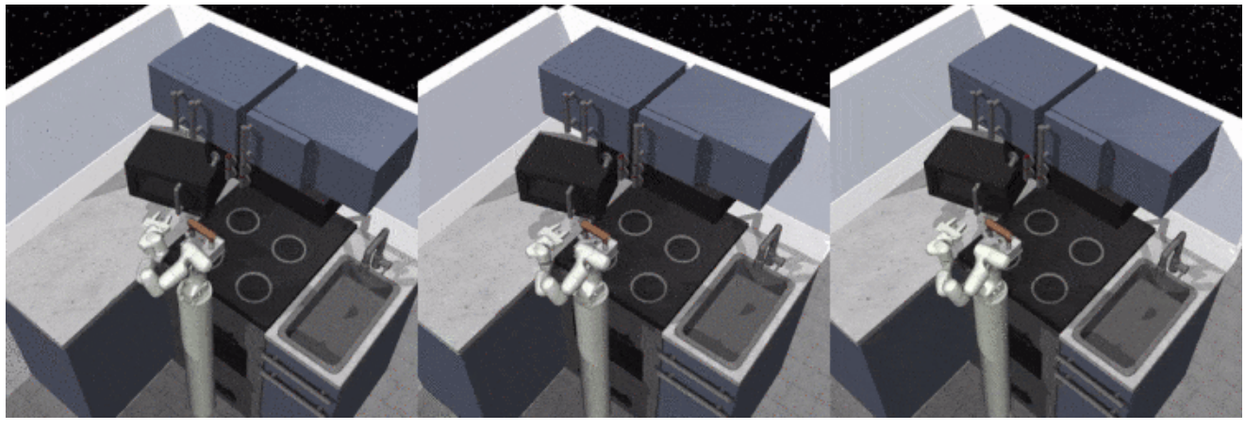


Long Horizon Problems

- **Problems**:
  - Complex **observation** space
  - Difficult **decision** space
  - Very **sparse** reward function



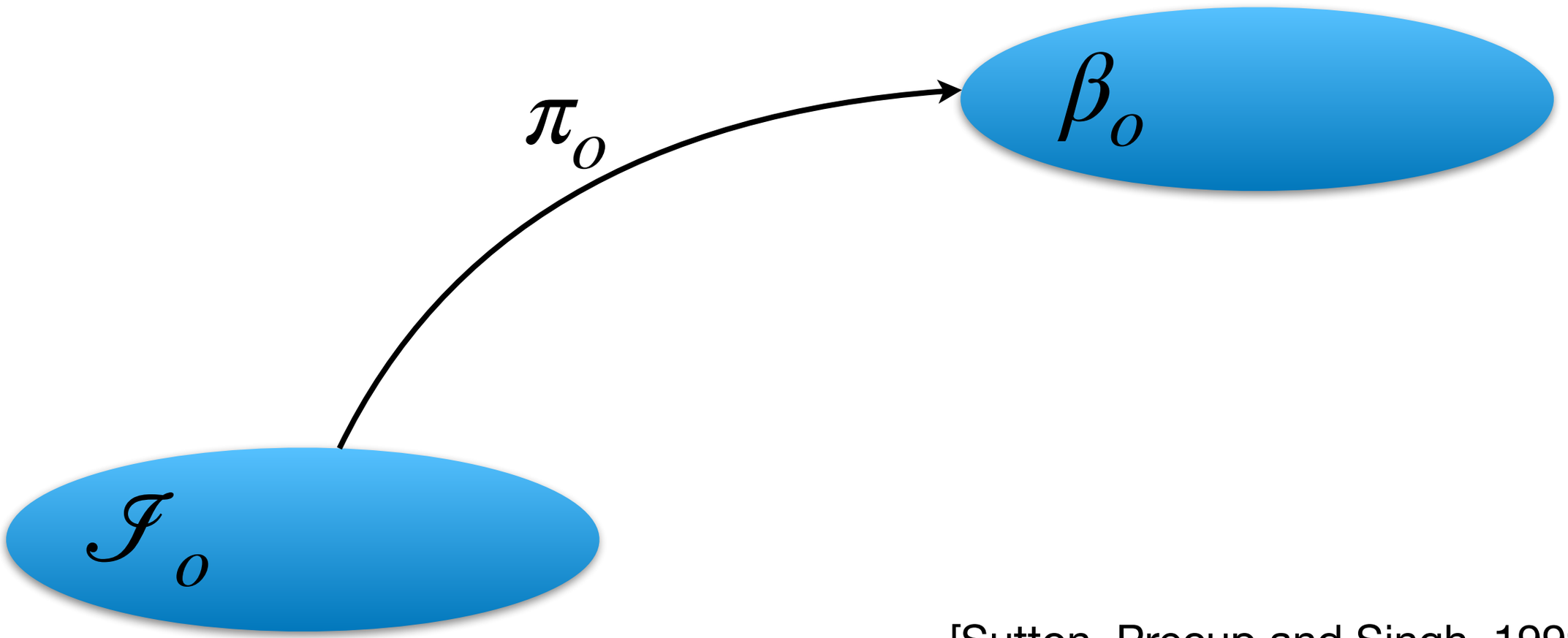
- **Solutions**:
  - State and action **abstractions**
  - **Propose your own goals** and learn to reach them



Options as a basis for state and action abstractions

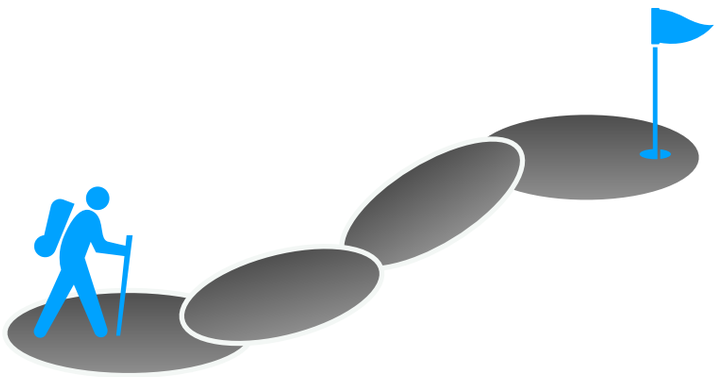
Skills = modular closed loop control policies

$$\mathcal{O} = (\mathcal{I}_o, \pi_o, \beta_o)$$

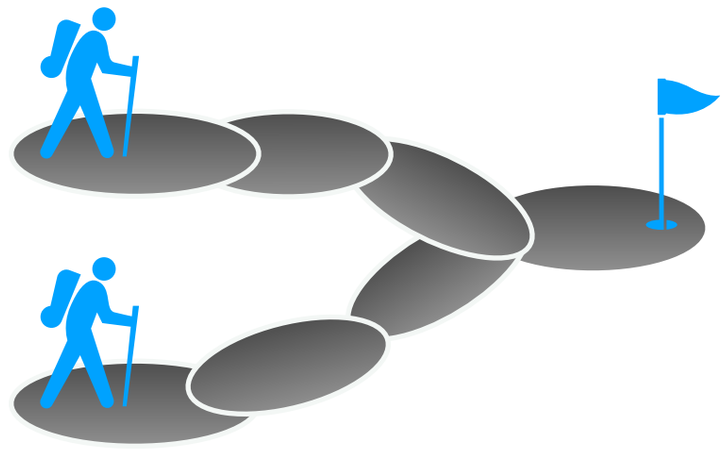


[Sutton, Precup and Singh, 1999]

Skill Discovery

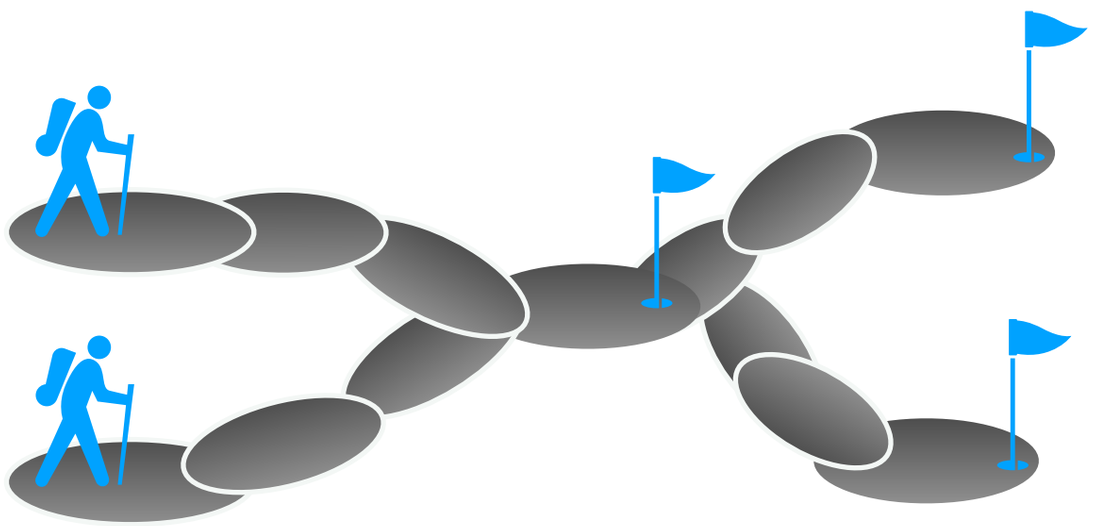


Skill chain



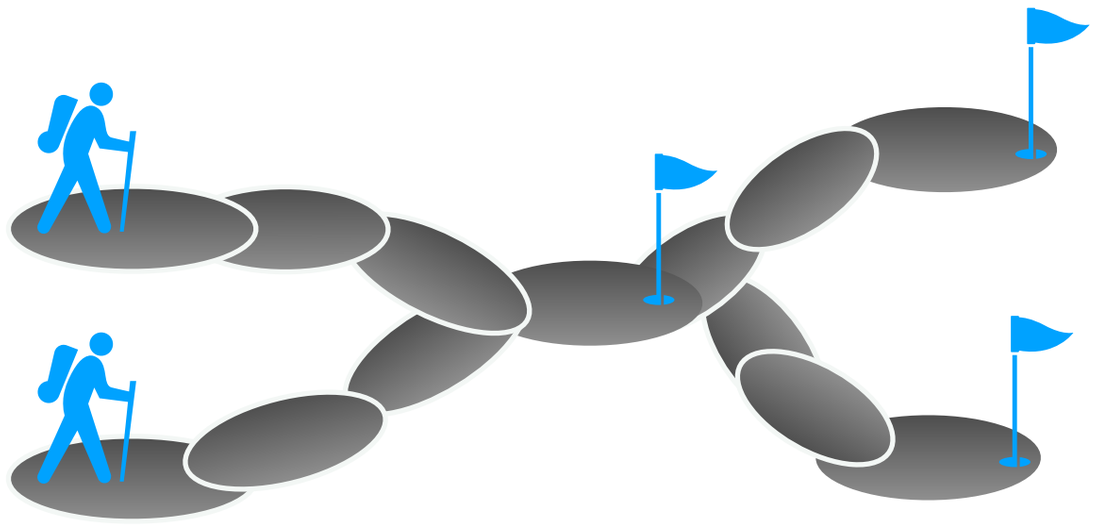
Skill tree

Deep Skill Chaining Algorithm (Konidaris & Barto, ‘09; Bagaria & Konidaris ‘20)



Skill graph

Skill Graph: Definition



Edges

Skill graph

$$G = (\mathscr{V}, E, \mathscr{W})$$

Discovered Goals

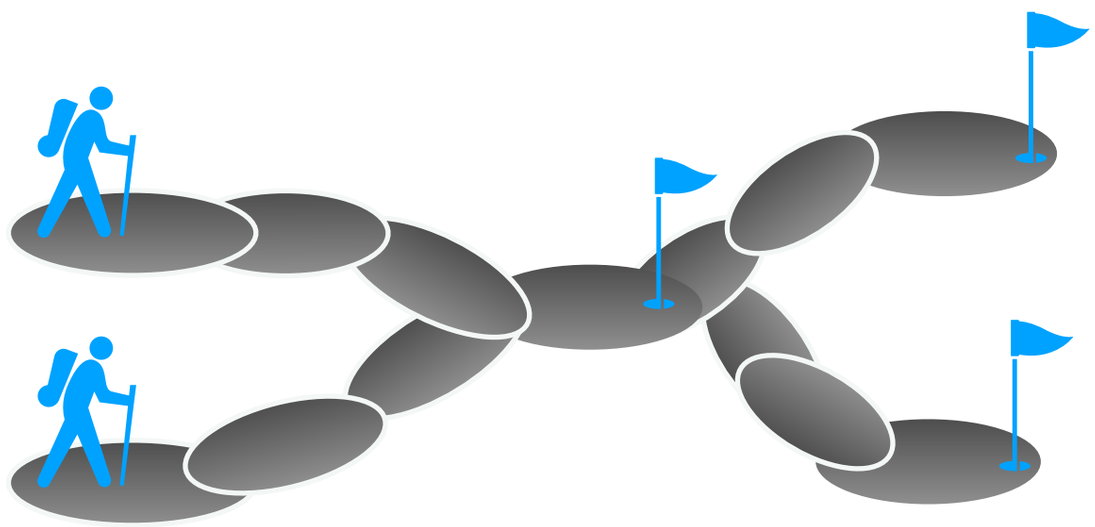
$$\epsilon_g : s \rightarrow \{0,1\}$$

Options

$$o = (I_o, \pi_o, \beta_o)$$

Vertices / nodes

Skill Graph: Usefulness

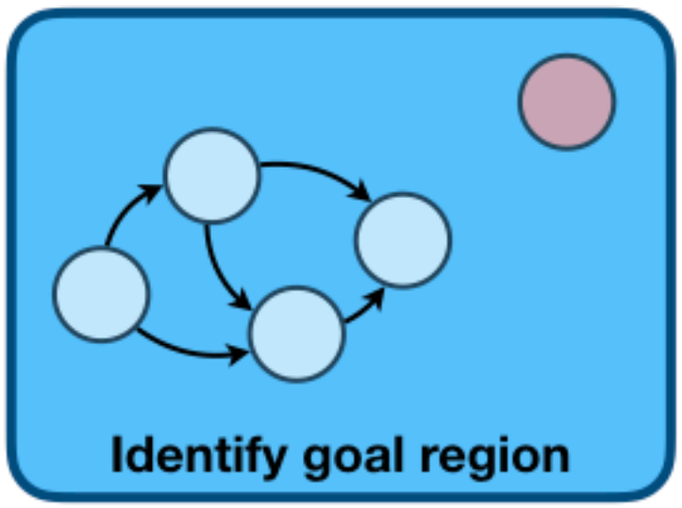


- **Training time:** Graph construction procedure ensured coverage **(i.e, exploration)**

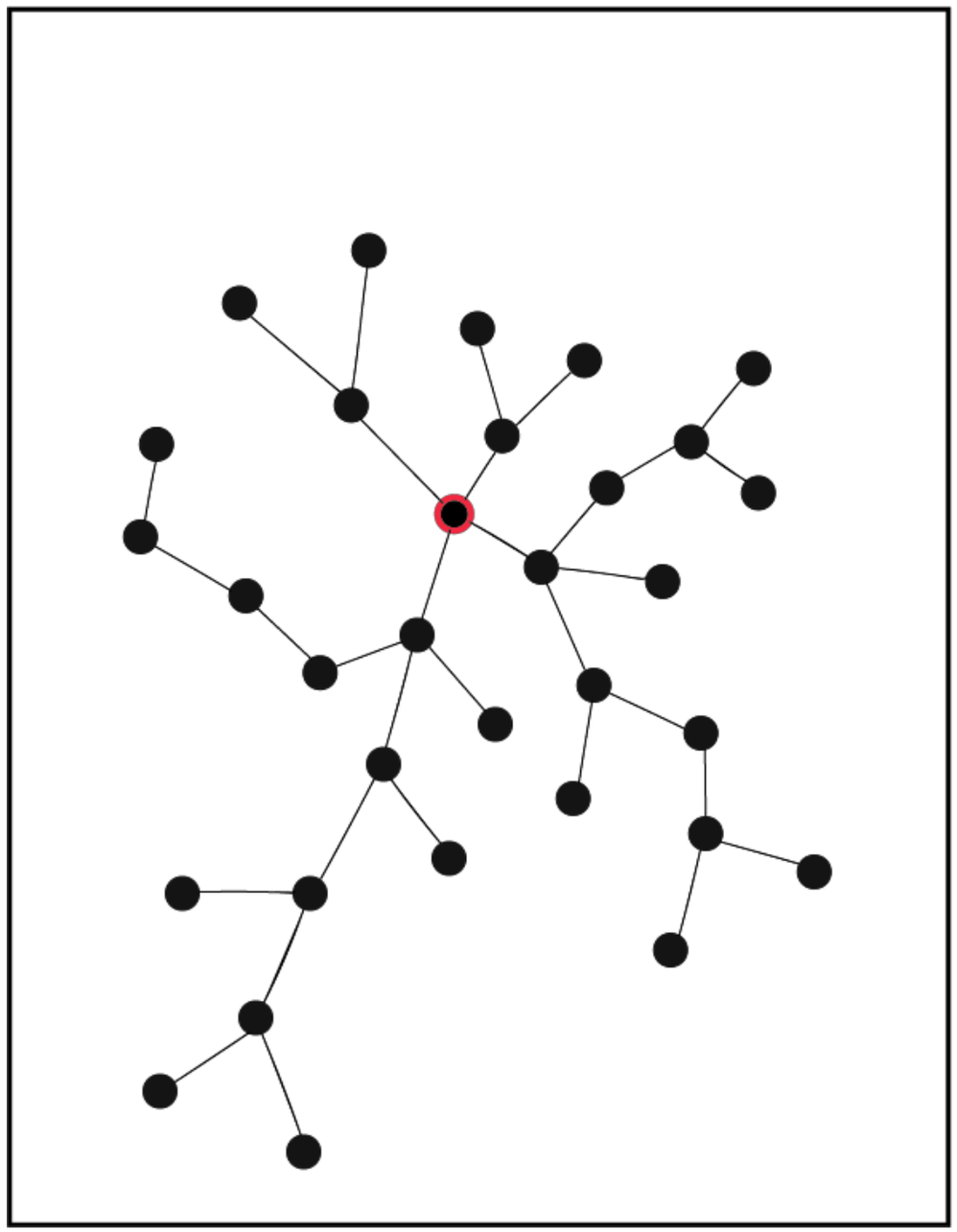
- **Test time:** If the goal is **inside** the graph, use **planning** to reach it

- **Test time:** If the goal is **outside** the graph, **plan to the nearest node**, then **switch to learning**

Algorithm Overview



Graph Expansion Algorithm

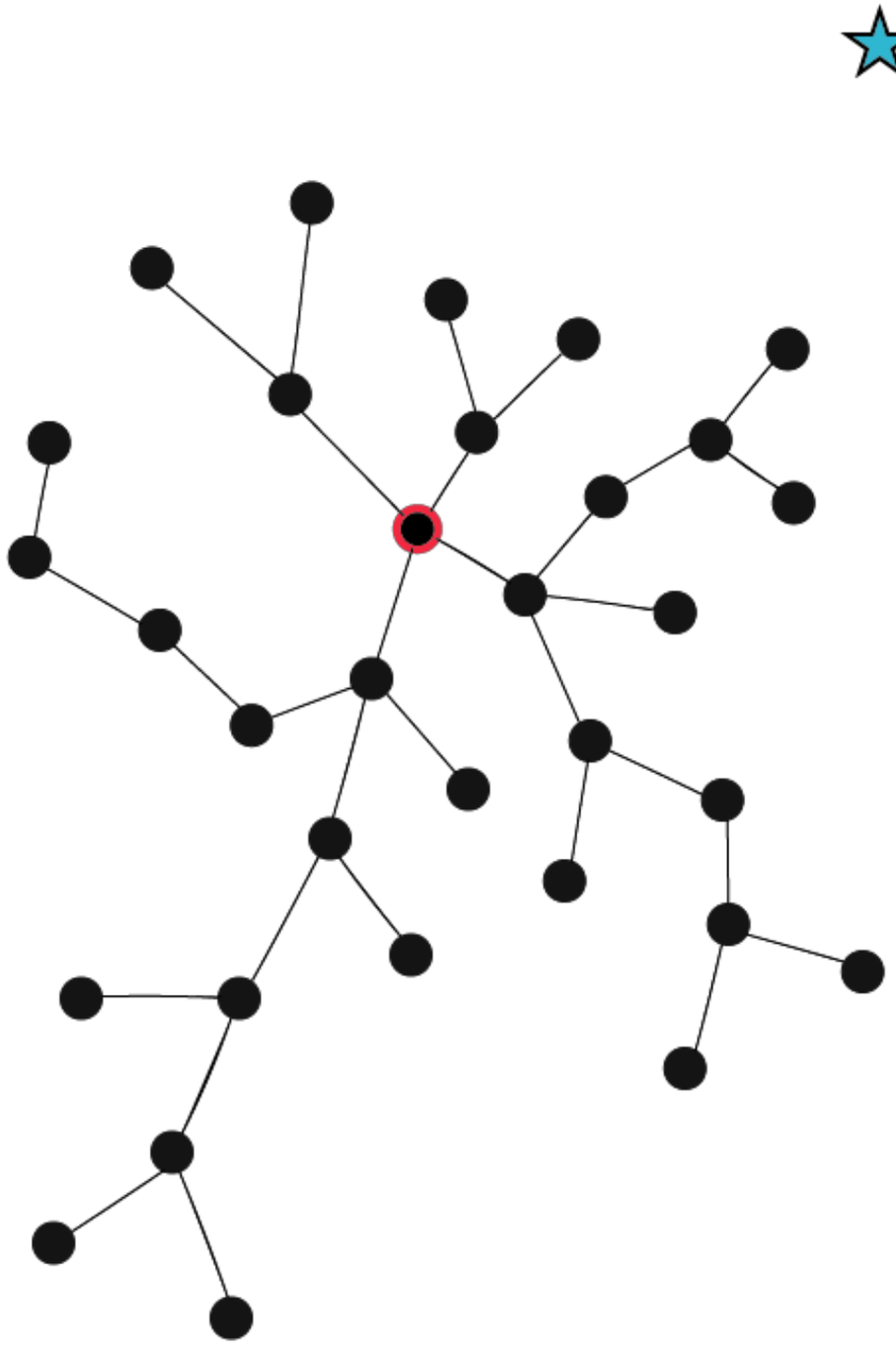


Exploration objective

How can we **add a node** such that we **maximally increase coverage** of the state-space?

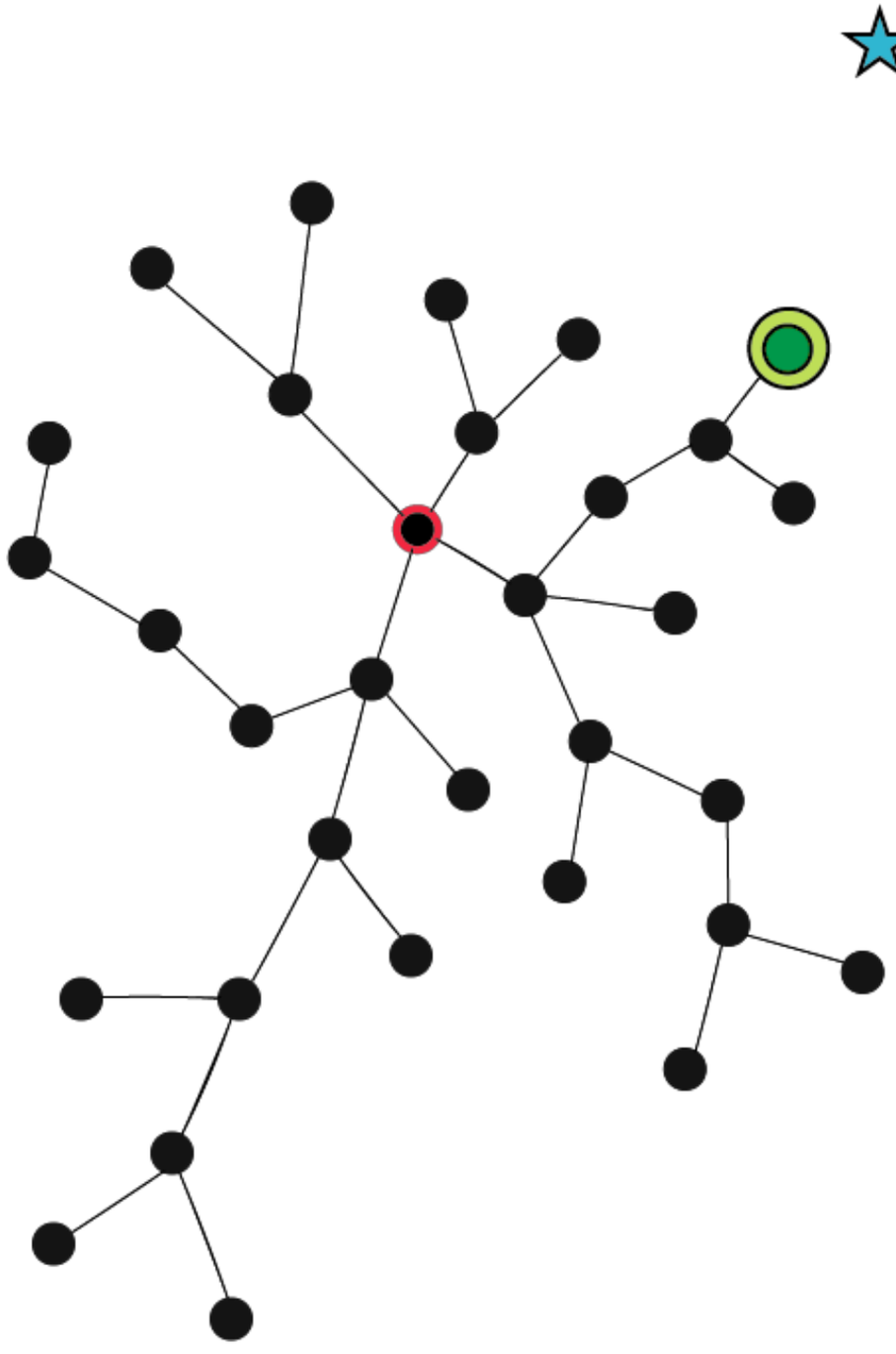
Random Sampling!
(RRTs)

Graph Expansion Algorithm



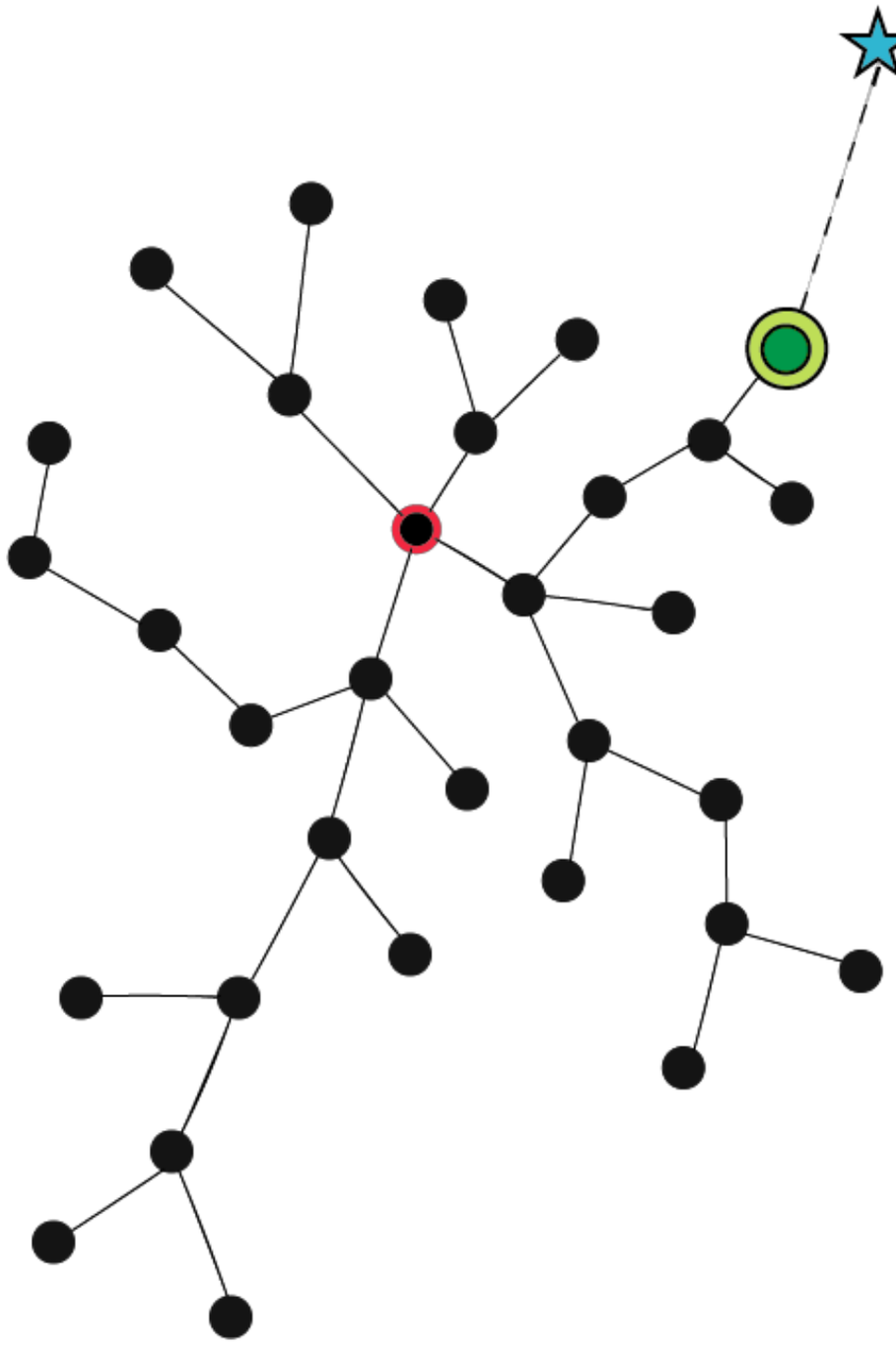
Randomly sample a state from the state-space

Graph Expansion Algorithm



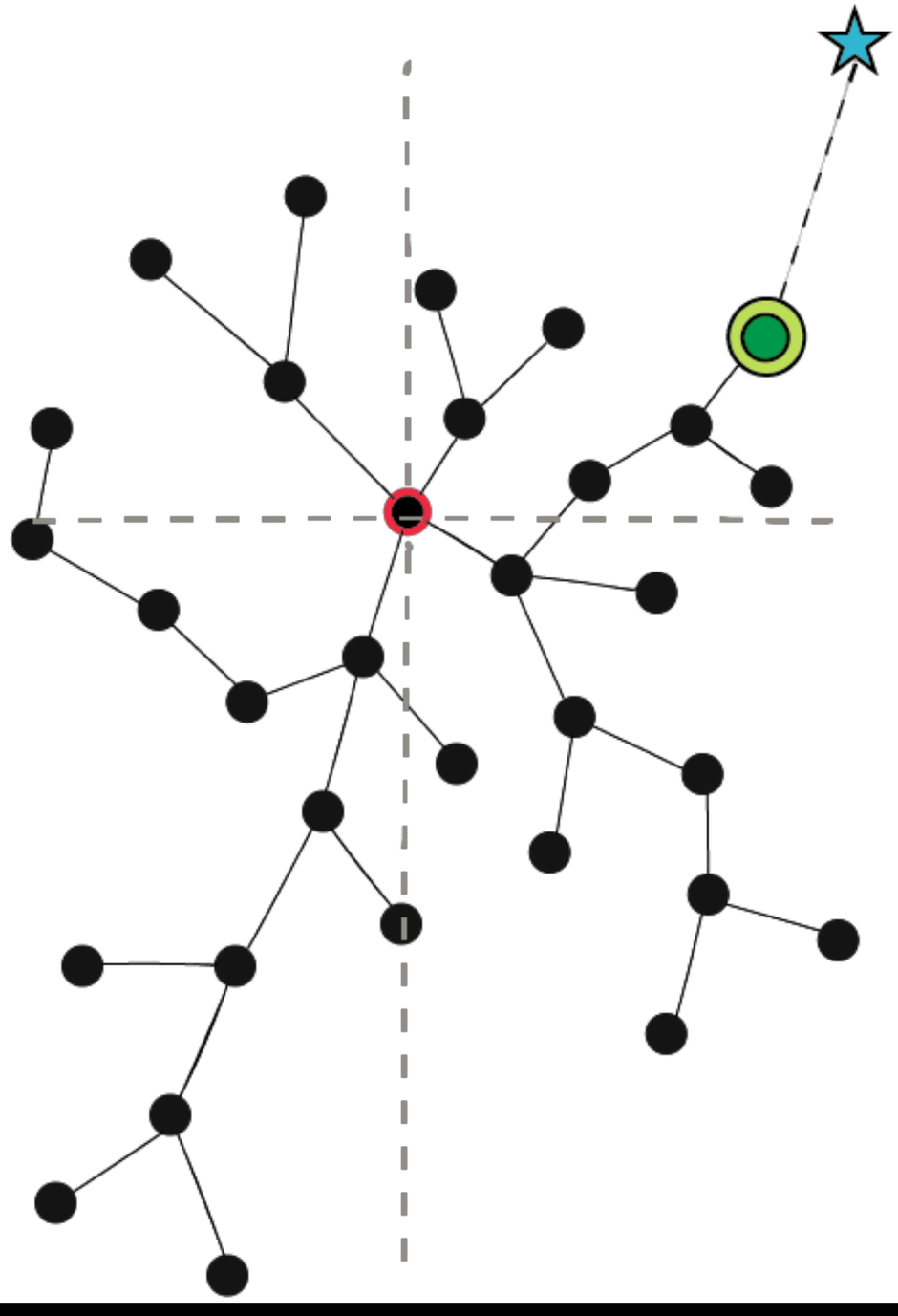
Identify the nearest neighbor in the graph

Graph Expansion Algorithm



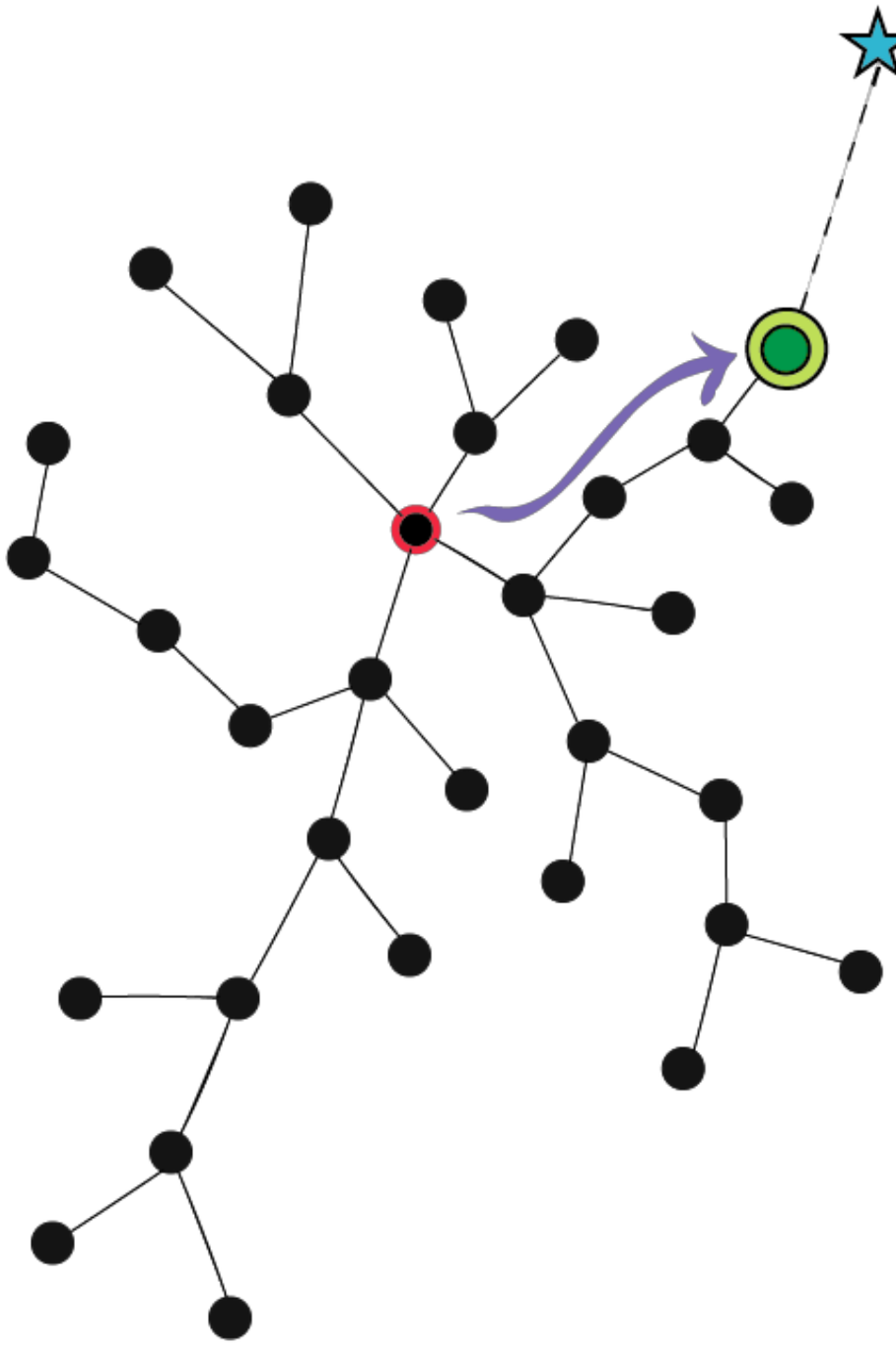
Extend the graph in the direction of the random goal

Expansion towards Frontier



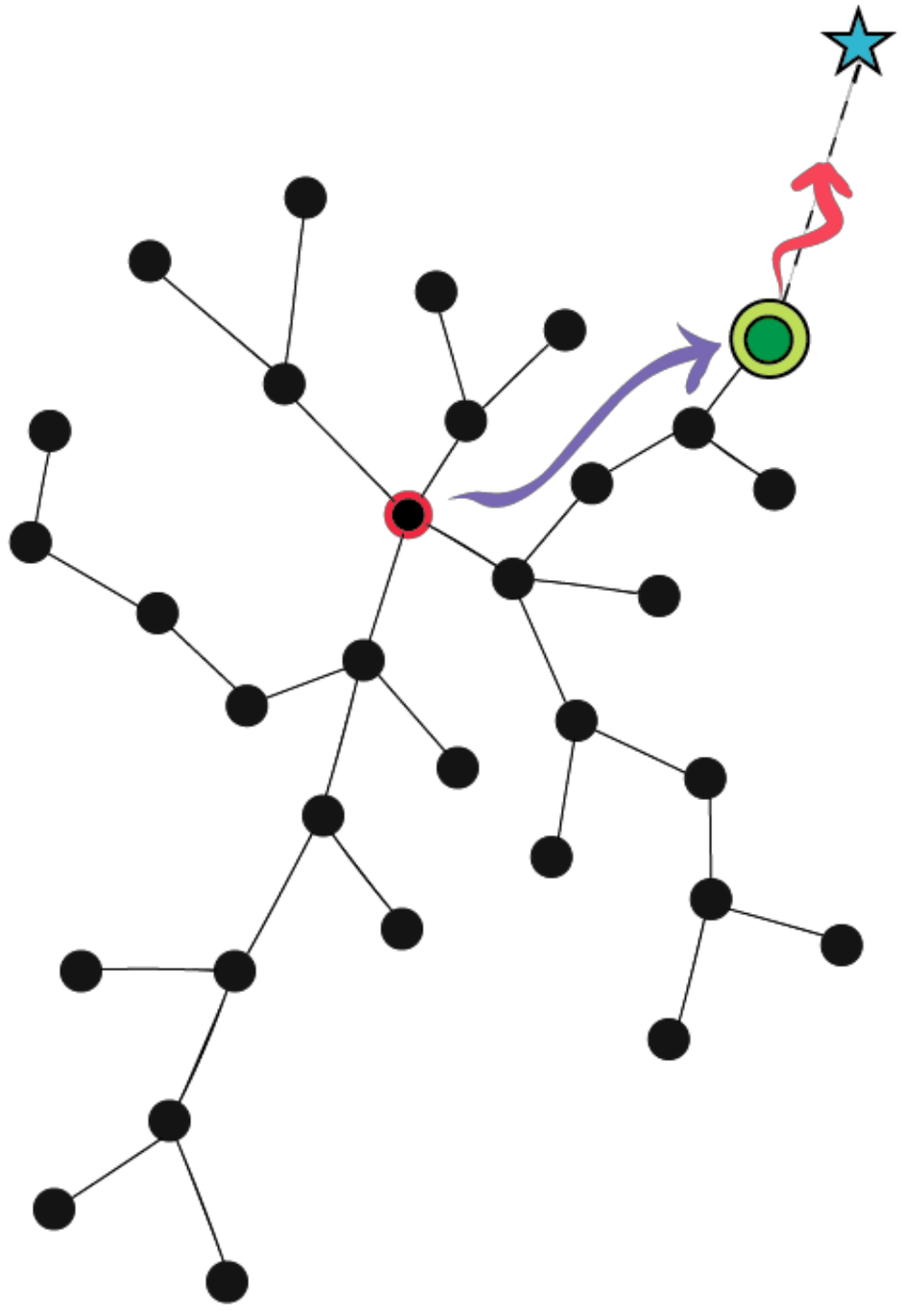
Why does it work? The node with the biggest voronoi region is chosen for expansion

Graph Expansion Algorithm



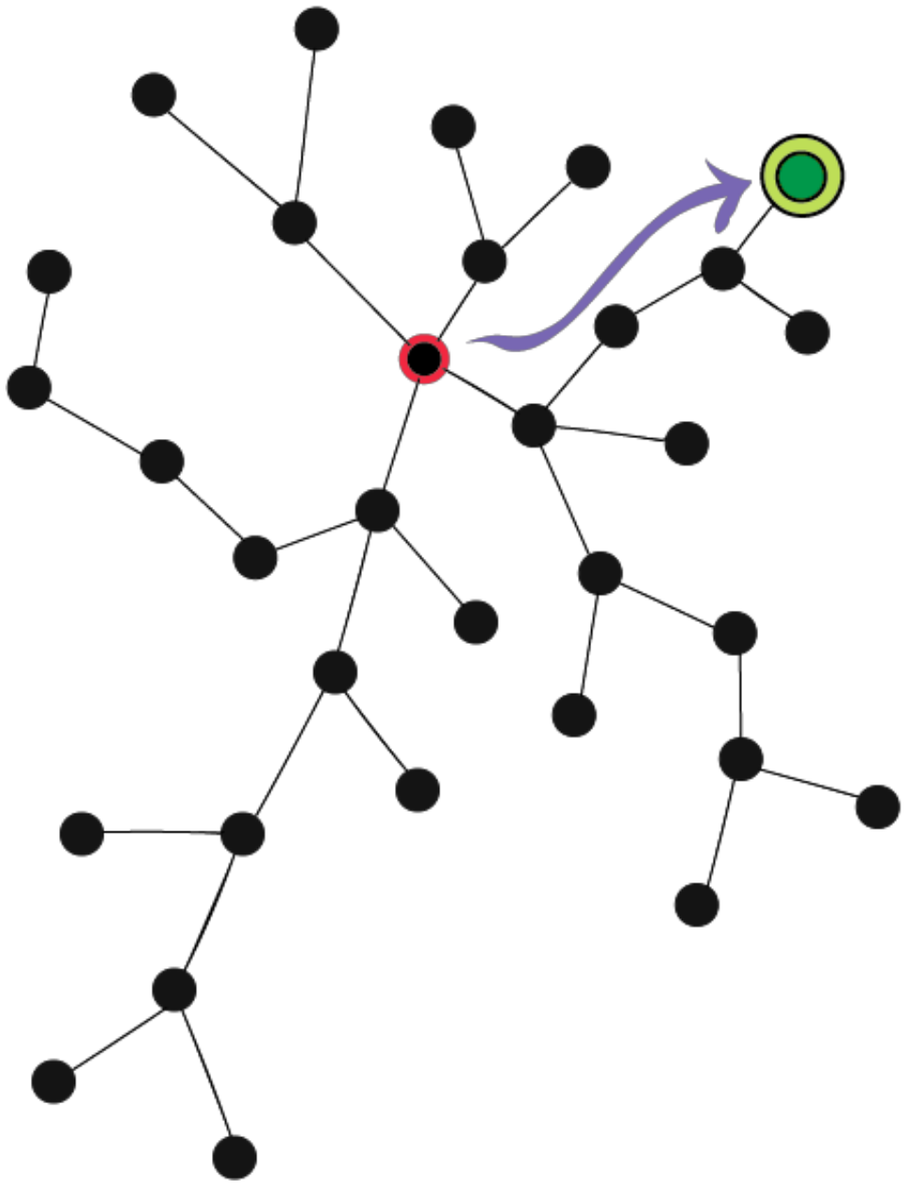
Use planner inside the graph to reach the nearest neighbor

Graph Expansion Algorithm



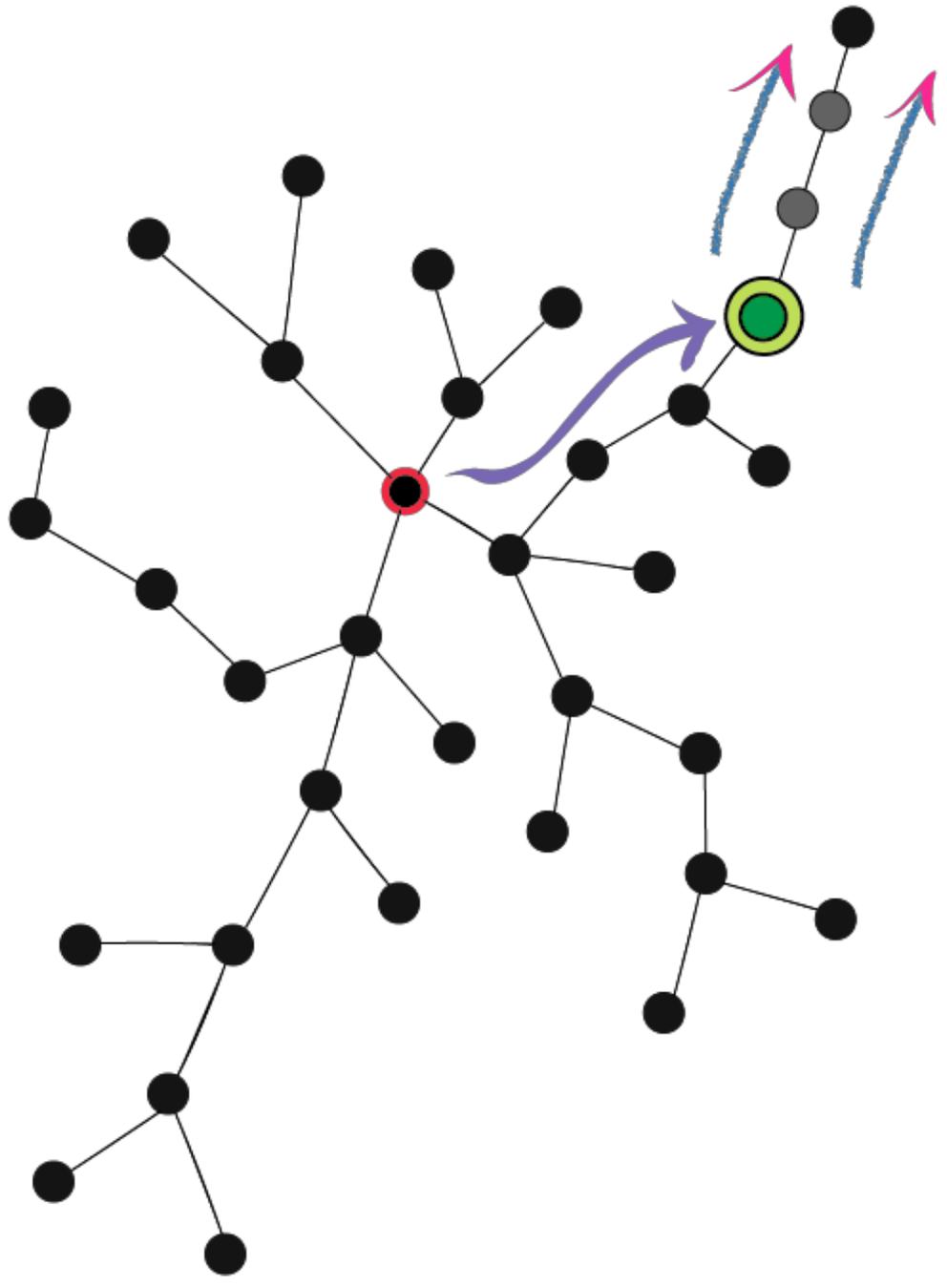
Use model-based RL to move K steps in the *direction* of goal

Graph Expansion Algorithm



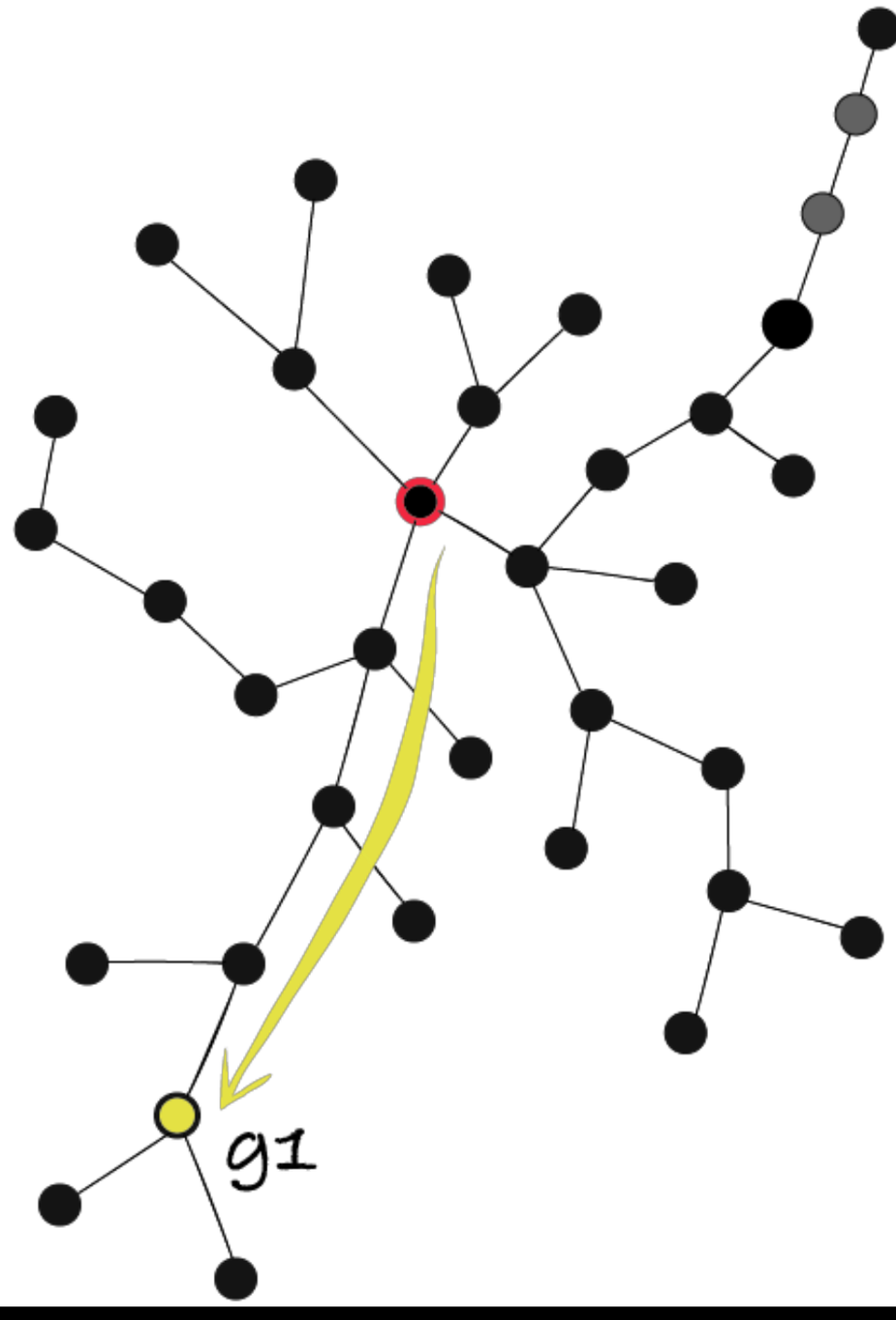
Add the resulting state as a node in the graph

Graph Expansion Algorithm



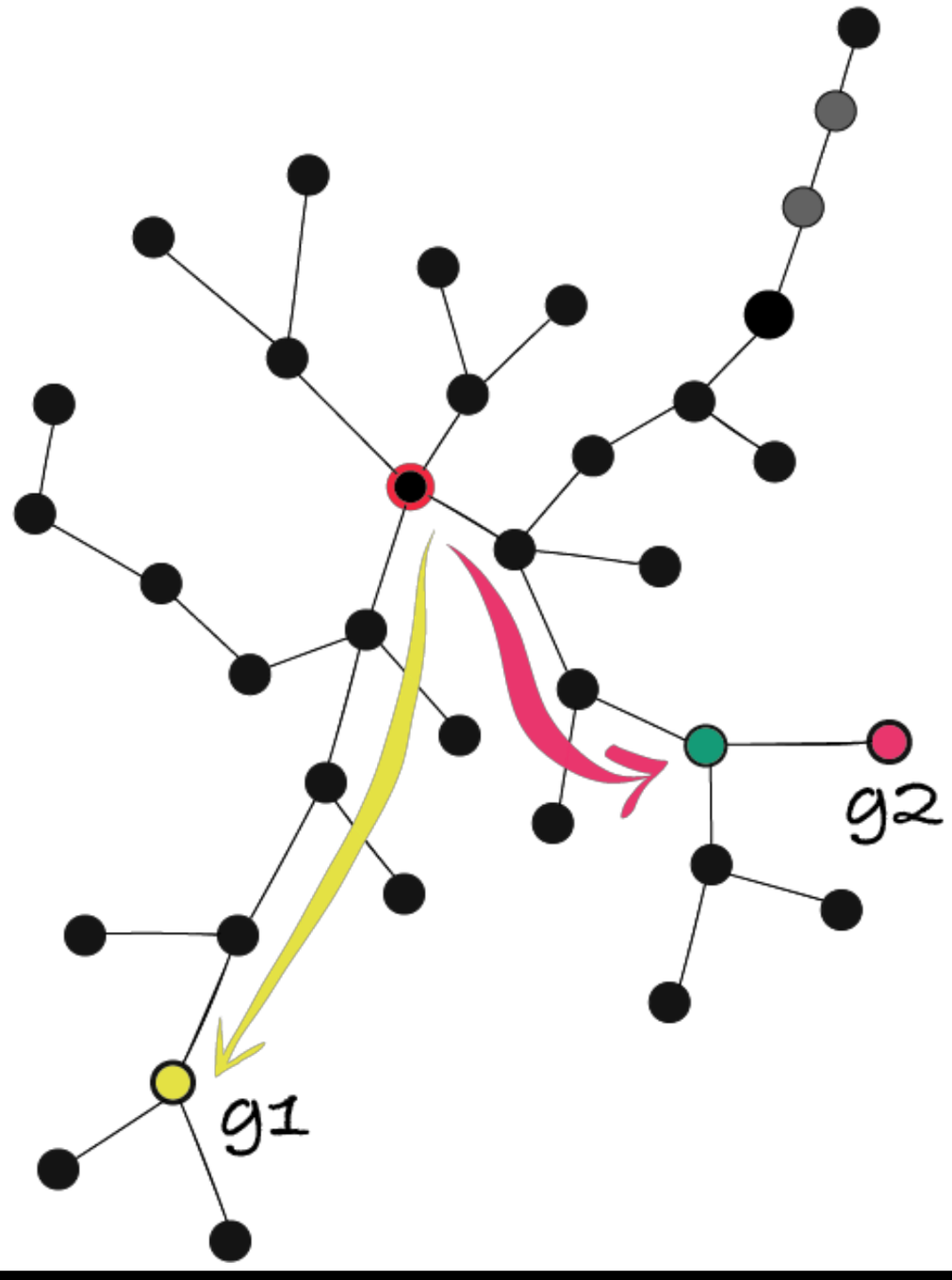
Use deep skill chaining to connect the new goal node to the existing graph

Graph Expansion Algorithm



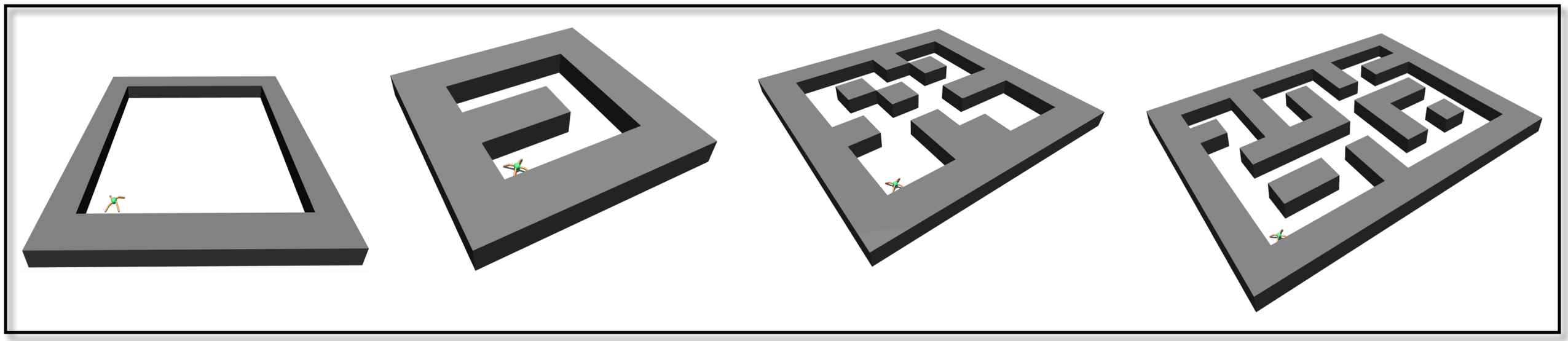
Test time: if the goal is inside the graph, just plan with learned skills

Graph Expansion Algorithm



Test time: if the goal is outside the graph, plan to the nearest node and then switch to DSC

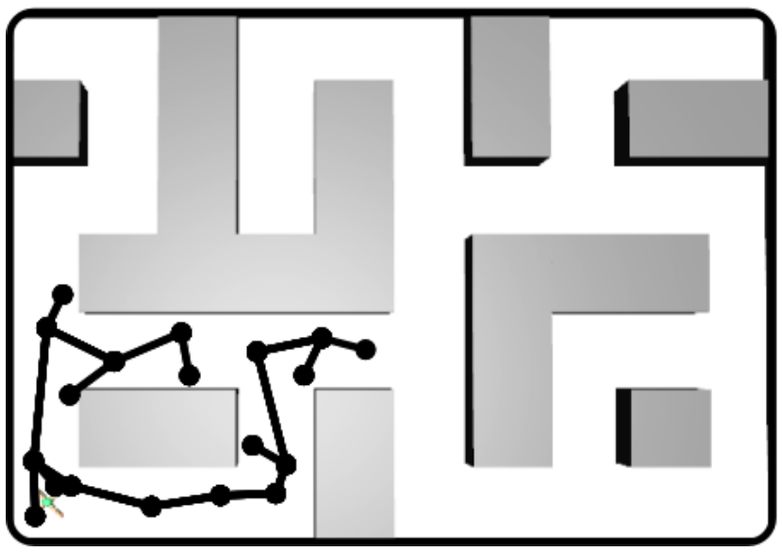
Experimental Setup



Training Time: No reward function - unsupervised training

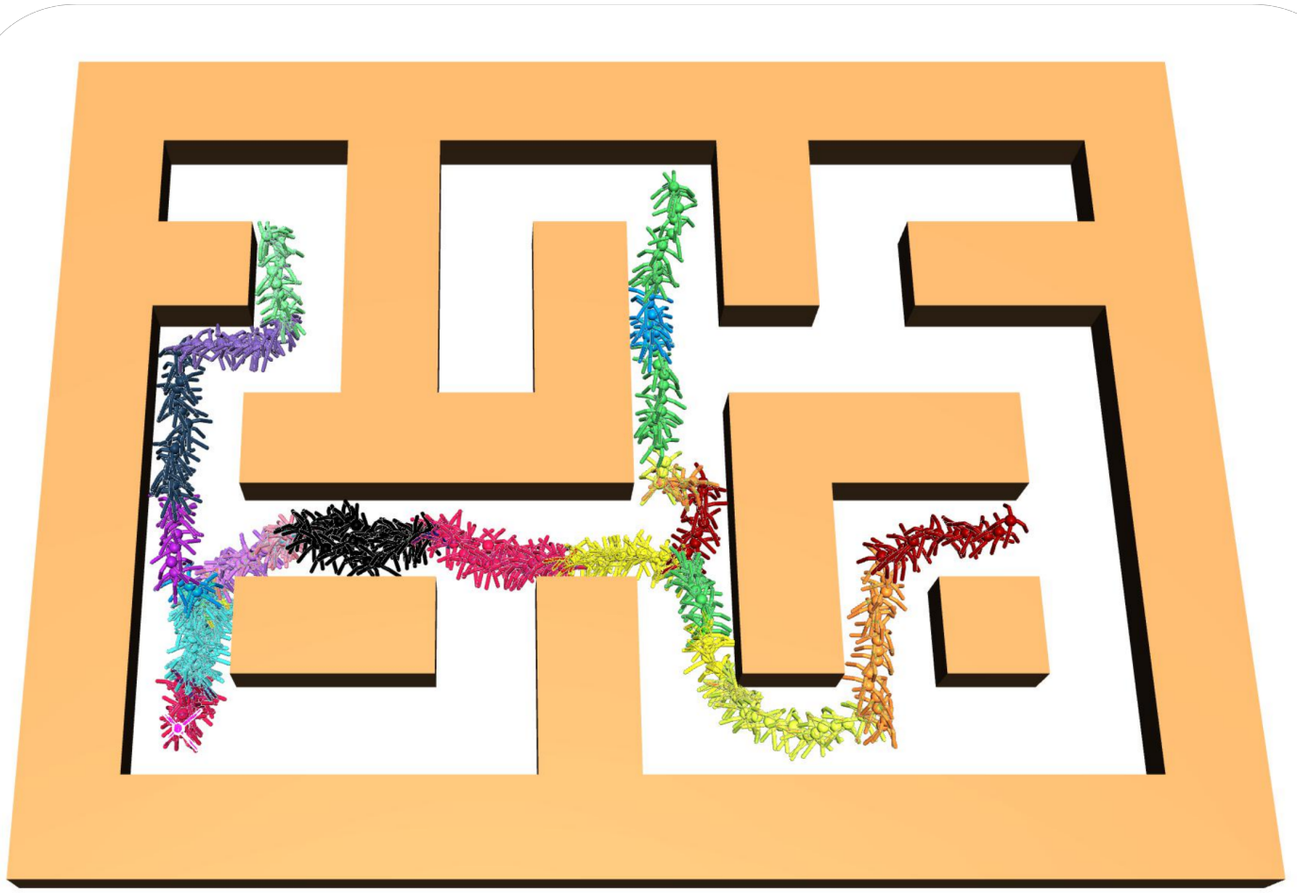
Testing Time: Rewarded for reaching random goal states from random start states

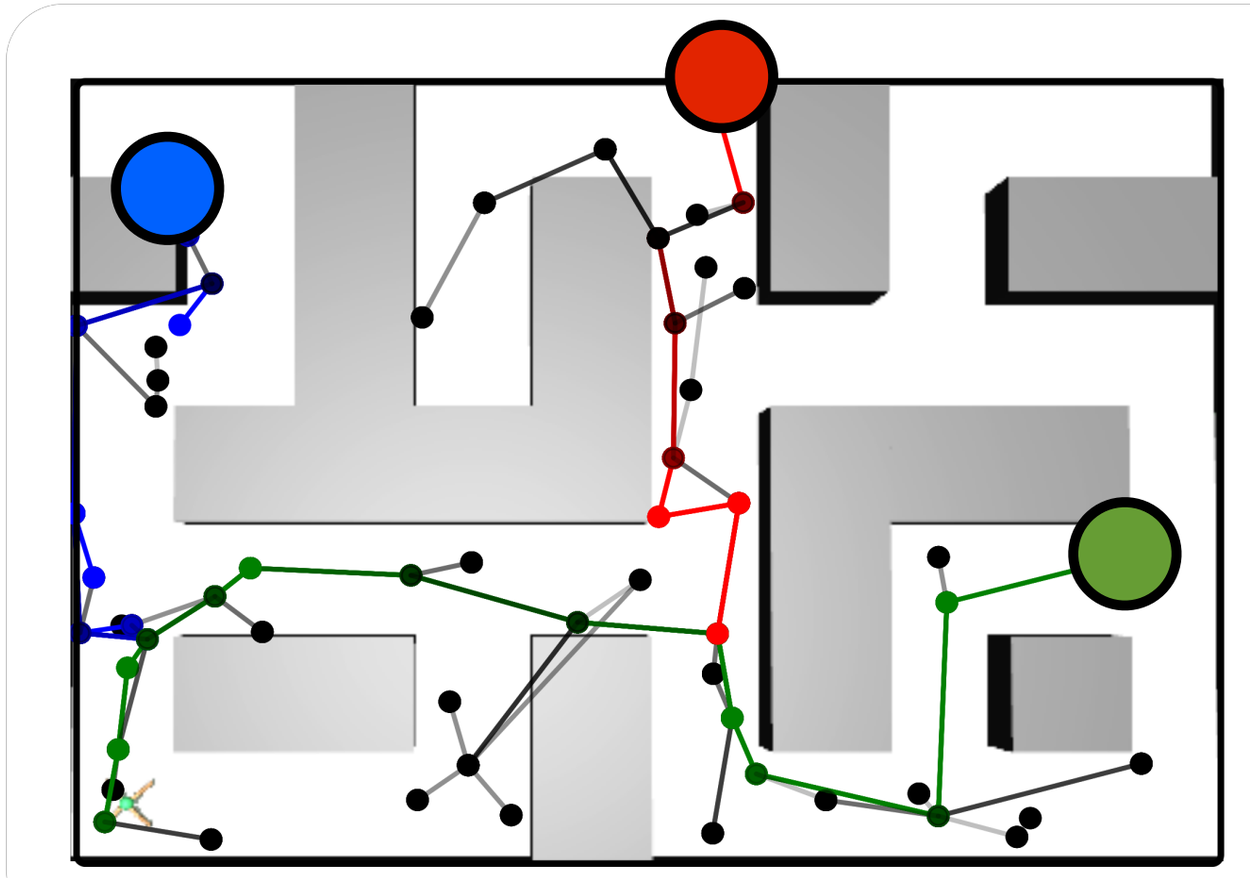
Incremental Graph Expansion



To visualize nodes, we plot the median state of their termination conditions

Solution Trajectories

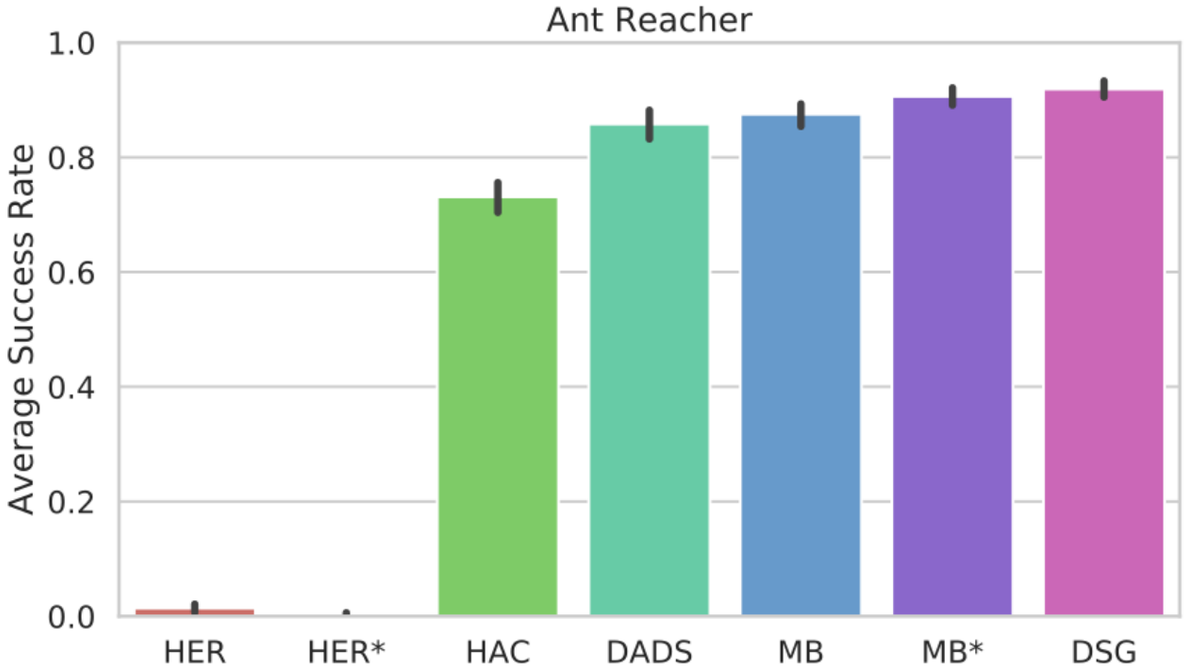


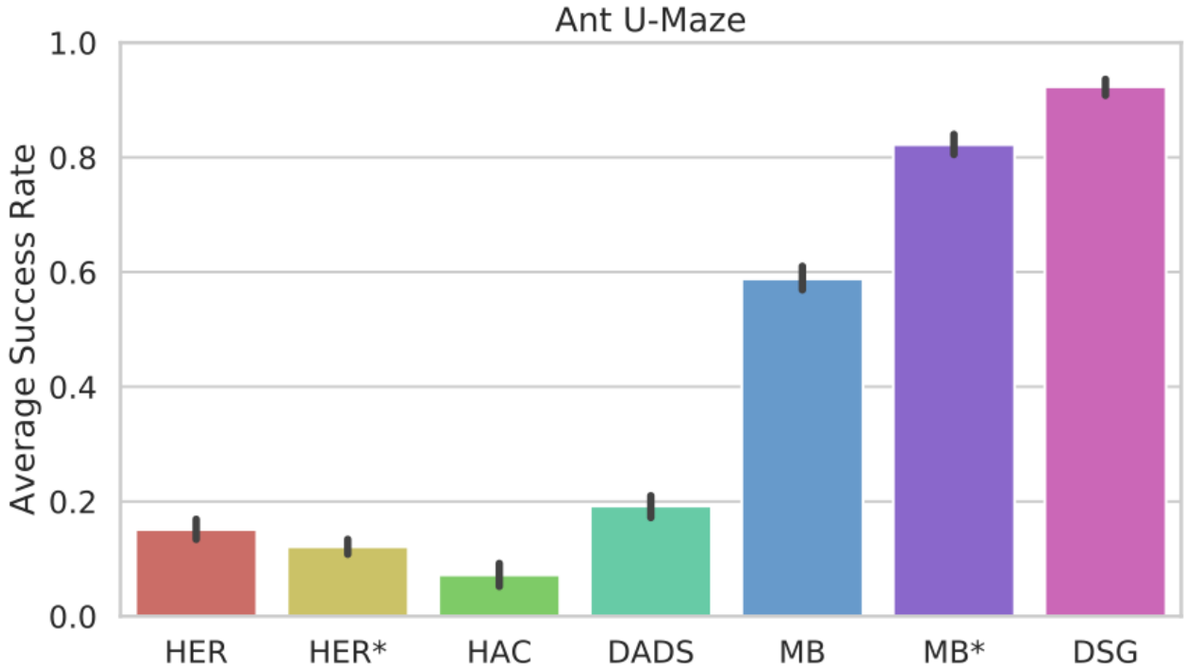


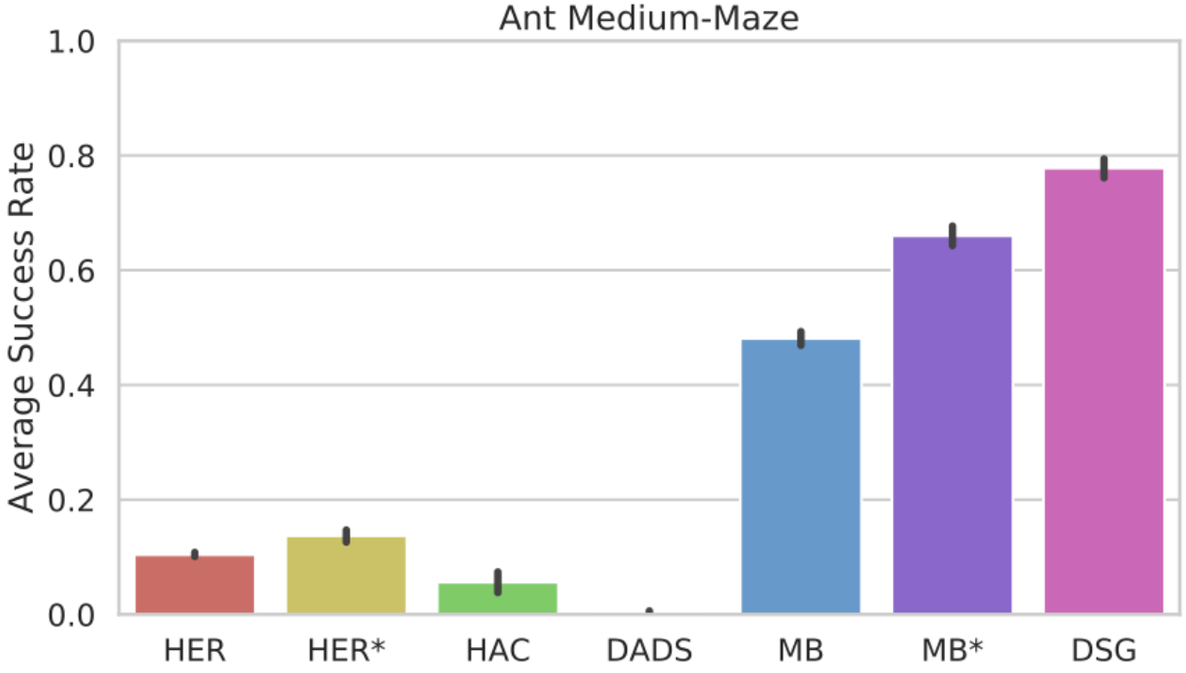
Comparative Analysis

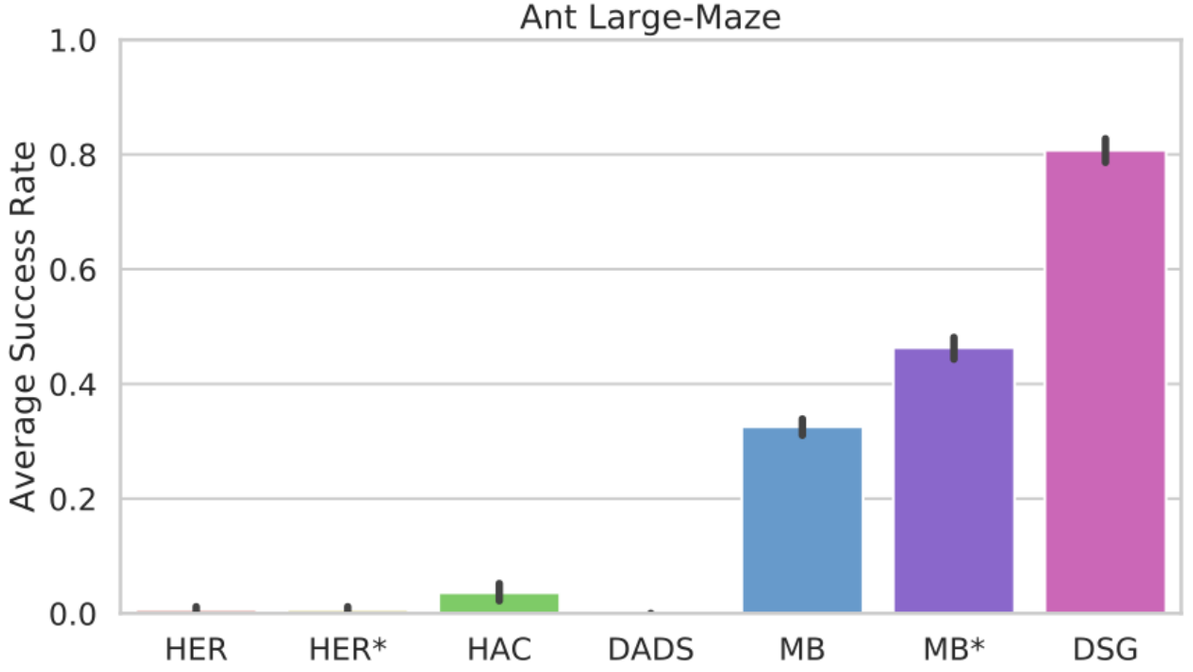
- Flat model-free: Hindsight Experience Replay **(HER, HER*)**
- Flat model-based: **MB, MB*** [1]
- Hierarchical model-free: Hierarchical Actor-Critic **(HAC)**
- Hierarchical model-based: Dynamics Aware Unsupervised Discovery of Skills **(DADS)**

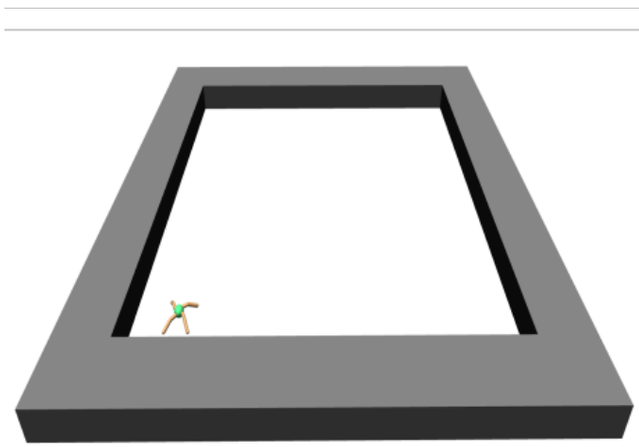
[1] Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, ICRA 2018

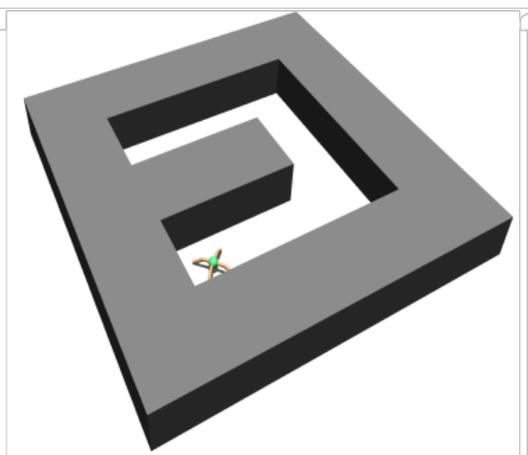


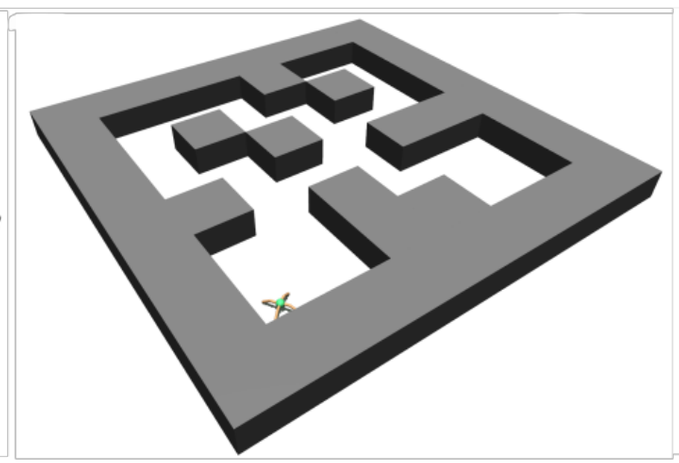







Ant-Reacher


Ant U-Maze


Ant Medium-Maze


Ant Large-Maze

Conclusion

- Skill graphs abstract **large continuous** MDPs into small **discrete** ones suitable for **planning**

- Skill graph **expands incrementally** outward from the start state — high-level exploration

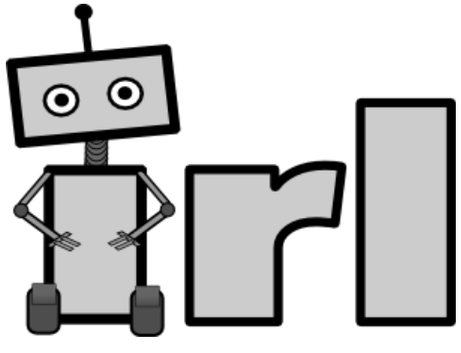
- DSG uses **planning** to get to the **frontier** and then explores

Conclusion

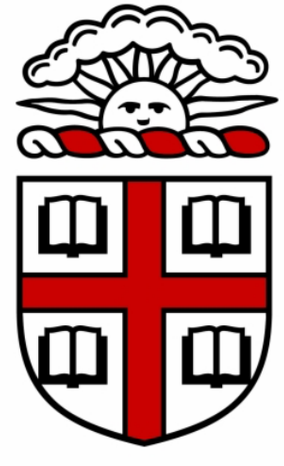
- Skill graphs abstract **large continuous** MDPs into small **discrete** ones suitable for **planning**

- Skill graph **expands incrementally** outward from the start state — high-level exploration

- DSG uses **planning** to get to the **frontier** and then explores



Thank you!



Thank you