Laplacian Representations for Decision-Time Planning


Dikshant Shehmar


Matthew Schlegel


Matthew E. Taylor


Marlos C. Machado

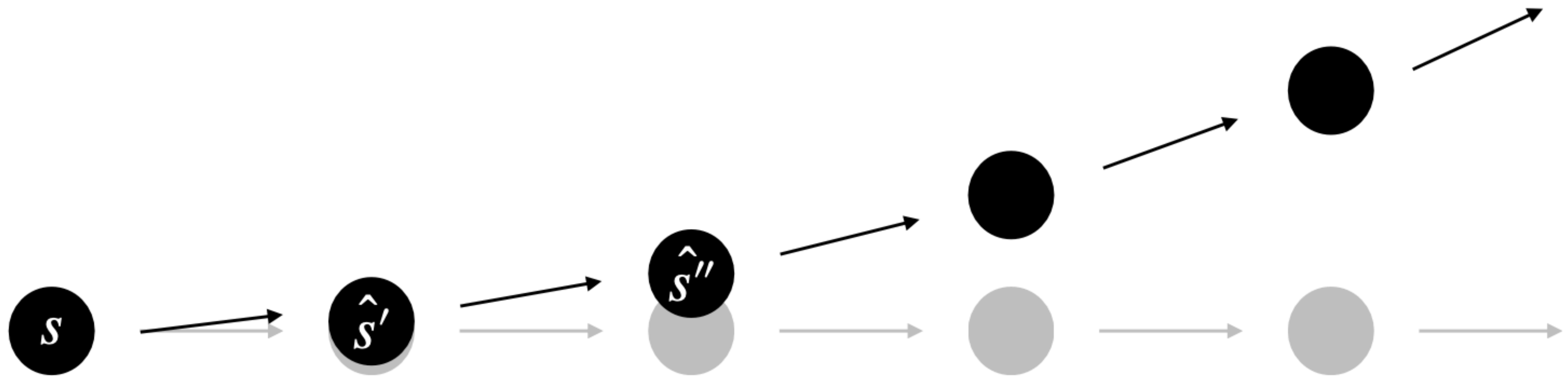
UNIVERSITY OF ALBERTA

Motivation

- Planning with a learned model is challenging because:

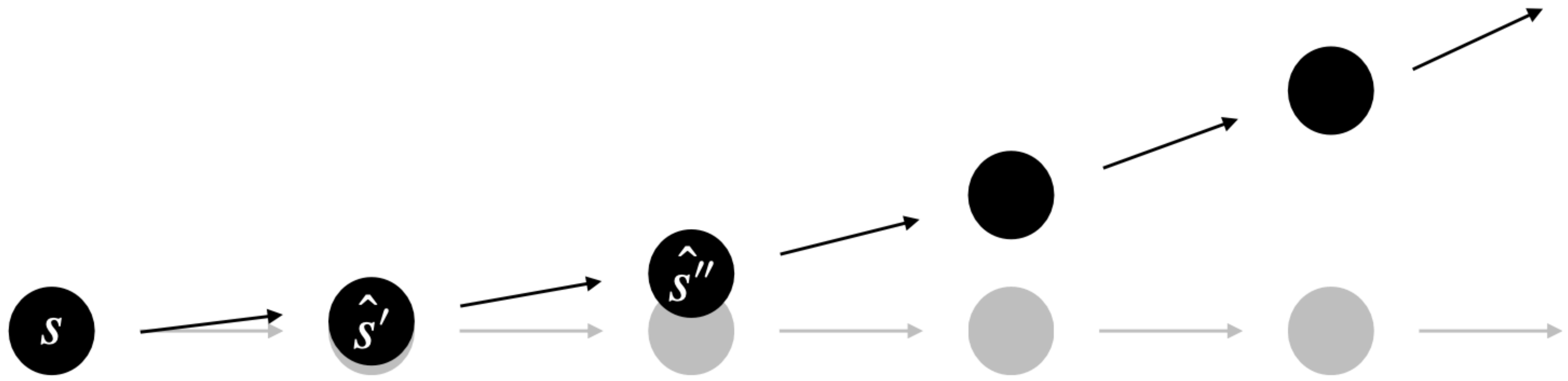
Motivation

- Planning with a learned model is challenging because:
  - Compounding error problem — small prediction errors accumulate over long horizon

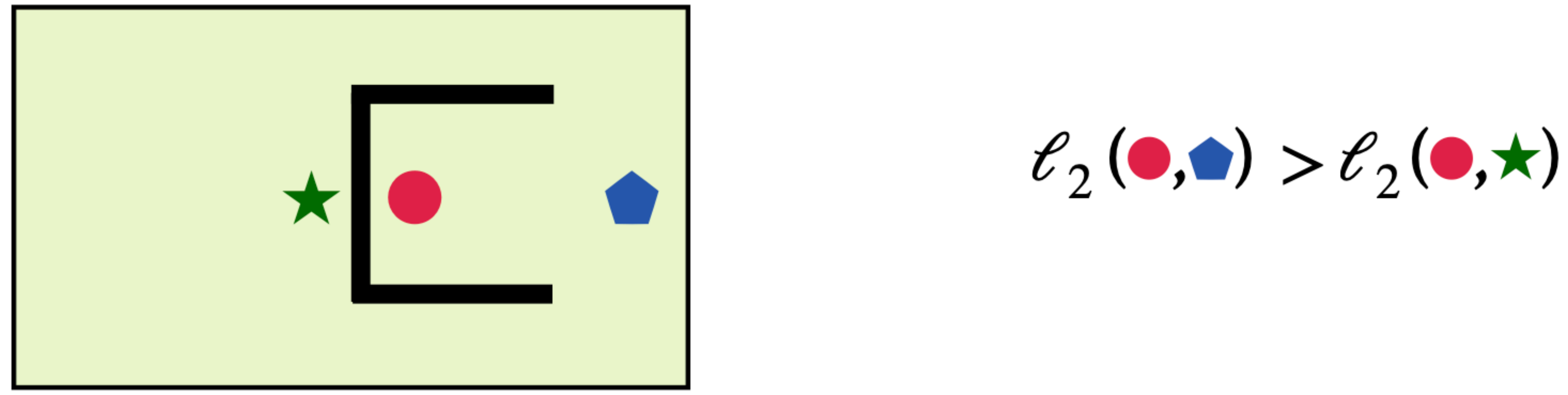


Motivation

- Planning with a learned model is challenging because:
  - Compounding error problem — small prediction errors accumulate over long horizon



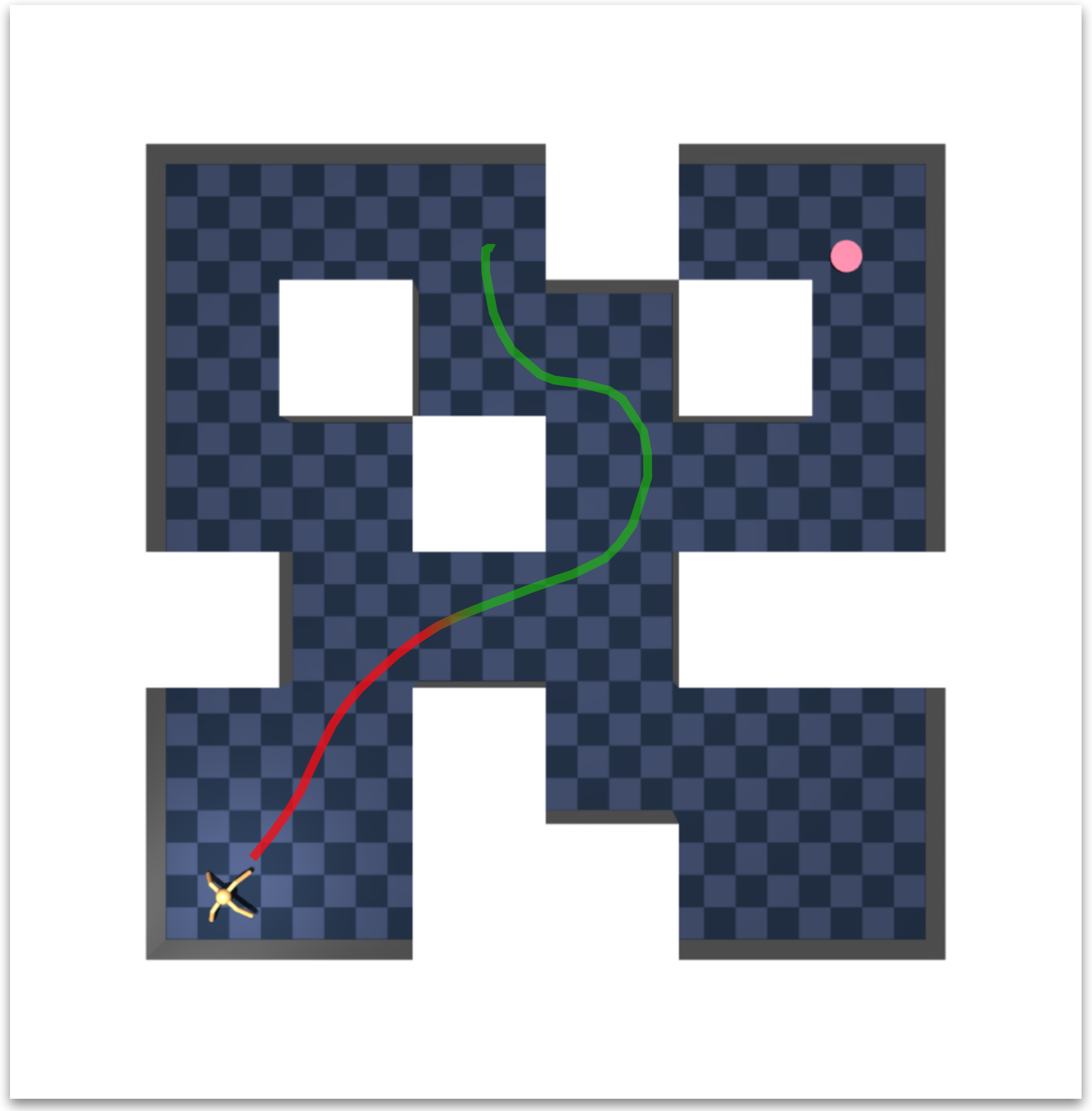
  - Metric mismatch — ℓ_2 distance in state space doesn't capture structure of environment



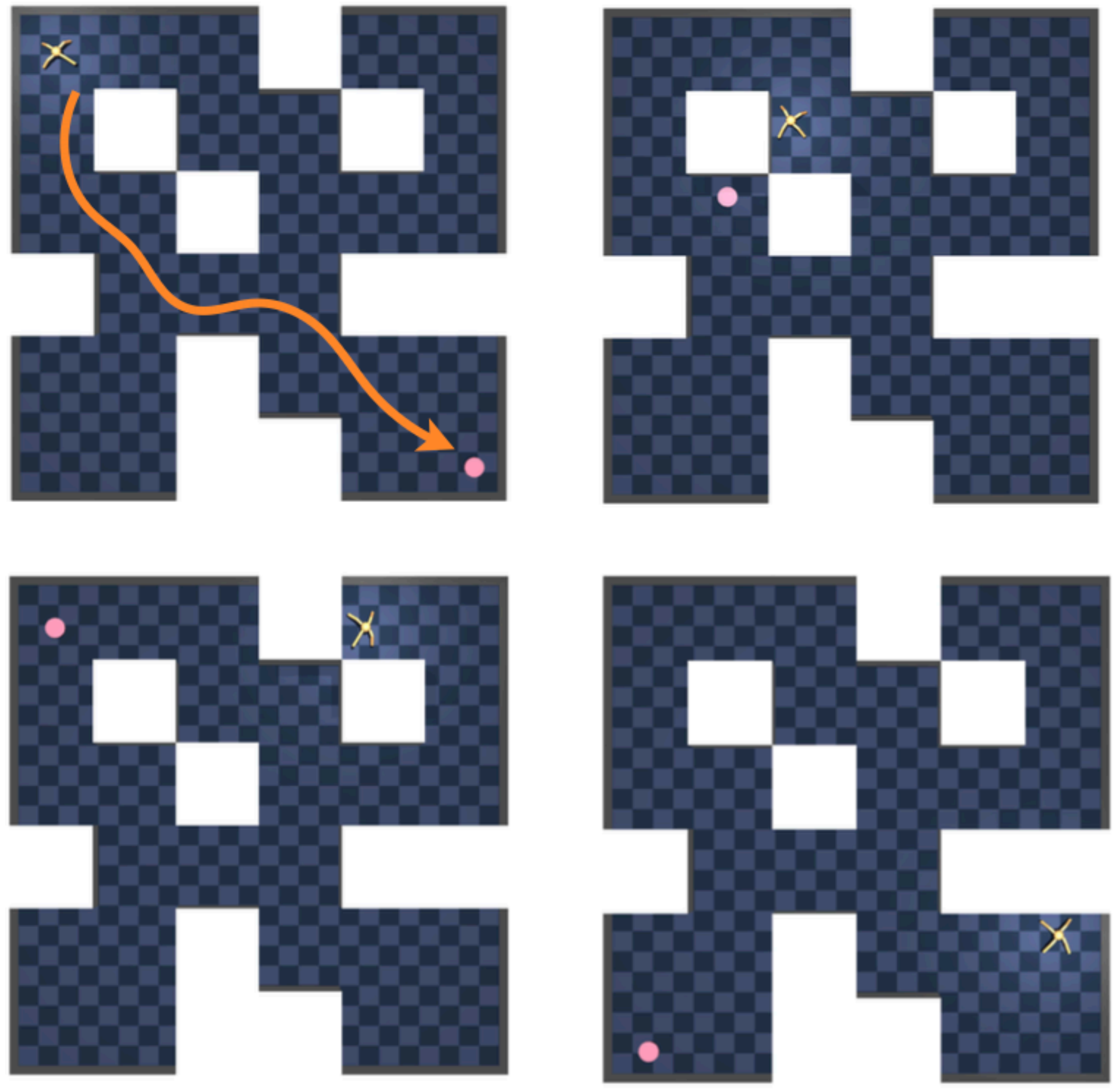
How can we plan with a learned model to solve long-horizon problems?

Problem Setting

Given fixed dataset of transitions, learn a policy that can reach any goal state from any start state



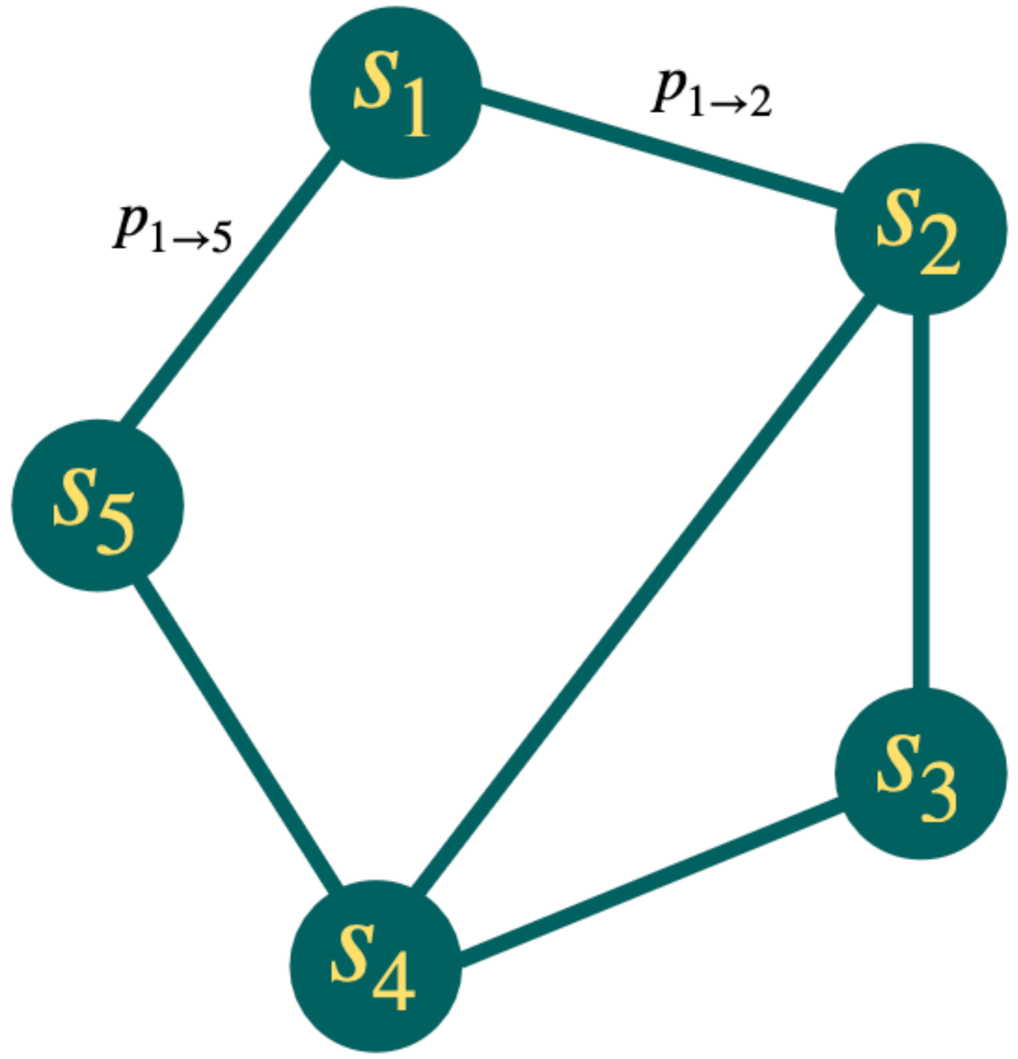
Offline dataset



Downstream tasks

Laplacian Representation

- Graph Laplacian matrix is defined as: $\mathbf{L} = \mathbf{I} - \mathbf{P}_\pi$



$$\mathbf{P}_\pi = \begin{bmatrix} 0 & p_{1\to2} & 0 & 0 & p_{1\to5} \\ p_{2\to1} & 0 & p_{2\to3} & p_{2\to4} & 0 \\ 0 & p_{3\to2} & 0 & p_{3\to4} & 0 \\ 0 & p_{4\to2} & p_{4\to3} & 0 & p_{4\to5} \\ p_{5\to1} & 0 & 0 & p_{5\to4} & 0 \end{bmatrix}$$

Laplacian Representation

- Graph Laplacian matrix is defined as: $\mathbf{L} = \mathbf{I} - \mathbf{P}_\pi$

- Eigenvectors from the eigendecomposition: $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$

- Laplacian representation is a state representation mapping $\psi : \mathcal{S} \to \mathbb{R}^d$ defined from the eigenvectors of the graph Laplacian

Laplacian Representation

Laplacian representation is a state representation mapping $\psi : \mathcal{S} \to \mathbb{R}^d$ defined from the eigenvectors of the graph Laplacian

Approximating eigenvectors ($\mathbf{u}$) using Augmented Lagrangian Laplacian Objective(ALLO)

$$\max_{\beta} \min_{\mathbf{u}} \sum_{i=1}^{d} \langle \mathbf{u}_i, \mathbf{L}\mathbf{u}_i \rangle + \sum_{j=1}^{d} \sum_{k=1}^{j} \beta_{jk} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right) + b \sum_{j=1}^{d} \sum_{k=1}^{j} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right)^2$$

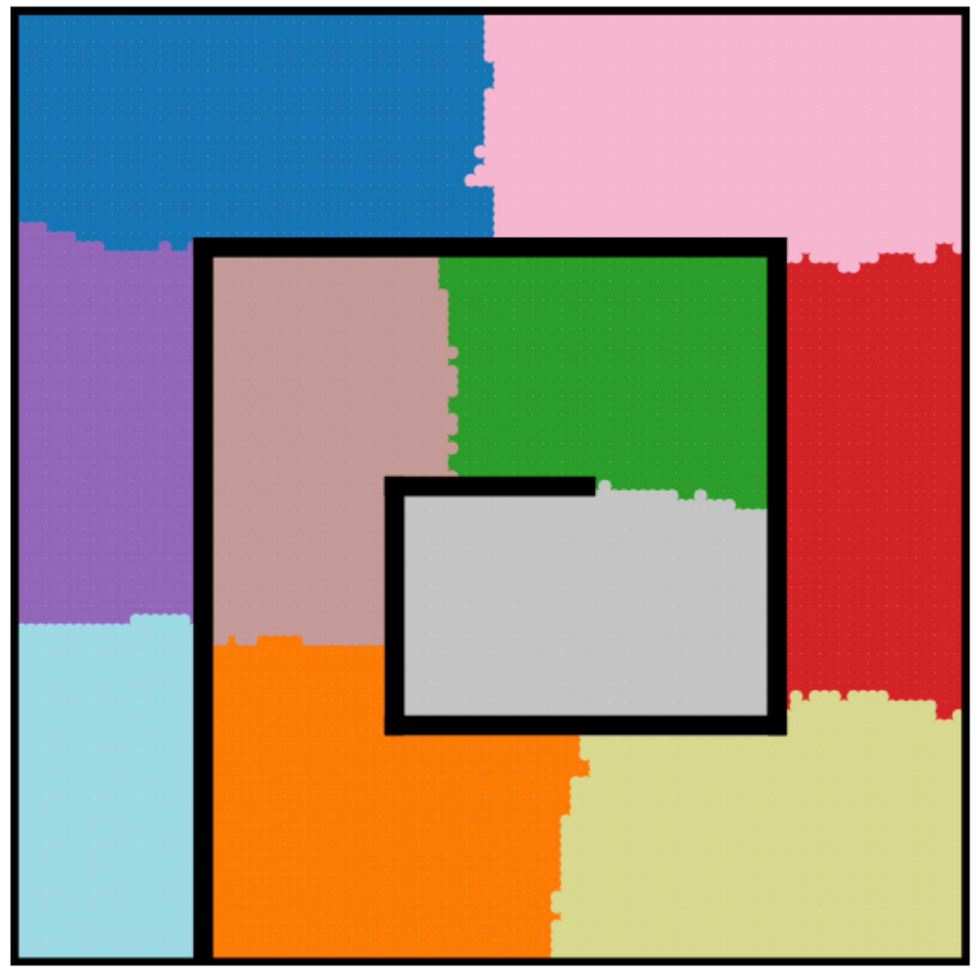
temporal proximity

orthonormality constraint

Source: Gomez et al., Proper Laplacian Representation Learning, ICLR 2024

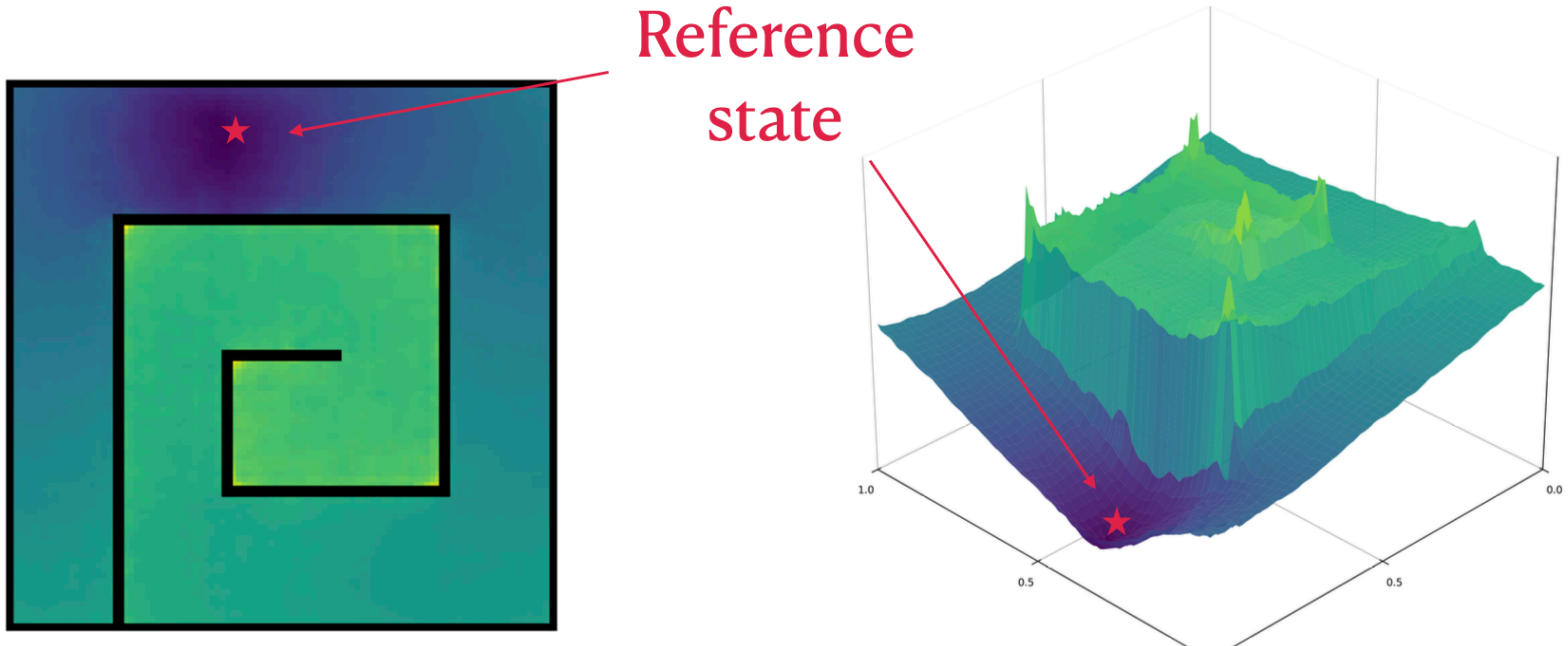
Laplacian Representation

Spectral Clustering



Clustering in Laplacian space

Commute Time Distance



ℓ_2 distance in Laplacian space

Augmented Laplacian Planning with Subgoals

LPS

ALPS

ALPS is a hierarchical planning algorithm with two key ideas:

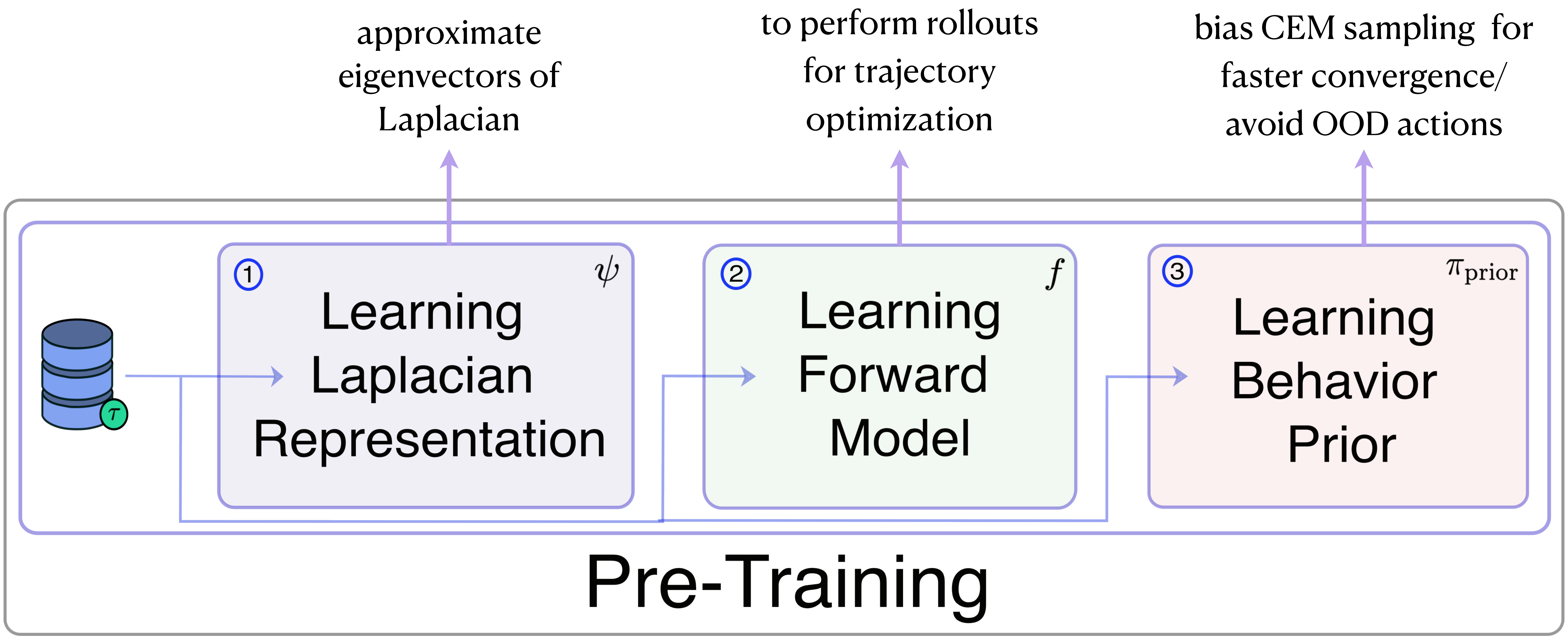
1. Learn a Laplacian representation that captures environment structure, then perform clustering in this latent space.
2. Plan hierarchically:
    i. High-level: Dijkstra's algorithm over clusters to find a sequence of subgoals
    ii. Low-level: CEM with the learned model to navigate b/w consecutive subgoals

Long horizons become a sequence of short, more reliable planning problems!

ALPS

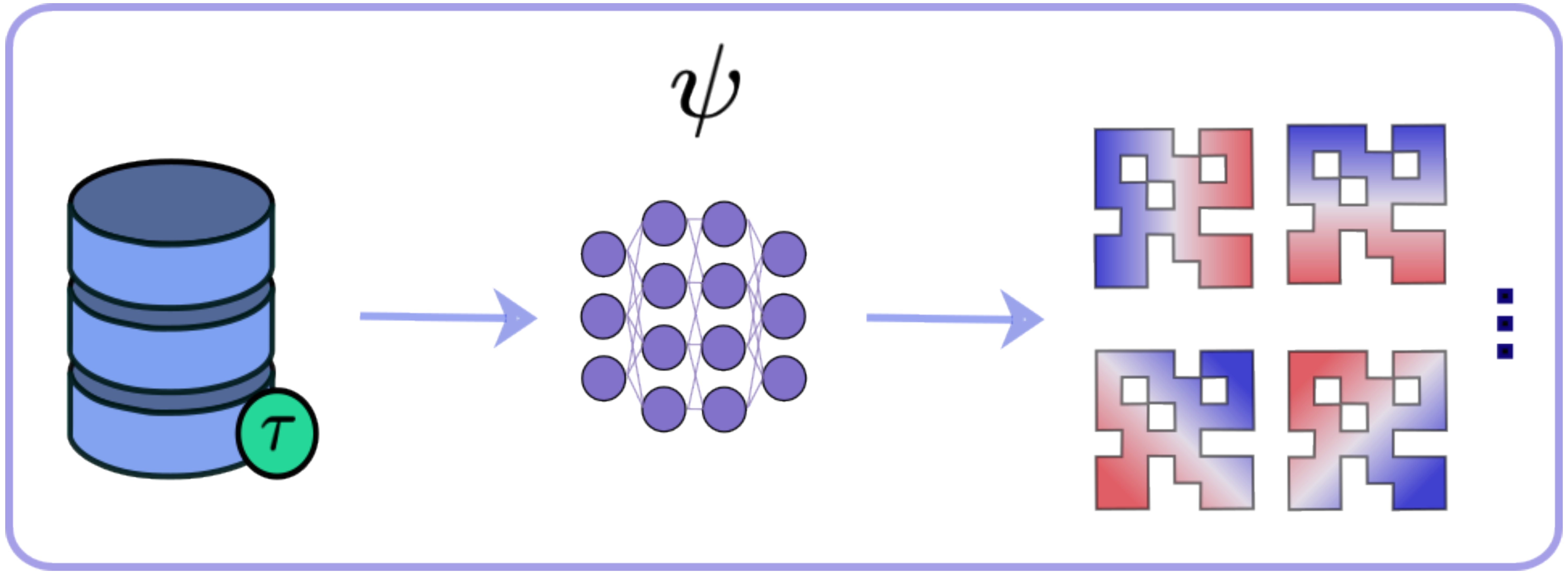
1
ψ
Learning Laplacian Representation
2
f
Learning Forward Model
3
π_{prior}
Learning Behavior Prior
4
Spectral Clustering
Building Cluster Graph
Pre-Training
High-Level Planning
Low-Level Planning
Decision-Time Planning

ALPS


approximate eigenvectors of Laplacian
to perform rollouts for trajectory optimization
bias CEM sampling for faster convergence/ avoid OOD actions
τ
1
ψ
Learning Laplacian Representation
2
f
Learning Forward Model
3
π_prior
Learning Behavior Prior
Pre-Training

ALPS

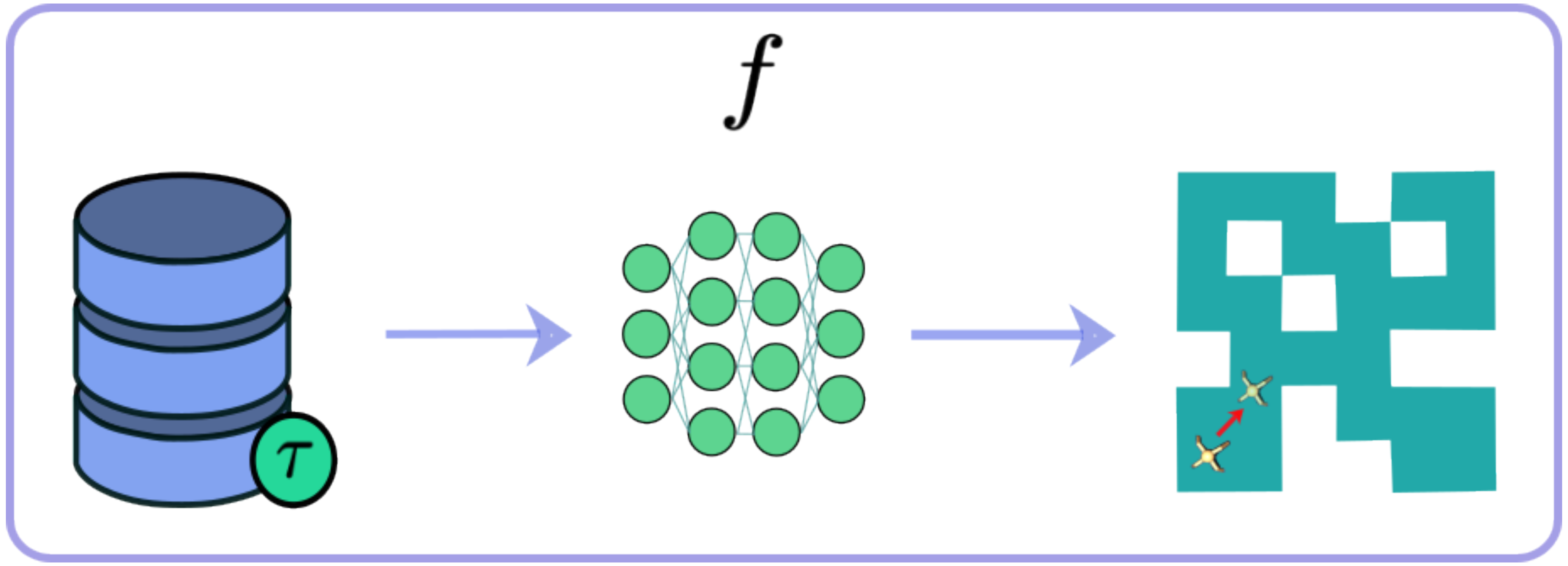
Learn Laplacian representation



$$\ell_{\text{ALLO}} = \max_{\boldsymbol{\beta}} \min_{\mathbf{u}} \sum_{i=1}^{d} \langle \mathbf{u}_i, \mathbf{L}\mathbf{u}_i \rangle + \sum_{j=1}^{d} \sum_{k=1}^{j} \beta_{jk} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right) + B \sum_{j=1}^{d} \sum_{k=1}^{j} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right)^2$$

ALPS

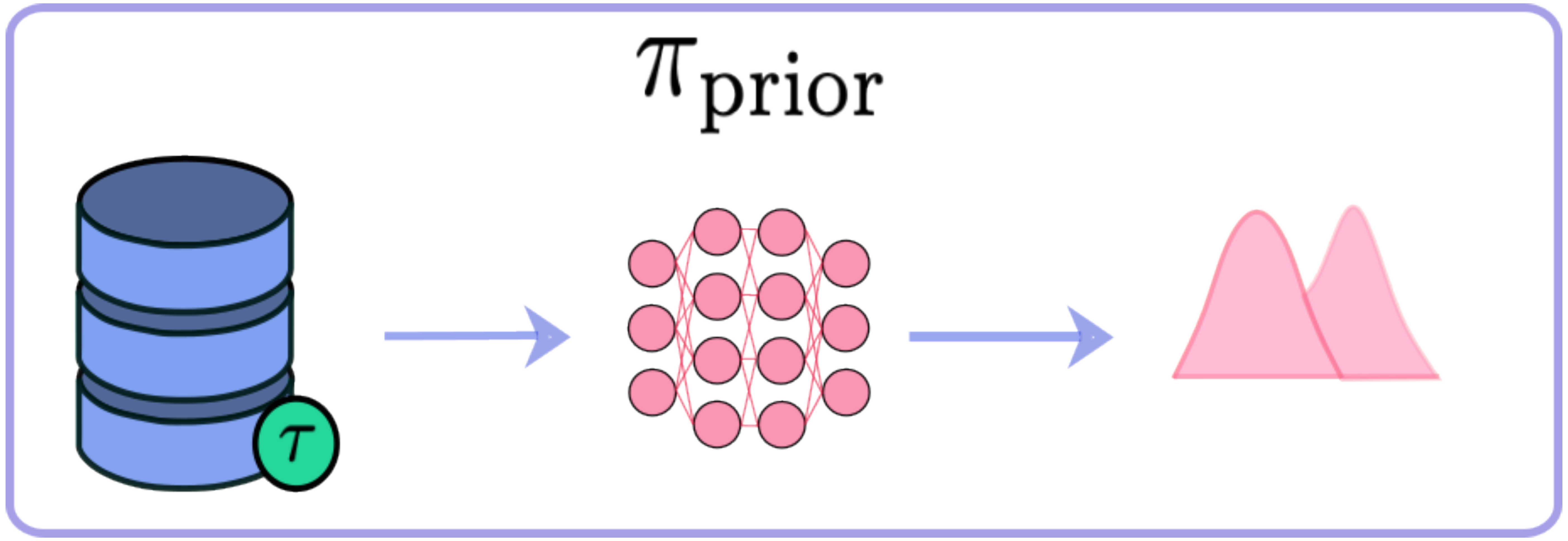
Learn forward model



$$\ell_{\text{forward}} = \mathbb{E}_{(S_t, A_{t:t+H-1}, S_{t+1:t+H}) \sim \mathcal{D}} \left[\frac{1}{H} \sum_{\tau=1}^{H} \| \hat{S}_{t+\tau} - S_{t+\tau} \|_2^2 \right]$$

ALPS

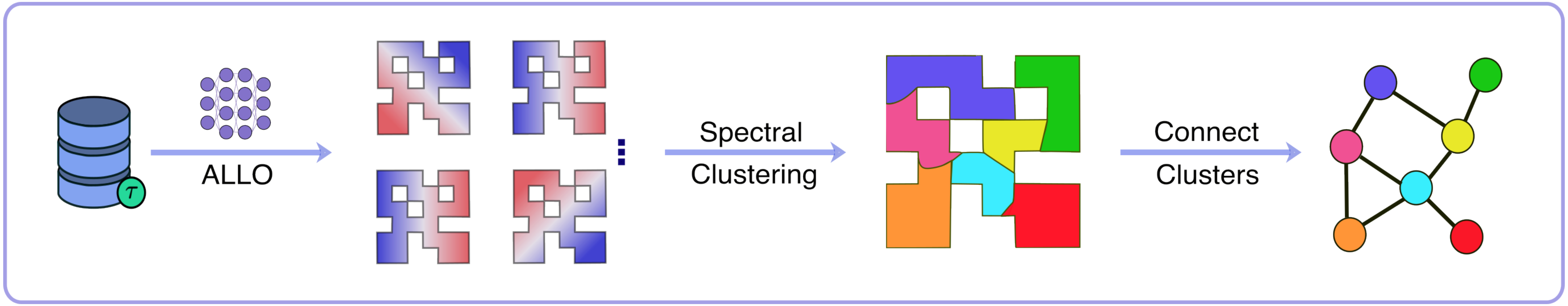
Learn behaviour prior



$$\ell_{\text{prior}} = \mathbb{E}_{(S_t, A_t, S_{t+k}) \sim \mathcal{D}}[\|\pi_{\text{prior}}(S_t, \psi(S_t), \psi(S_{t+k})) - A_t\|_2^2]$$

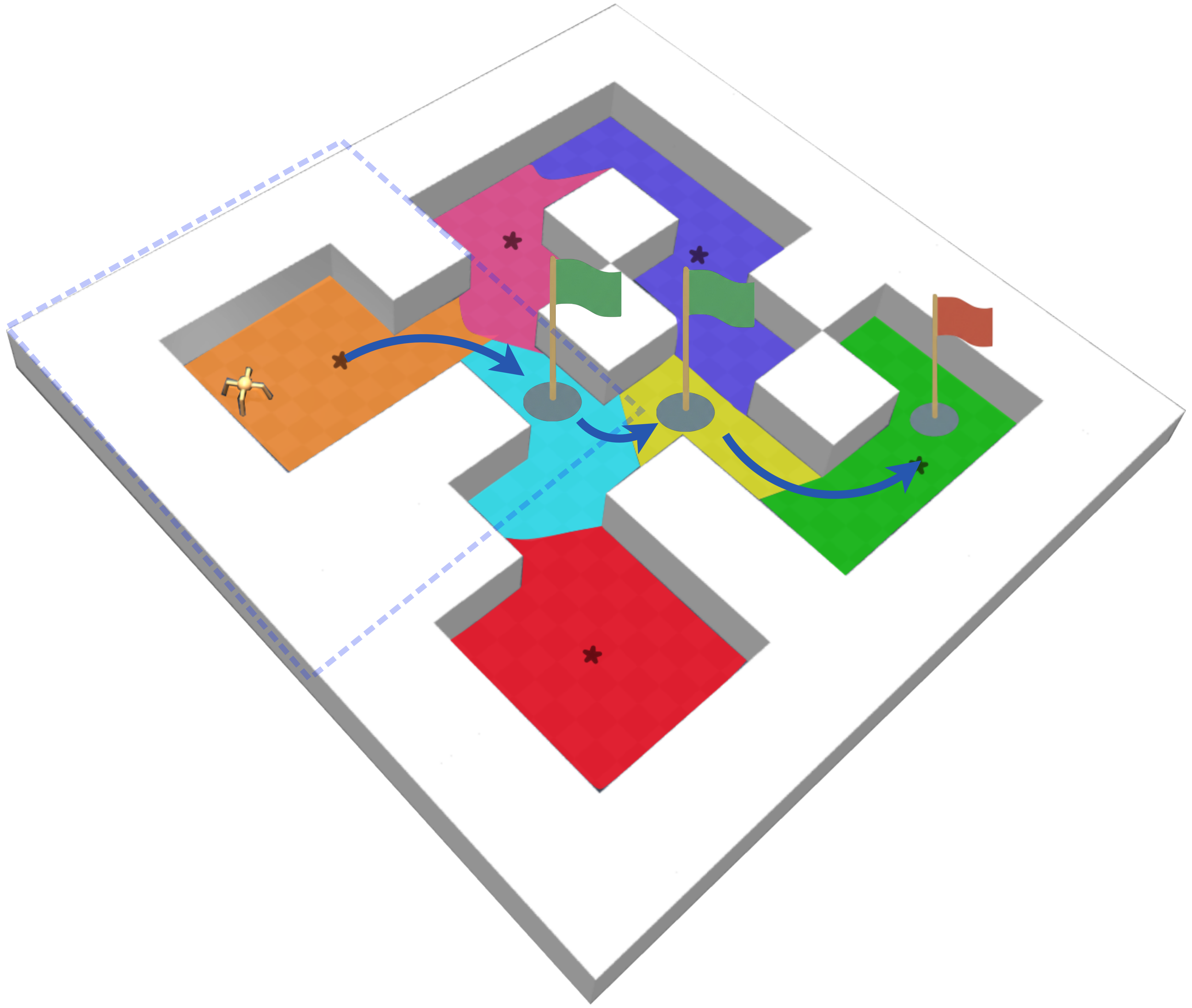
ALPS

Get cluster graph



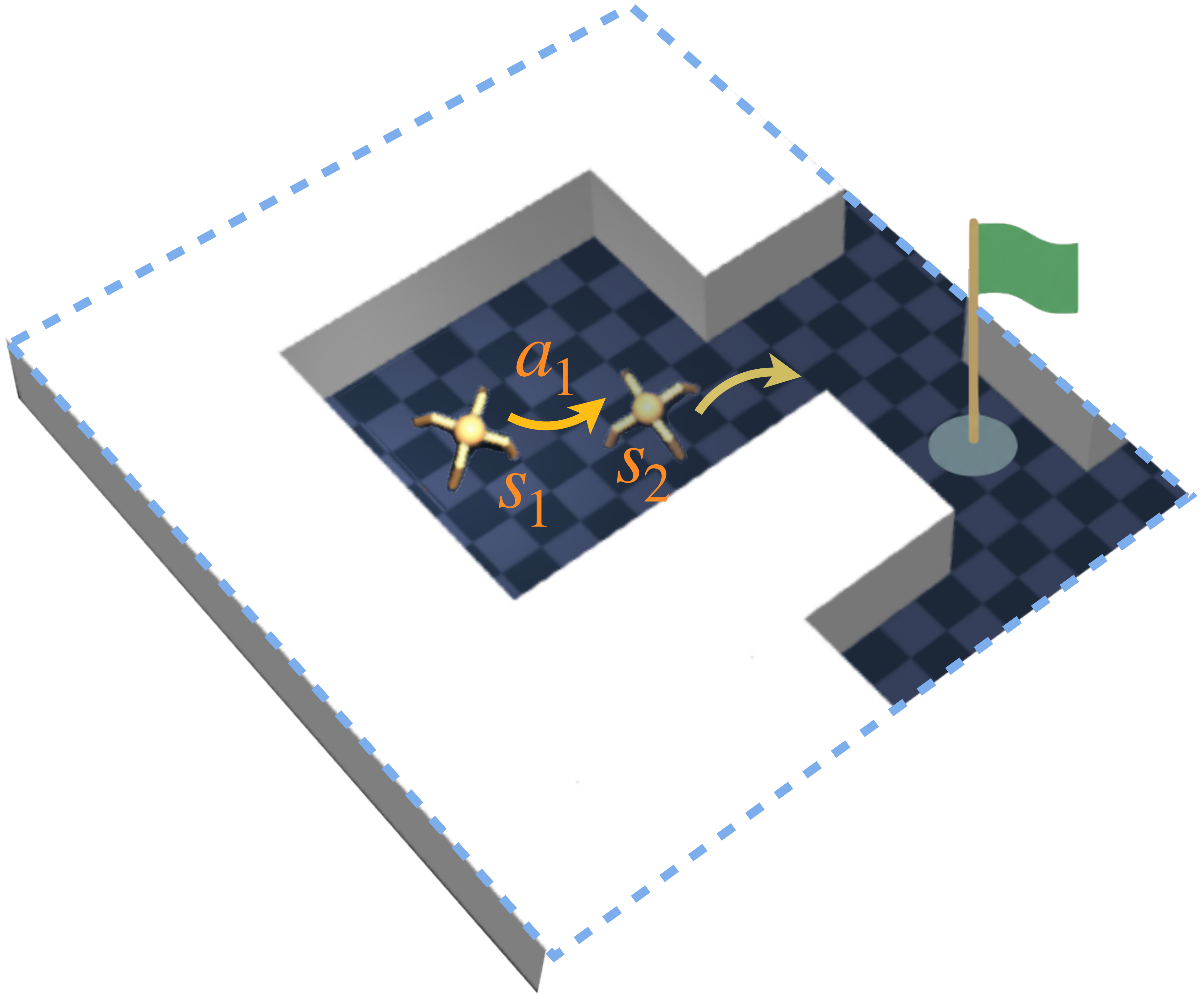
ALPS

Perform decision-time planning



ALPS

Perform decision-time planning

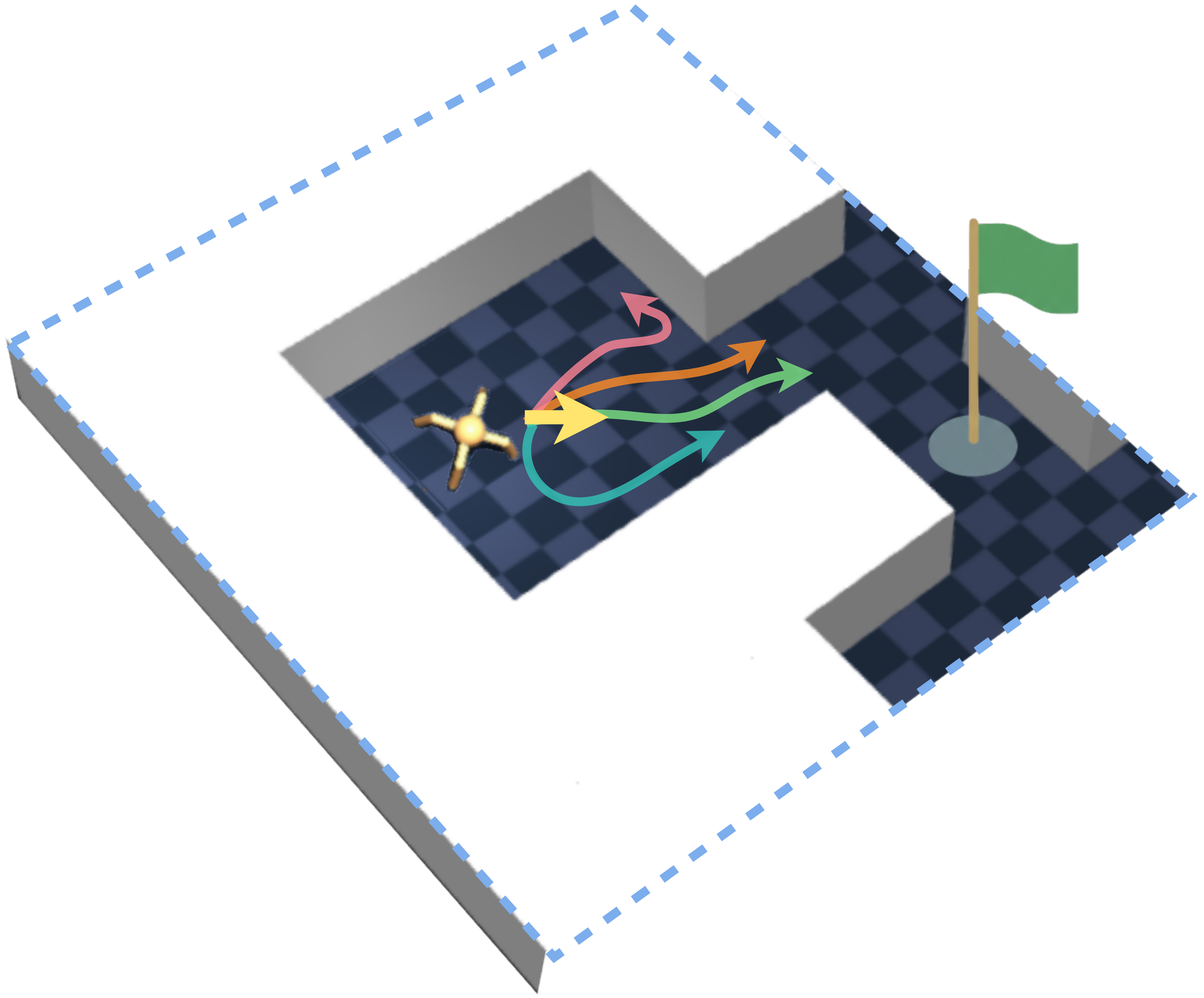


$$\pi_{\text{prior}}\left(s_1, \psi(s_1), \psi_{\text{sub}}\right) \rightarrow a_1$$

$$f(s_1, a_1) \rightarrow s_2$$

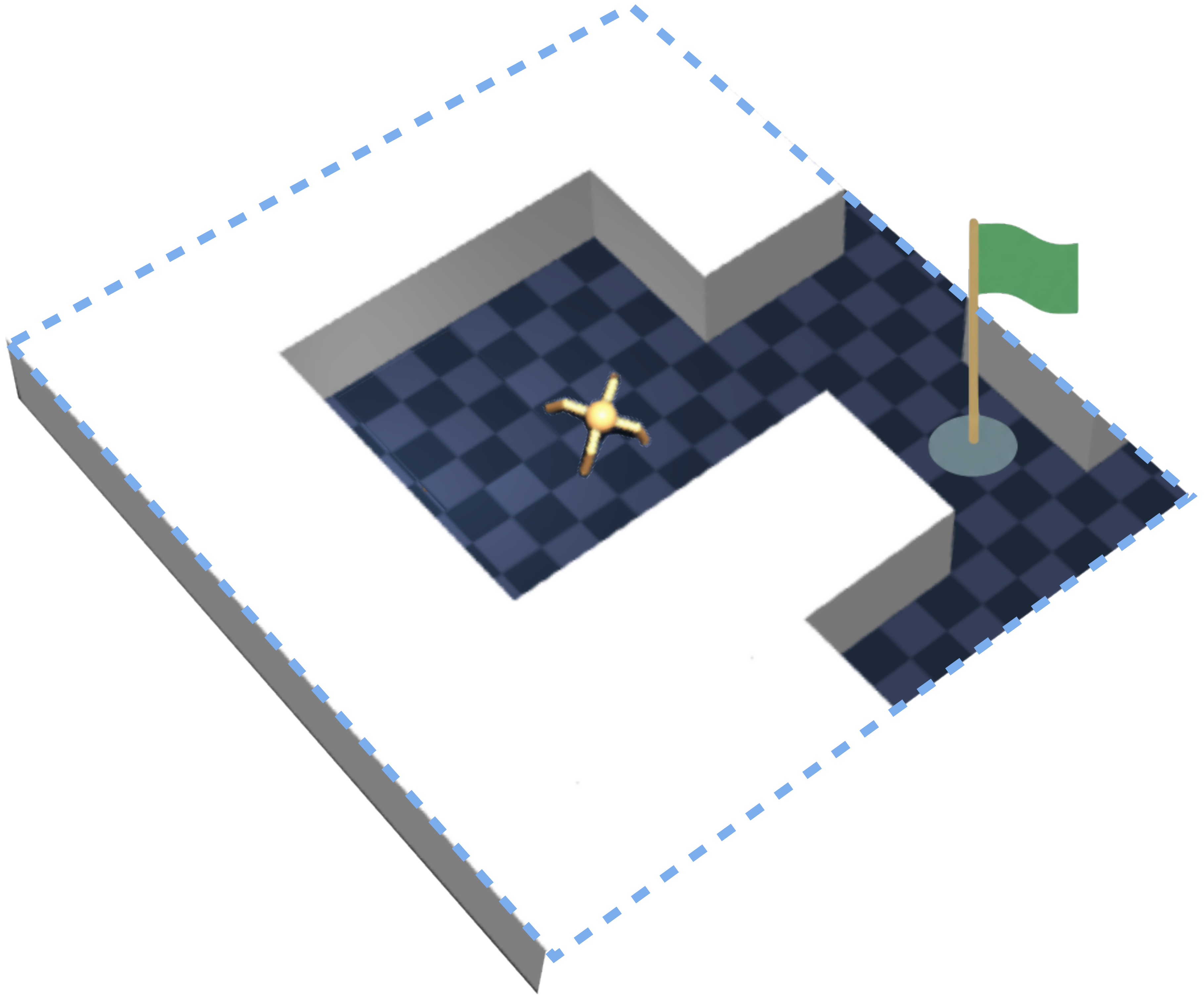
ALPS

Perform decision-time planning



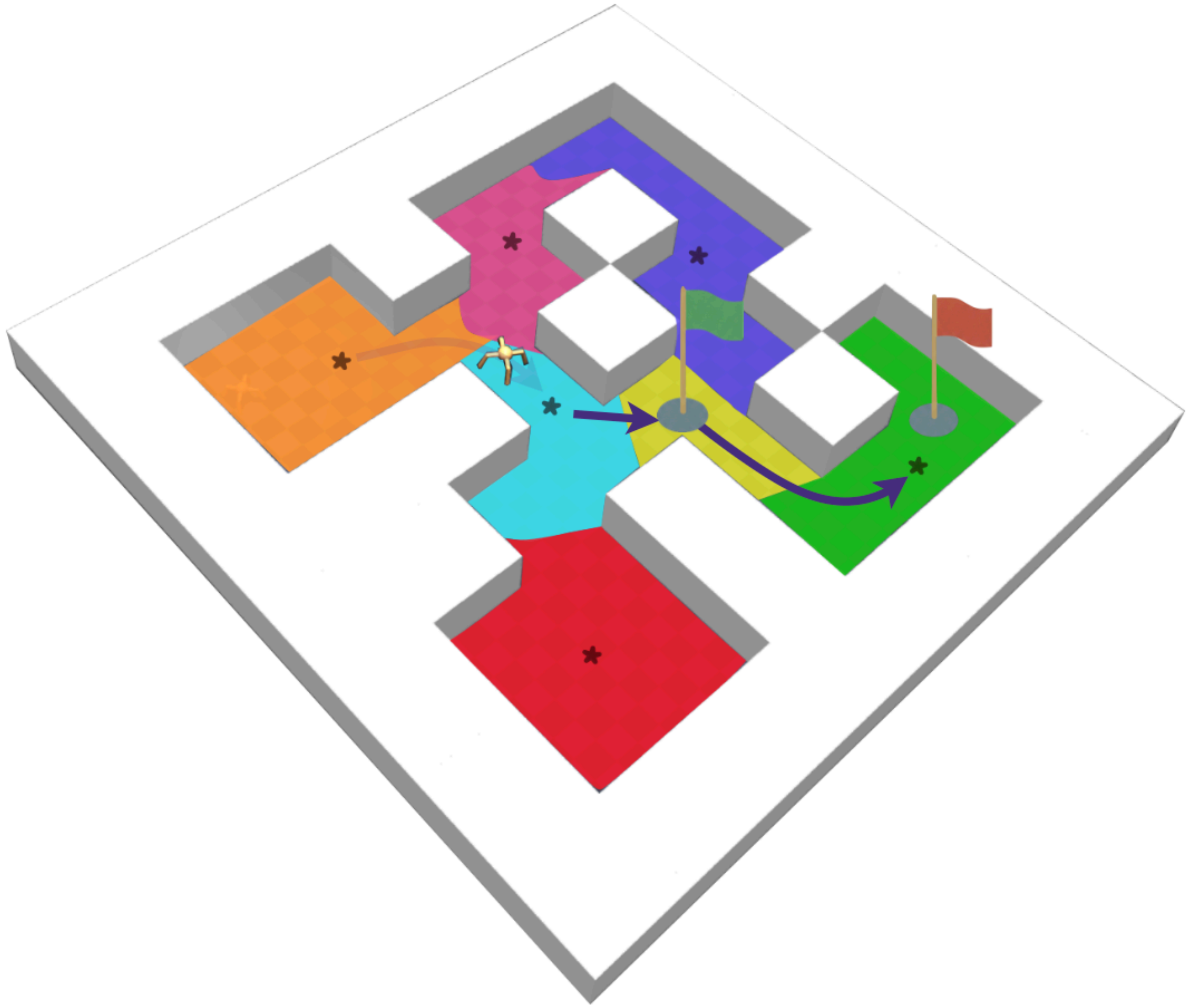
ALPS

Perform decision-time planning



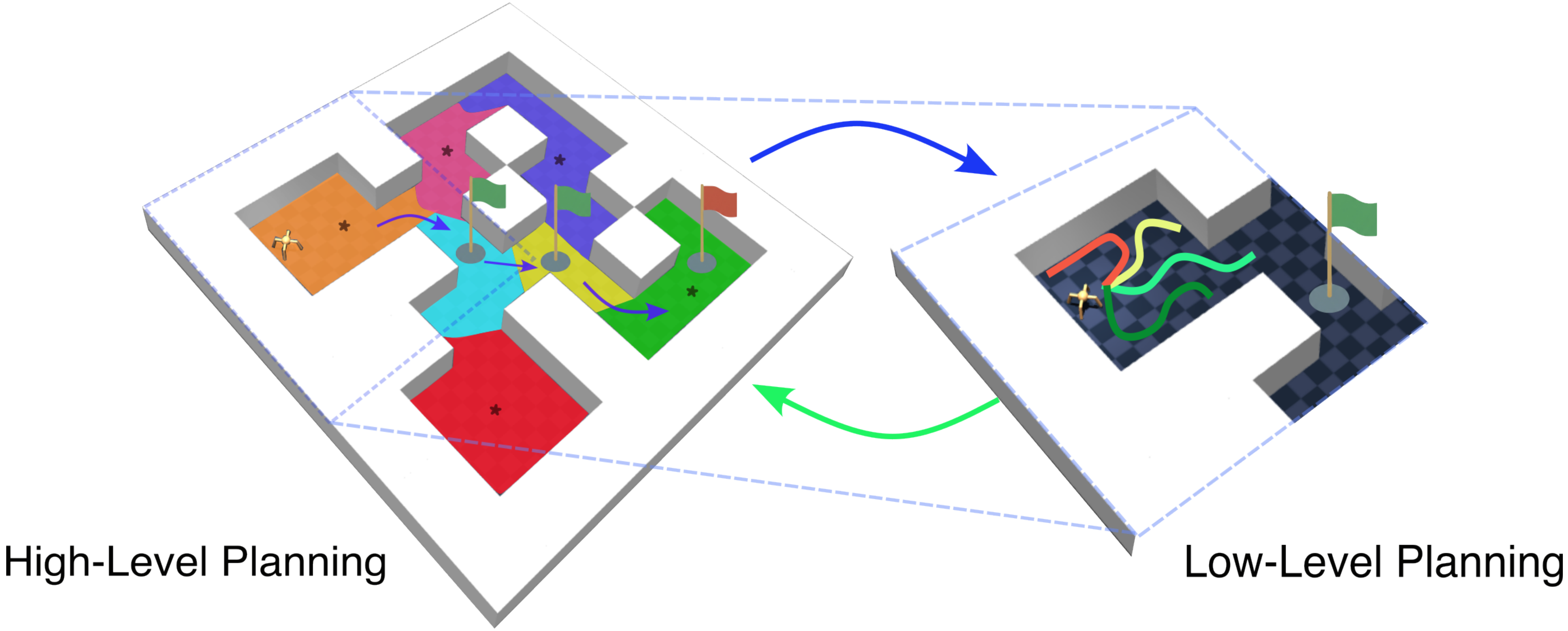
ALPS

Perform decision-time planning



ALPS

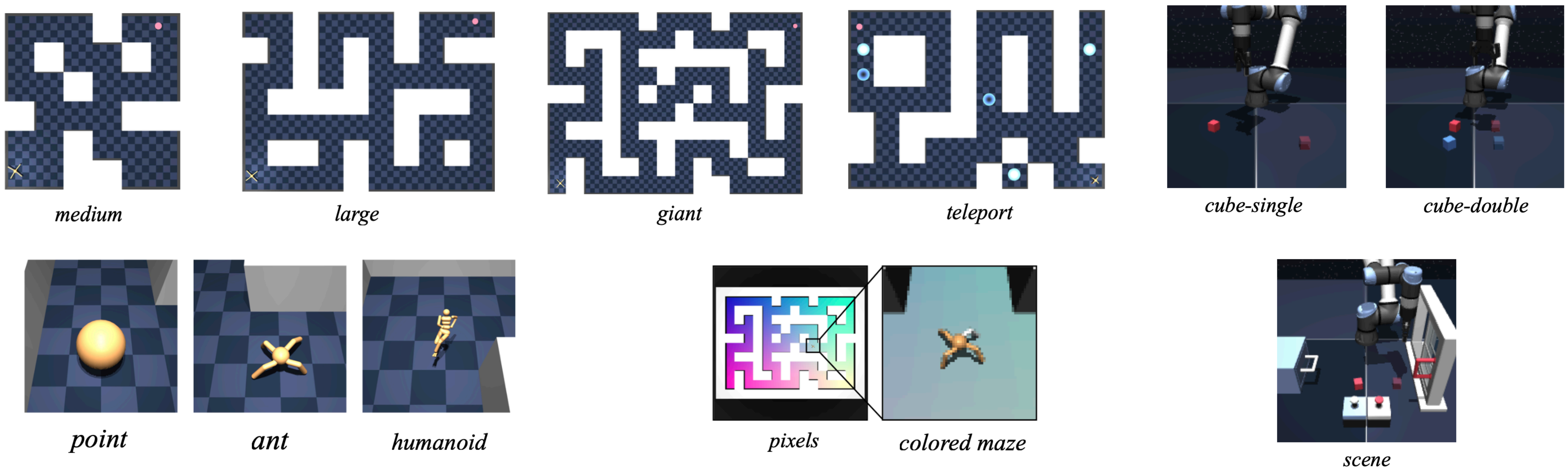
Perform decision-time planning



ALPS

1
ALLO - Eq. 1
ψ
Learning Laplacian Representation
2
Eq. 2
f
Learning Forward Model
3
Eq. 3
π_{prior}
Learning Behavior Prior
4
ALLO
Spectral Clustering
Connect Clusters
Building Cluster Graph
Pre-Training
5
High-Level Planning
Low-Level Planning
Decision-Time Planning

Experimental Domains

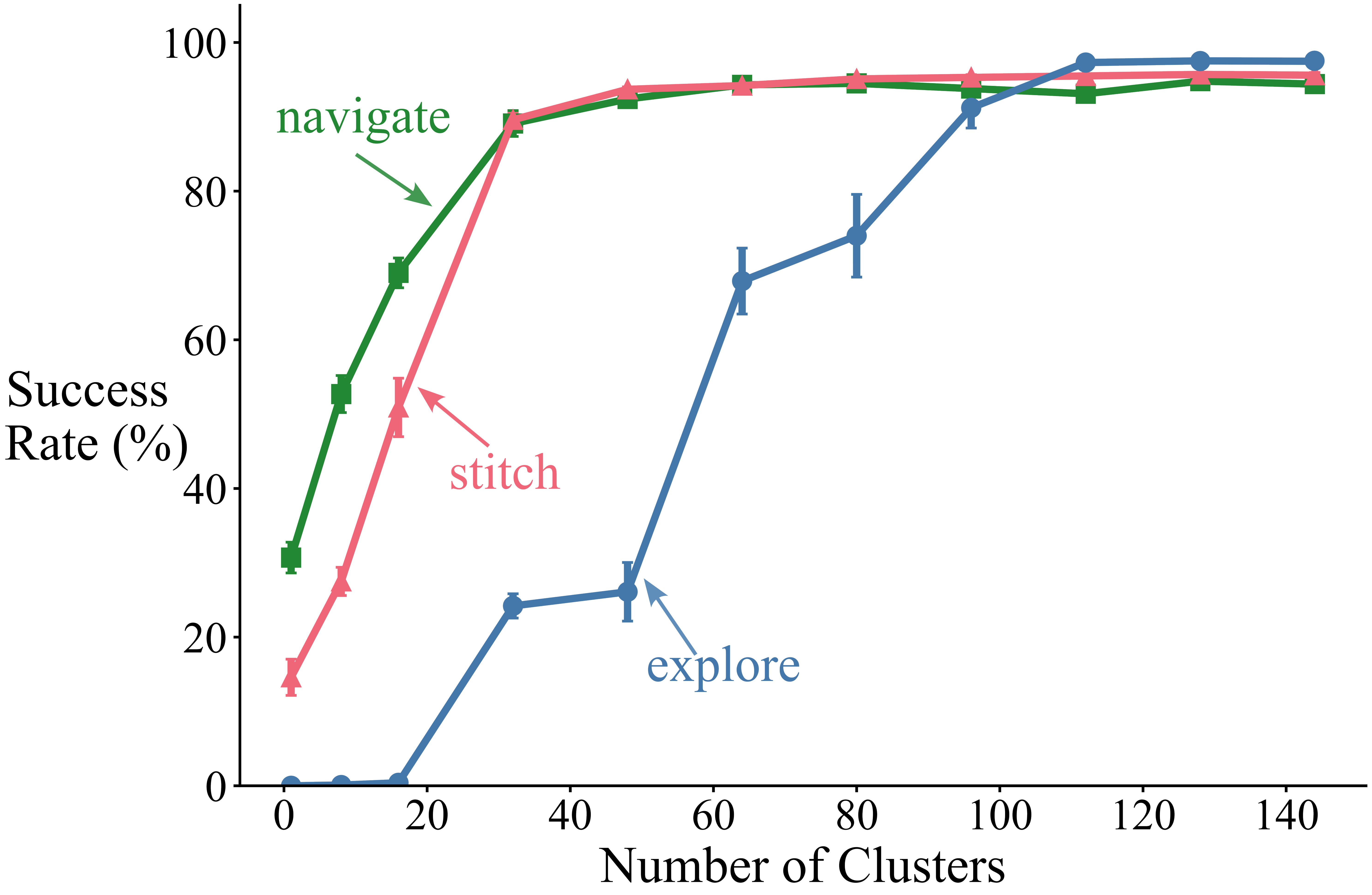


Source: Park et al., OGBench: Benchmarking Offline Goal-Conditioned RL, ICLR 2025

Results

Environment	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	ALPS
antmaze-giant-navigate	0 ± 0	0 ± 0	0 ± 0	14 ± 3	16 ± 3	65 ± 5	69 ± 9
antmaze-giant-stitch	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	2 ± 2	92 ± 3
antmaze-large-explore	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	4 ± 5	90 ± 15
visual-antmaze-giant-stitch	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	55 ± 6
humanoid-giant-navigate	0 ± 0	0 ± 0	0 ± 0	1 ± 0	3 ± 2	12 ± 4	67 ± 11
humanoid-giant-stitch	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	3 ± 2	62 ± 6
cube-single-play	6 ± 2	53 ± 4	68 ± 6	5 ± 1	19 ± 2	15 ± 3	68 ± 6
scene-play	5 ± 1	42 ± 4	51 ± 4	5 ± 1	19 ± 2	38 ± 3	26 ± 3

Results



Success Rate (%)
100
80
60
40
20
0
navigate
stitch
explore
0
20
40
60
80
100
120
140
Number of Clusters

Key Takeaways

- The Laplacian representation might be a promising starting point for model-based reinforcement learning

- Laplacian representation helps us in:
  - Breaking down the problem through spectral clustering
  - Encodes commute time distance for trajectory optimisation

- ALPS is a hierarchical planning algorithm that uses these properties of Laplacian representation to tackle long-horizon planning tasks

Thanks!

paper & code

Dikshant
Shehmar

Matthew
Schlegel

Matthew
E. Taylor

Marlos C.
Machado