Register at: essai.si

ESSAI & ACAI 2023
LJUBLJANA, SLOVENIA

MODEL UNCERTAINTY IN SEQUENTIAL DECISION MAKING

DAVID PARKER
University of Oxford

BRUNO LACERDA
University of Oxford

NICK HAWES
University of Oxford

Recap

- Sample based UMDPs consider a finite set of possible models
  - Enables modelling dependencies between transitions
  - Enables less conservative behaviour
  - Enables adaptive behaviour
  - Problem becomes hard to solve optimally
    - We looked at approximation techniques
- Regret is a suitable measure which trades-off robustness and conservatism
- We optimise for regret where we assume n-step rectangularity rather than $(1$-step) rectangularity
  - Consider n step dependencies

Course contents

- ~~Markov decision processes (MDPs) and stochastic games~~
  - ~~MDPs: key concepts and algorithms~~
  - ~~stochastic games: adding adversarial aspects~~

- ~~Uncertain MDPs~~
  - ~~MDPs + epistemic uncertainty, robust control, robust dynamic programming, interval MDPs, uncertainty set representation, challenges, tools~~

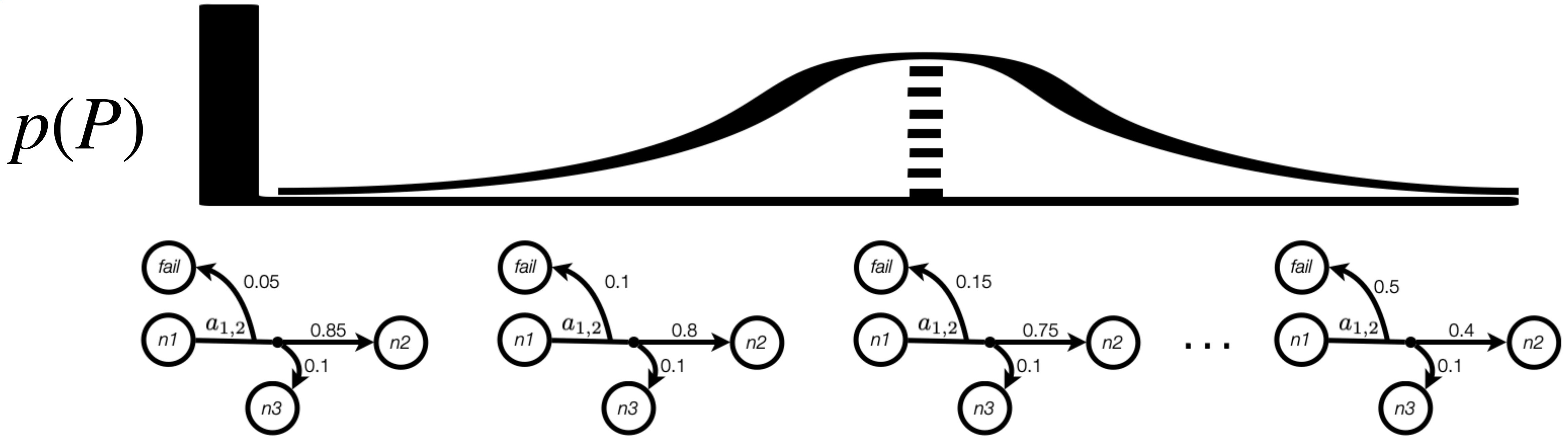
- ~~Sample-based uncertain MDPs~~
  - ~~removing the transition independence assumption~~

- Bayes-adaptive MDPs
  - maintaining a distribution over the possible models
  - usage in mission planning for robots

Bayes-adaptive MDPs

Adding prior over uncertainty set

$$\mathscr{M} = (S, s_0, A, \mathscr{P}, C, goal)$$



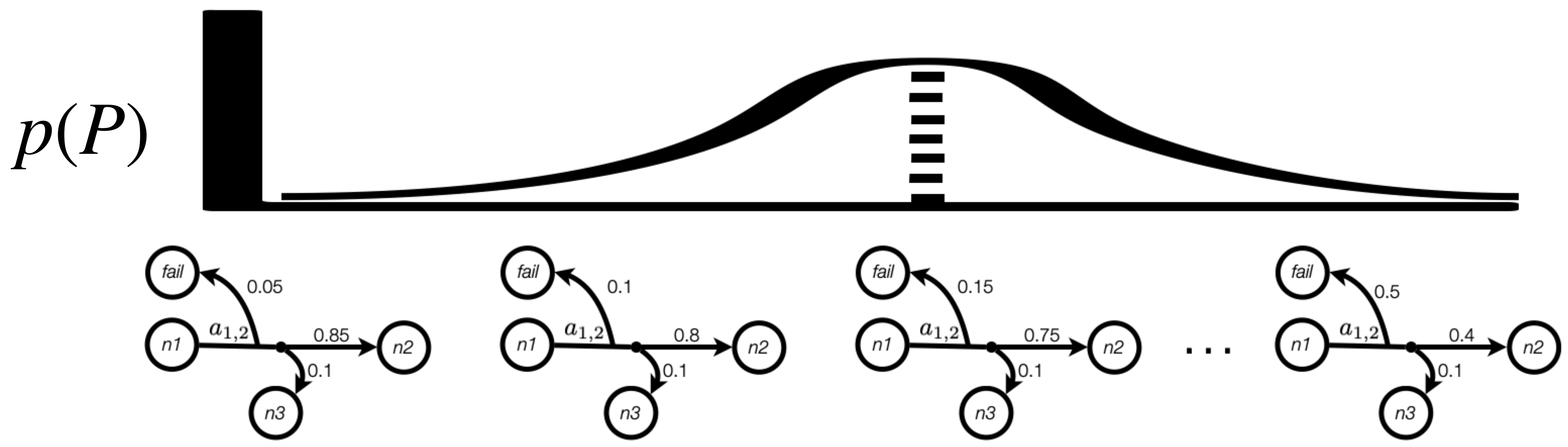
- Add prior $p(P)$ over $\mathscr{P}$

- Turns the problem into a model-based Bayes-adaptive reinforcement learning (RL) problem

- We do not make assumptions on uncertainty set $\mathscr{P}$ or the form of its prior
  - We will see how to work explicitly with a finite $\mathscr{P}$
  - An open question is what are suitable ways of maintaining and updating $p(P)$ when $\mathscr{P}$ is continuous and has dependencies
    - Problem specific
    - We will discuss a few approaches later

Bayes-adaptive MDP

$$\mathscr{M} = (S, s_0, A, \mathscr{P}, C, goal)$$



- Add prior $p(P)$ over $\mathscr{P}$
- The BAMDP for $\mathscr{M}$ is defined as $\mathscr{M}^+ = (S^+, A, s_0, P^+, C^+, goal^+)$, where:
  - $S^+ = (S \times A)^* \times S$ is the set of states
    - A state in the BAMDP is a state-action history (aka path) $s^+ = (s_0 a_0 s_1 a_1 \ldots s_{n-1} a_{n-1} s_n)$
    - We will also use $h \in (S \times A)^*$ and denote BAMDP states as $s^+ = (hs)$
  - The transition function is defined as $P^+(hs, a, hsas') = \int_{P \in \mathscr{P}} P(s, a, s') p(P \mid hs) dP$
    - For finite $\mathscr{P}$, $P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s') p(P \mid hs)$
  - $C^+(hs, a) = C(s, a)$
  - $hs \in goal^+$ if and only if $s \in goal$

Calculating a posterior the uncertainty set

- Using Bayes rule, we can recursively compute the posterior over the uncertainty set given the observed history
  - This is our belief over which is the real model

$$p(P \mid h) = \frac{p(h \mid P)p(P)}{p(h)}$$

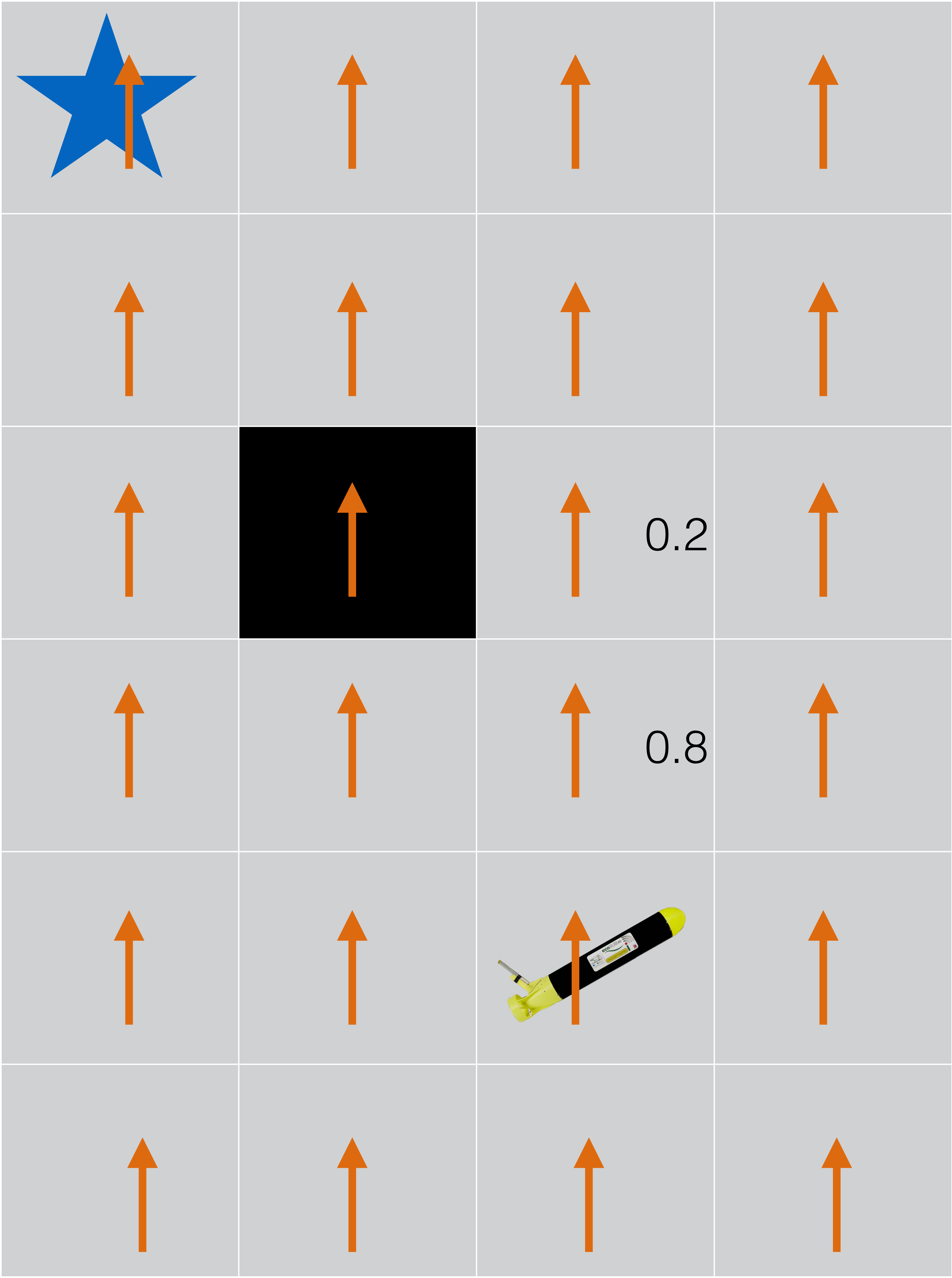
$$p(P \mid s_0) = p(P)$$

$$p(P \mid hsas') = \frac{P(s,a,s')p(P \mid hs)}{\sum_{P' \in \mathscr{P}} P'(s,a,s')p(P' \mid hs)}$$

Example

North currents - P_N

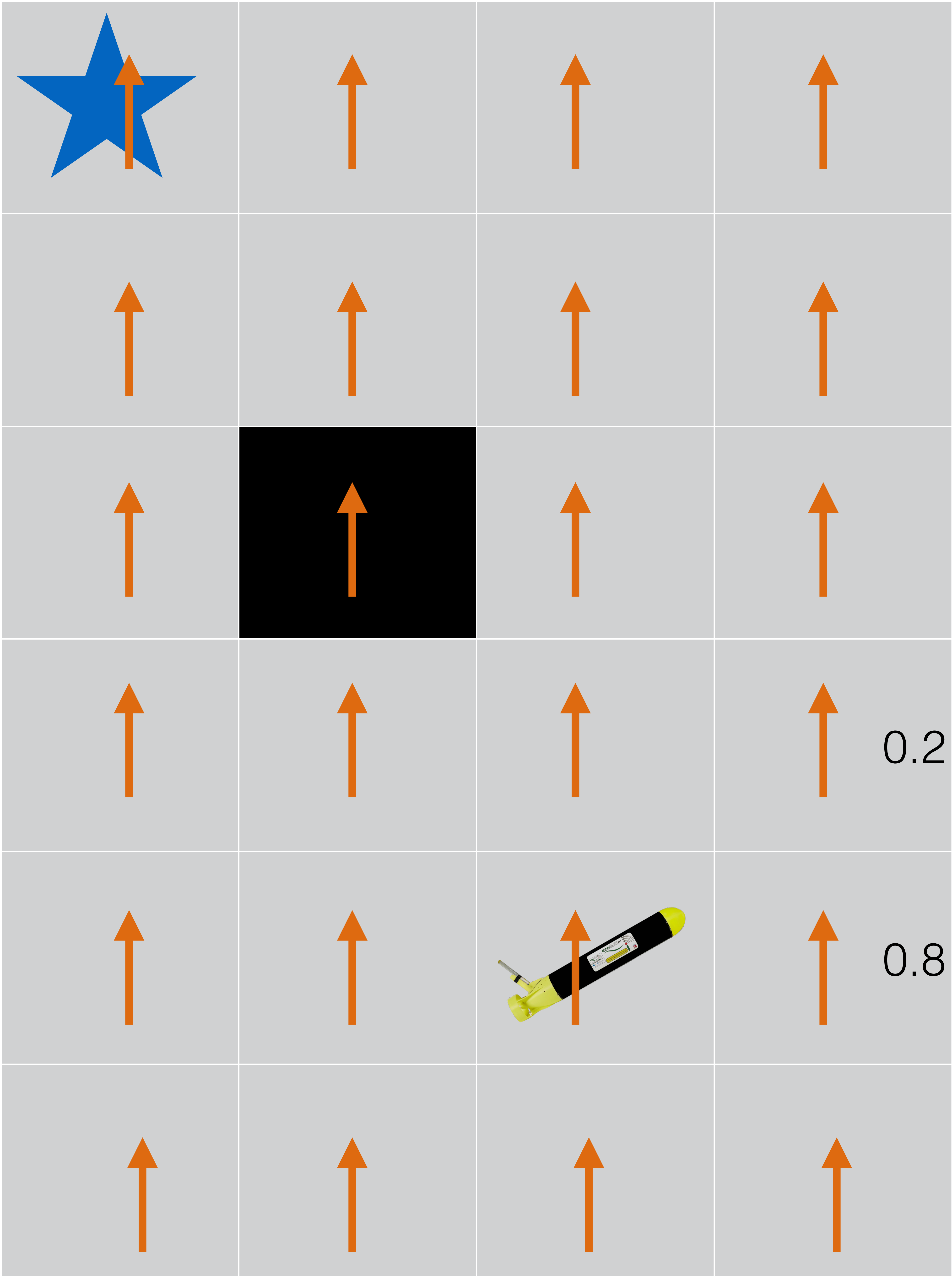
- Action: move up (N)



Example

North currents - P_N

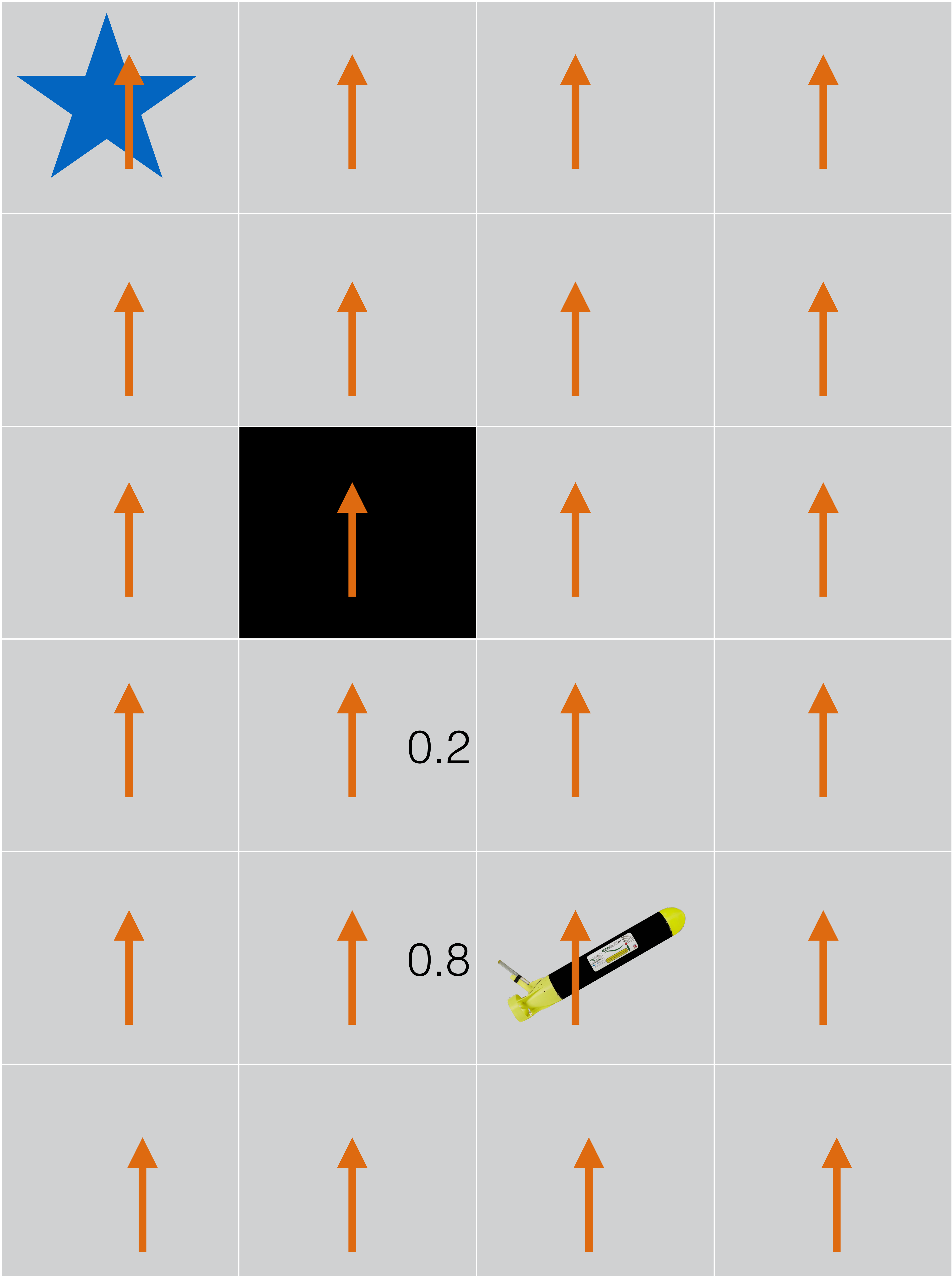
- Action: move east (E)



Example

North currents - P_N

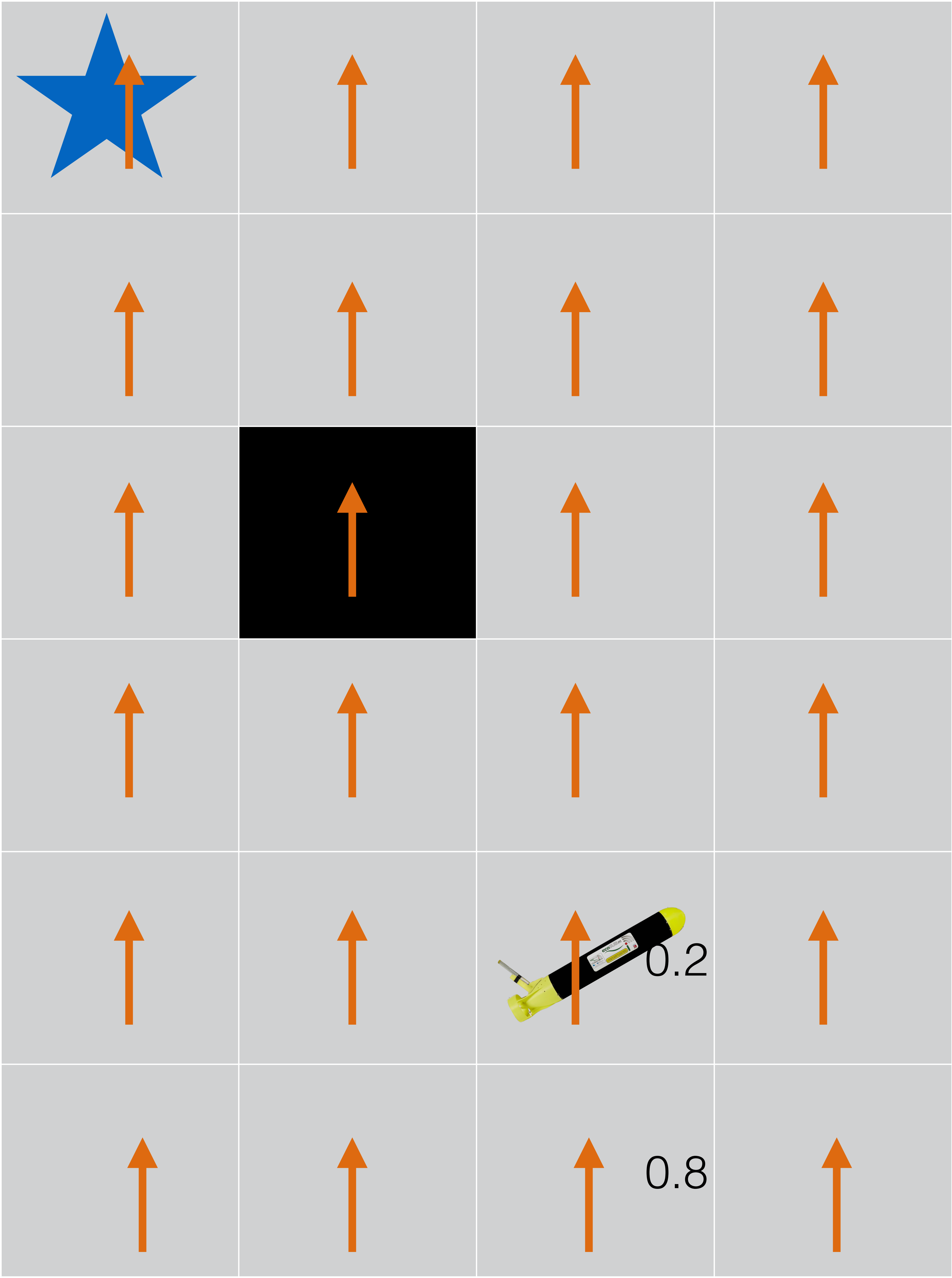
- Action: move west (W)



Example

North currents - P_S

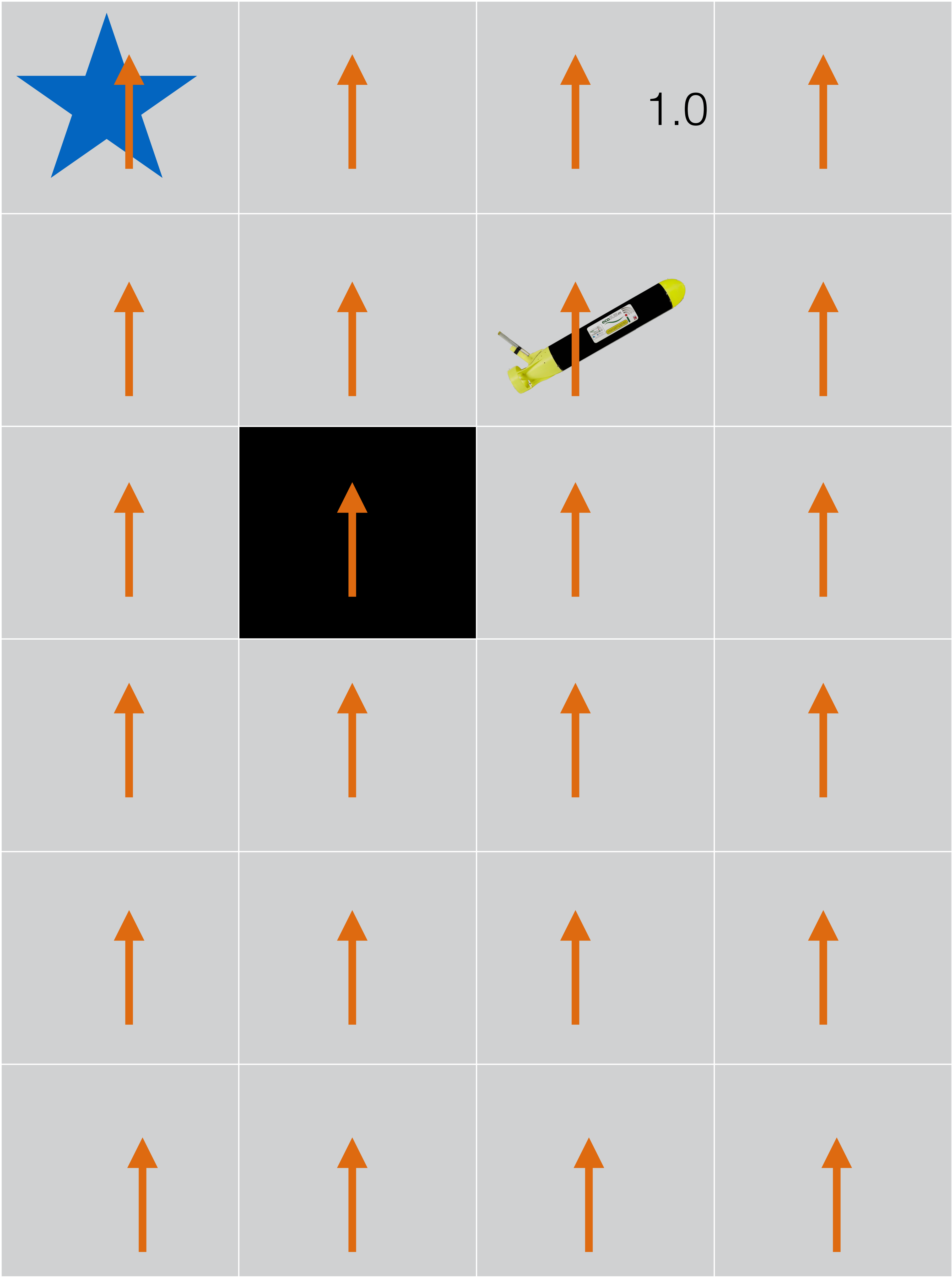
- Action: move south (S)



Example

North currents - P_N

- Action: move up (N)



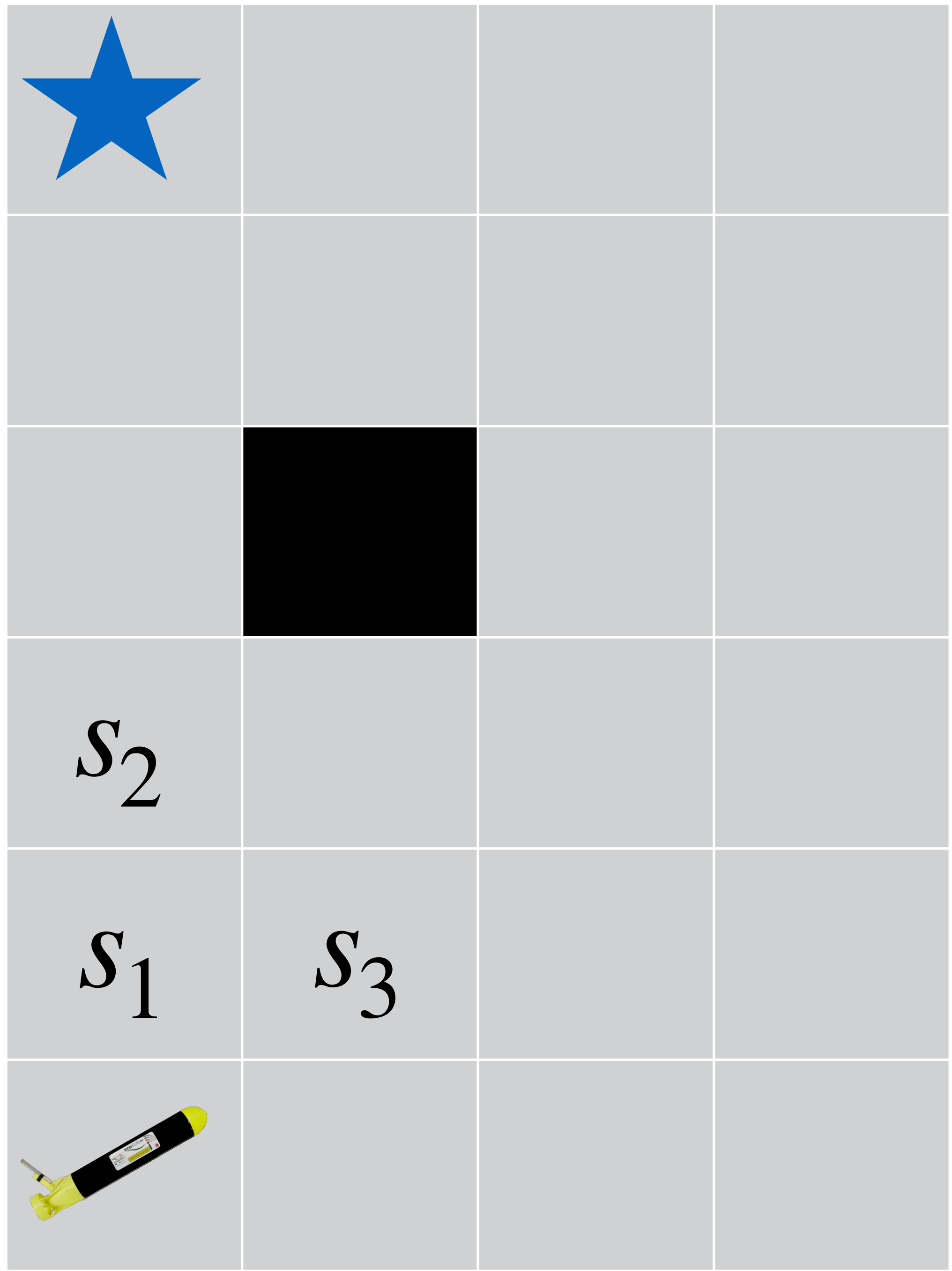
Example

$$P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s') p(P \mid hs)$$

$$p(P \mid s_0) = p(P)$$

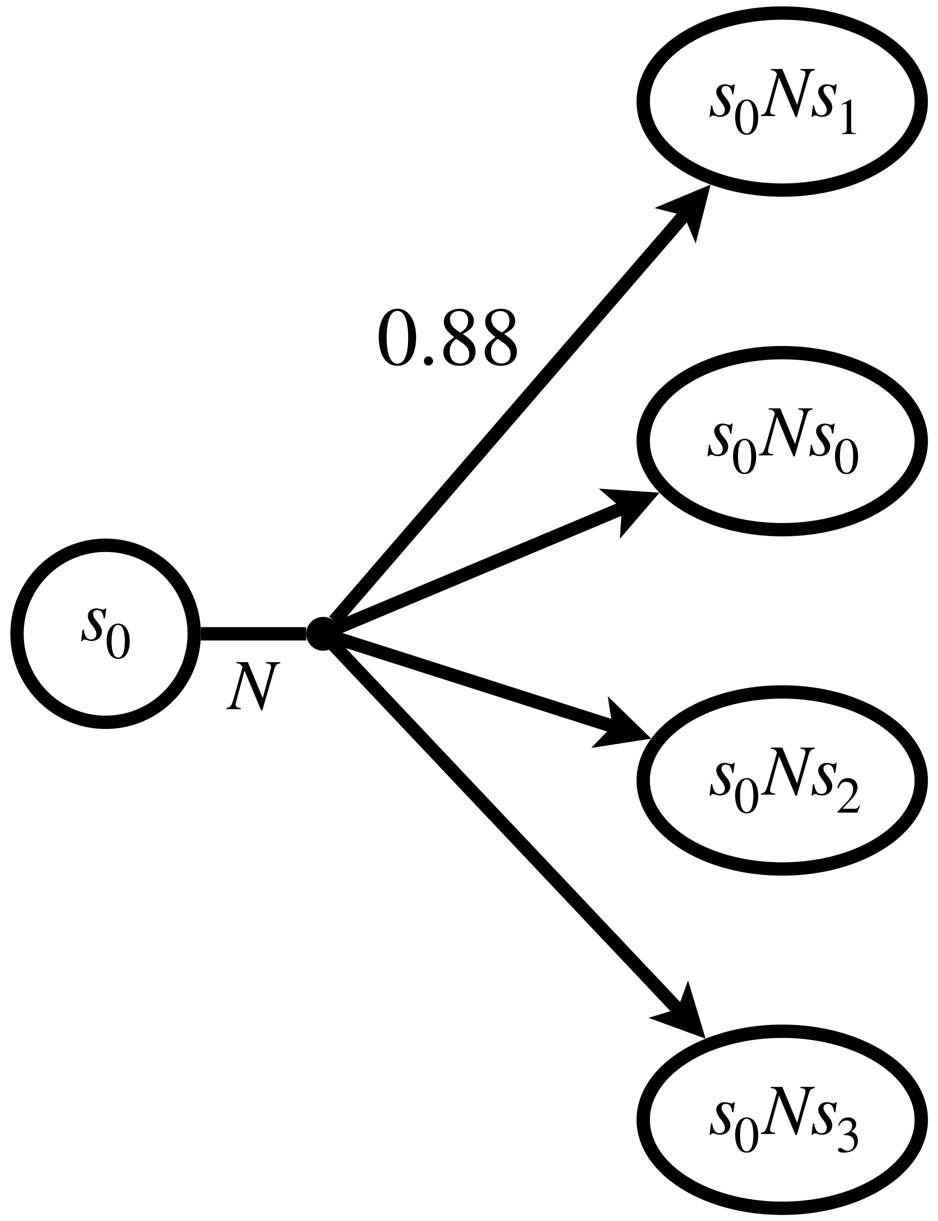
$$p(P \mid hsas') = \frac{P(s, a, s') p(P \mid h)}{\sum_{P' \in \mathscr{P}} P'(s, a, s') p(P' \mid h)}$$

$$p(P \mid s_0) = p(P) = [P_Z : 0.2, \quad P_N : 0.2, \quad P_S : 0.2, \quad P_W : 0.2, \quad P_E : 0.2]$$



$$\mathscr{P} = \{P_Z, P_N, P_S, P_W, P_E\}$$

Example



$$P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s')p(P \mid hs)$$

$$p(P \mid s_0) = p(P)$$

$$p(P \mid hsas') = \frac{P(s, a, s')p(P \mid h)}{\sum_{P' \in \mathscr{P}} P'(s, a, s')p(P' \mid h)}$$

$$p(P \mid s_0) = p(P) = [P_Z : 0.2,\ P_N : 0.2,\ P_S : 0.2,\ P_W : 0.2,\ P_E : 0.2]$$

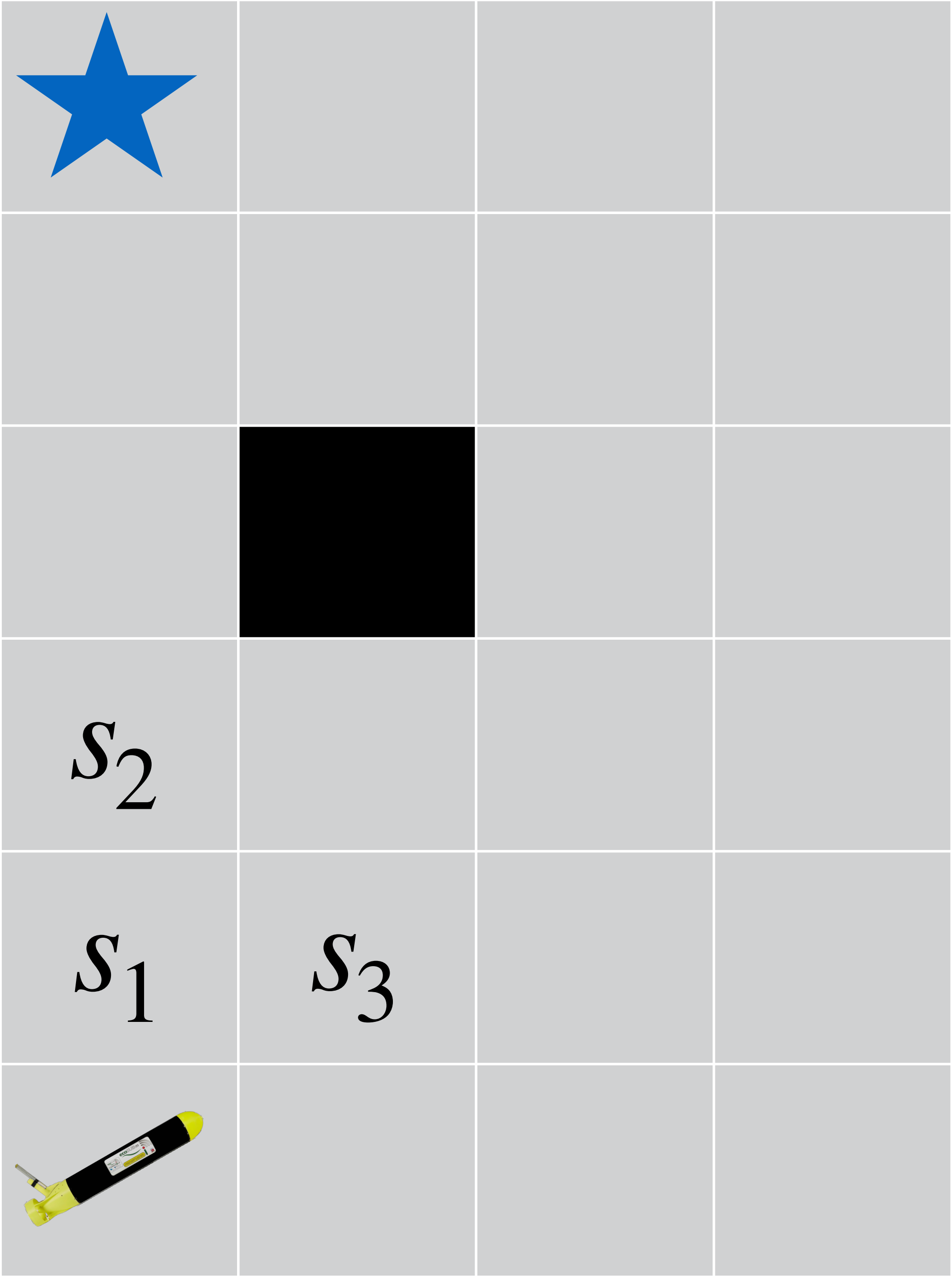
$$P^+(s_0, N, s_0Ns_1) =$$

$$\sum_{P \in \mathscr{P}} P(s_0, N, s_1)p(P \mid s_0) = 1.0 \cdot 0.2 + 0.8 \cdot 0.2 + 0.8 \cdot 0.2 + 1.0 \cdot 0.2 + 0.8 \cdot 0.2 = 0.88$$

$$p(P_N \mid s_0Ns_1) = p(P_S \mid s_0Ns_1) = p(P_E \mid s_0Ns_1) = \frac{0.8 \cdot 0.2}{0.88} \approx 0.182$$

$$p(P_Z \mid s_0Ns_1) = p(P_W \mid s_0Ns_1) = \frac{1.0 \cdot 0.2}{0.88} \approx 0.227$$

$$p(P \mid s_0Ns_1) = [P_Z : 0.227,\ P_N : 0.182,\ P_S : 0.182,\ P_W : 0.227,\ P_E : 0.182]$$



$$\mathscr{P} = \{P_Z, P_N, P_S, P_W, P_E\}$$

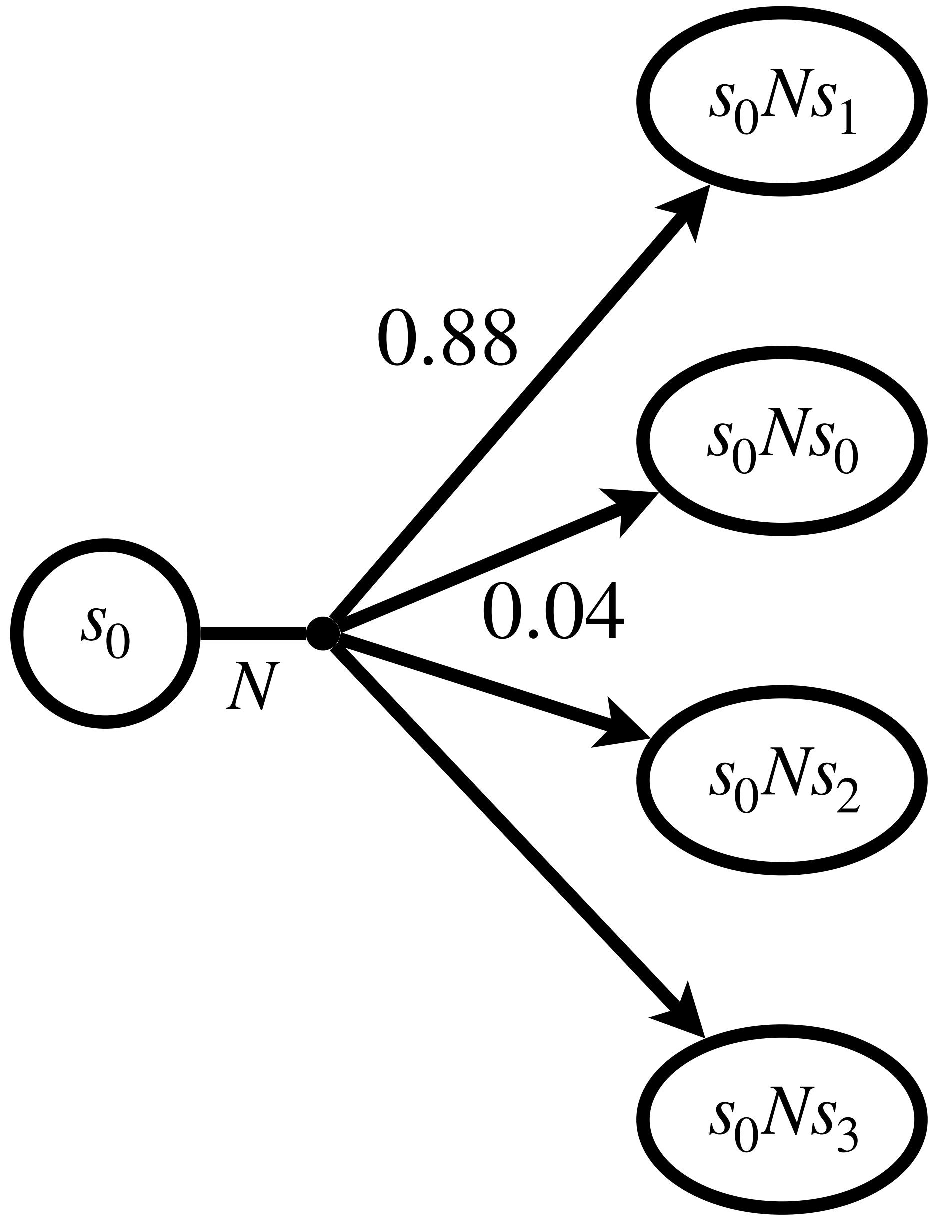
Example

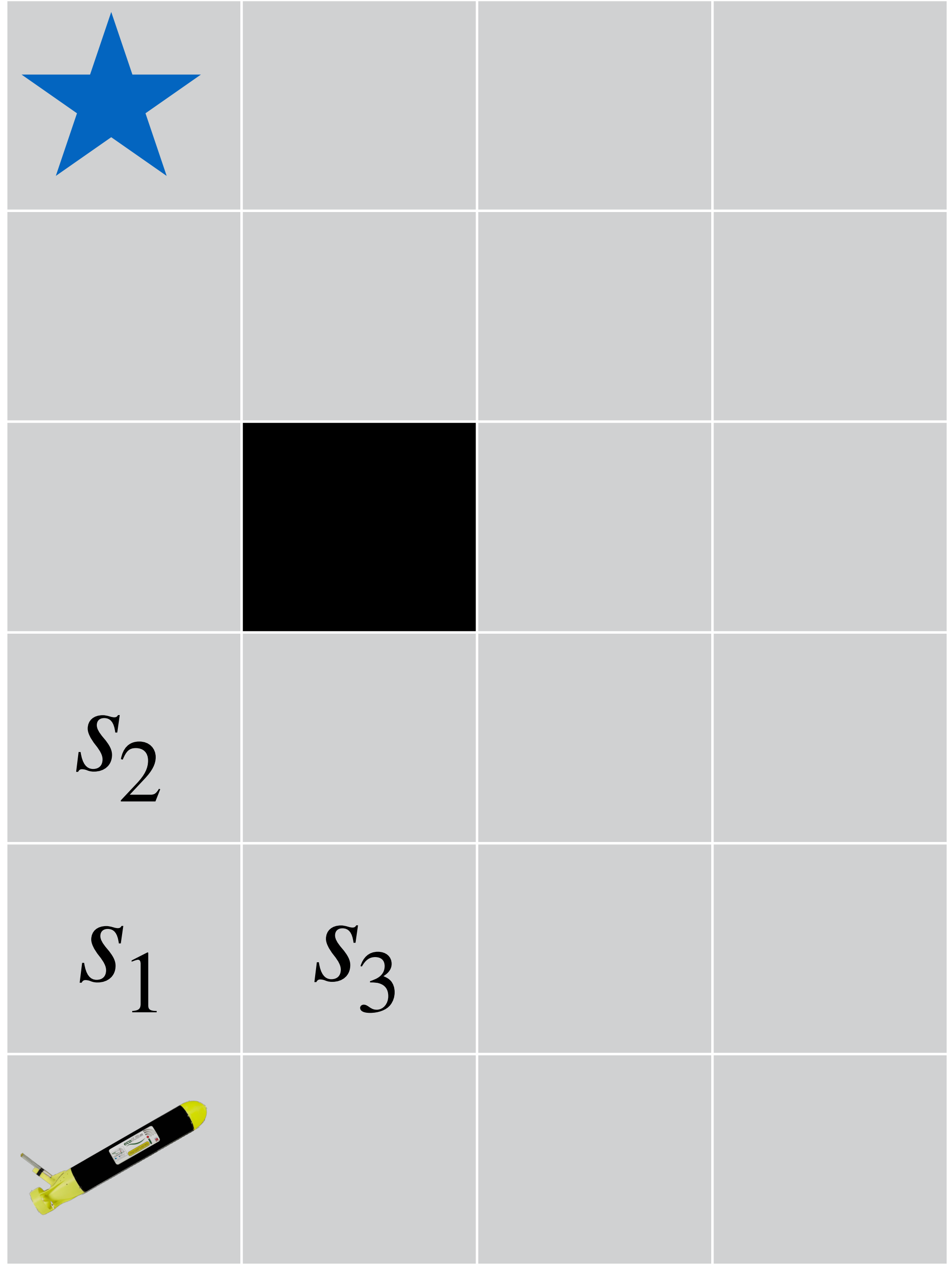
$$P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s')p(P \mid hs)$$

$$p(P \mid s_0) = p(P)$$

$$p(P \mid hsas') = \frac{P(s, a, s')p(P \mid h)}{\sum_{P' \in \mathscr{P}} P'(s, a, s')p(P' \mid h)}$$

$$p(P \mid s_0) = p(P) = [P_Z : 0.2, \; P_N : 0.2, \; P_S : 0.2, \; P_W : 0.2, \; P_E : 0.2]$$





$$P^+(s_0, N, s_0Ns_0) =$$

$$\sum_{P \in \mathscr{P}} P(s_0, N, s_0)p(P \mid s_0) = 0.0 \cdot 0.2 + 0.0 \cdot 0.2 + 0.2 \cdot 0.2 + 0.0 \cdot 0.2 + 0.0 \cdot 0.2 = 0.04$$

$$p(P_Z \mid s_0Ns_0) = p(P_N \mid s_0Ns_0) = p(P_W \mid s_0Ns_0) = p(P_E \mid s_0Ns_0) = \frac{0.0 \cdot 0.2}{0.04} = 0$$

$$p(P_S \mid s_0Ns_0) = \frac{0.2 \cdot 0.2}{0.04} = 1$$

$$p(P \mid s_0Ns_0) = [P_Z : 0, \; P_N : 0, \; P_S : 1, \; P_W : 0, \; P_E : 0]$$

$$\mathscr{P} = \{P_Z, P_N, P_S, P_W, P_E\}$$

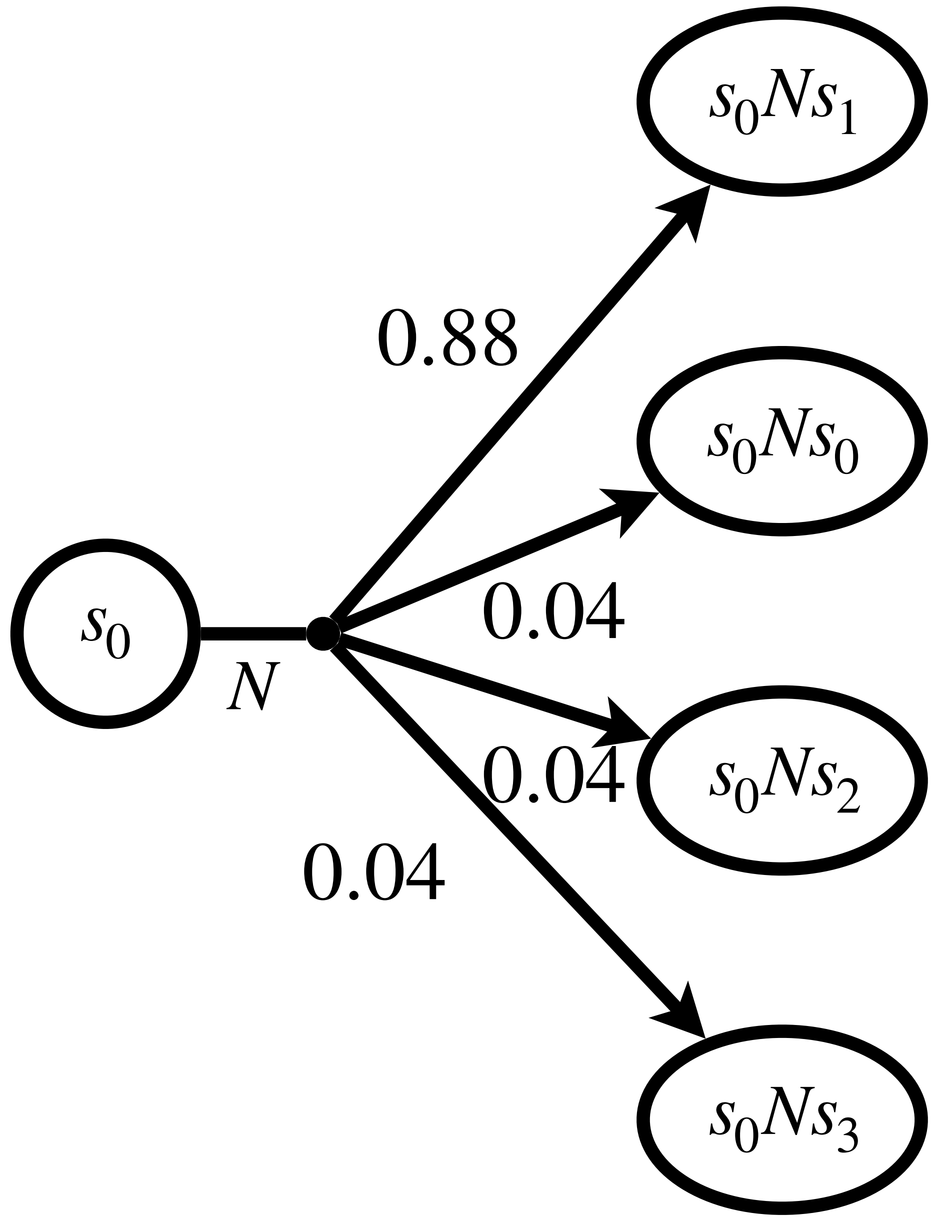
Example

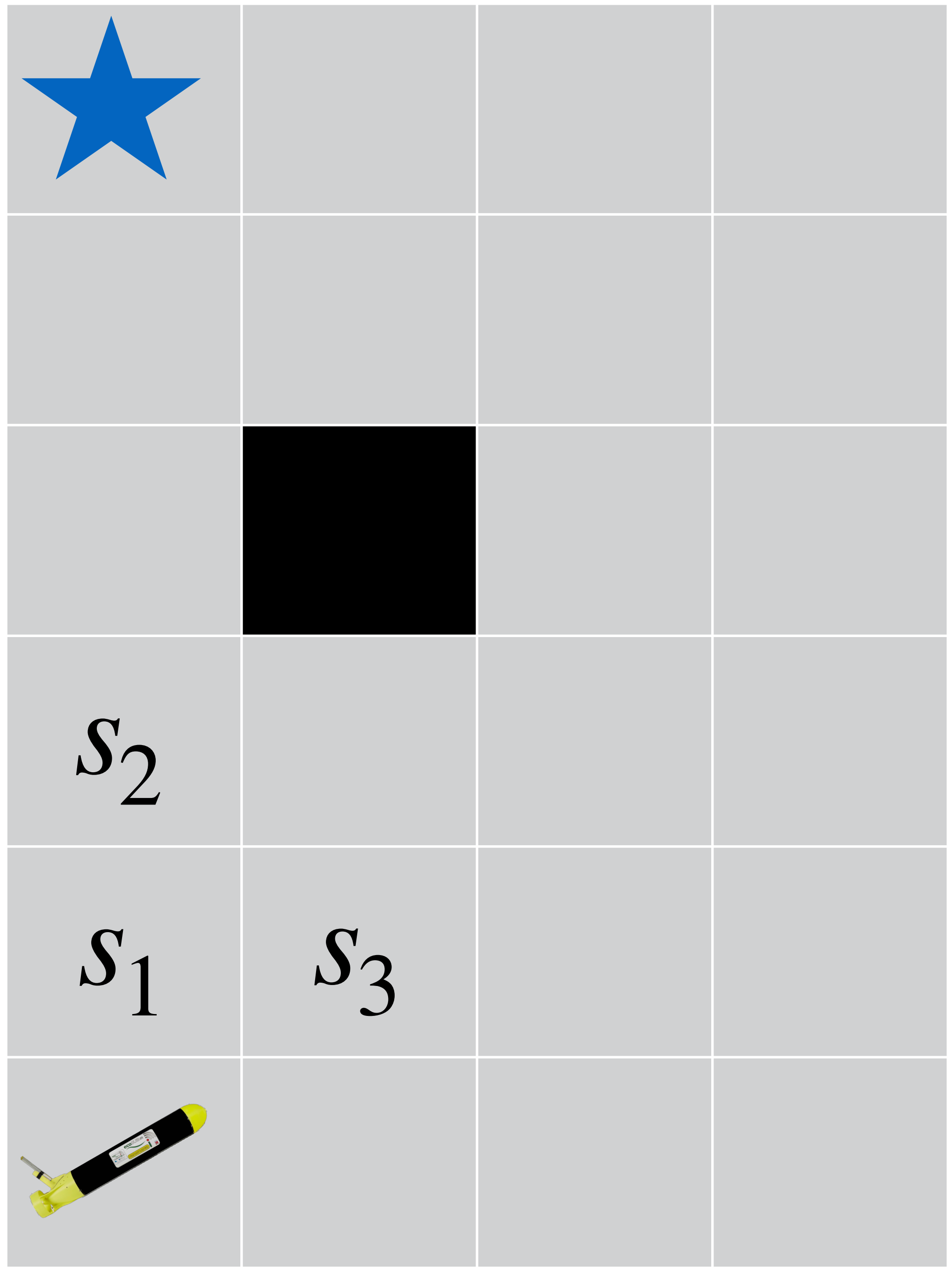
$$P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s')p(P \mid hs)$$

$$p(P \mid s_0) = p(P)$$

$$p(P \mid hsas') = \frac{P(s, a, s')p(P \mid h)}{\sum_{P' \in \mathscr{P}} P'(s, a, s')p(P' \mid h)}$$

$$p(P \mid s_0) = p(P) = [P_Z : 0.2, P_N : 0.2, P_S : 0.2, P_W : 0.2, P_E : 0.2]$$





$$P^+(s_0, N, s_0Ns_0) =$$

$$\sum_{P \in \mathscr{P}} P(s_0, N, s_0)p(P \mid s_0) = 0.0 \cdot 0.2 + 0.0 \cdot 0.2 + 0.2 \cdot 0.2 + 0.0 \cdot 0.2 + 0.0 \cdot 0.2 = 0.04$$

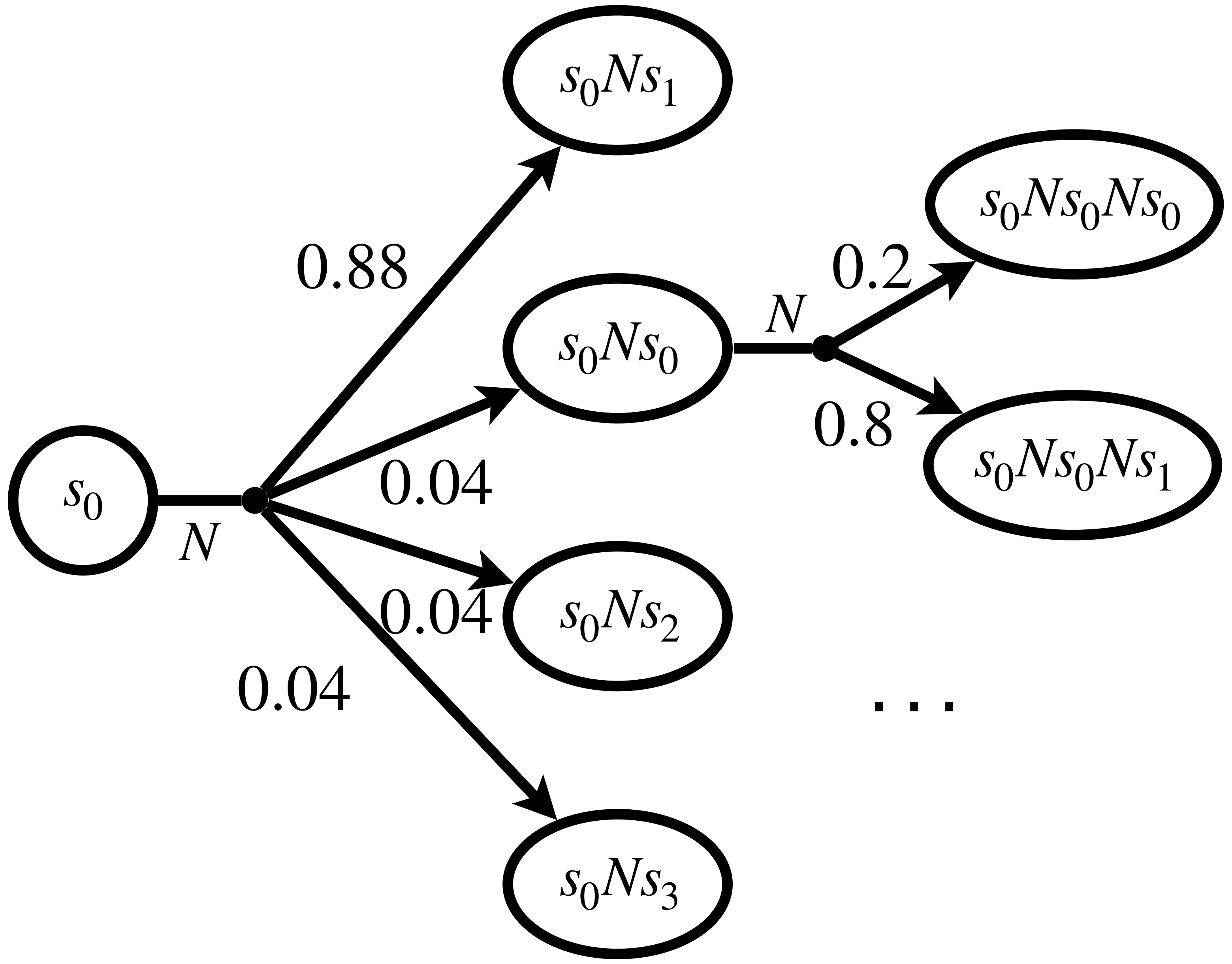
$$p(P_Z \mid s_0Ns_0) = p(P_N \mid s_0Ns_0) = p(P_W \mid s_0Ns_0) = p(P_E \mid s_0Ns_0) = \frac{0.0 \cdot 0.2}{0.04} = 0$$

$$p(P_S \mid s_0Ns_0) = \frac{0.2 \cdot 0.2}{0.04} = 1$$

$$p(P \mid s_0Ns_0) = [P_Z : 0, P_N : 0, P_S : 1, P_W : 0, P_E : 0]$$

$$\mathscr{P} = \{P_Z, P_N, P_S, P_W, P_E\}$$

Example



$$P^+(hs, a, hsas') = \sum_{P \in \mathscr{P}} P(s, a, s')p(P \mid hs)$$

$$p(P \mid s_0) = p(P)$$

$$p(P \mid hsas') = \frac{P(s, a, s')p(P \mid h)}{\sum_{P' \in \mathscr{P}} P'(s, a, s')p(P' \mid h)}$$

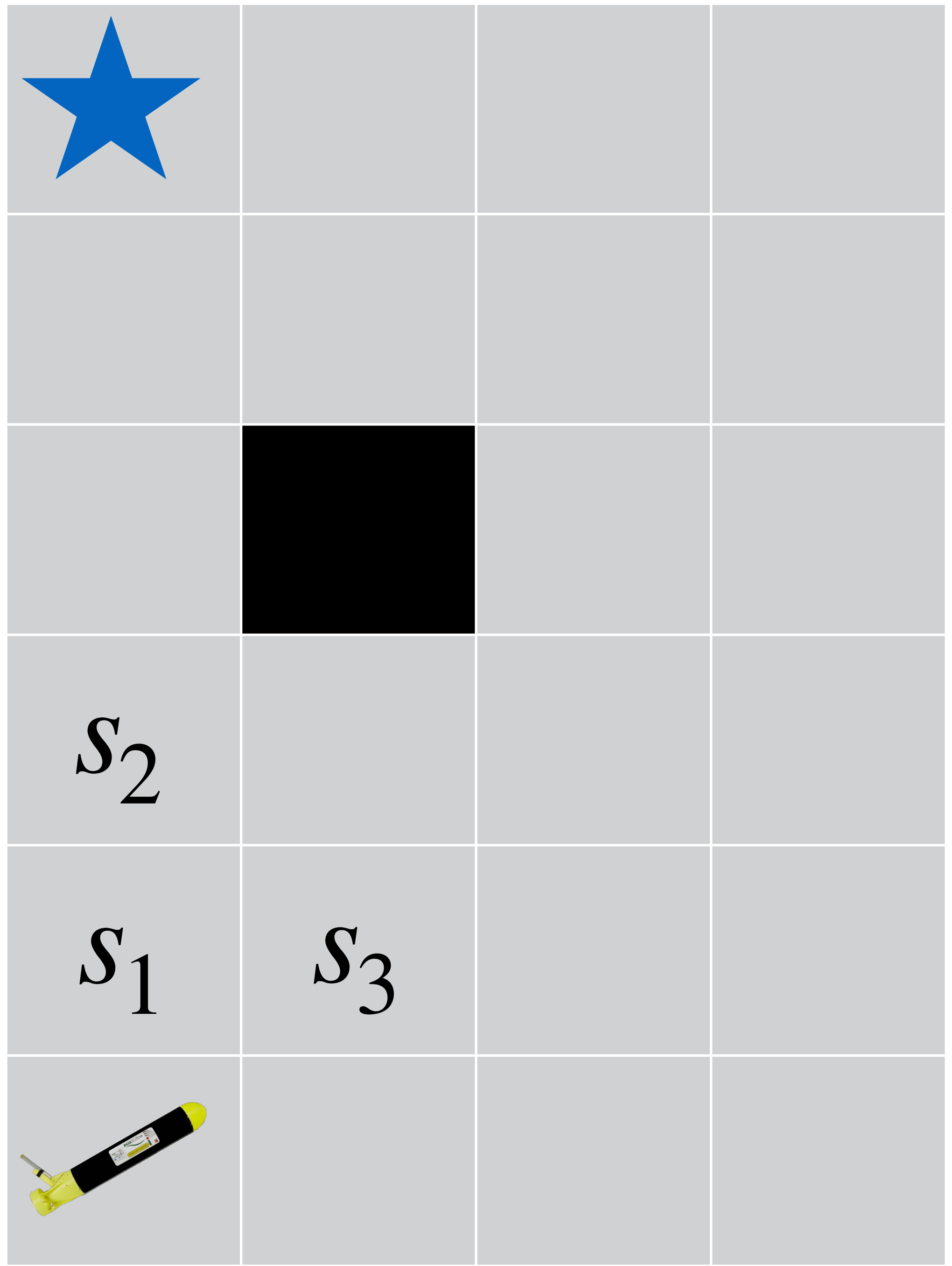
$$p(P \mid s_0Ns_0) = [P_Z : 0,\; P_N : 0,\; P_S : 1,\; P_W : 0,\; P_E : 0]$$

$$P^+(s_0Ns_0, N, s_0Ns_0Ns_1) =$$

$$\sum_{P \in \mathscr{P}} P(s_0, N, s_1)p(P \mid s_0Ns_0) = 1.0 \cdot 0.0 + 0.8 \cdot 0.0 + 0.8 \cdot 1.0 + 1.0 \cdot 0.0 + 0.8 \cdot 0.0 = 0.8$$

$$P^+(s_0Ns_0, N, s_0Ns_0Ns_0) =$$

$$\sum_{P \in \mathscr{P}} P(s_0, N, s_0)p(P \mid s_0Ns_0) = 1.0 \cdot 0.0 + 0.0 \cdot 0.0 + 0.2 \cdot 1.0 + 0.0 \cdot 0.0 + 0.0 \cdot 0.0 = 0.2$$



$$\mathscr{P} = \{P_Z, P_N, P_S, P_W, P_E\}$$

Bayes-adaptive MDP

- Optimally solves the exploration (improving belief over model) and exploitation (use current belief to achieve the goal) problem

- Possible models + prior can be viewed as a partially observable MDP (POMDP)
  - Agent state fully observable
  - Latent feature is the model we are executing in
  - Observation set is the set of agent states
  - The BAMDP is the belief MDP of this POMDP
  - If the environment is dynamic then we need to model the problem as a POMDP (specifically a mixed-observability MDP)

- BAMDP state-space is infinite
  - One can use adaptations of POMDP techniques
  - We will look into one such technique, based on Monte-Carlo Tree Search, named Bayes-adaptive Monte Carlo Planning (BAMCP)

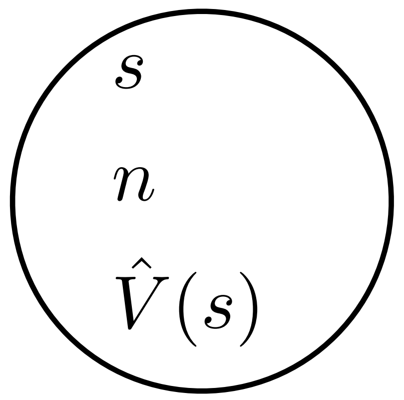
Monte-Carlo Tree Search

MCTS

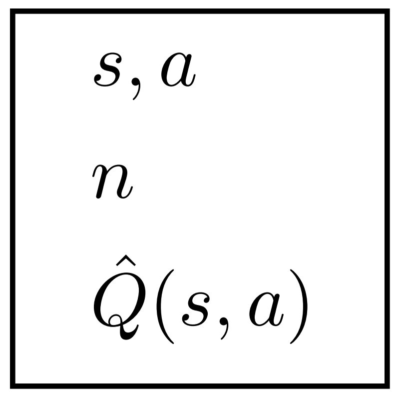
- In many cases it is expensive or difficult to enumerate states, or there is no access to an explicit transition function, but can simulate the transitions between states
    - Use a Monte-Carlo (i.e sampling-based) approach to approximate the value function
- Monte-Carlo Tree Search (MCTS) is a trial-based tree search algorithm that has been extremely successful approximating solutions (e.g. AlphaGo)
    - Allows for online (interleaving planning and execution) or offline planning
    - Under certain configurations, provides PAC guarantees - “with probability 0.95 the solution from x trials is within 5% of optimal”
    - In the limit (i.e. given infinite samples), produces the optimal value function, but can also function as an anytime algorithm

MCTS

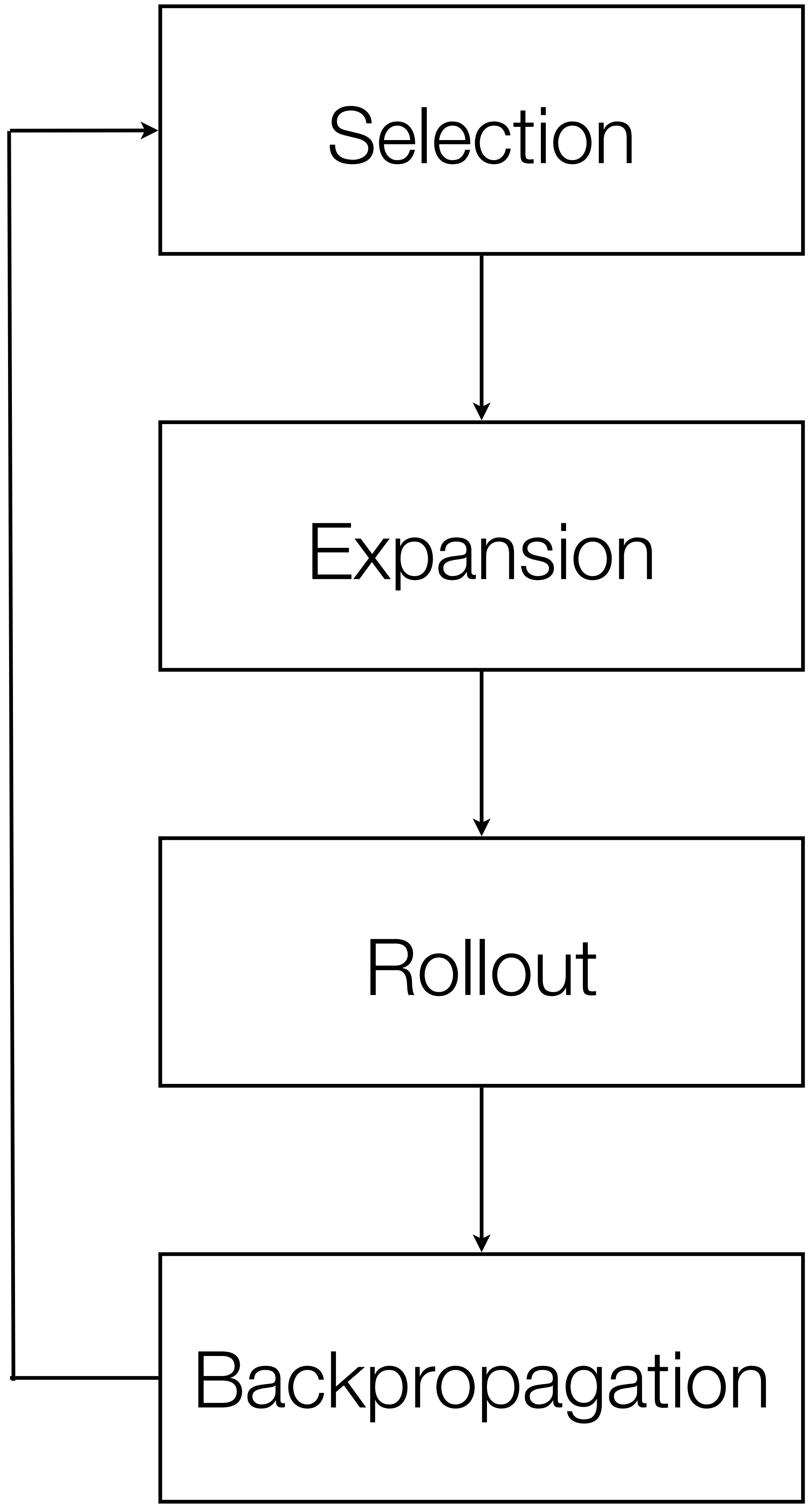
- We introduce MCTS for MDPs
- Two types of search nodes
  - Decision nodes - correspond to states and are use to keep estimate of $V(s)$

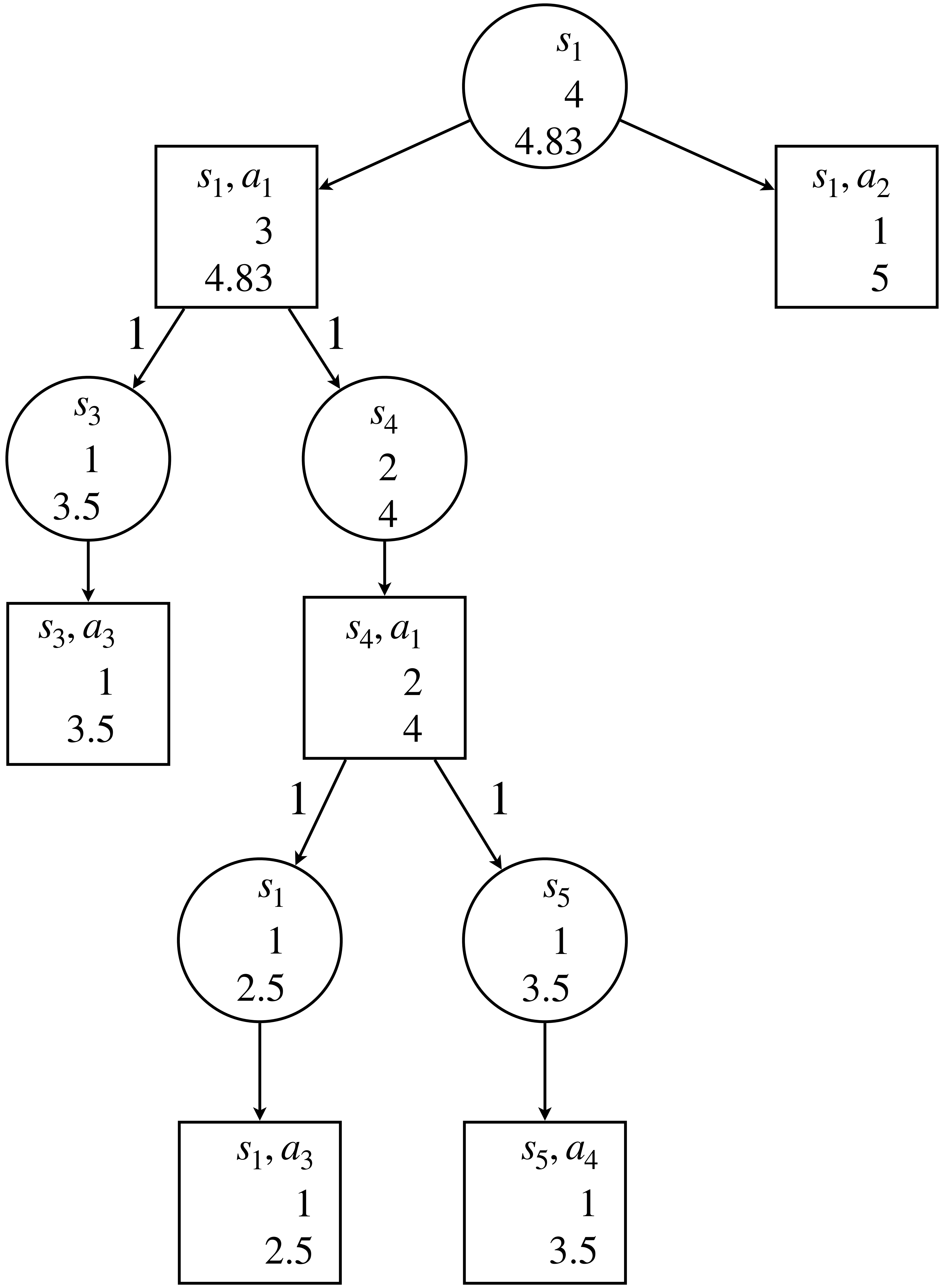


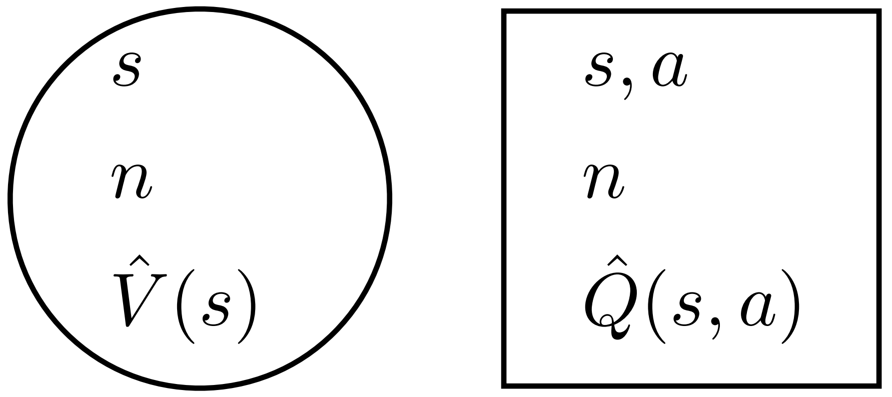
  - Chance nodes - correspond to state-action pairs and are used to keep estimate of $Q(s, a)$



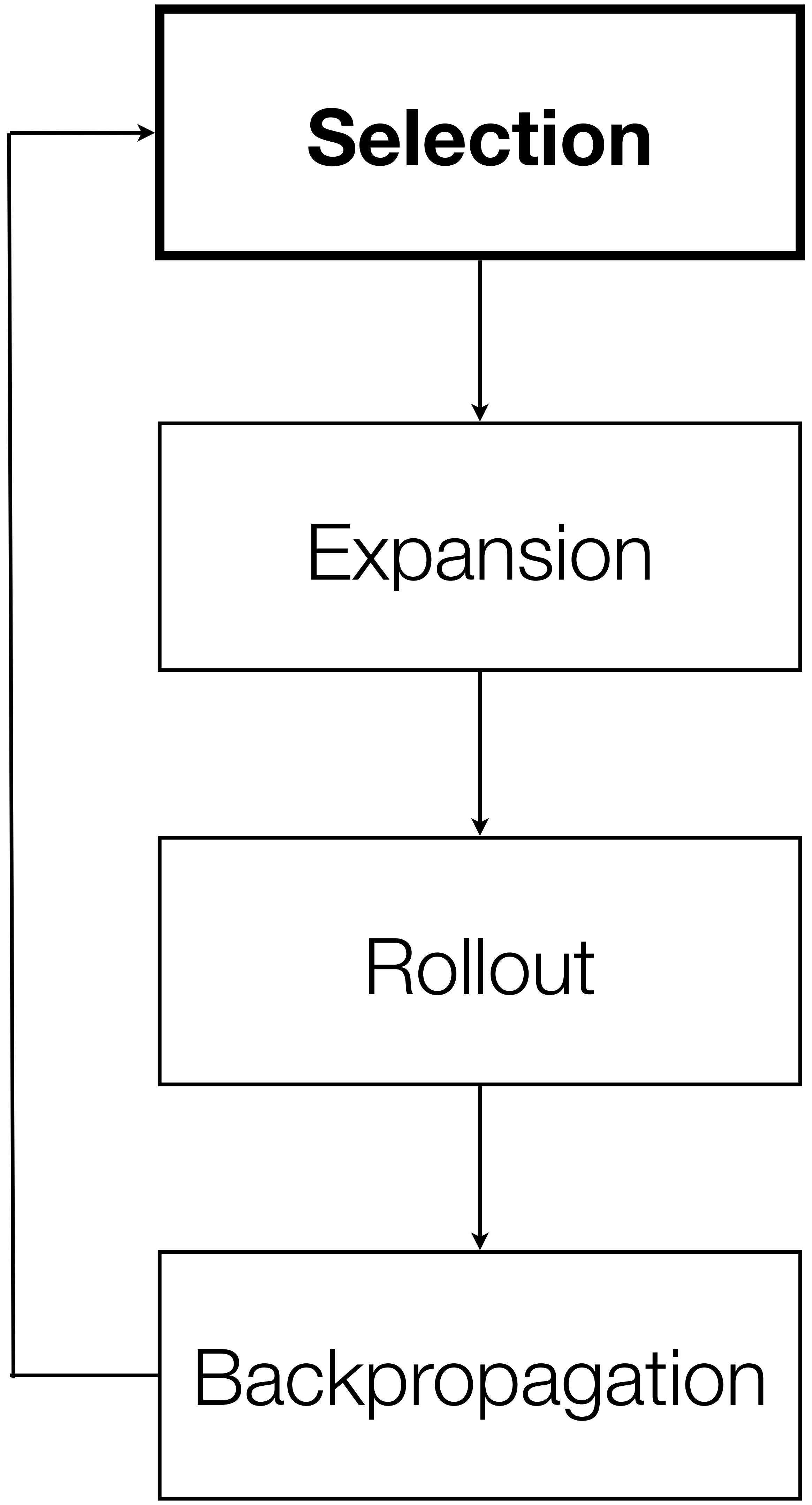
MCTS

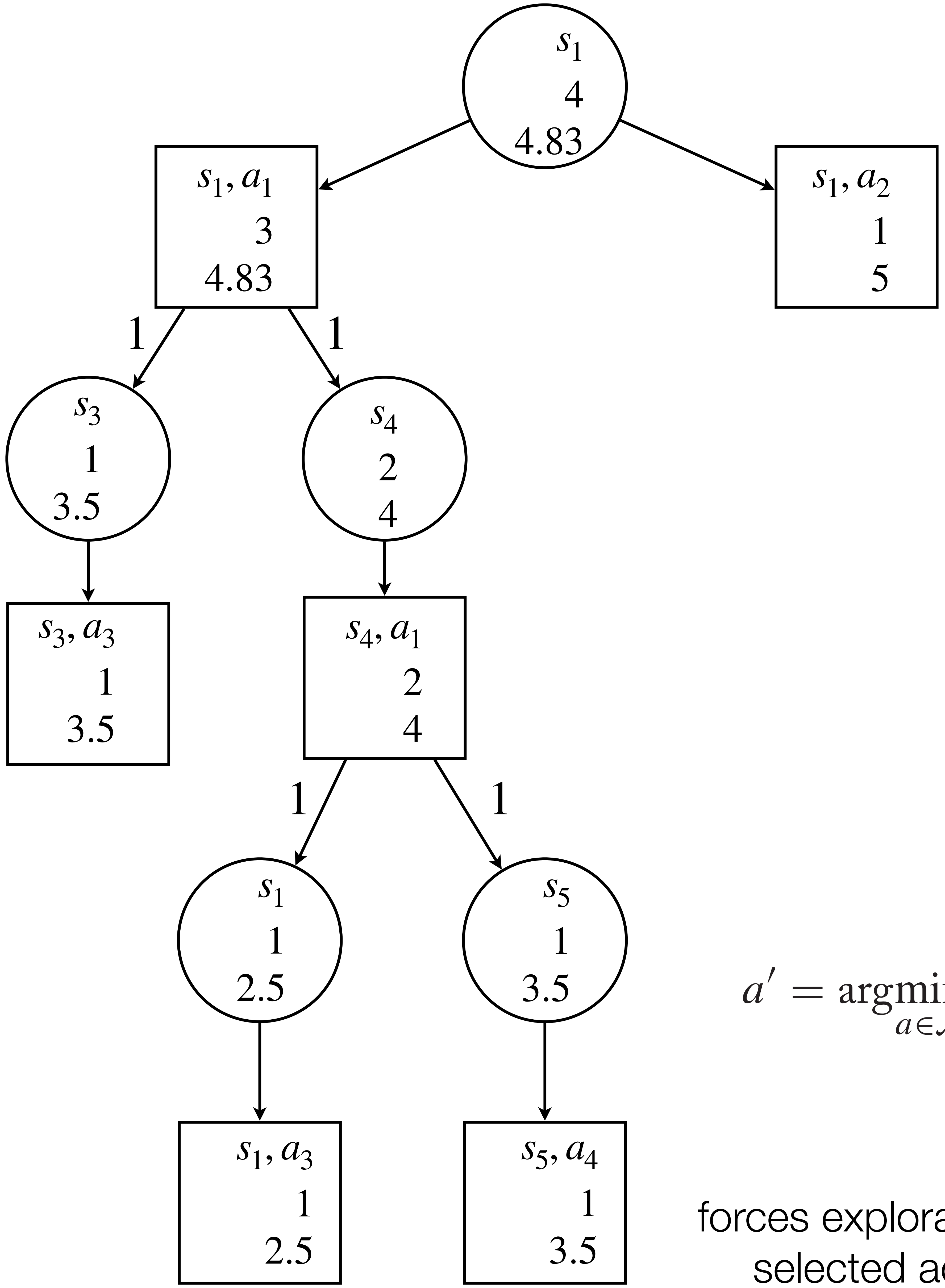

Selection
Expansion
Rollout
Backpropagation


s_1
4
4.83
s_1, a_1
3
4.83
s_1, a_2
1
5
1
1
s_3
1
3.5
s_4
2
4
s_3, a_3
1
3.5
s_4, a_1
2
4
1
1
s_1
1
2.5
s_5
1
3.5
s_1, a_3
1
2.5
s_5, a_4
1
3.5


s
n
$\hat{V}(s)$
s, a
n
$\hat{Q}(s, a)$

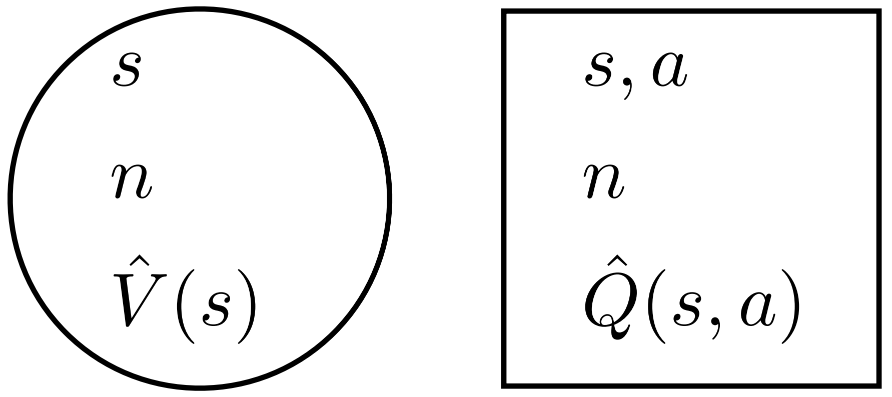
MCTS





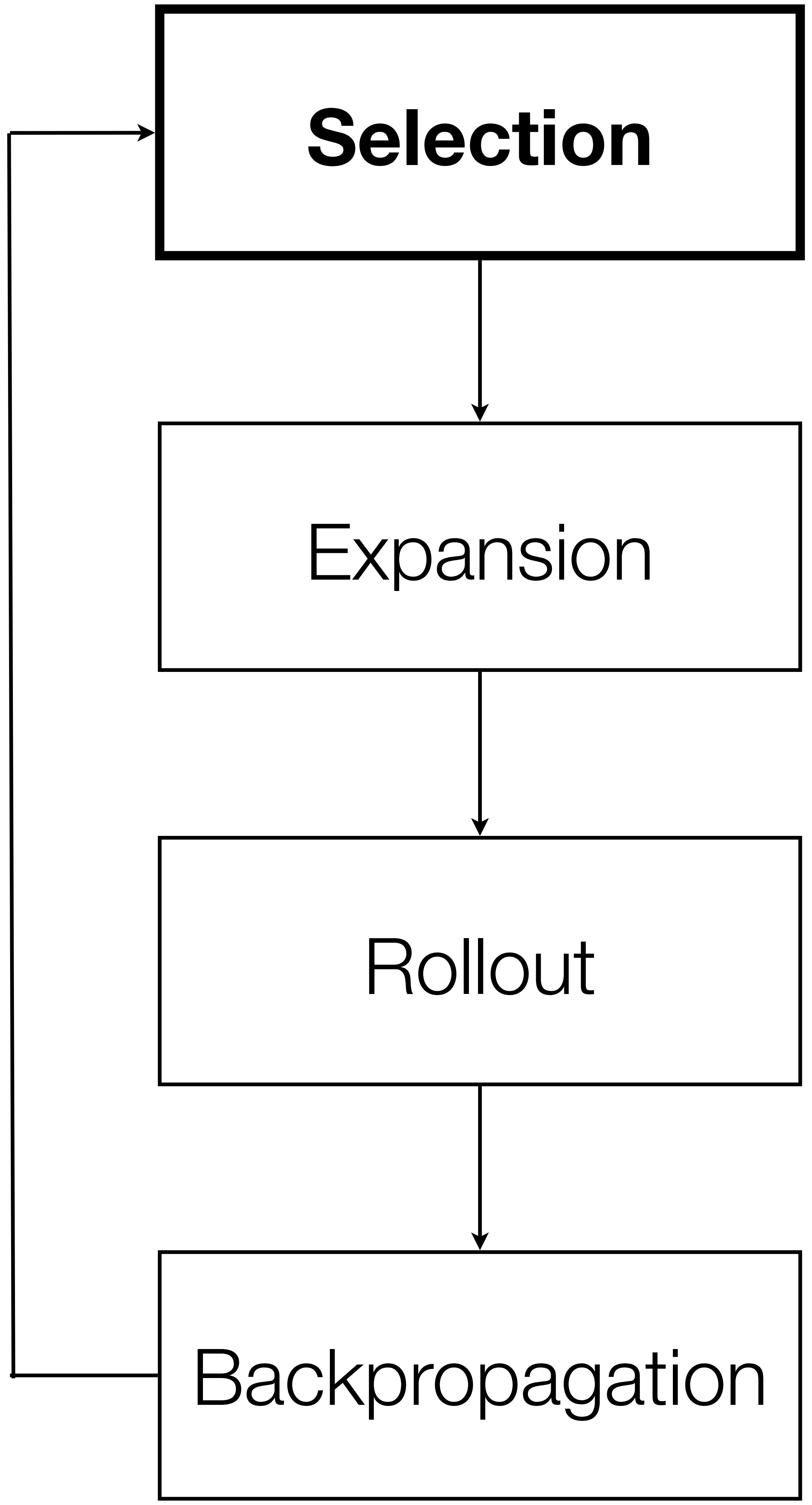
$$a' = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left\{ \hat{Q}(s, a) - C \sqrt{\frac{\ln(n_s)}{n_{s,a}}} \right\}$$

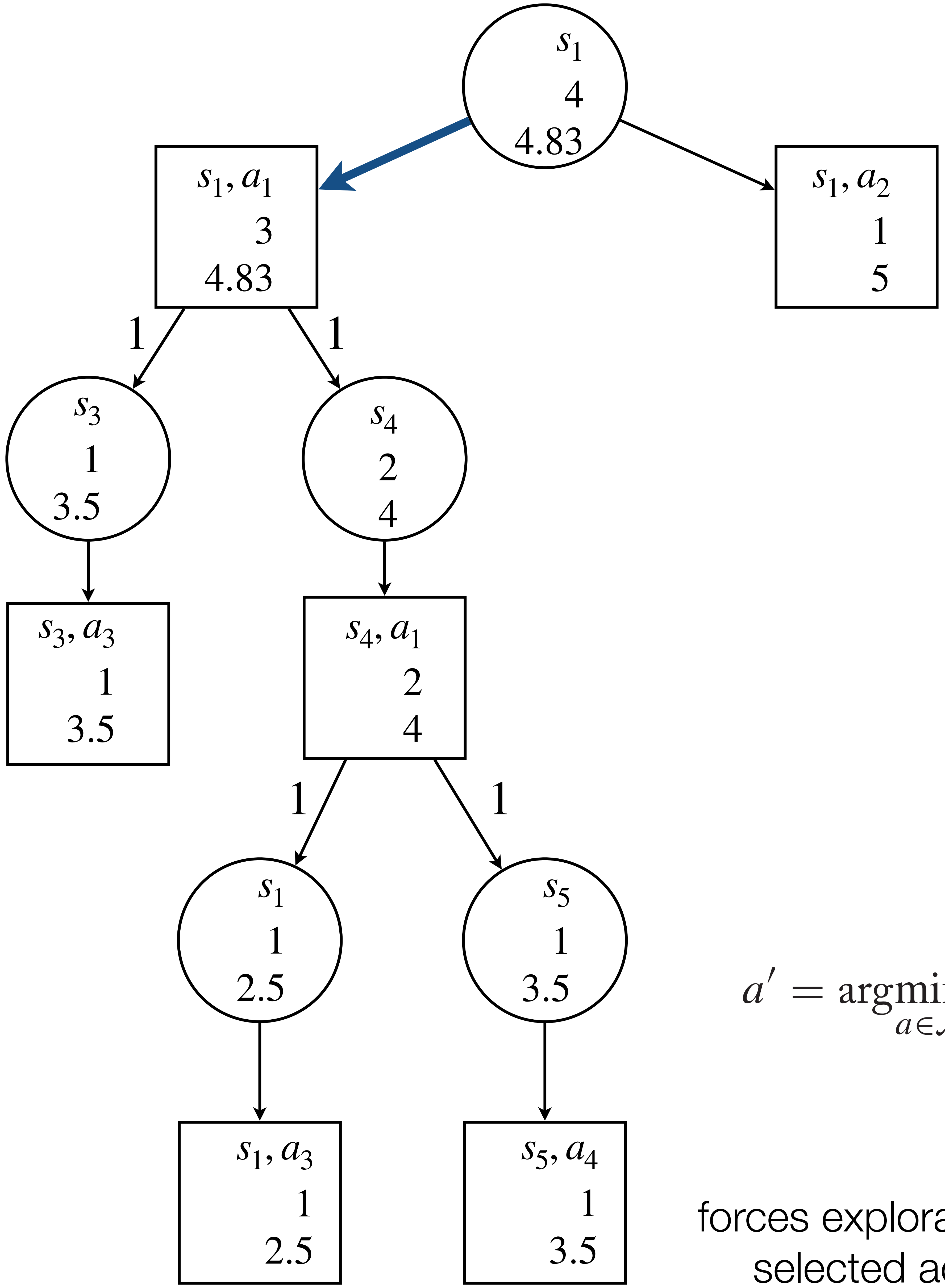
forces exploration by making under-selected actions look cheaper



Upper confidence bound applied to trees (UCT)

MCTS

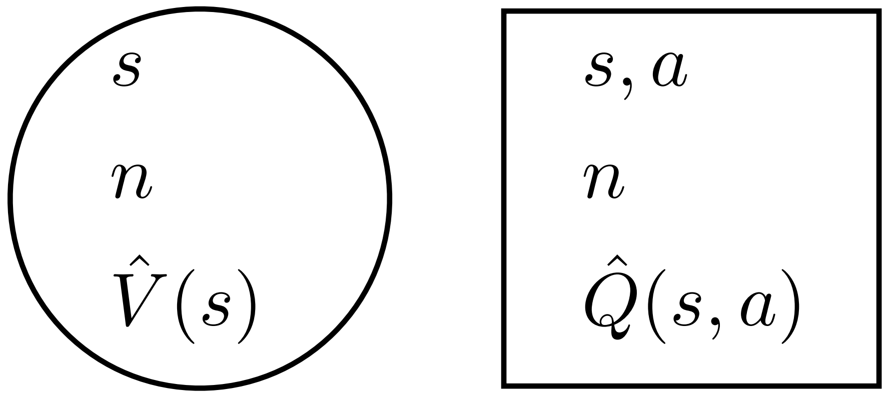




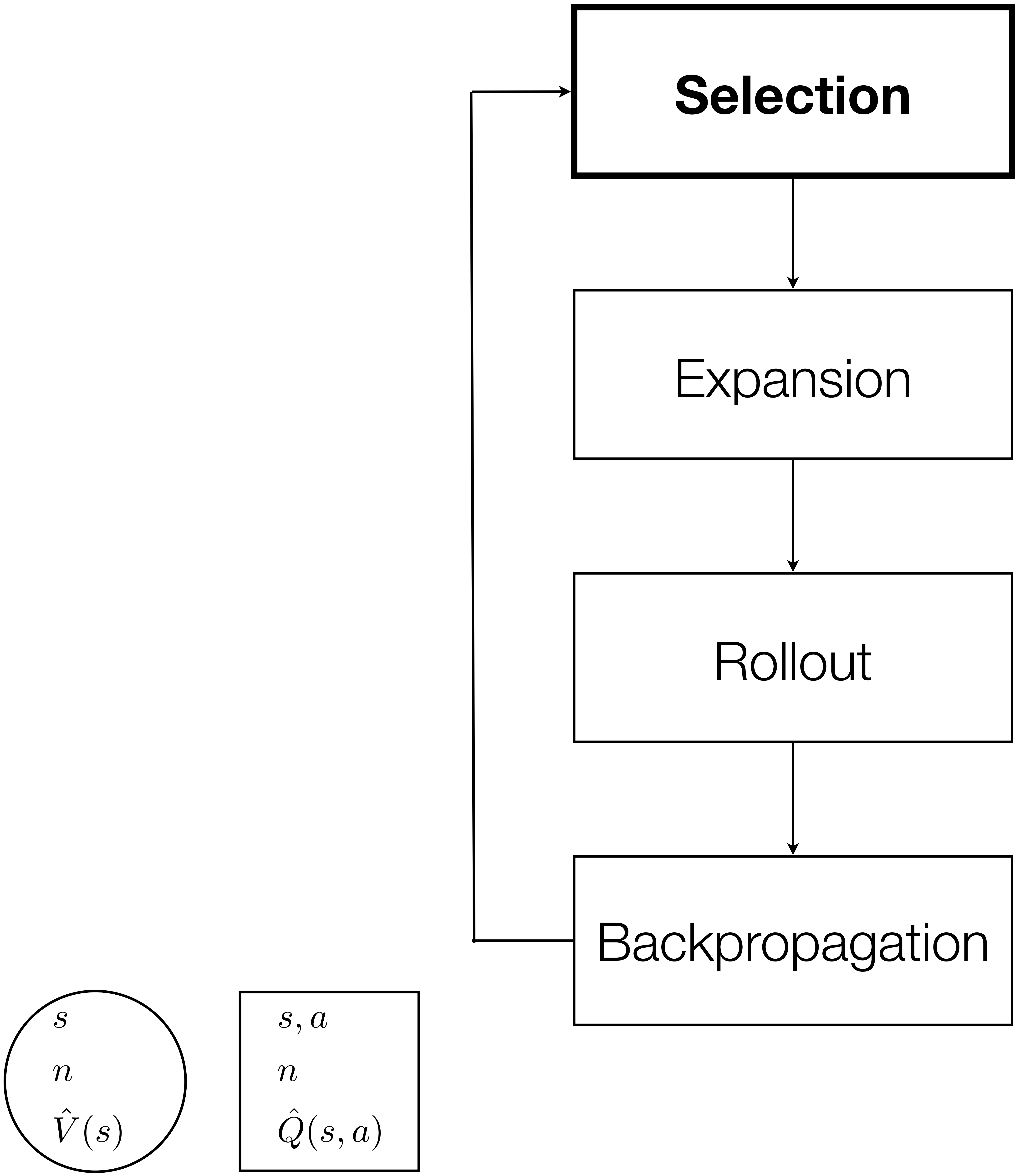
$$a' = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left\{ \hat{Q}(s, a) - C\sqrt{\frac{\ln(n_s)}{n_{s,a}}} \right\}$$

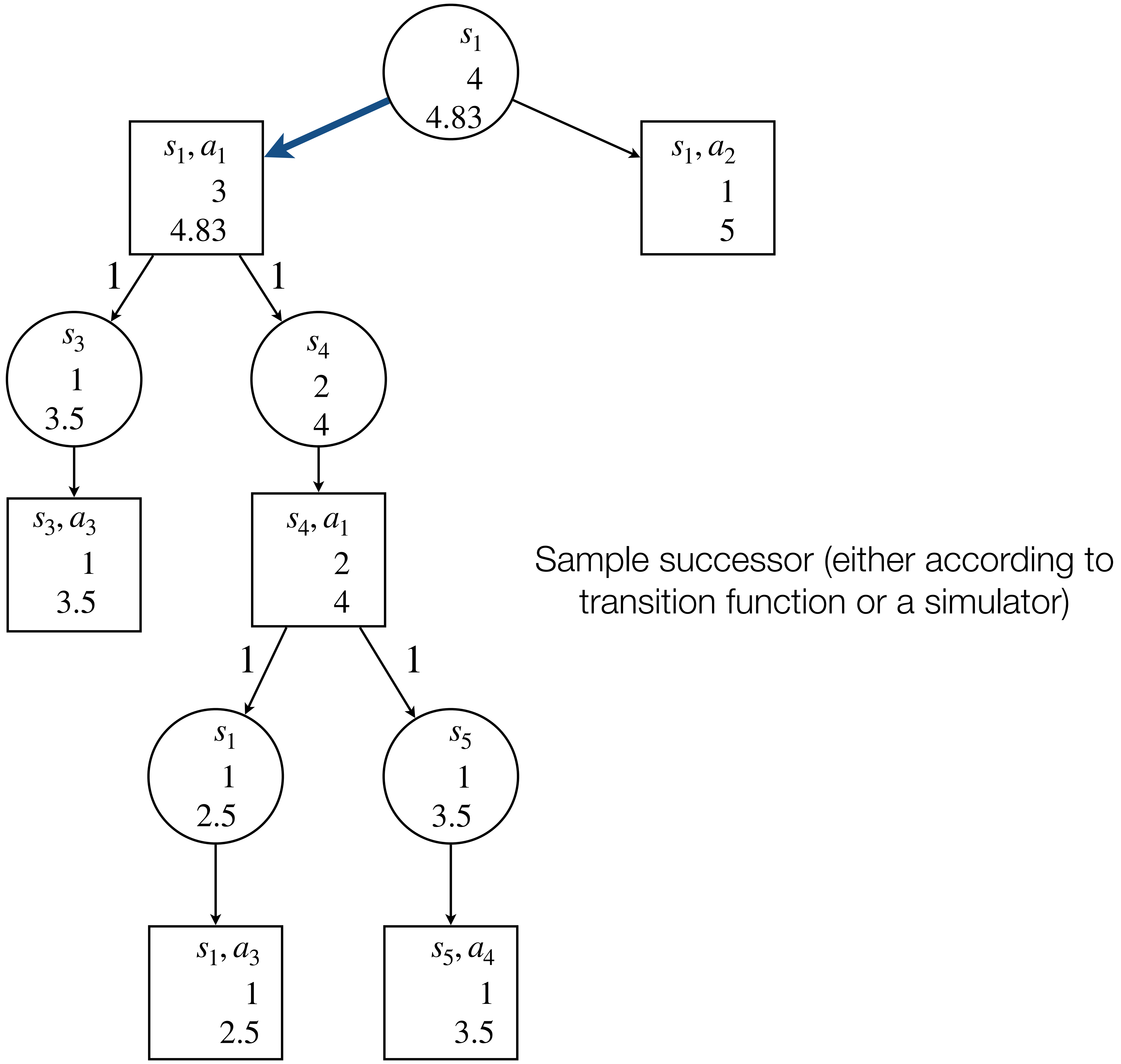
forces exploration by making under-selected actions look cheaper

Upper confidence bound applied to trees (UCT)

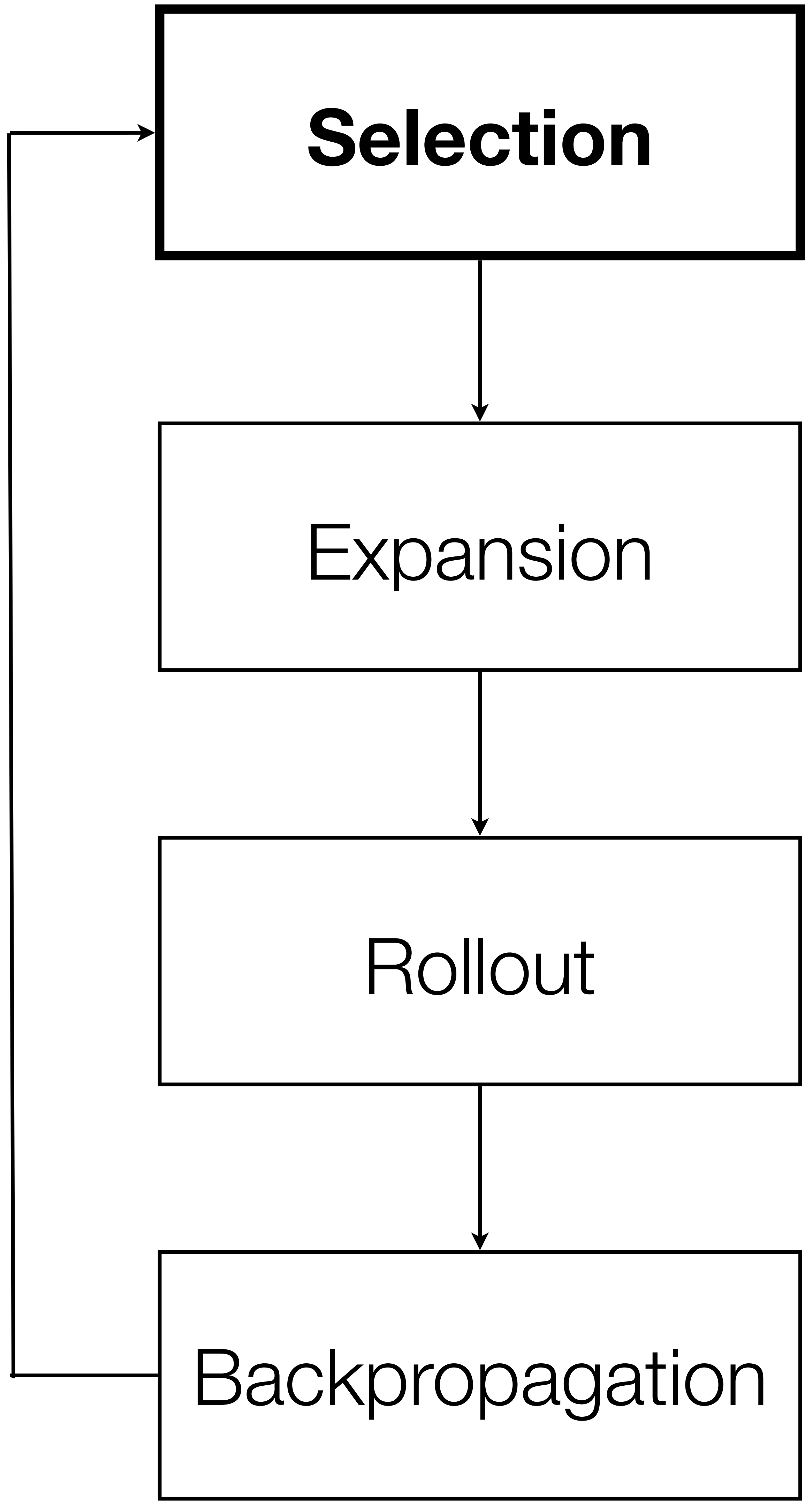


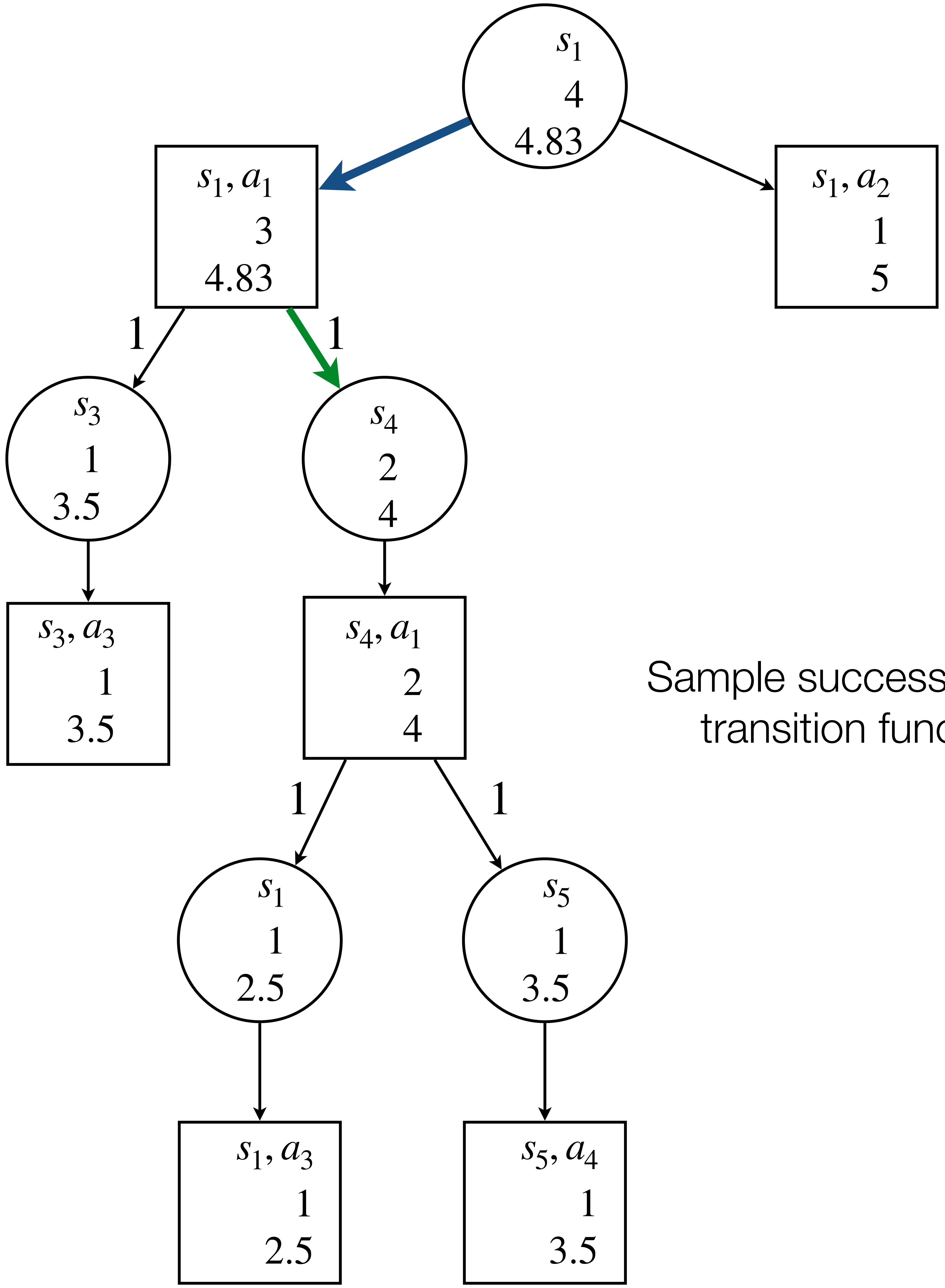
MCTS


Selection
Expansion
Rollout
Backpropagation
s
n
$\hat{V}(s)$
s, a
n
$\hat{Q}(s, a)$

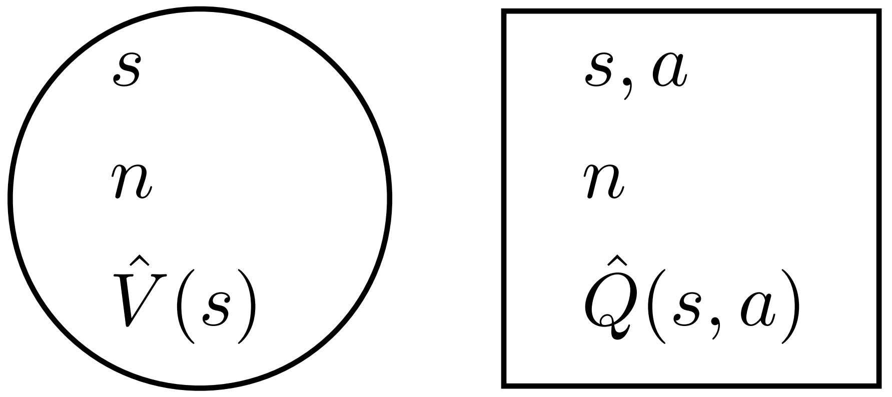

s_1 4 4.83
s_1, a_1 3 4.83
s_1, a_2 1 5
1
1
s_3 1 3.5
s_4 2 4
s_3, a_3 1 3.5
s_4, a_1 2 4
1
1
s_1 1 2.5
s_5 1 3.5
s_1, a_3 1 2.5
s_5, a_4 1 3.5
Sample successor (either according to transition function or a simulator)

MCTS

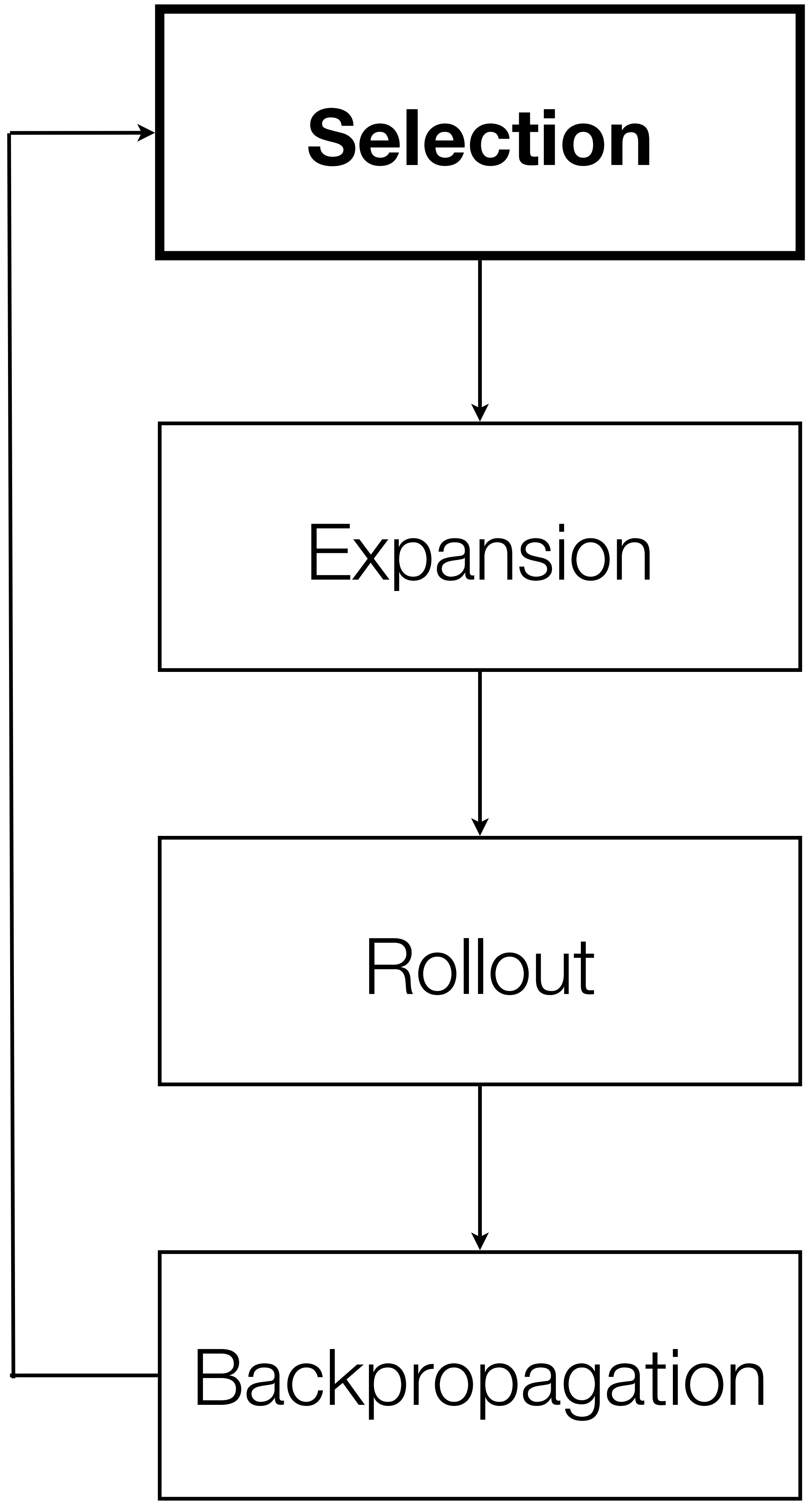


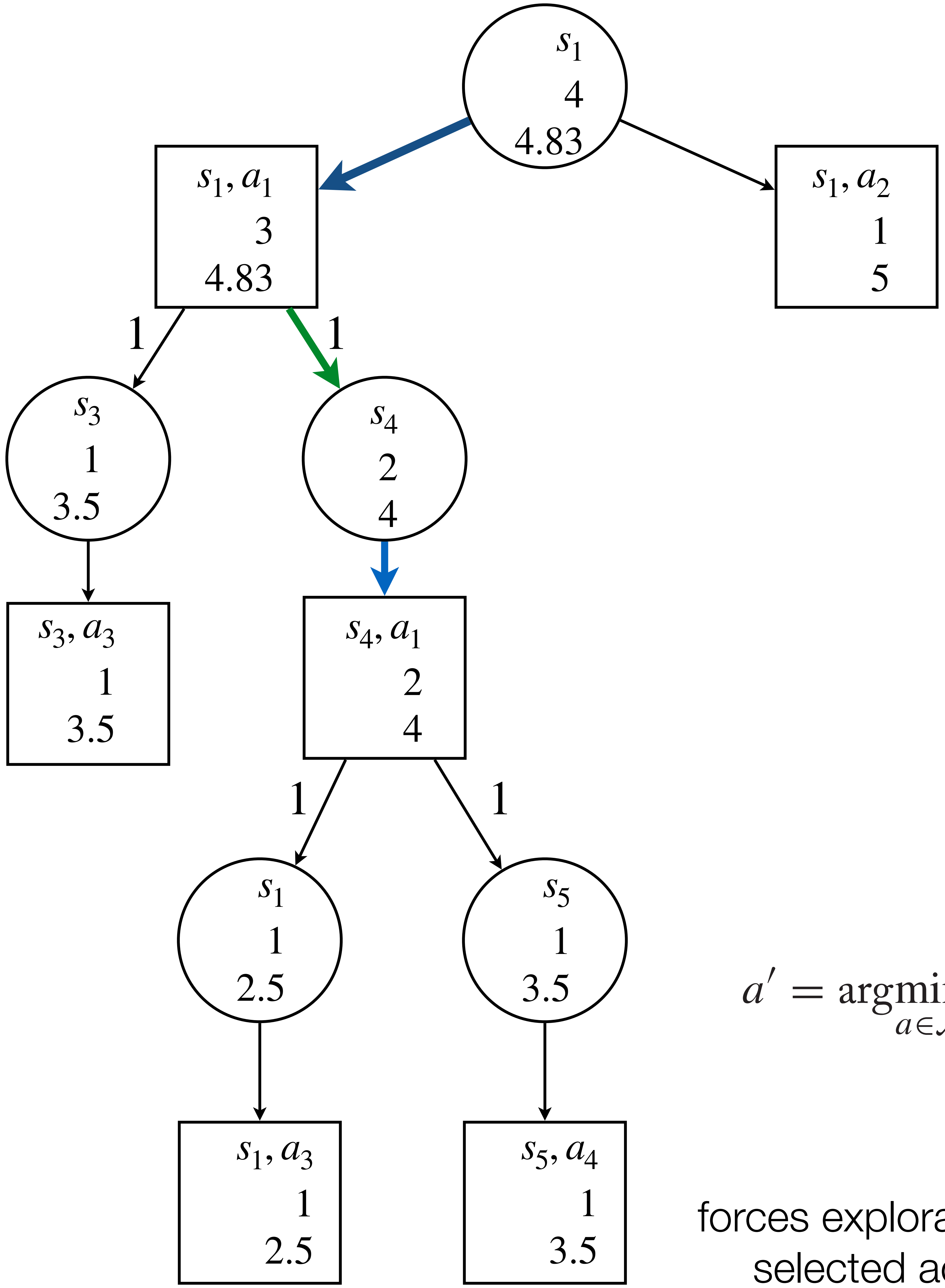


Sample successor (either according to transition function or a simulator)



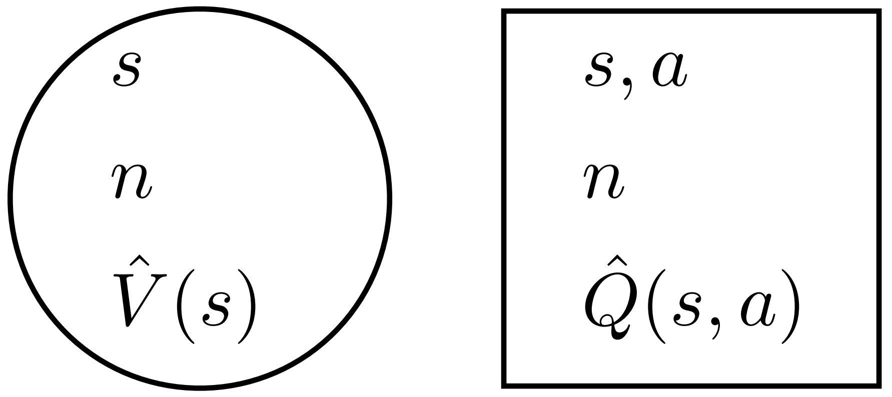
MCTS





$$a' = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left\{ \hat{Q}(s, a) - C\sqrt{\frac{\ln(n_s)}{n_{s,a}}} \right\}$$

forces exploration by making under-selected actions look cheaper

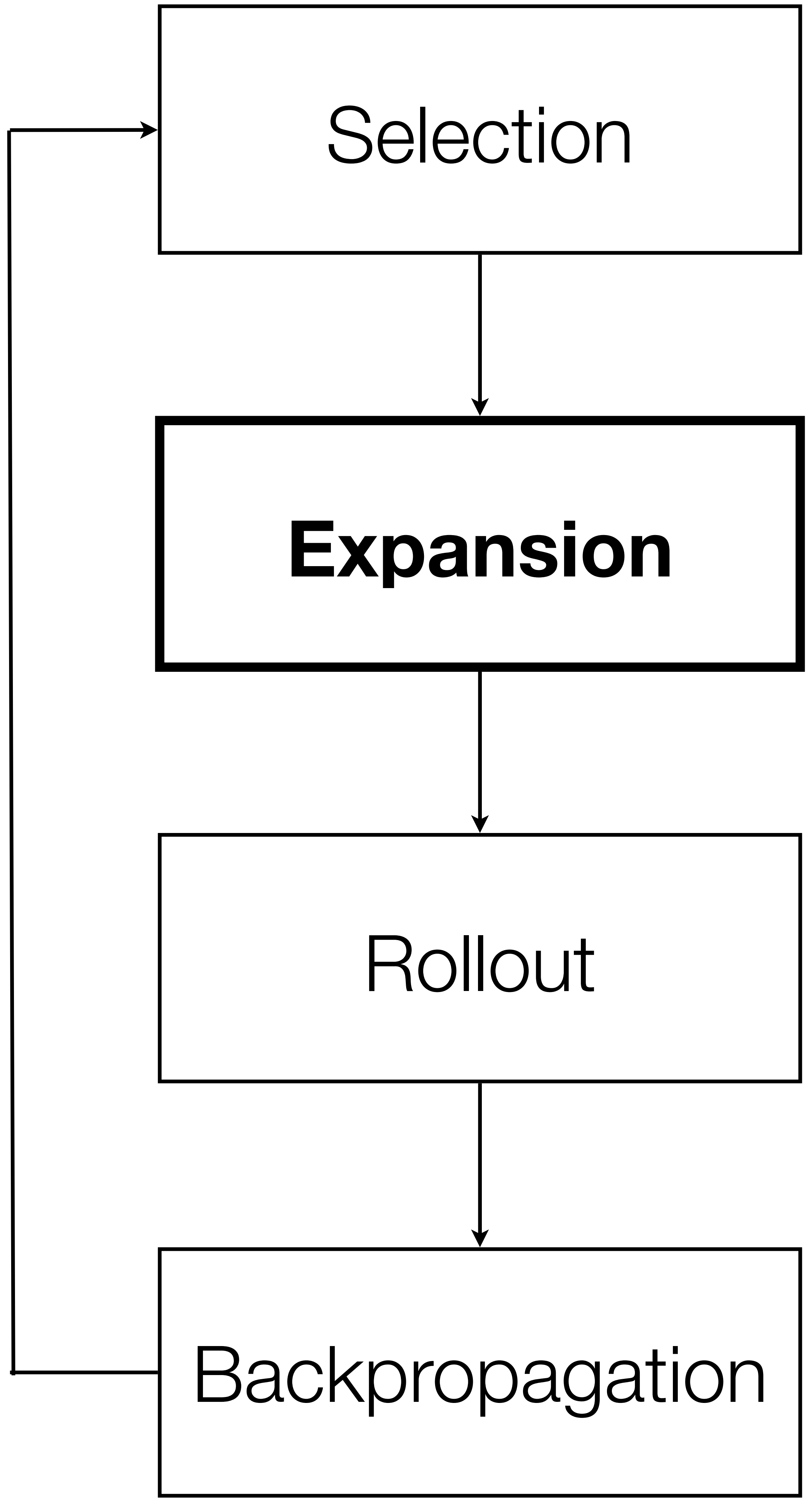


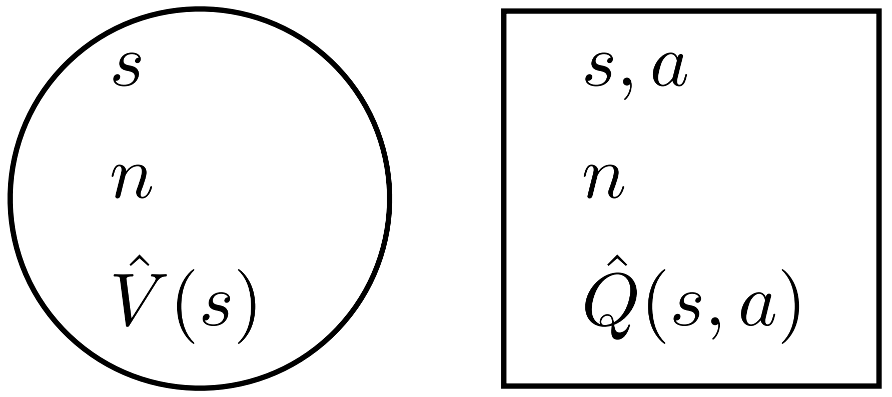
Upper confidence bound applied to trees (UCT)

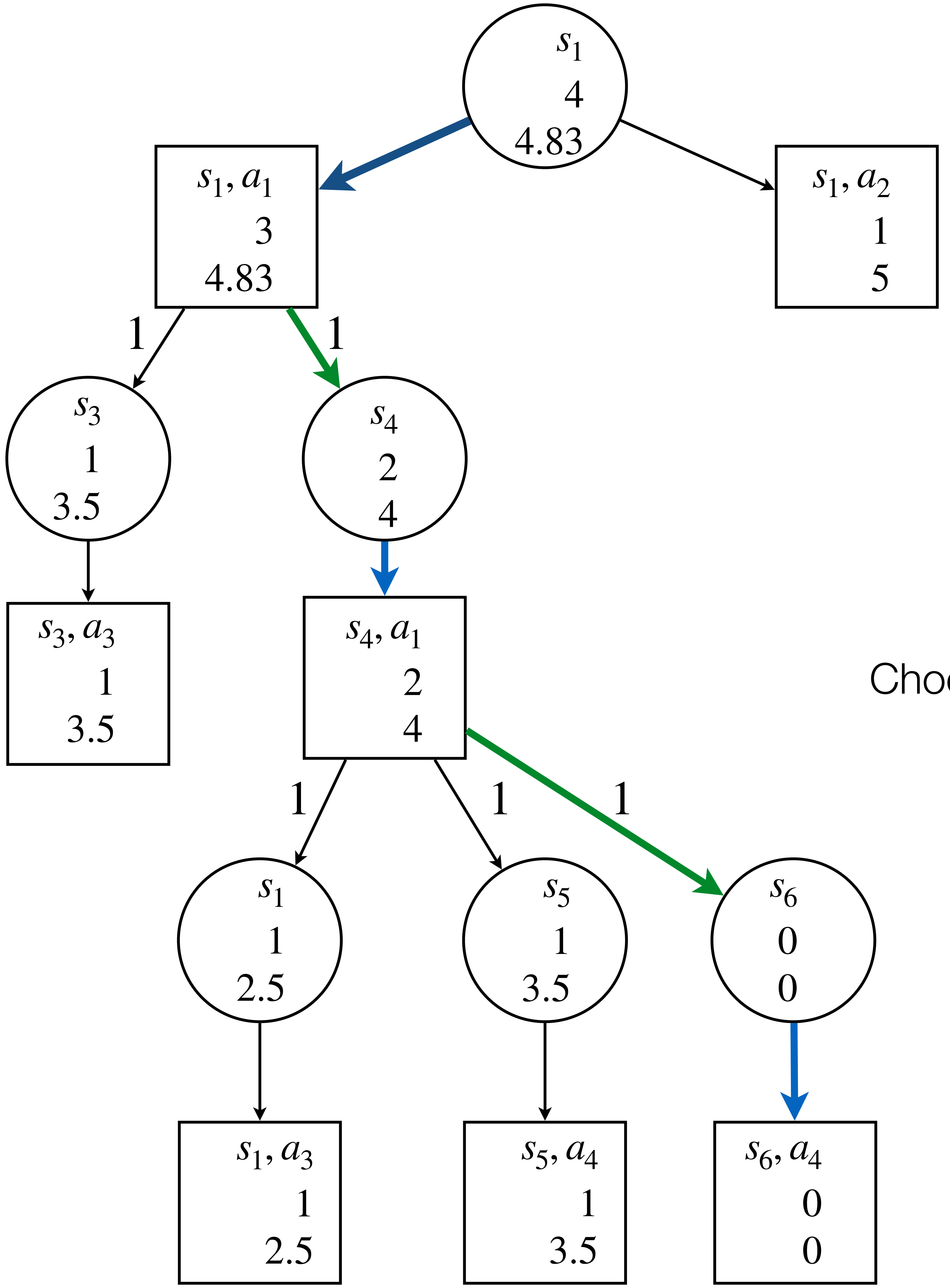
MCTS


Selection
Expansion
Rollout
Backpropagation
s_1 4 4.83
s_1, a_1 3 4.83
s_1, a_2 1 5
s_3 1 3.5
s_4 2 4
s_3, a_3 1 3.5
s_4, a_1 2 4
Sample successor (either according to transition function or a simulator)
s_1 1 2.5
s_5 1 3.5
s_6 0 0
s_1, a_3 1 2.5
s_5, a_4 1 3.5
1
1
1
1
1
s n $\hat{V}(s)$
s, a n $\hat{Q}(s, a)$

MCTS

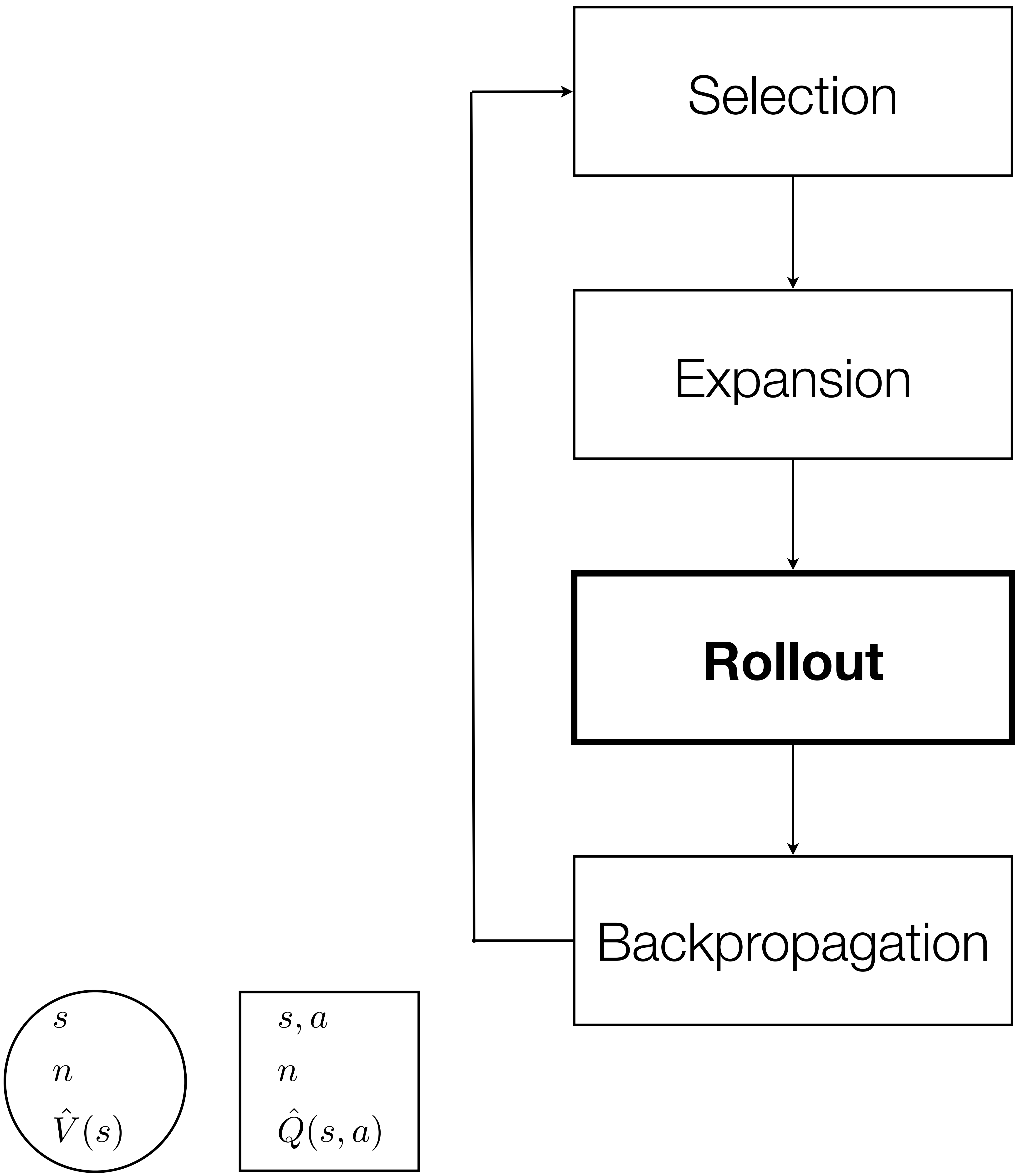


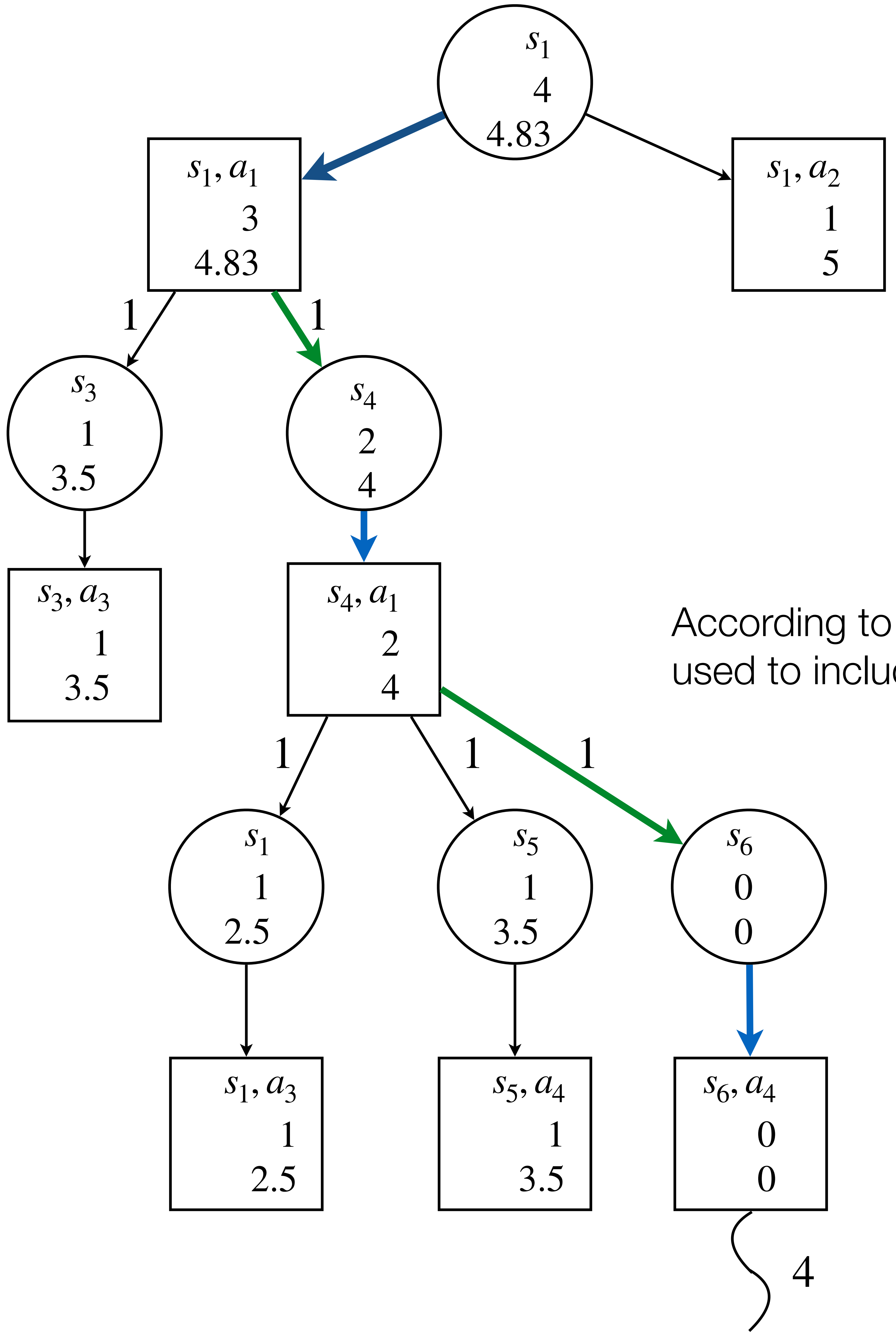




Choose random action

MCTS



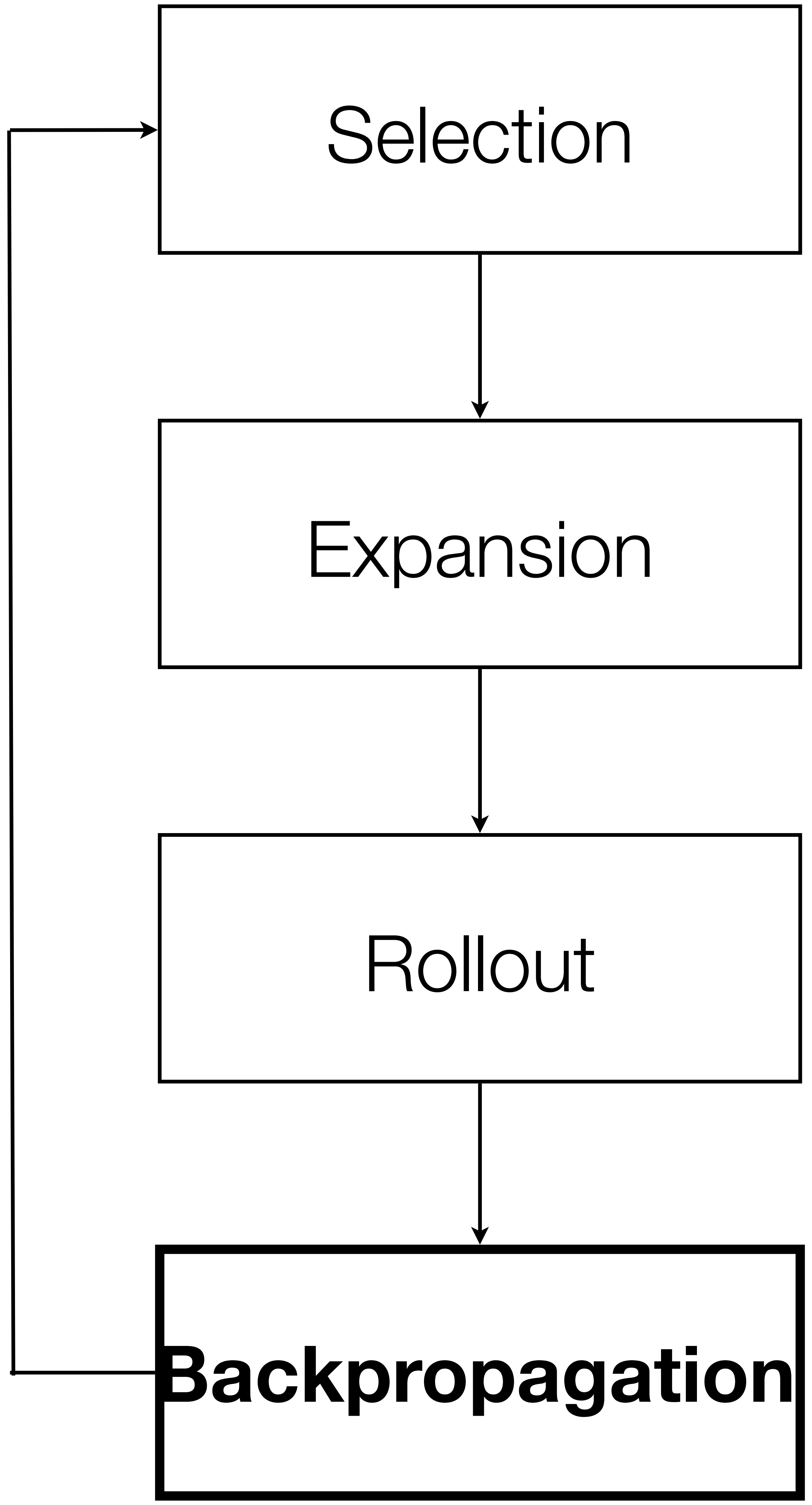


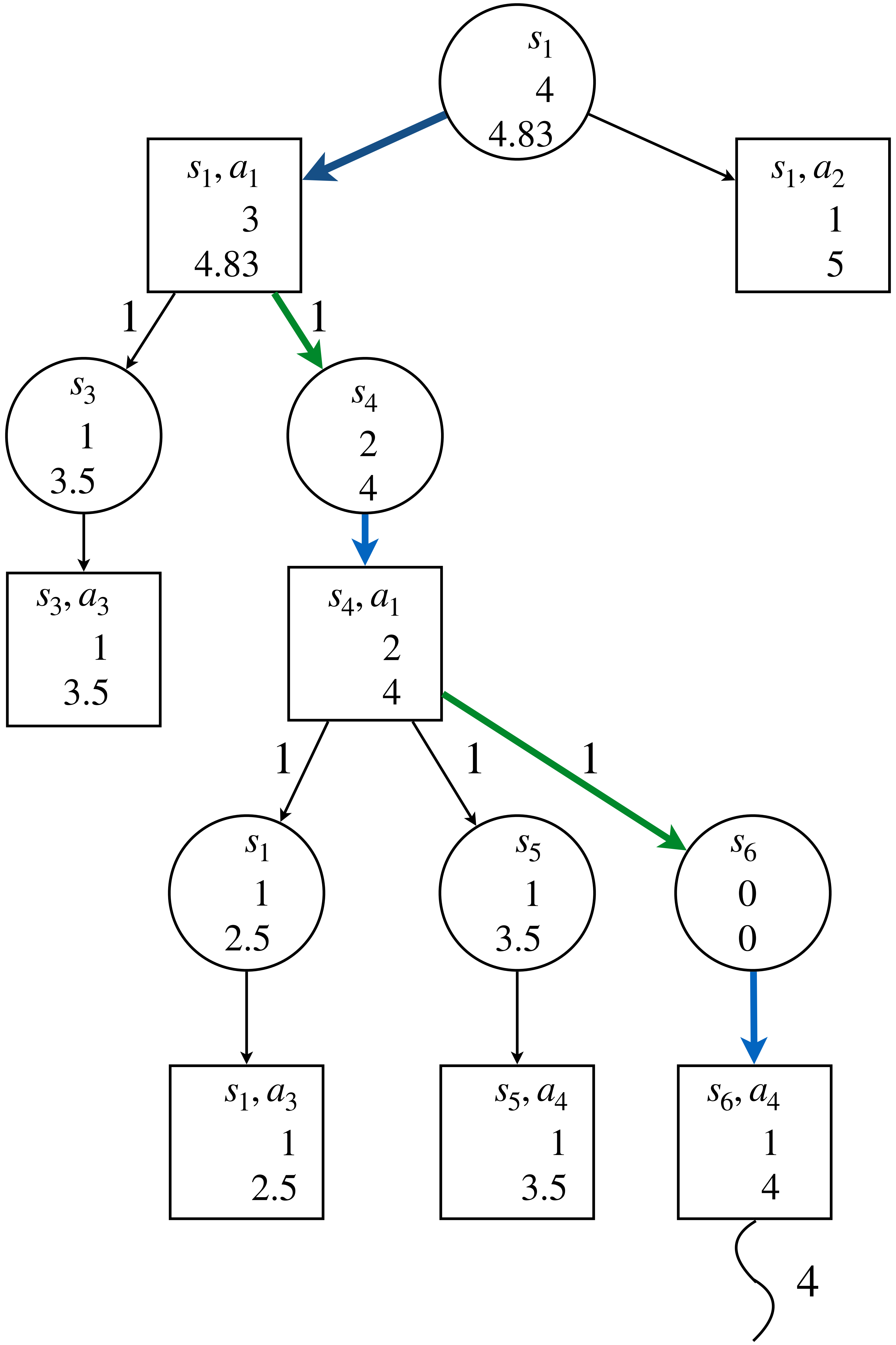
According to rollout policy - can be used to include domain knowledge

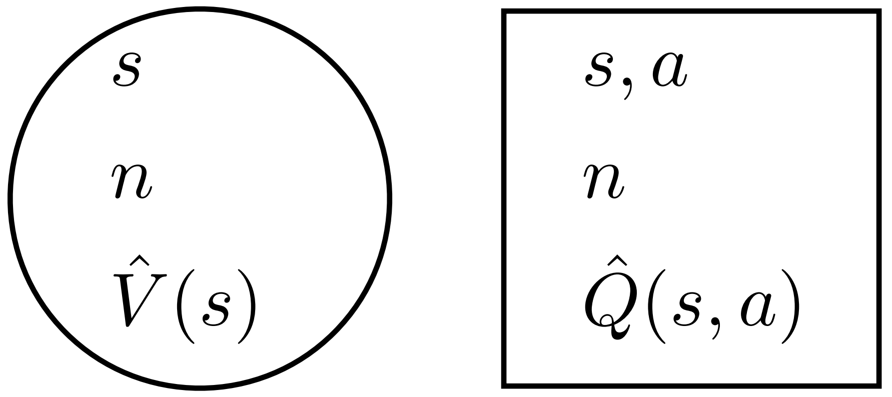
MCTS


Selection
Expansion
Rollout
Backpropagation
s_1 4 4.83
s_1, a_1 3 4.83
s_1, a_2 1 5
1
1
s_3 1 3.5
s_4 2 4
s_3, a_3 1 3.5
s_4, a_1 2 4
According to rollout policy - can be used to include domain knowledge
1
1
1
s_1 1 2.5
s_5 1 3.5
s_6 0 0
s_1, a_3 1 2.5
s_5, a_4 1 3.5
s_6, a_4 0 0
4
s n $\hat{V}(s)$
s, a n $\hat{Q}(s, a)$

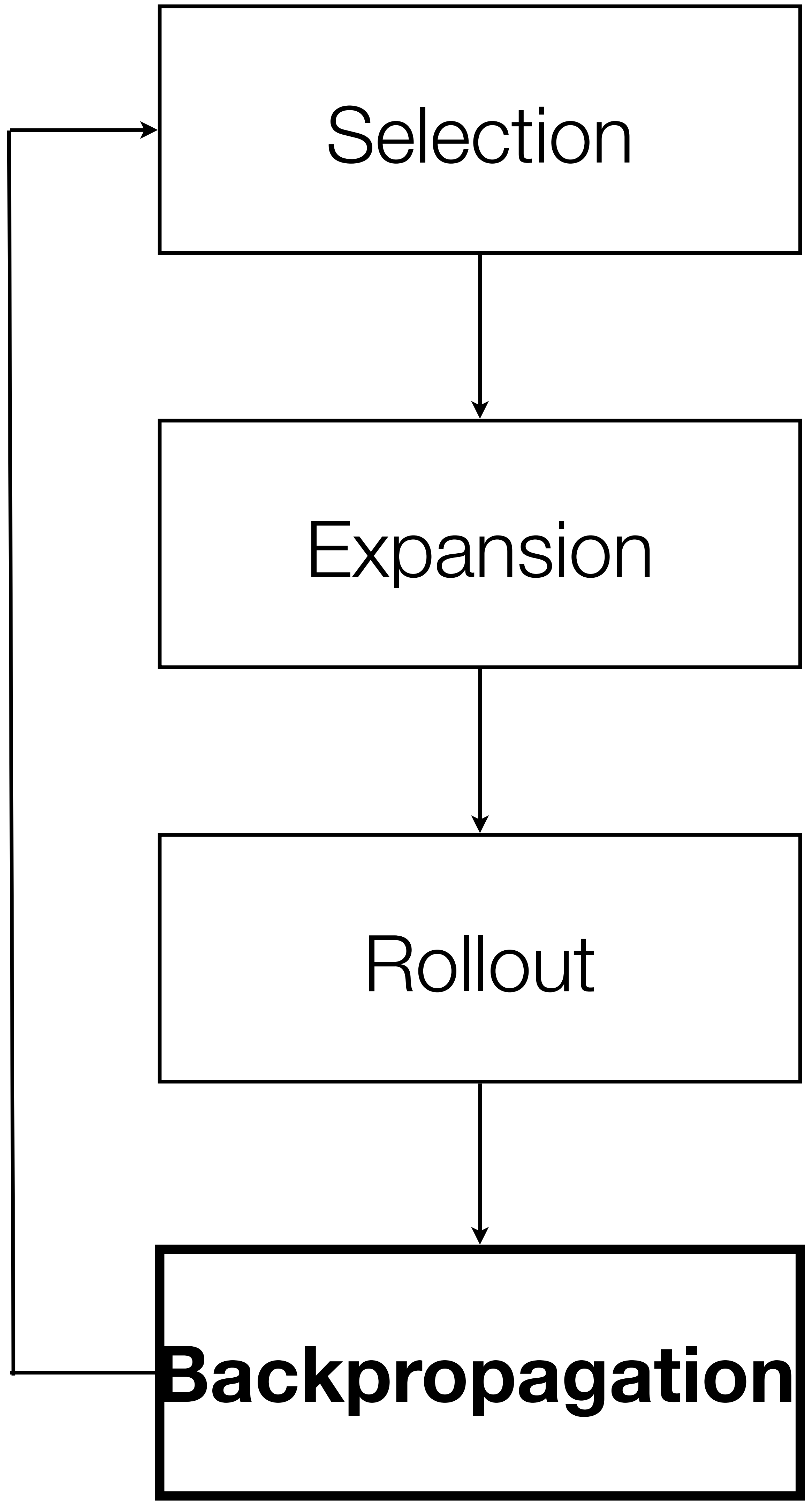
MCTS

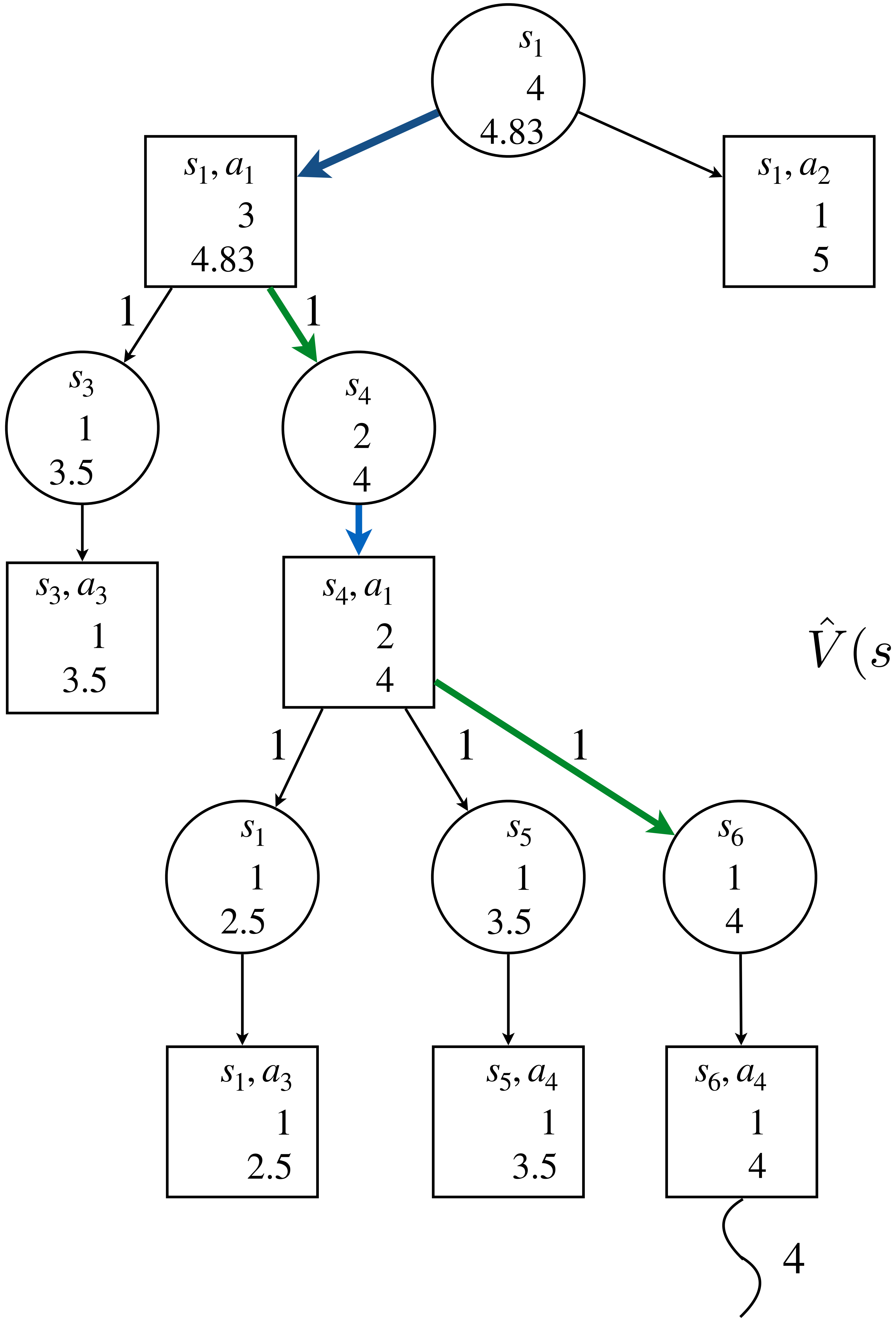

Selection
Expansion
Rollout
Backpropagation


s_1
4
4.83
s_1, a_1
3
4.83
s_1, a_2
1
5
1
1
s_3
1
3.5
s_4
2
4
s_3, a_3
1
3.5
s_4, a_1
2
4
1
1
1
s_1
1
2.5
s_5
1
3.5
s_6
0
0
s_1, a_3
1
2.5
s_5, a_4
1
3.5
s_6, a_4
1
4
4

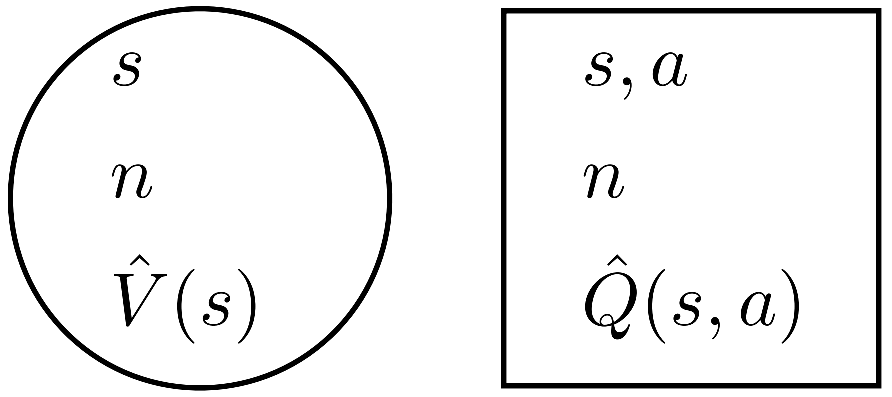

s
n
$\hat{V}(s)$
s, a
n
$\hat{Q}(s, a)$

MCTS

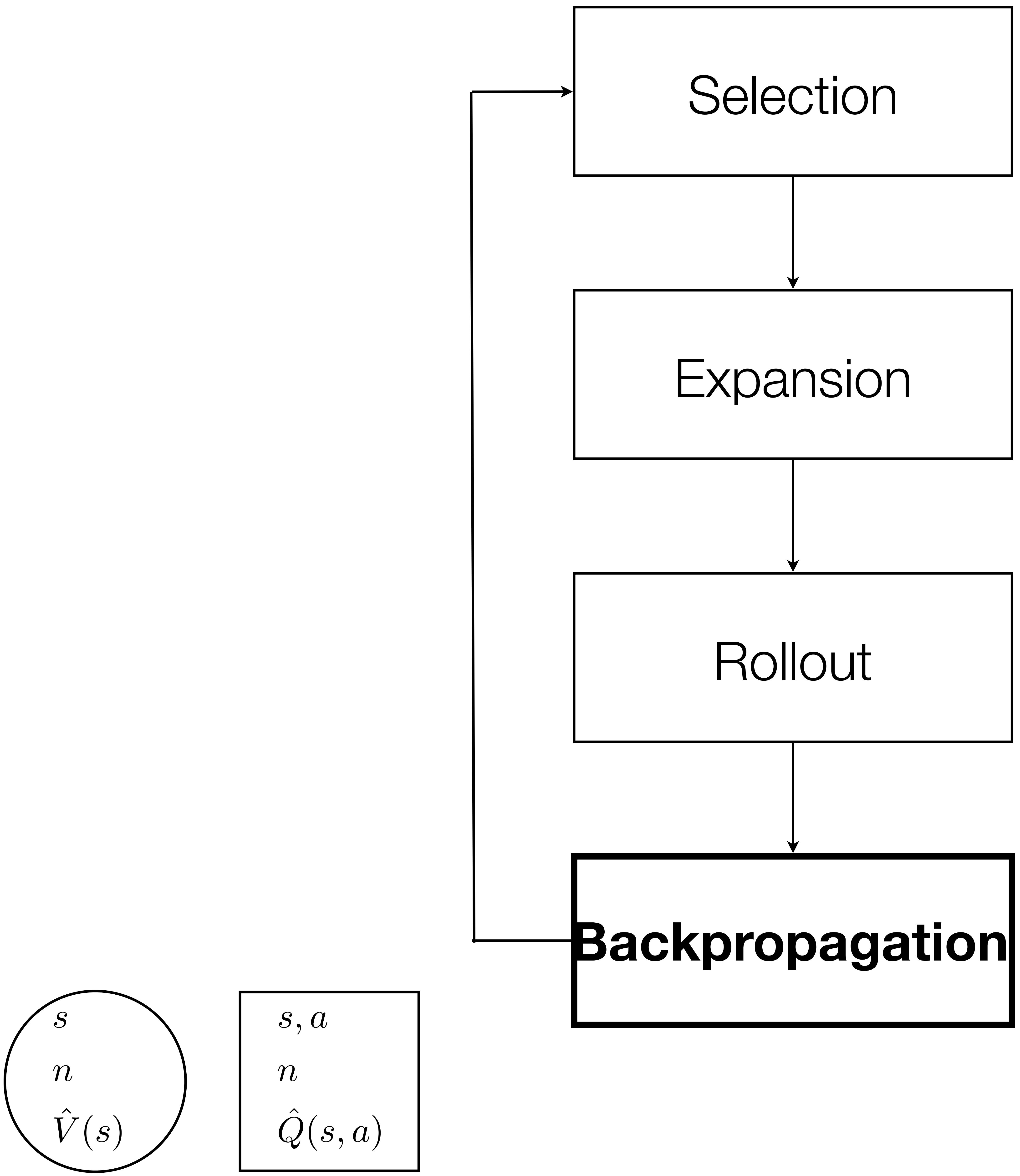


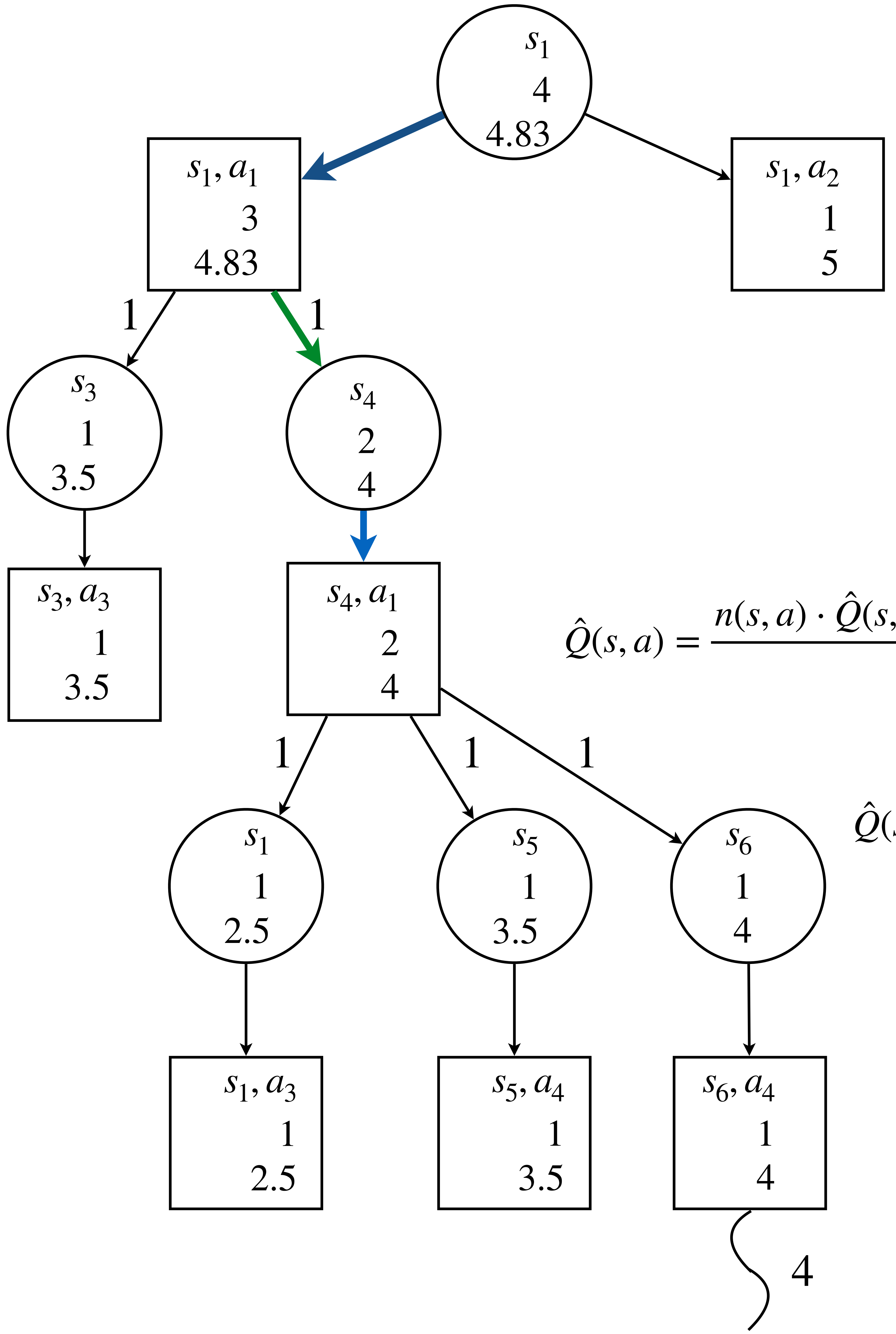


$$\hat{V}(s) = \min_a \hat{Q}(s, a)$$



MCTS

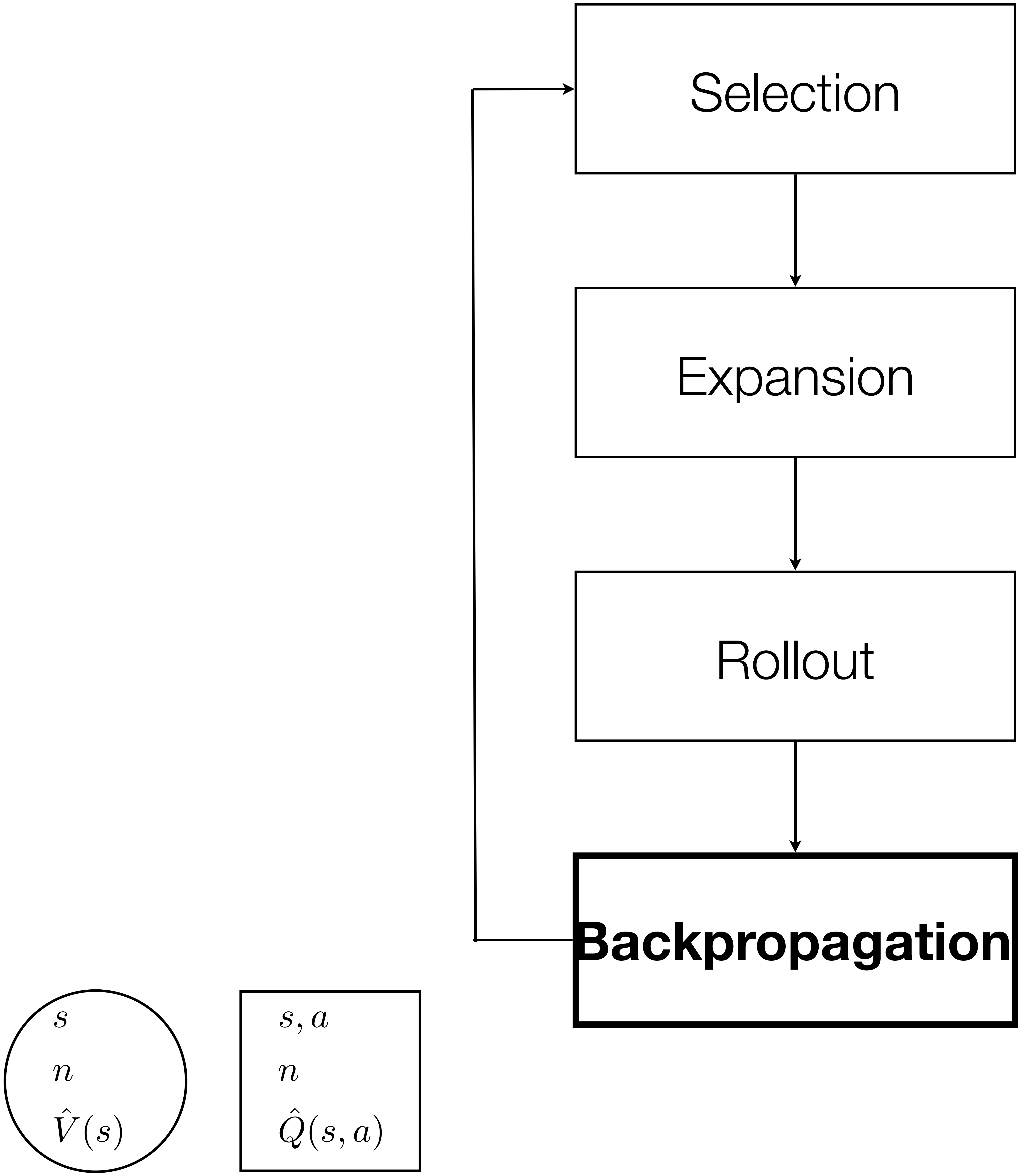


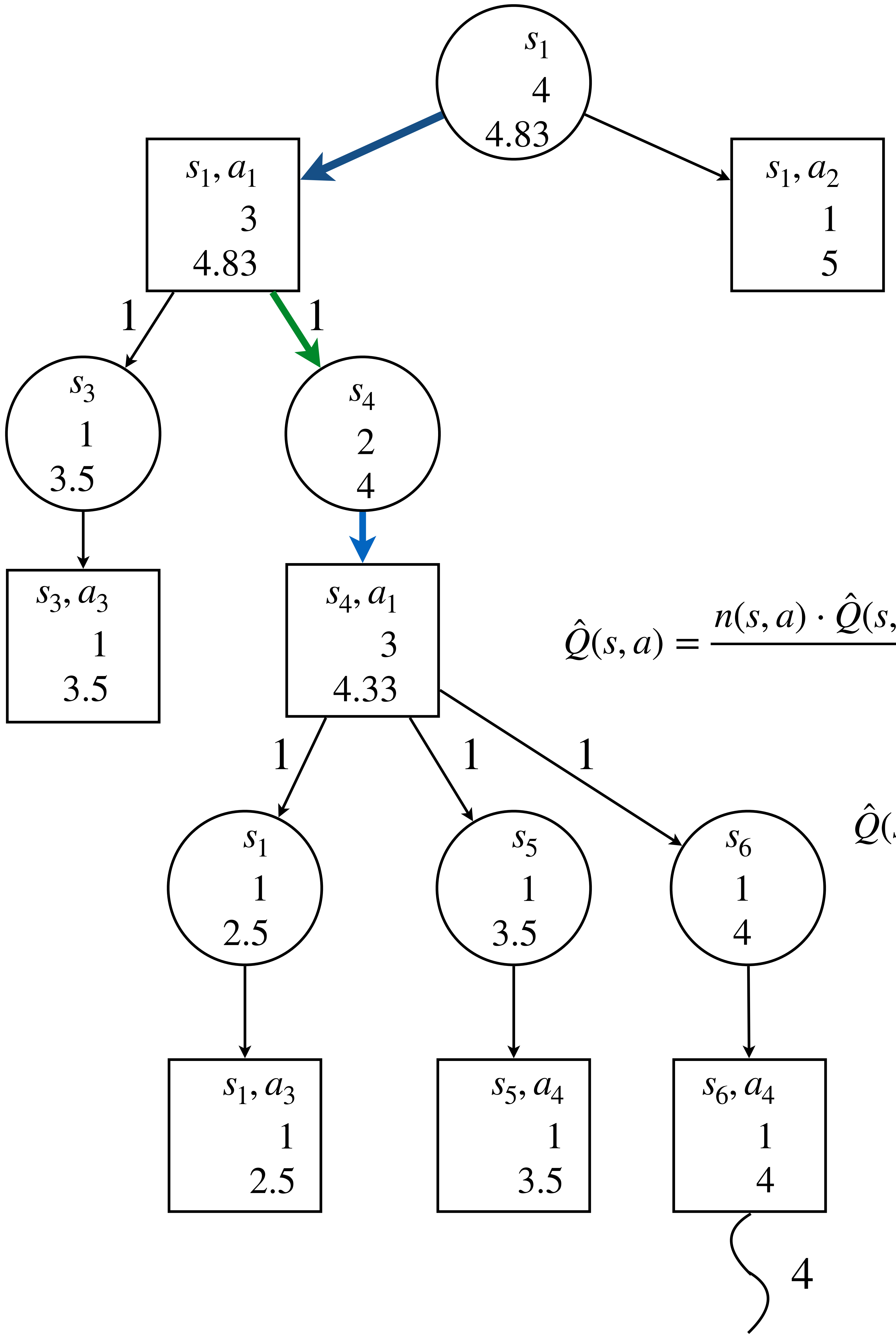


$$\hat{Q}(s,a) = \frac{n(s,a) \cdot \hat{Q}(s,a) + cumulCostFrom(s,a)}{n(s,a)+1}$$

$$\hat{Q}(s_4, a_1) = \frac{2 \cdot 4 + 5}{3} \approx 4.33$$

MCTS

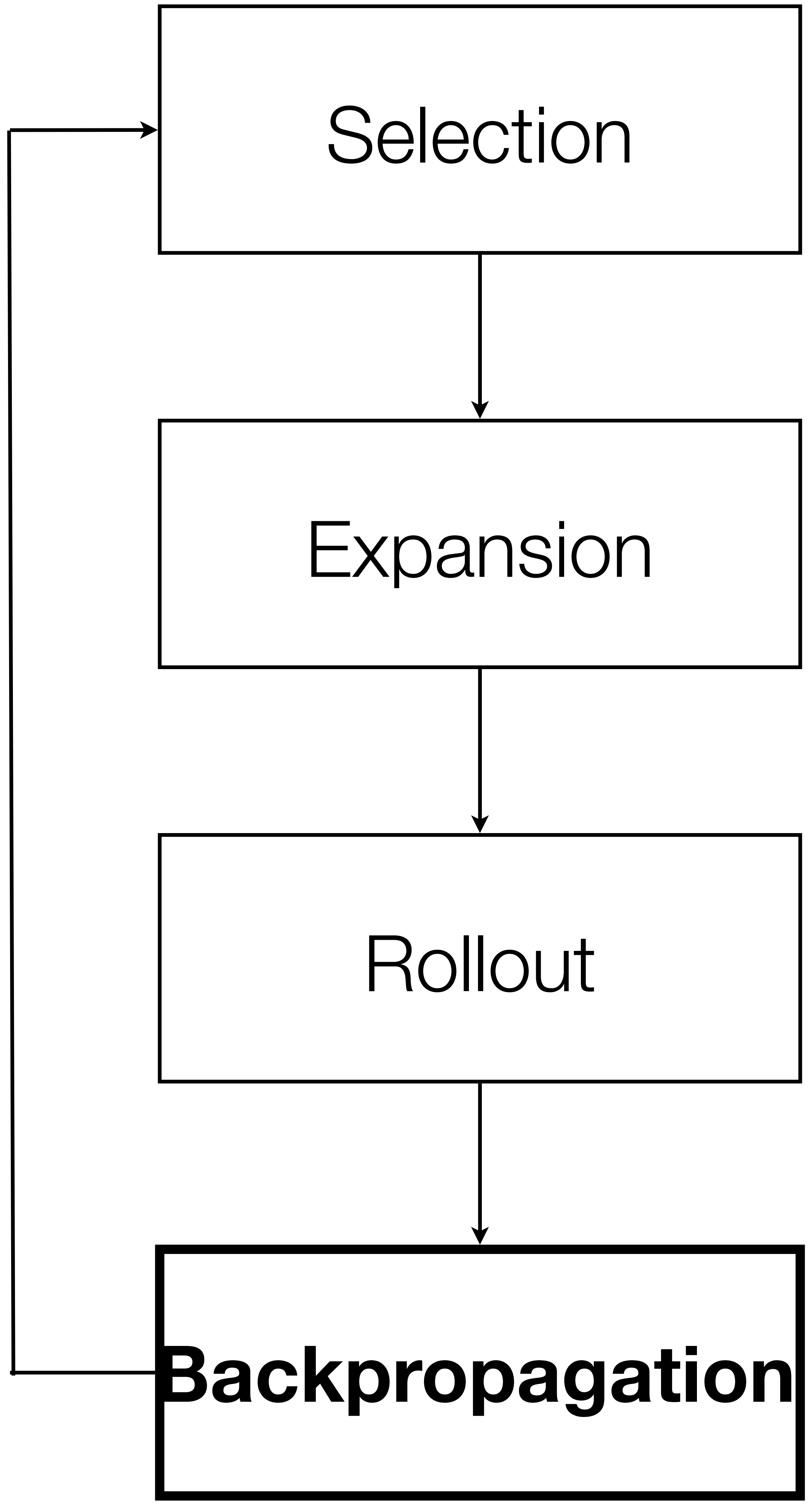


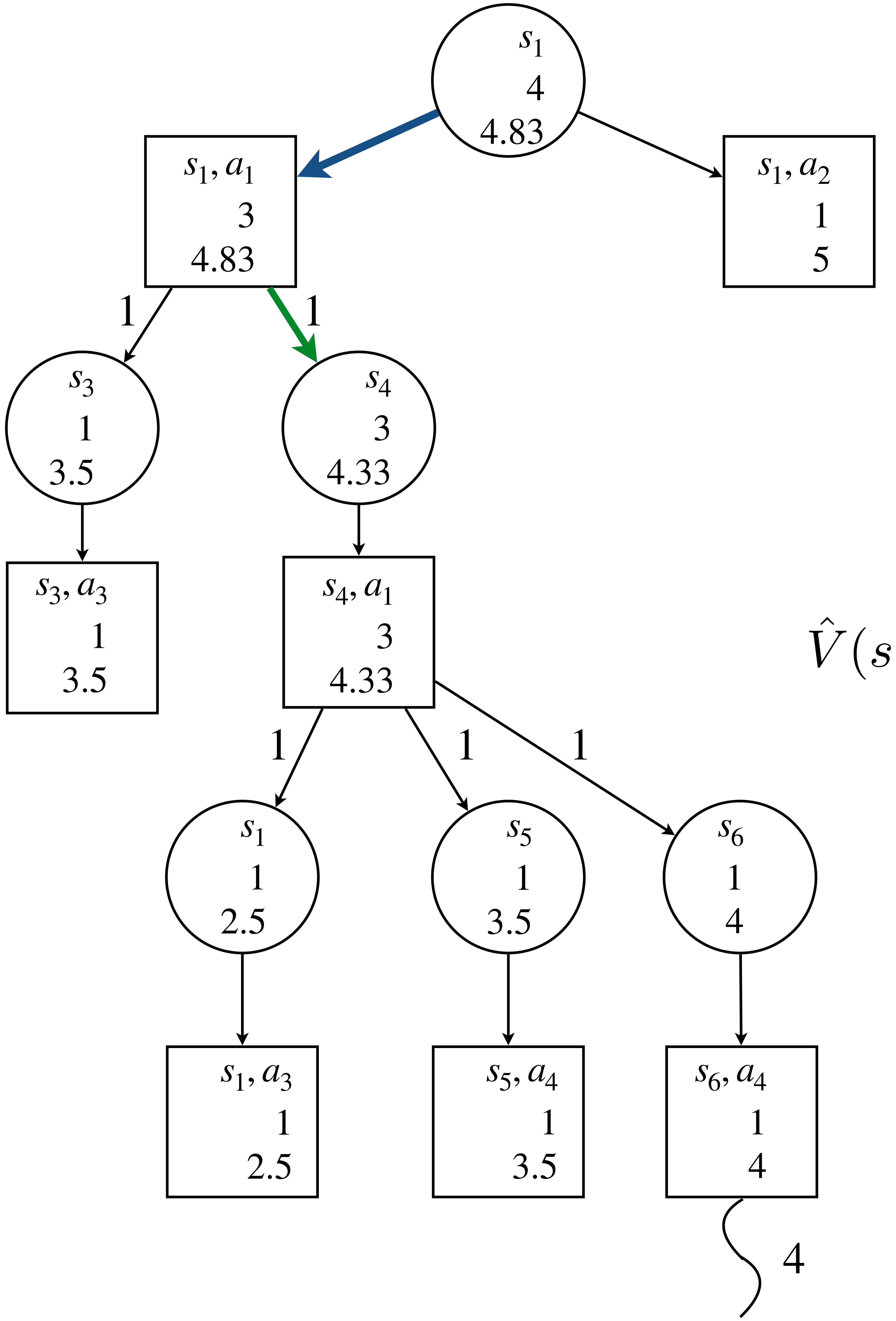


$$\hat{Q}(s,a) = \frac{n(s,a)\cdot\hat{Q}(s,a) + cumulCostFrom(s,a)}{n(s,a)+1}$$

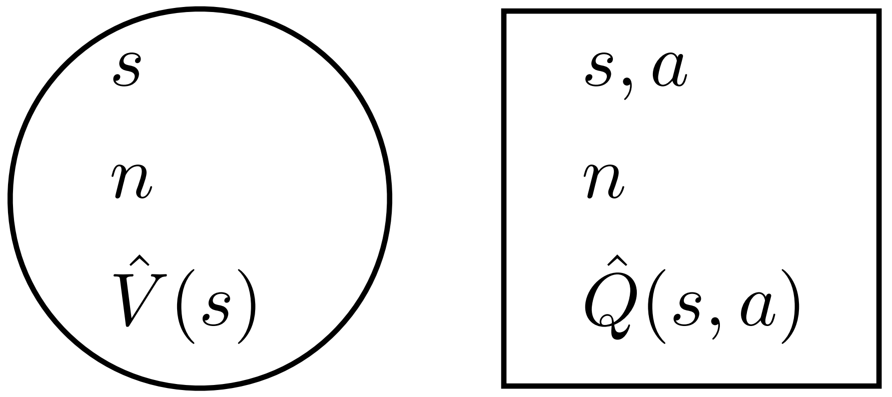
$$\hat{Q}(s_4,a_1) = \frac{2\cdot 4 + 5}{3} \approx 4.33$$

MCTS

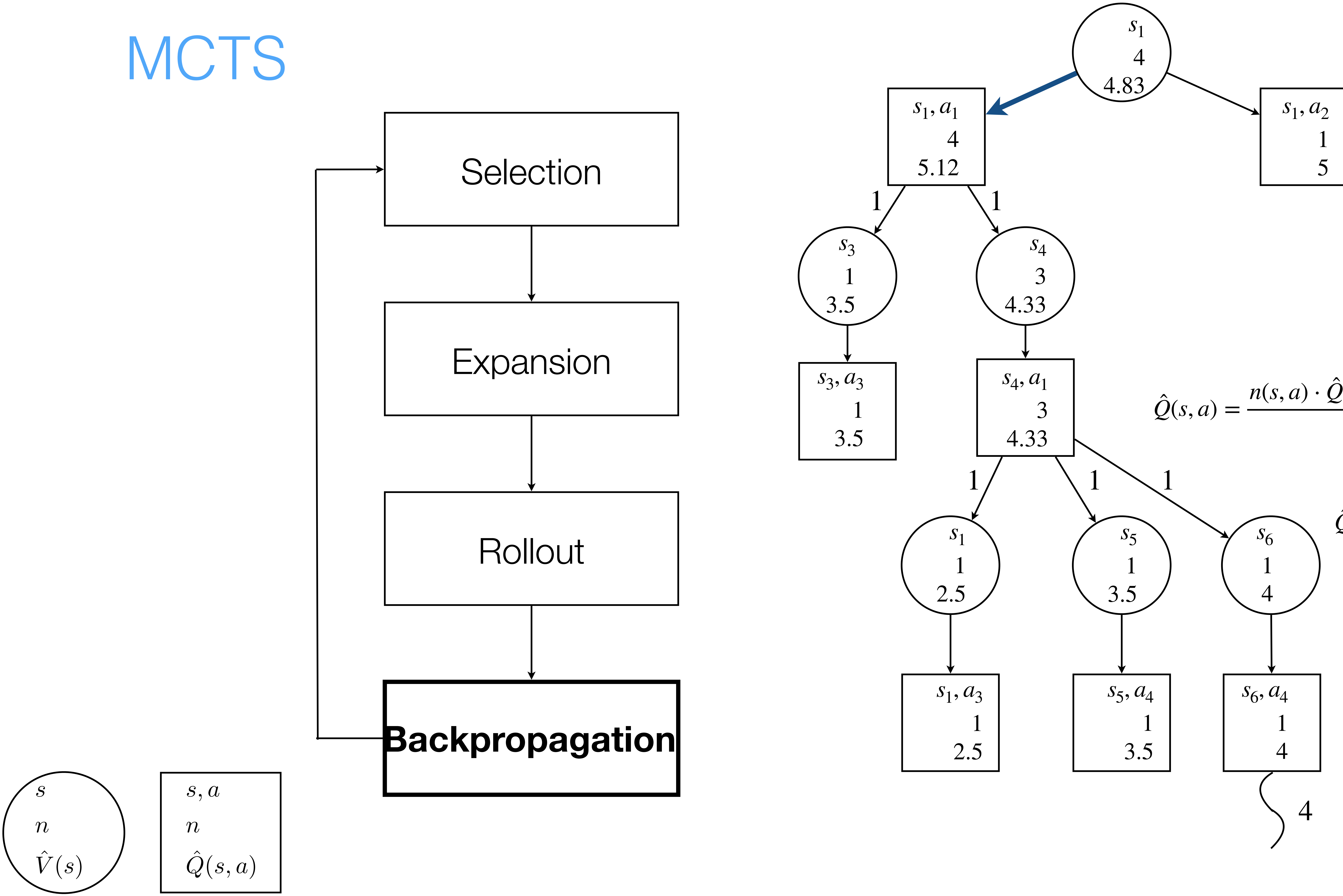




$$\hat{V}(s) = \min_a \hat{Q}(s, a)$$



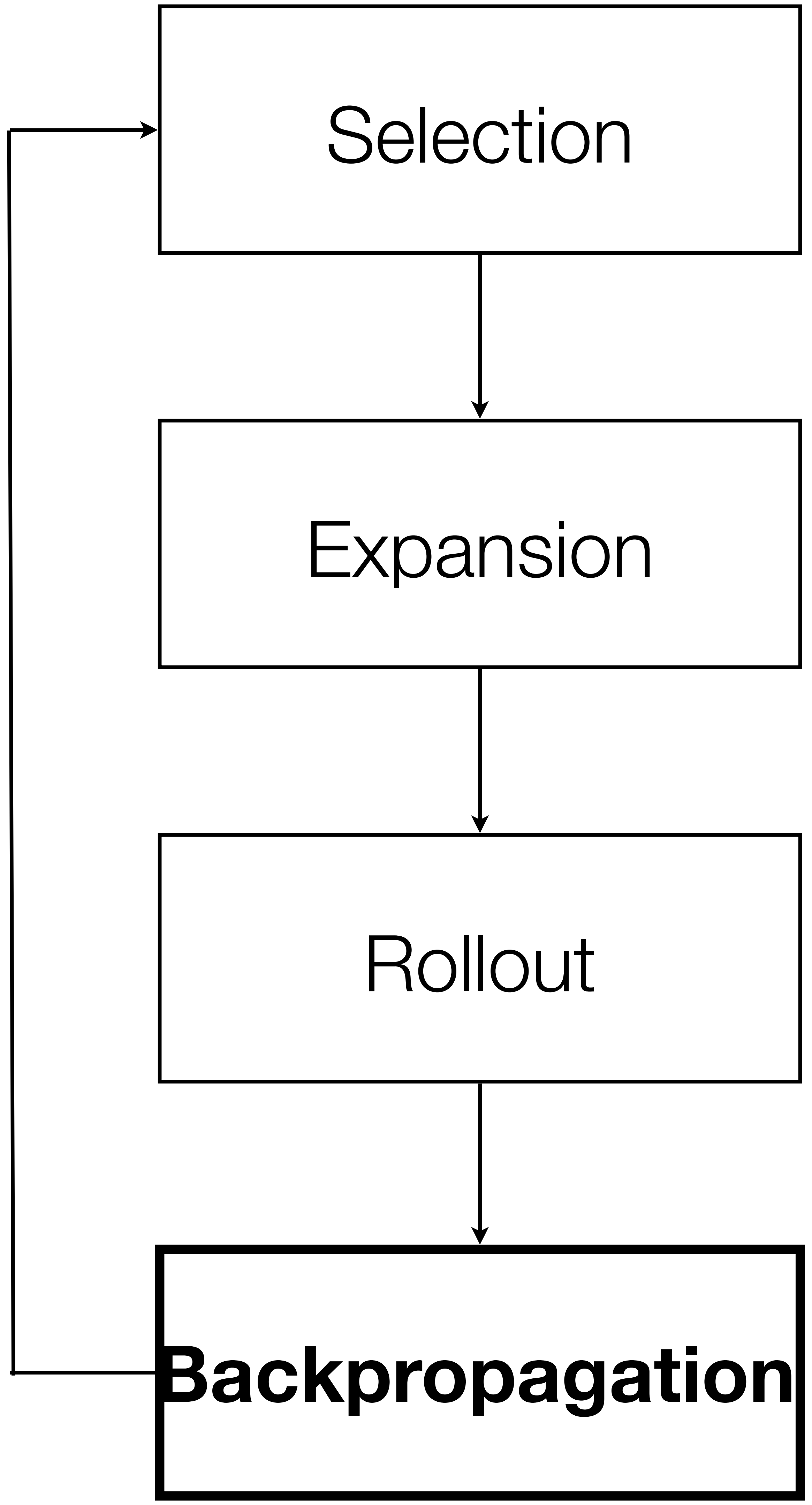
MCTS

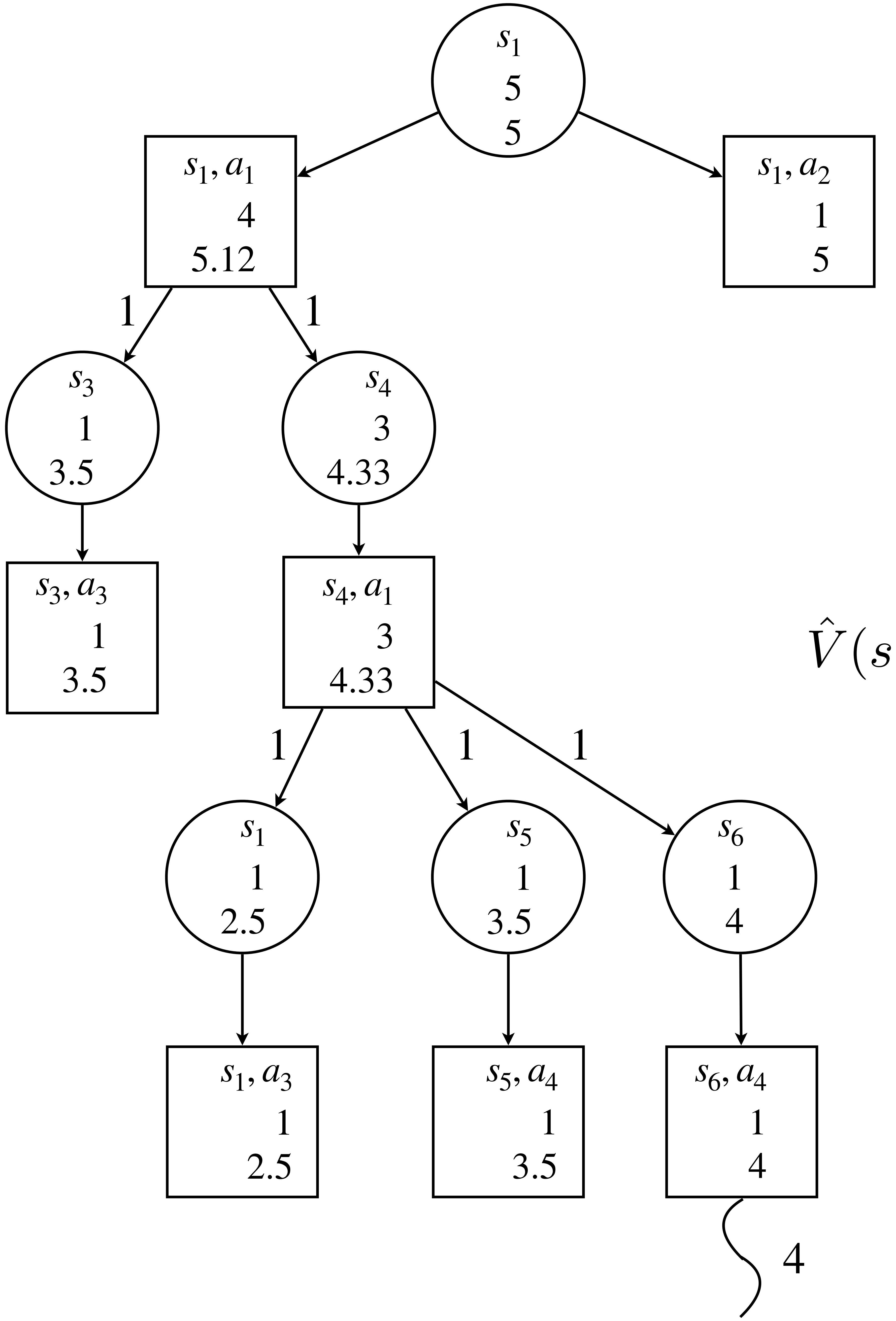


$$\hat{Q}(s,a) = \frac{n(s,a) \cdot \hat{Q}(s,a) + cumulCostFrom(s,a)}{n(s,a)+1}$$

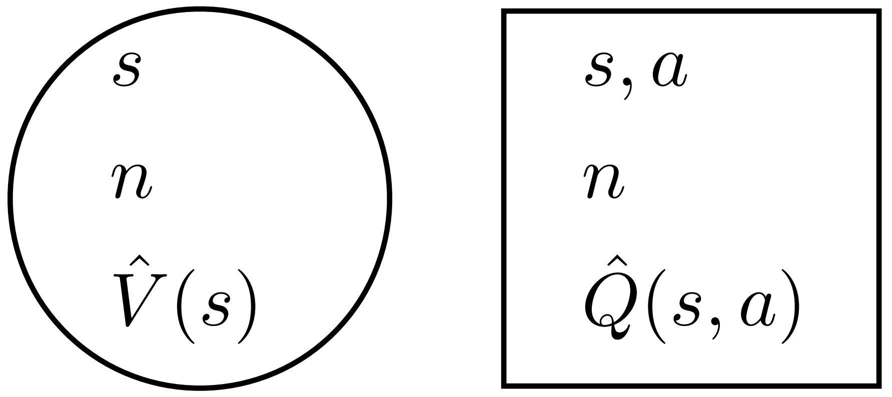
$$\hat{Q}(s_1,a_1) = \frac{3 \cdot 4.83 + 6}{4} \approx 5.12$$

MCTS



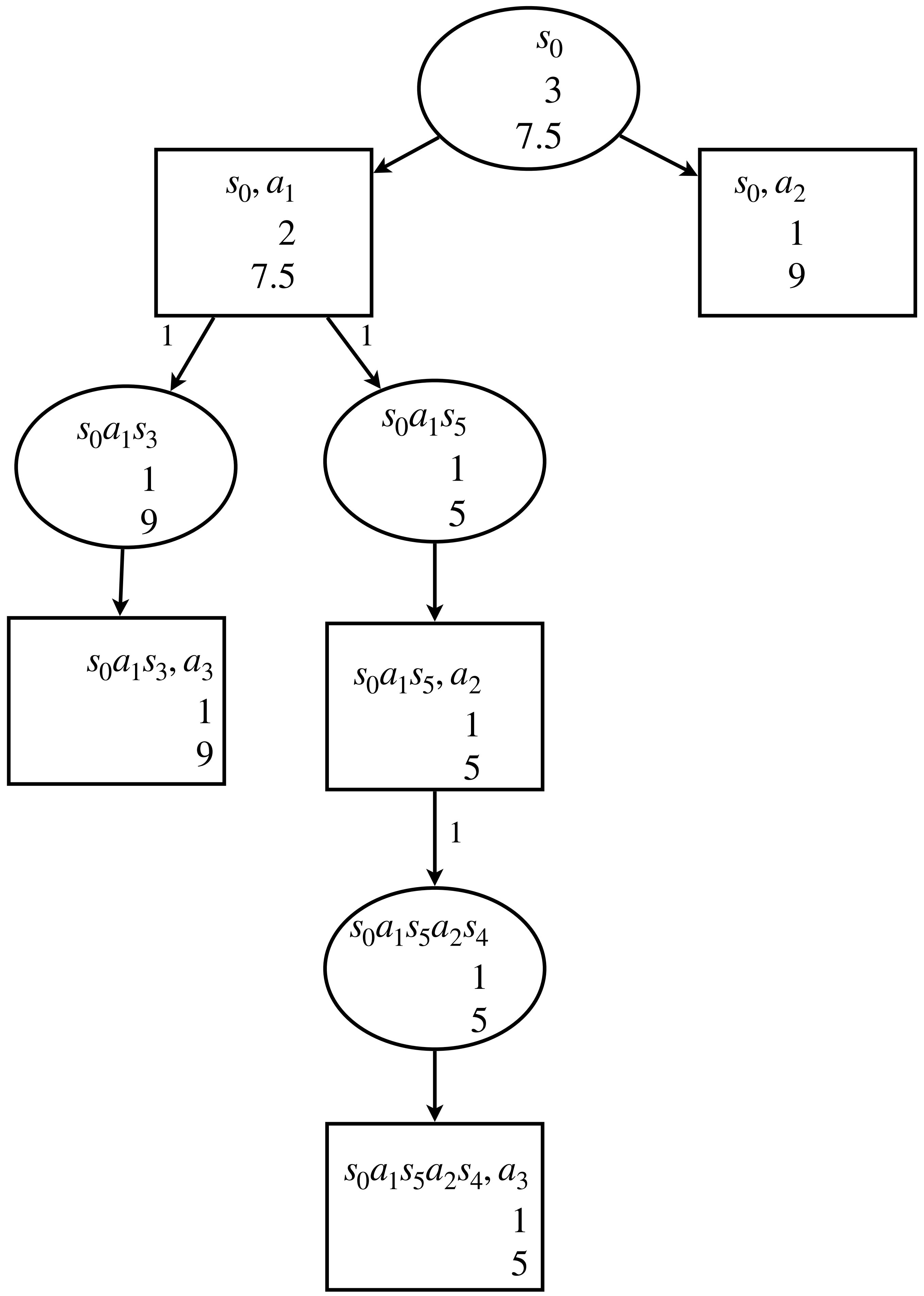


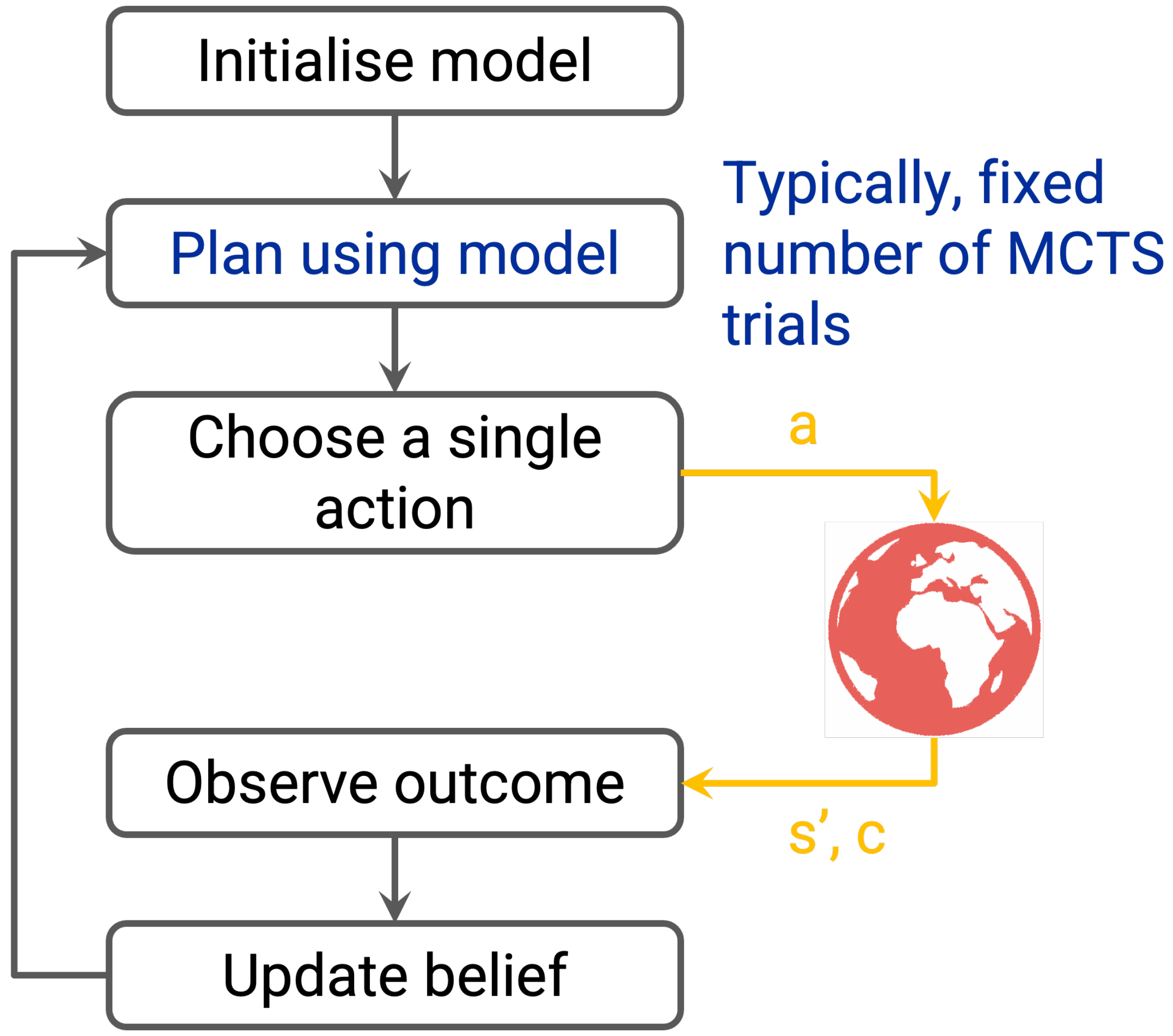
$$\hat{V}(s) = \min_a \hat{Q}(s,a)$$



Bayes-adaptive Monte-Carlo Planning

1. Repeat (until goal reached)
   1. Repeat (until timeout)
      1. Sample P according to $p(P)$ (root sampling)
      2. Run MCTS trial under P
   2. Execute action in the environment according to search tree
   3. Observe outcome and update $p(P)$ accordingly





Summary

- Putting a prior over the uncertainty set yields a model based Bayes-adaptive RL problem

- The problem can be encoded into a specific type of belief MDP, names Bayes-adaptive MDP

- To plan for BAMDPs, we use an MCTS algorithm which incrementally builds and approximates the BAMDP solution

- Until now, we have not discussed an aspect that has been central in the previous 4 lectures
    - Robustness to model uncertainty
    - BAMCP optimises in expectation
    - We will address robustness in a BAMDP context in the end of this lecture

References

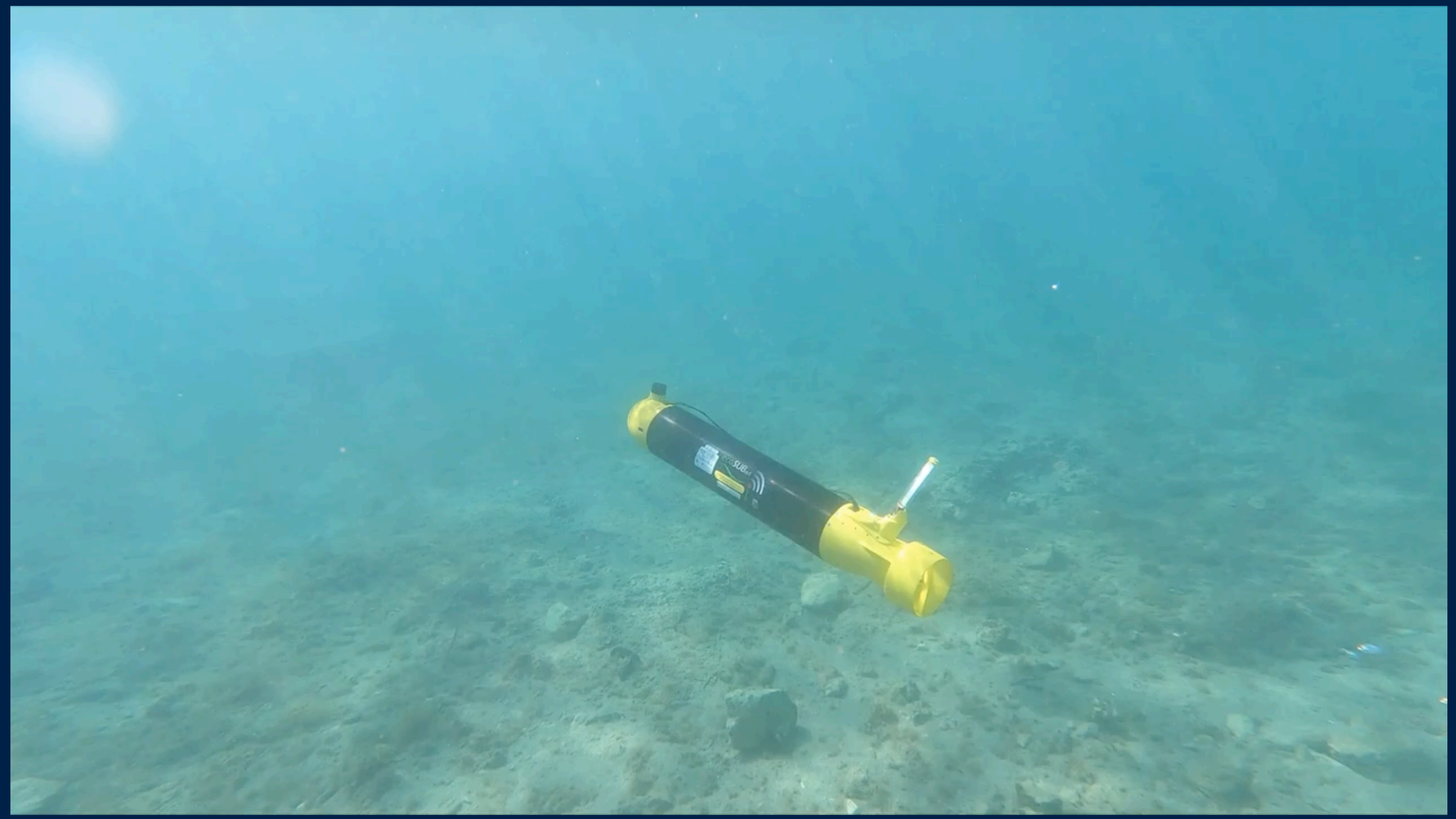
- Bayes-adaptive MDPs

  - M. O. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*, PhD Thesis, University of Massachusetts Amherst, 2002.
  - A. Guez, D. Silver, P. Dayan. *Efficient Bayes-Adaptive Reinforcement Learning using Sample-Based Search*, NeurIPS, 2012.

Epistemically Uncertain Robots

GOAL-ORIENTED
AUTONOMOUS
LONG-LIVED SYSTEMS
OXFORD ROBOTICS INSTITUTE

Sequential decision-making** techniques to allow **long-lived autonomous robots** to achieve their **goals, under uncertainty



MOBILE ROBOTICS GROUP
OXFORD ROBOTICS INSTITUTE

DYNAMIC ROBOT SYSTEMS GROUP
OXFORD ROBOTICS INSTITUTE

A²I
APPLIED ARTIFICIAL INTELLIGENCE LAB
OXFORD ROBOTICS INSTITUTE

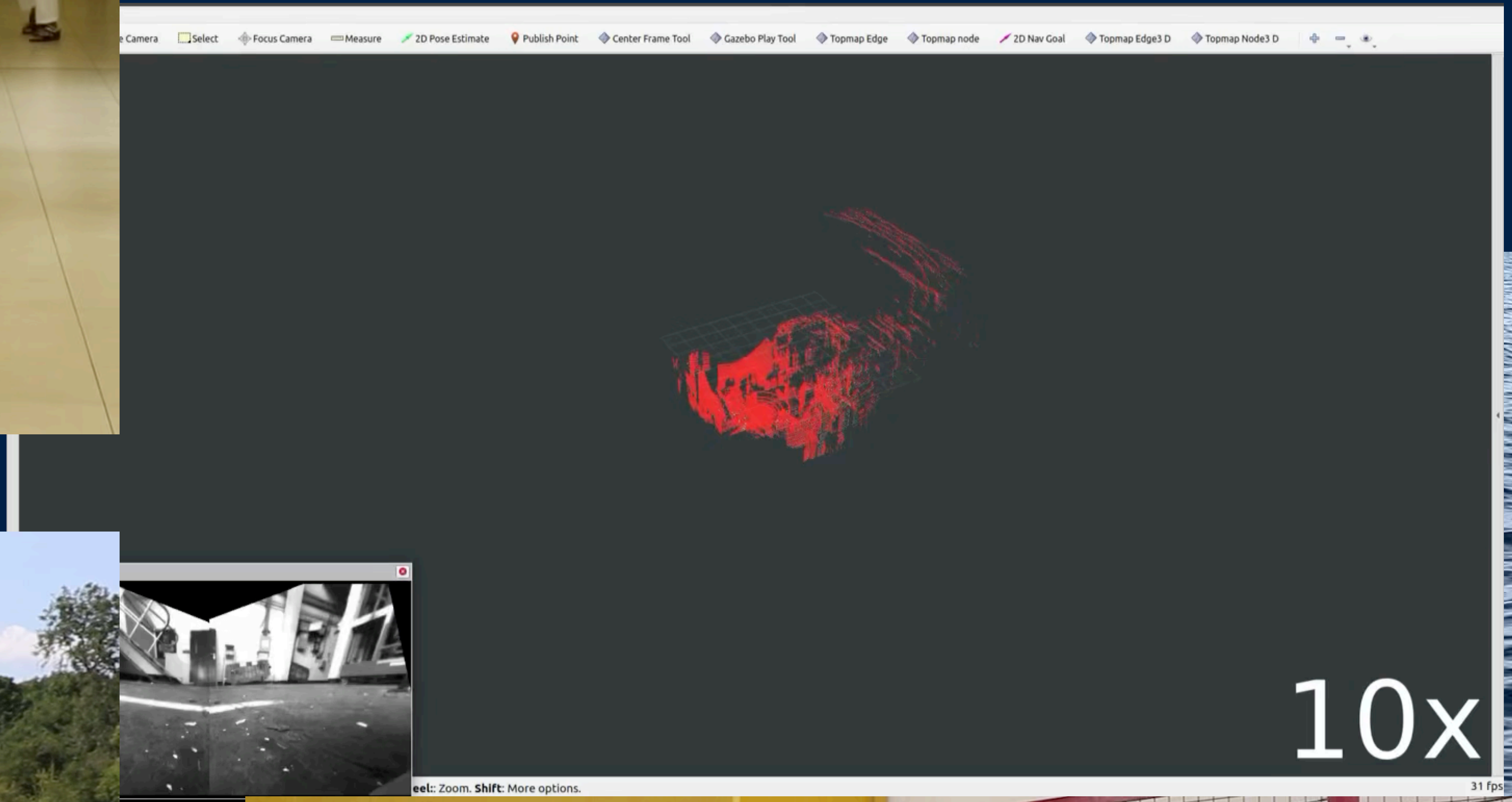
GOAL-ORIENTED
AUTONOMOUS
LONG-LIVED SYSTEMS
OXFORD ROBOTICS INSTITUTE

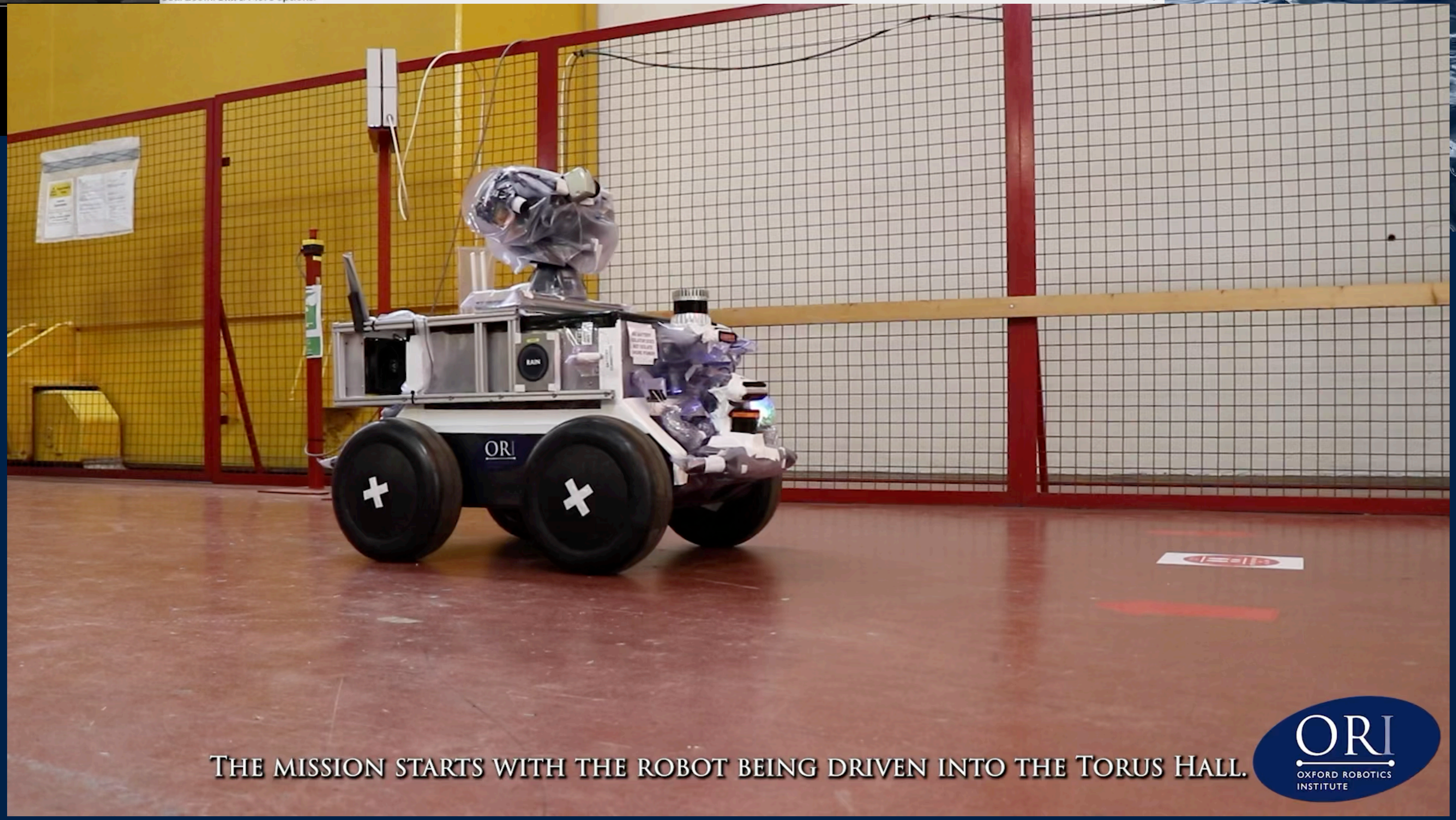
ESP
ESTIMATION, SEARCH & PLANNING GROUP
OXFORD ROBOTICS INSTITUTE

SOFT ROBOTICS LAB
OXFORD ROBOTICS INSTITUTE

COGNITIVE ROBOTICS GROUP
OXFORD ROBOTICS INSTITUTE

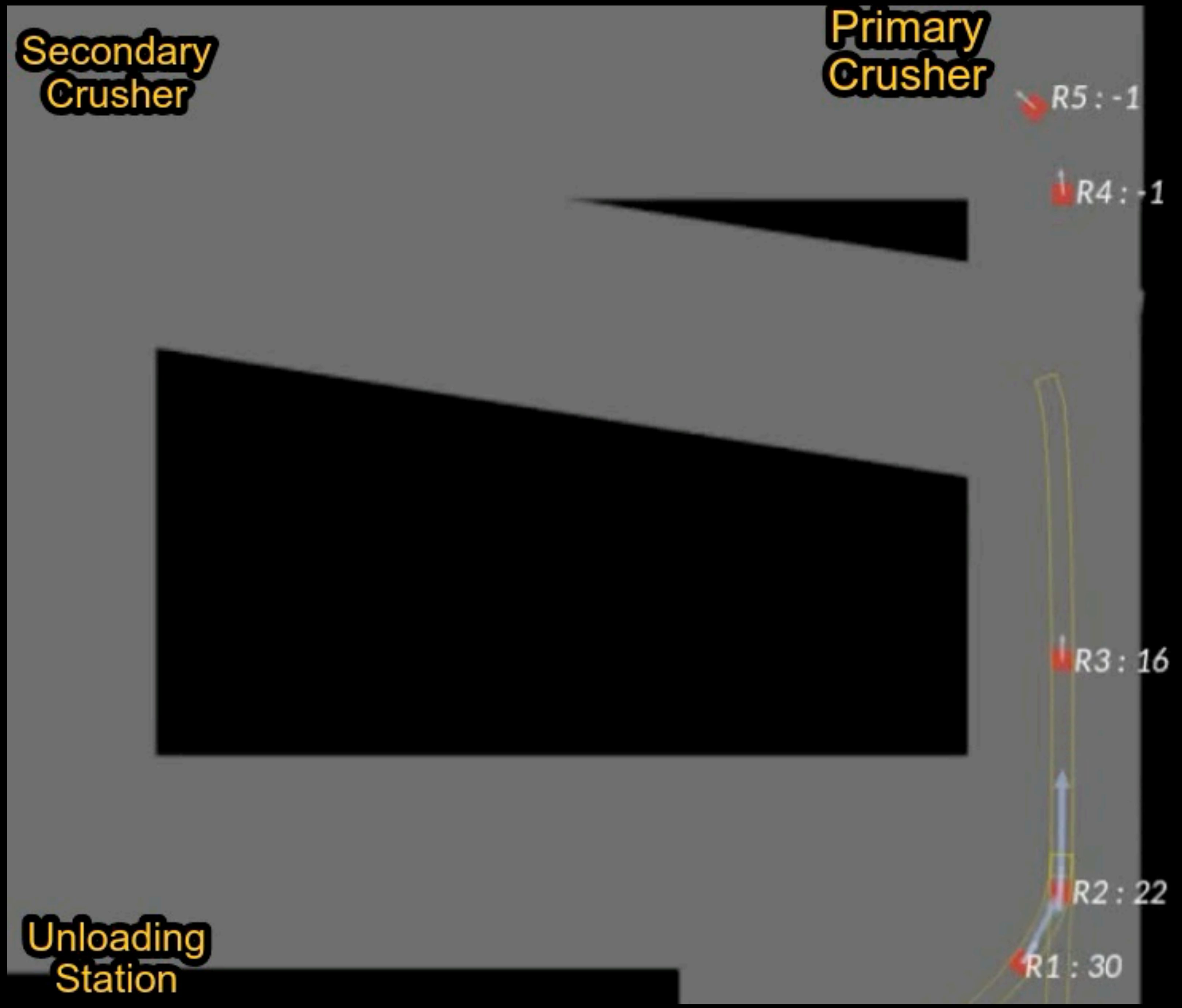
Mission planning for **autonomous systems** with **probabilistic guarantees** and **rich specifications**

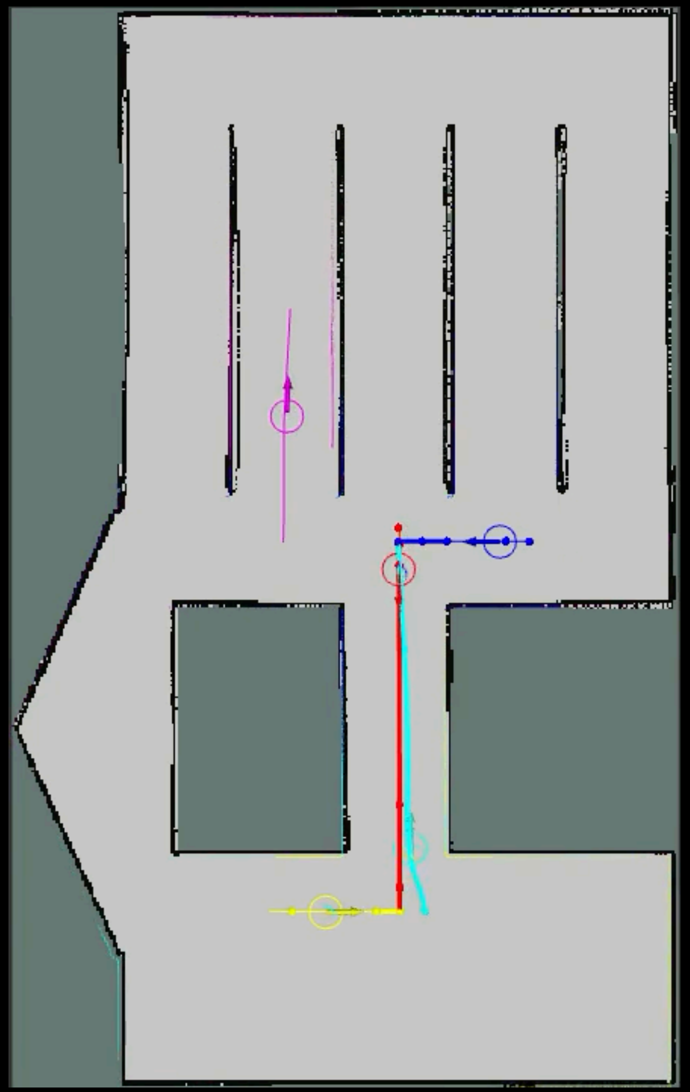


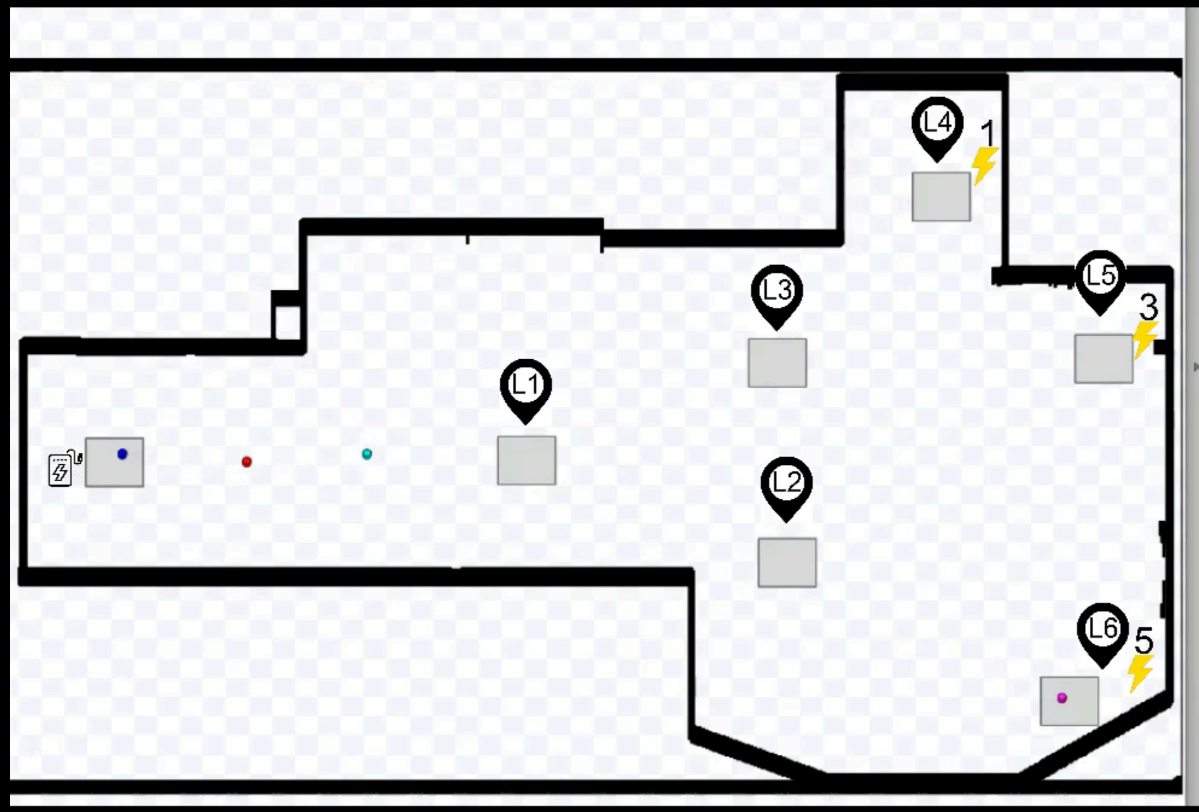


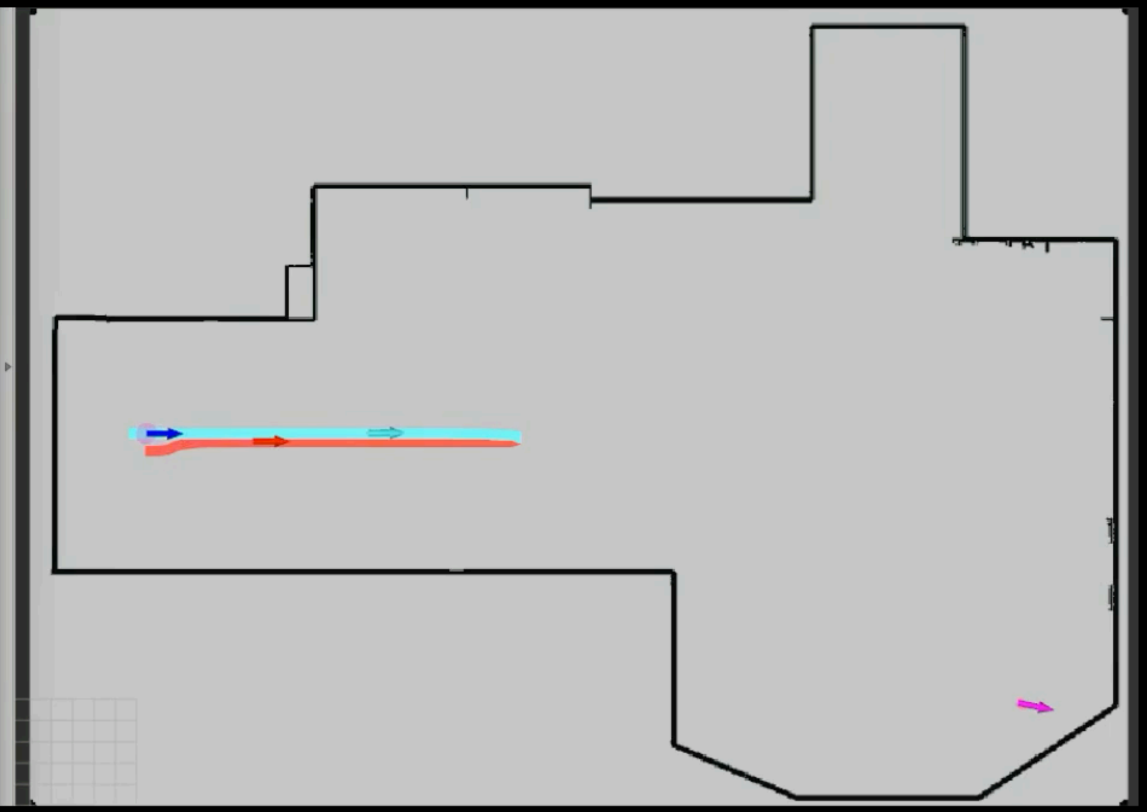
GOAL-ORIENTED AUTONOMOUS LONG-LIVED SYSTEMS
OXFORD ROBOTICS INSTITUTE

Multi-robot coordination with **team guarantees**, **resource constraints** and **continuous time**









Mission: F (WayPoint27 & F WayPoint28)

F (WayPoint59 & F WayPoint58), F WayPoint8

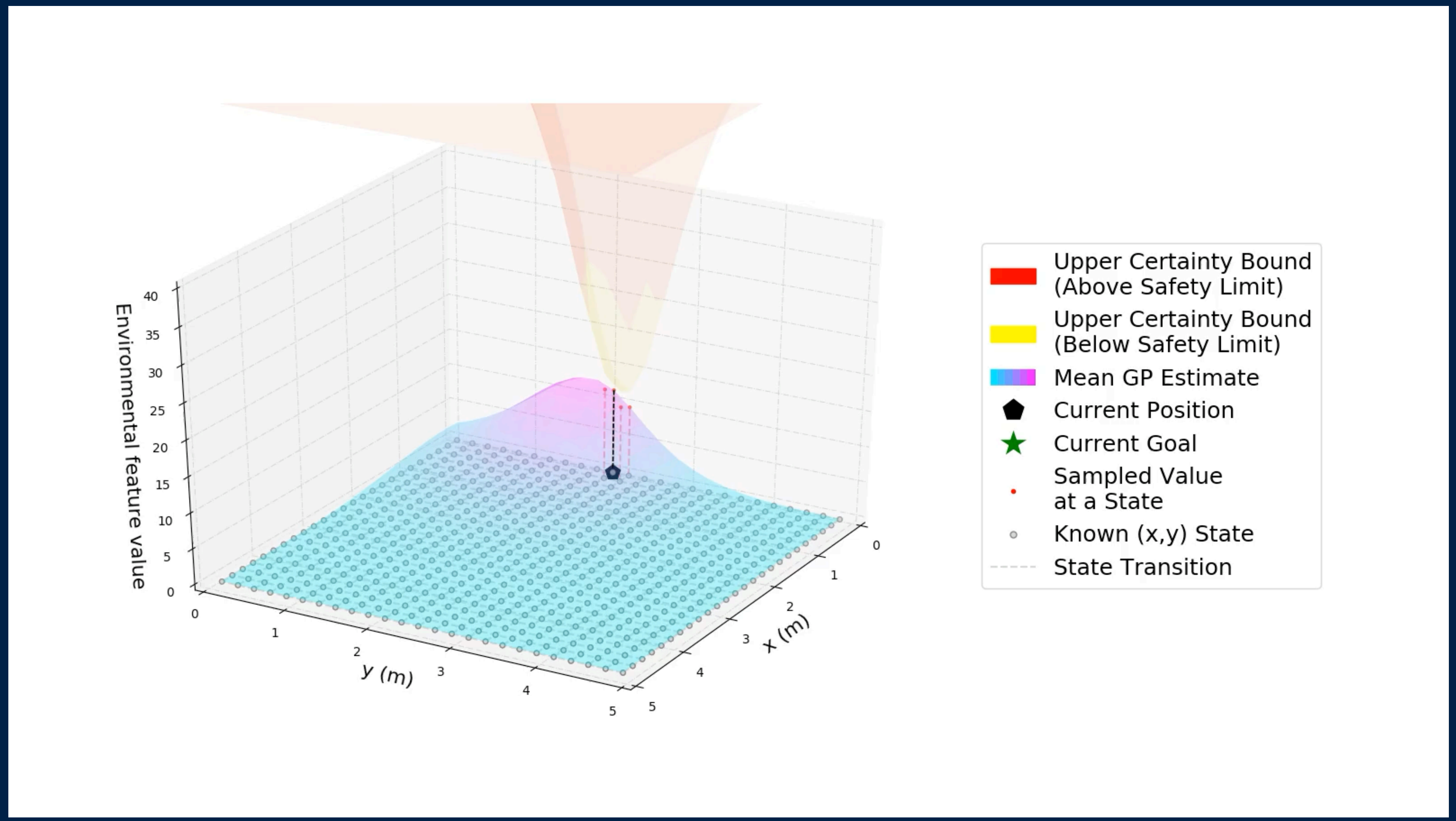
F WayPoint22, F WayPoint36, F WayPoint47,

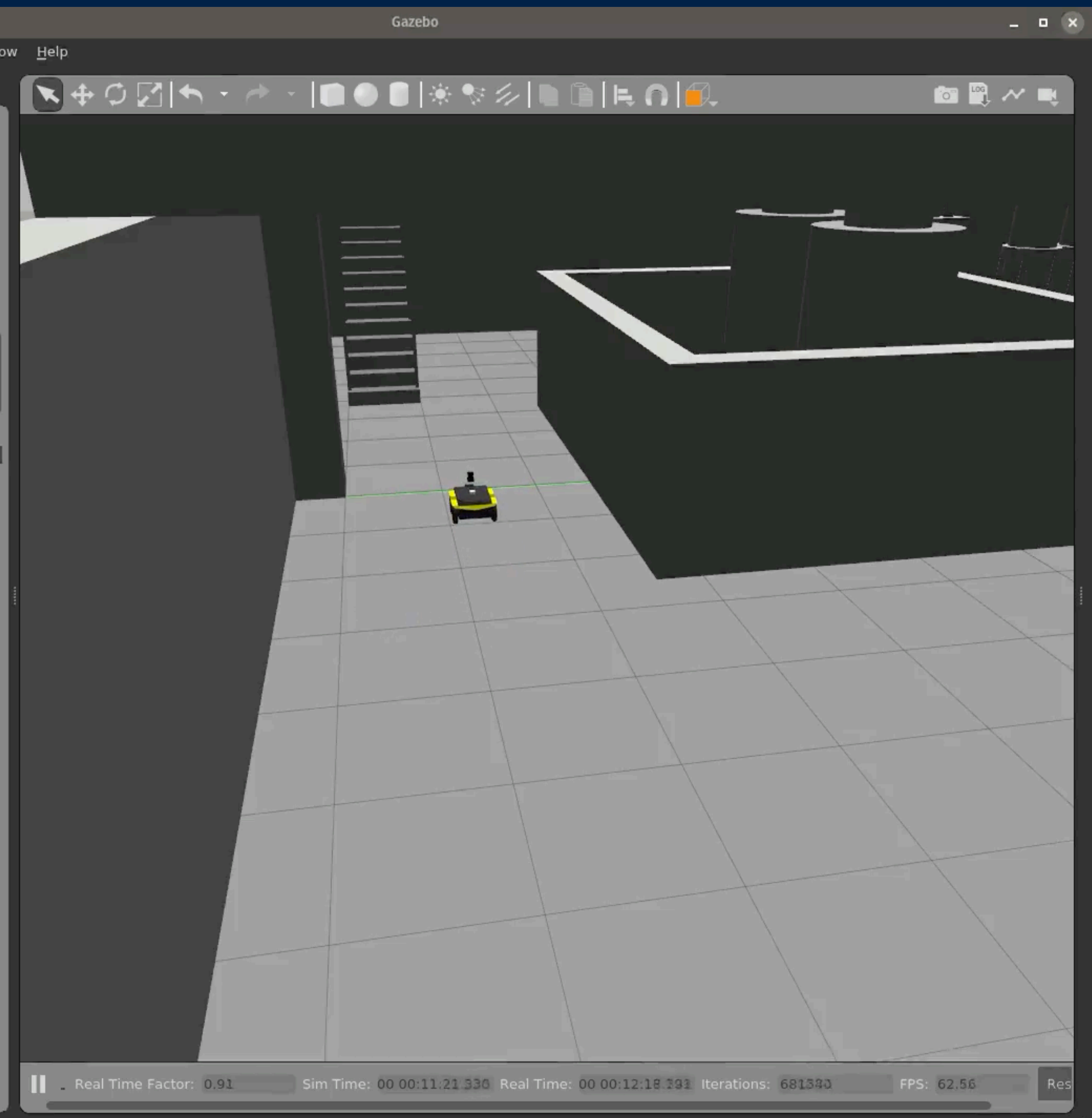
G !WayPoint4 & G !WayPoint51 & G !WayPoint26

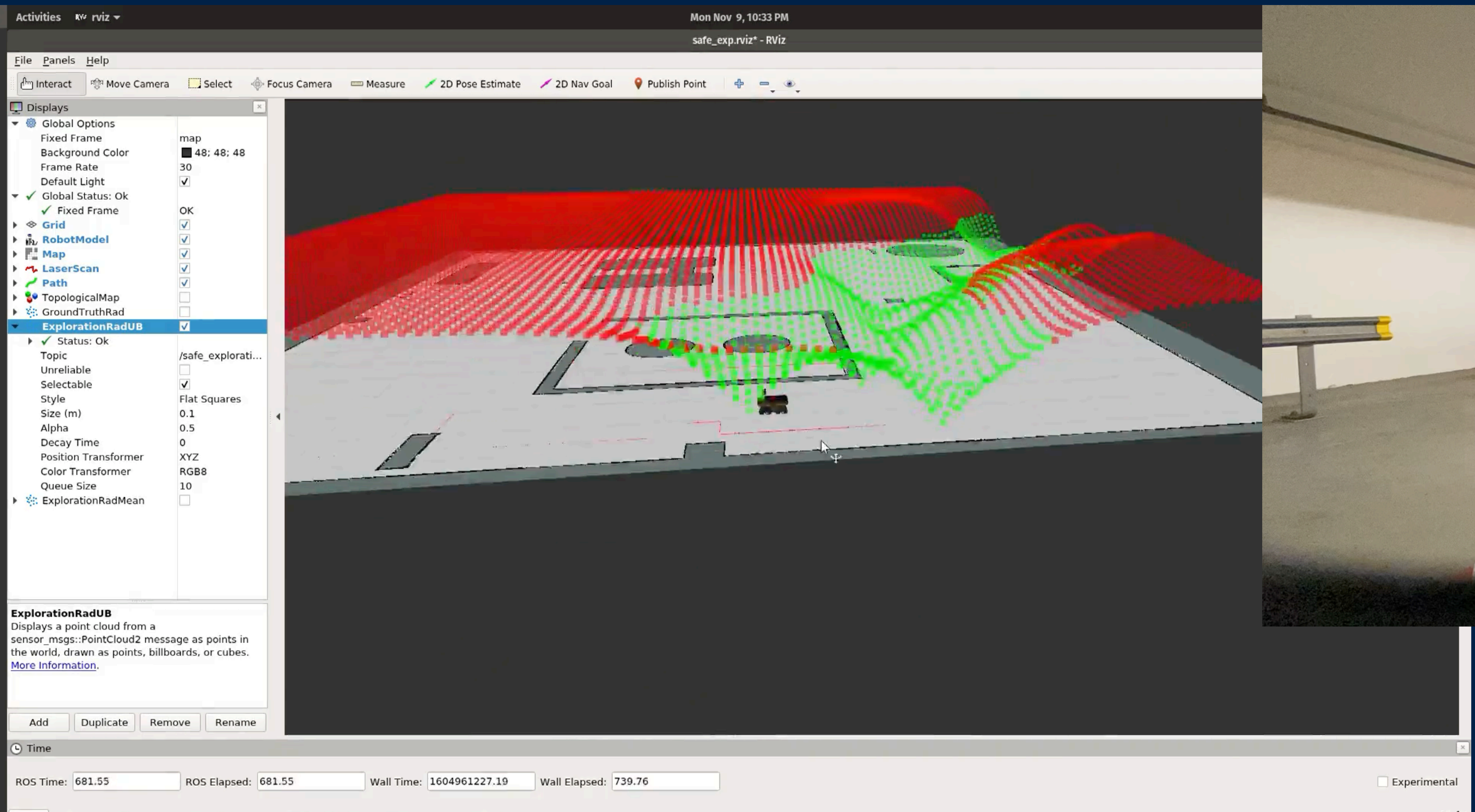
GOAL-ORIENTED
AUTONOMOUS
LONG-LIVED SYSTEMS
OXFORD ROBOTICS INSTITUTE

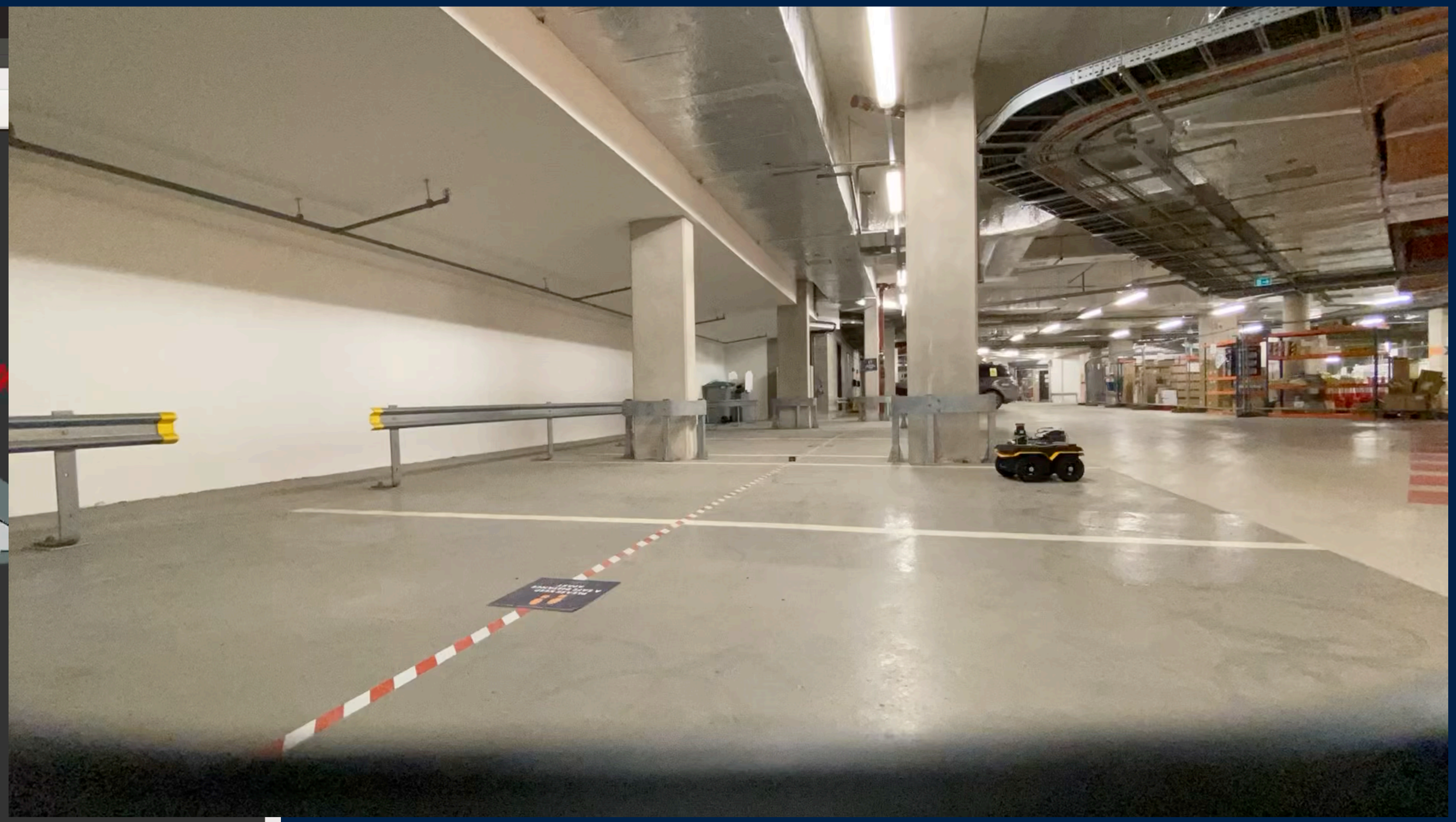
GOAL-ORIENTED AUTONOMOUS LONG-LIVED SYSTEMS
OXFORD ROBOTICS INSTITUTE

Planning with models acquired online or through learning

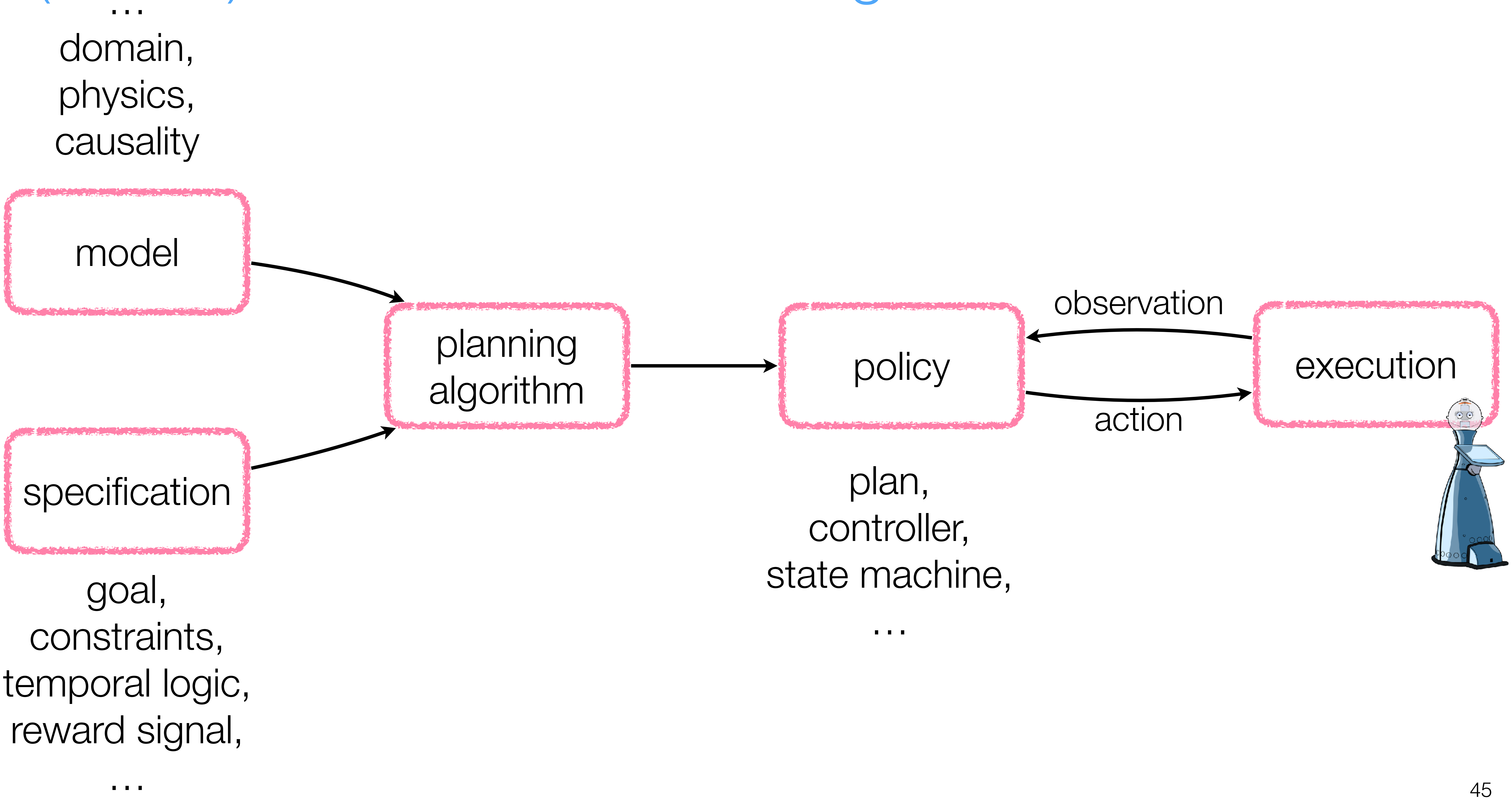




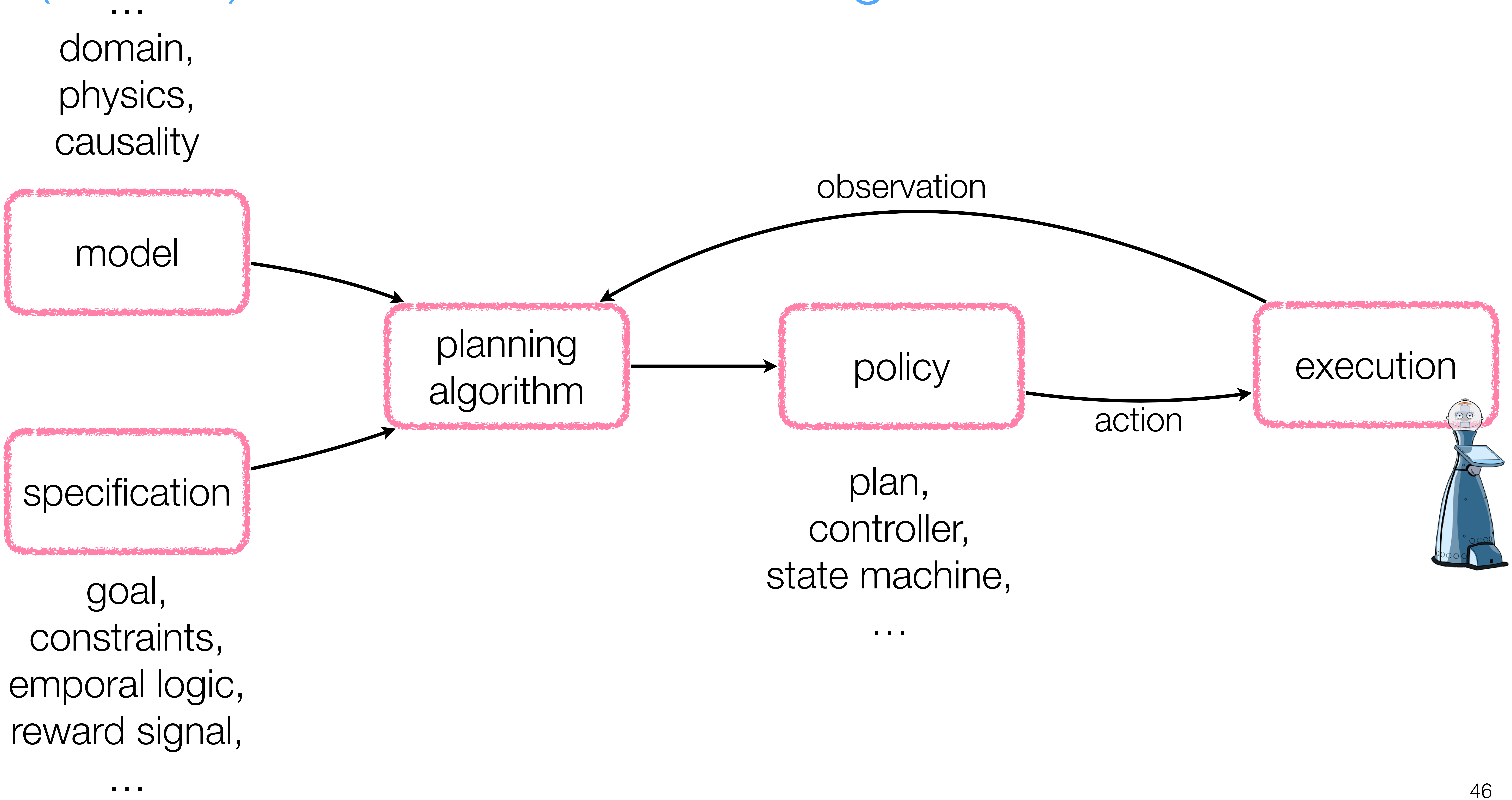




(Offline) Robot Mission Planning



(Online) Robot Mission Planning



Position Statement

Successful long-term robotic autonomy requires:

1. Data-driven model learning

2. Modelling and planning approaches that explicitly reason about the epistemic uncertainty inherent to models learnt from data

3. Incorporating rich specifications that go beyond typical reward maximisation in expectation

Position Statement

Successful long-term robotic autonomy requires:

1. Data-driven model learning

2. Modelling and planning approaches that explicitly reason about the epistemic uncertainty inherent to models learnt from data

3. Incorporating rich specifications that go beyond typical reward maximisation in expectation

Using data to populate MDPs

Long-Term Autonomy

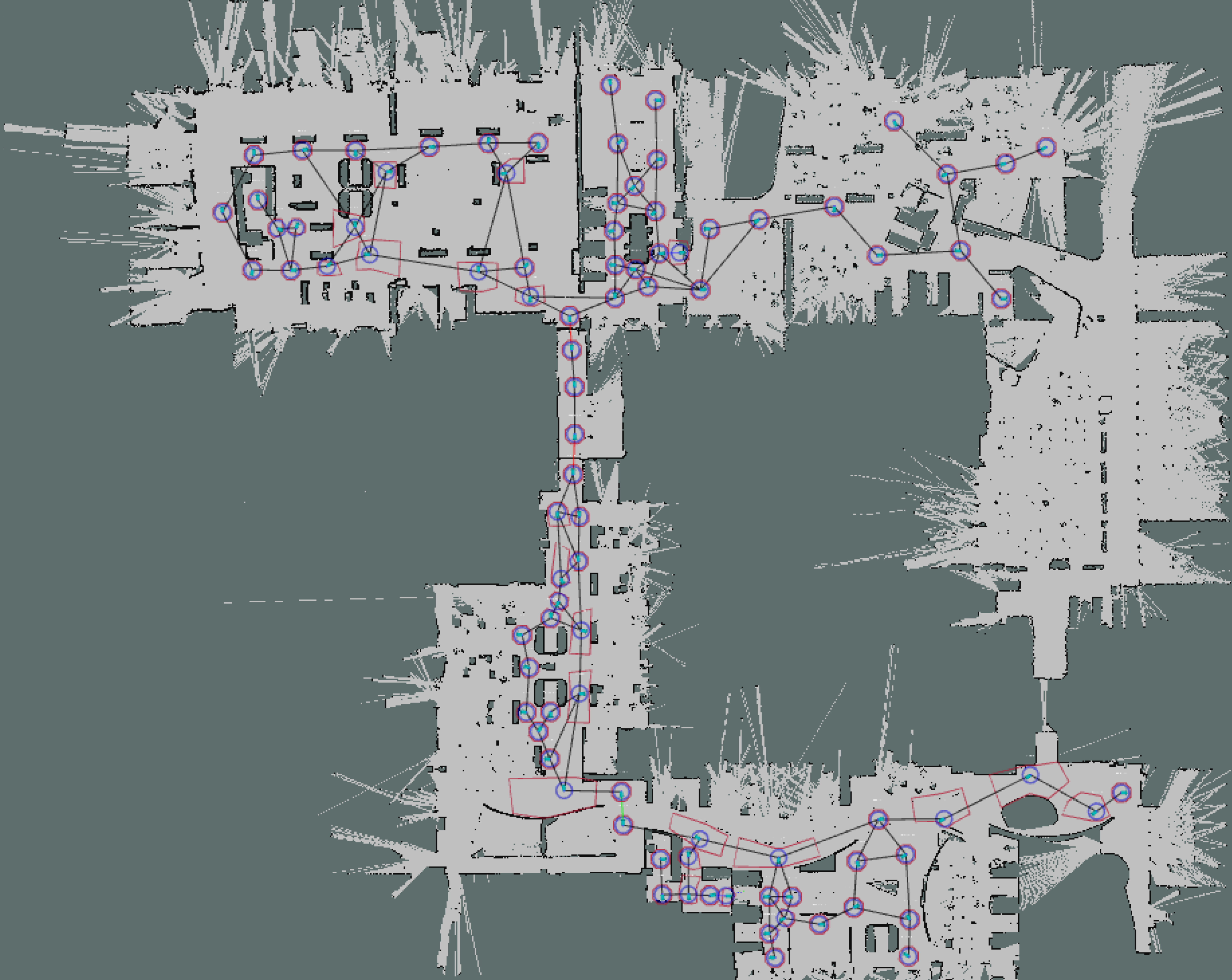
- Robots are deployed for months of unsupervised autonomous behaviour in real end-user environments
- Long- and short-term variation in tasks, resources and environments requires planning

TSC, Milton Keynes, UK

Haus der Barmherzigkeit, Vienna, Austria

G4S Security, Tewkesbury, UK

N. Hawes et al. “The STRANDS project: Long-term autonomy in everyday environments”. IEEE Robotics & Automation Magazine, 24(3), 2017.



Transport Systems Catapult, Milton Keynes, UK

object search
activity learning
docking to charge
door passing


fail!
Food Waste
fail!
SCITOS G5
planning must take into account the uncertainty associated with (at least) success and duration
uncertain durations

Markov decision Processes

$$\mathcal{M} = \langle S, A, T, C \rangle$$

states

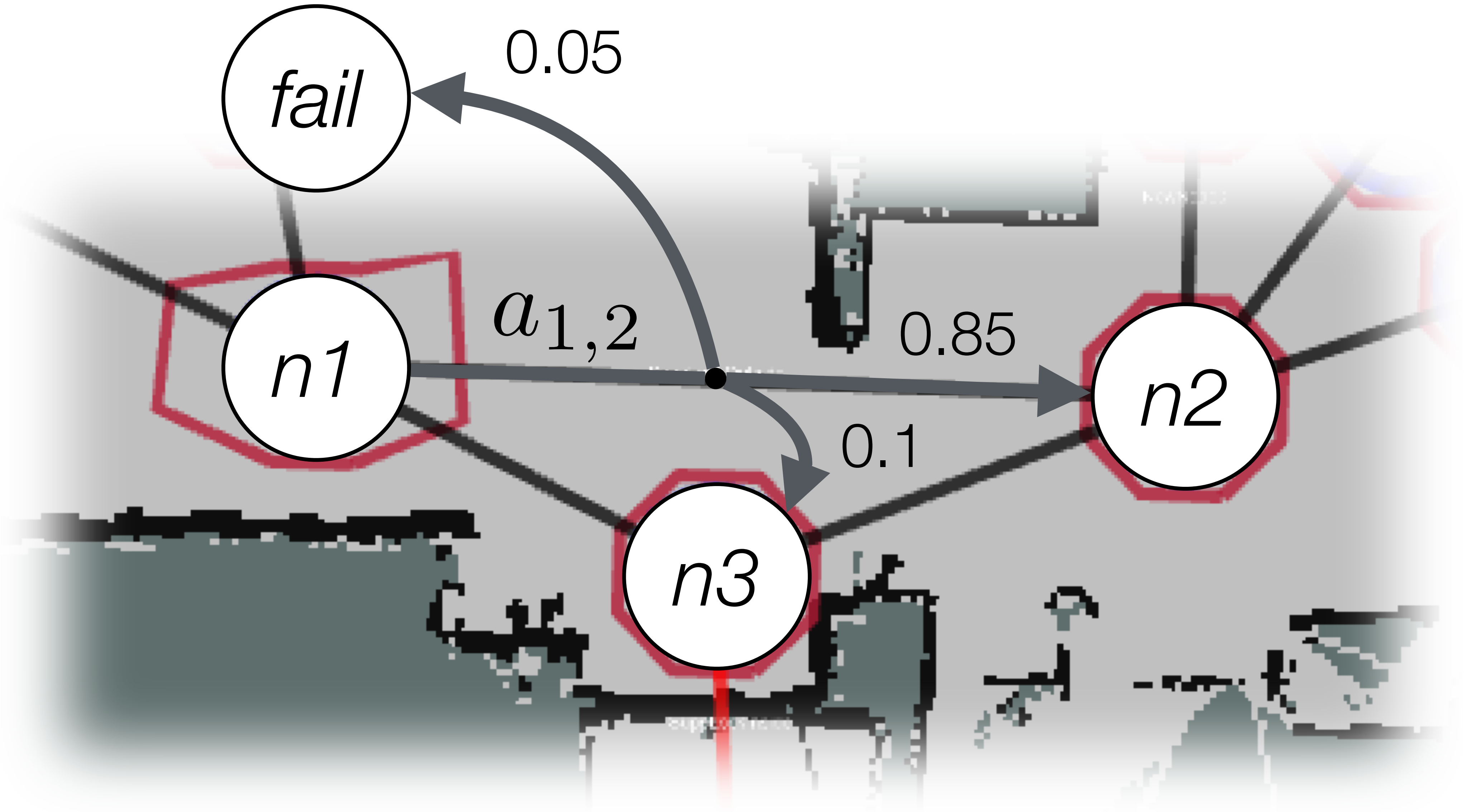
actions

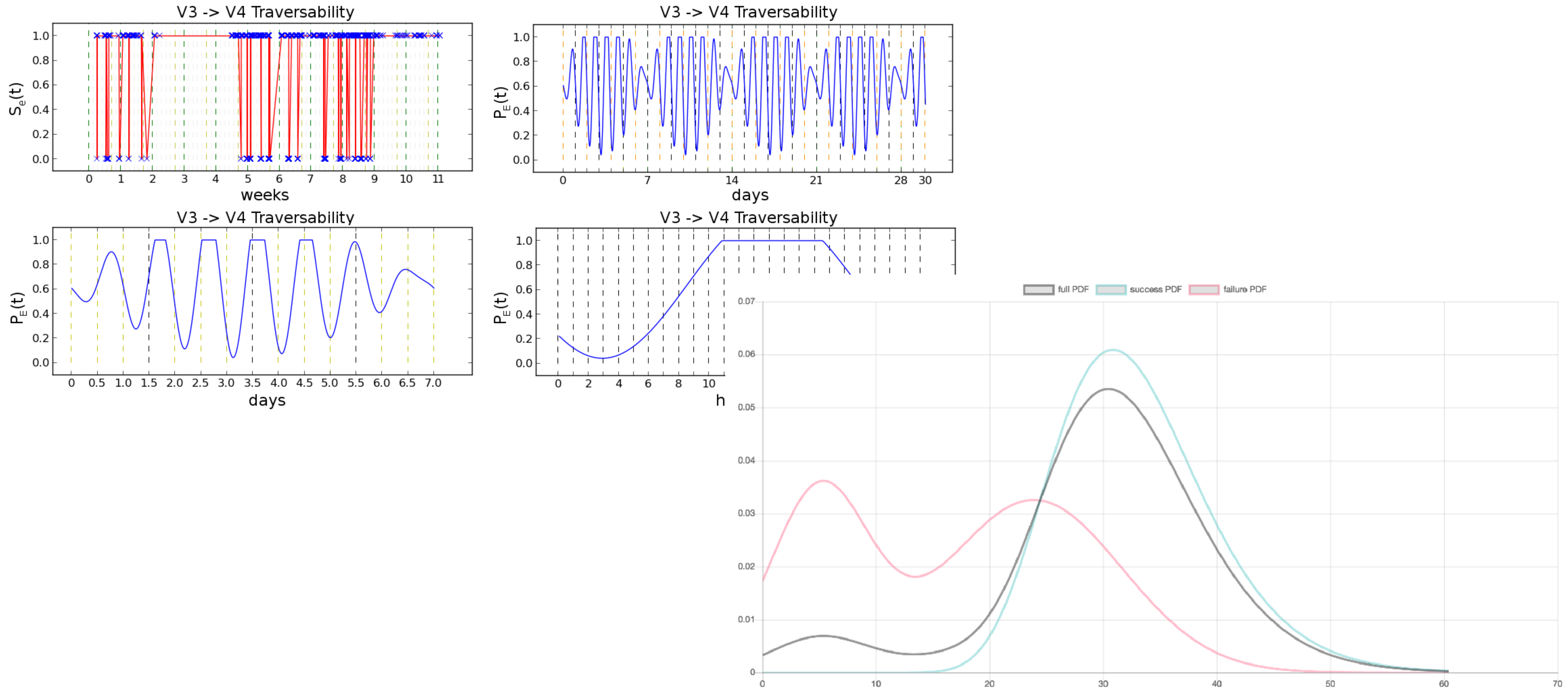
transition probs

action costs

defined by the map
and task

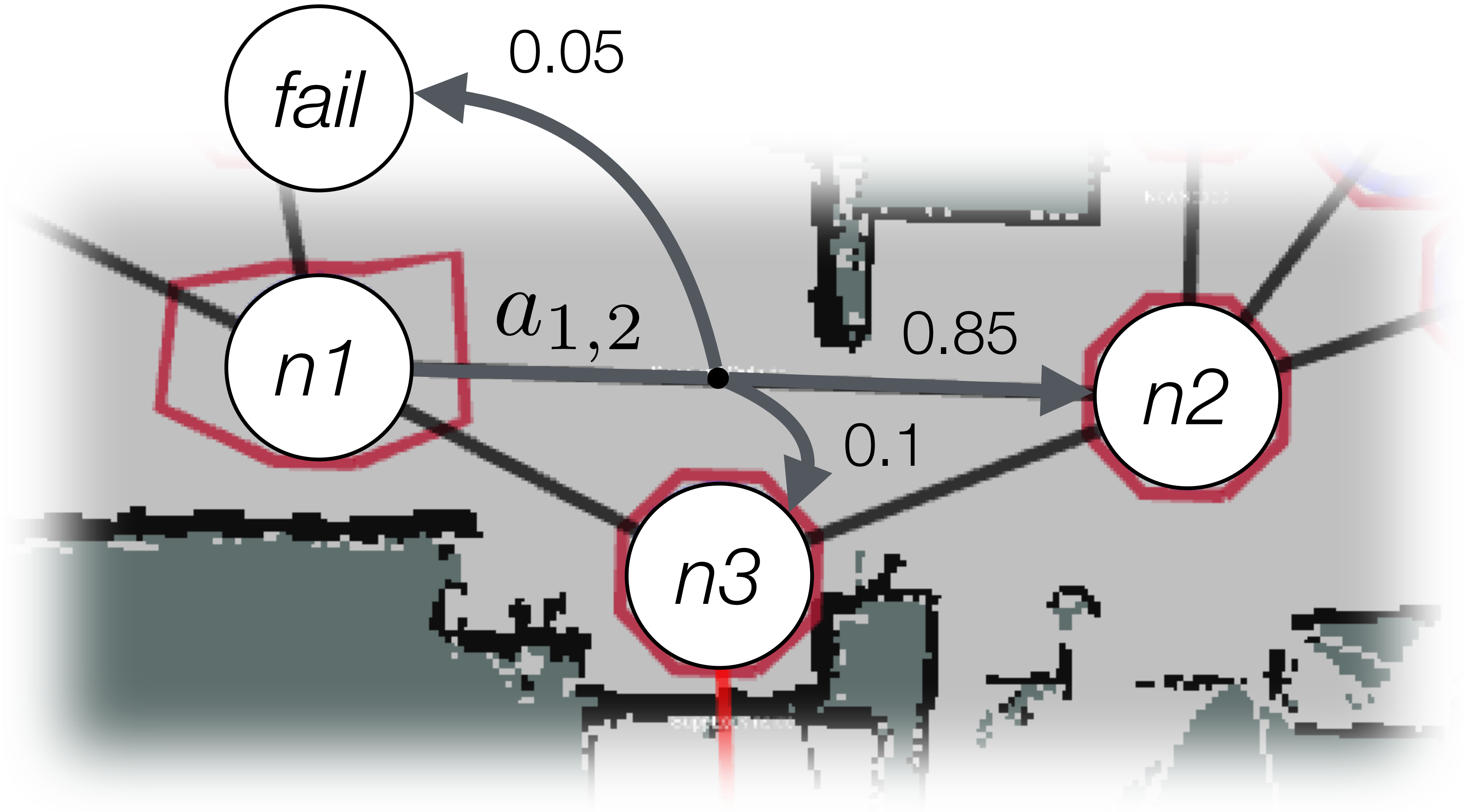
learnt from long-term
experience data





B. Lacerda, F. Faruq, D. Parker, and N. Hawes. “Probabilistic planning with formal performance guarantees for mobile service robots”. The International Journal of Robotics Research, 38(9), 2019.

Problem Specification - Partial Satisfiability

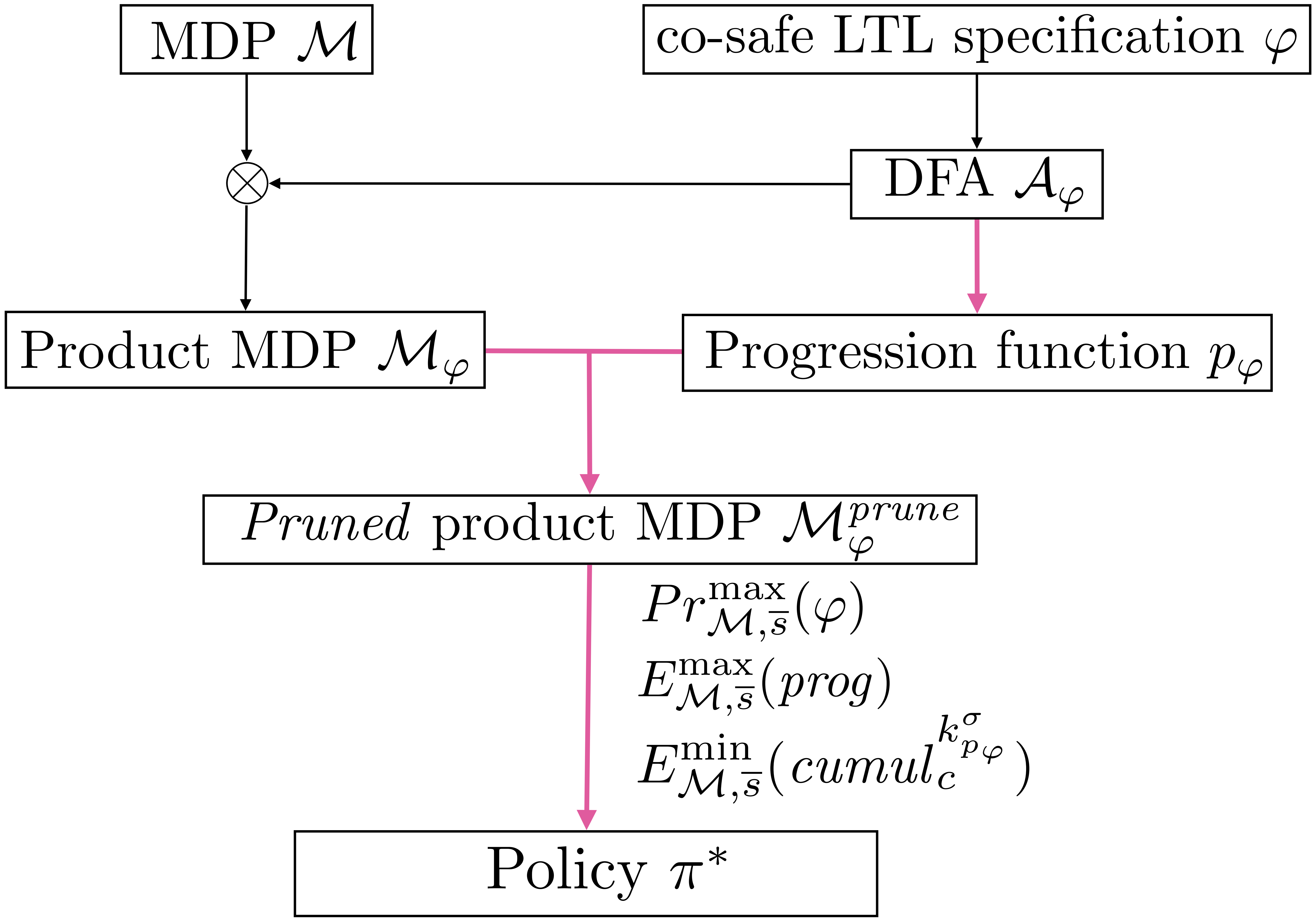


\+ co-safe spec φ

1. **Be robust:** Maximise probability of visiting a sequence of states that satisfies the spec
2. **Do as much as possible:** Even when the overall spec becomes unachievable (e.g., because of a task that is to be executed behind a closed door), continue executing and achieve as much of the spec as possible
3. **Be efficient:** Minimise expected time to execute the part of the task that is possible

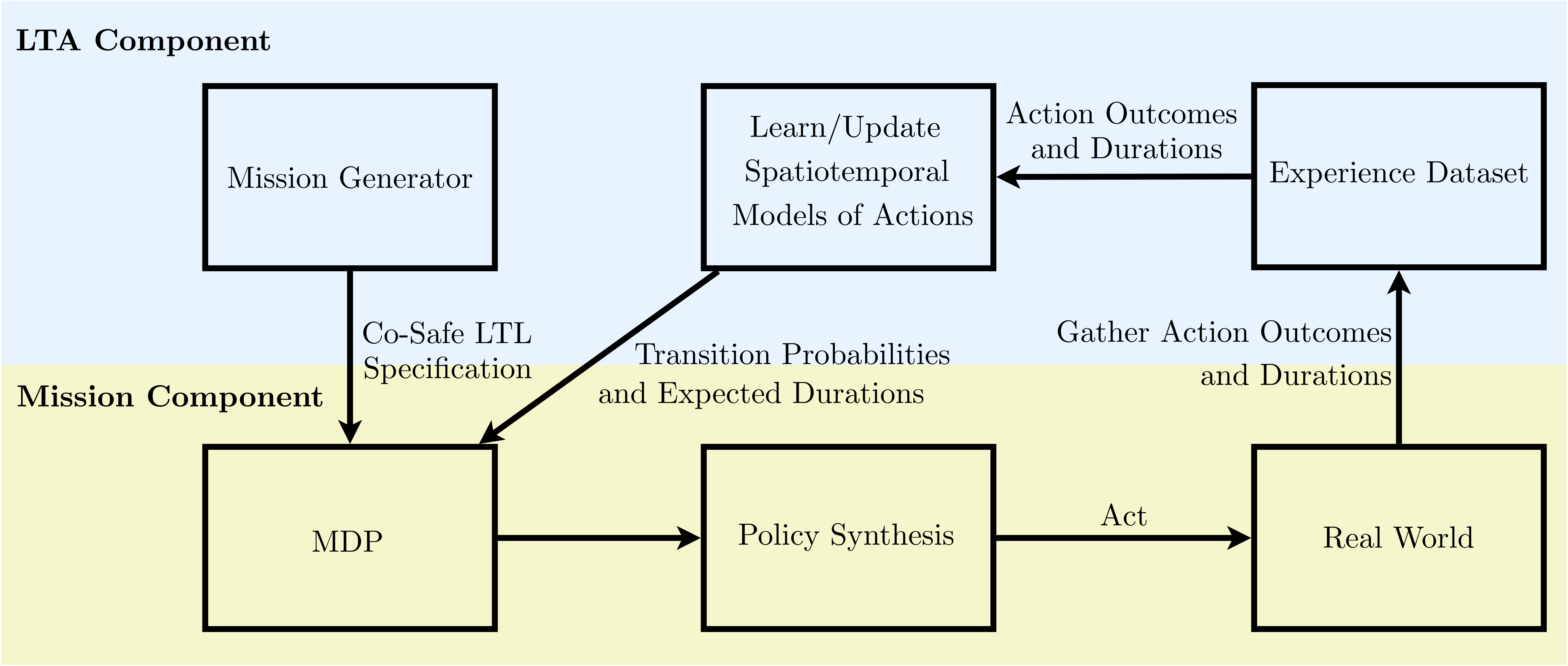
B. Lacerda, F. Faruq, D. Parker, and N. Hawes. “Probabilistic planning with formal performance guarantees for mobile service robots”. The International Journal of Robotics Research, 38(9), 2019.

Solution Diagram



B. Lacerda, F. Faruq, D. Parker, and N. Hawes. "Probabilistic planning with formal performance guarantees for mobile service robots". The International Journal of Robotics Research, 38(9), 2019.

Long-lived Mission Planning and Learning

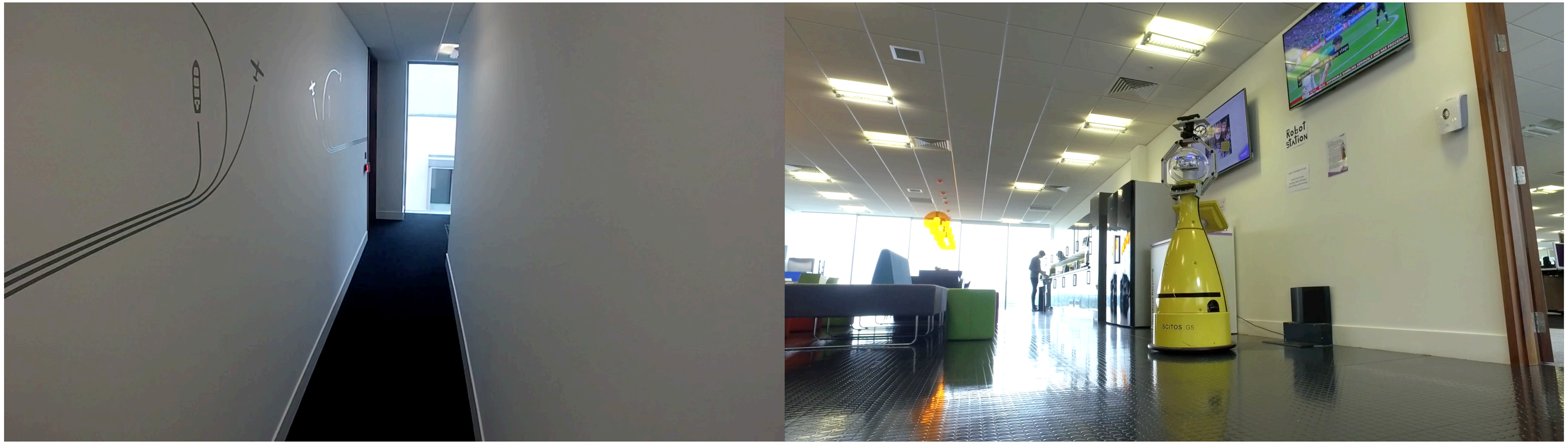


- **Data:** Action outcomes and durations
- **Model:** MDP
- **Specification:** Partially satisfiable co-safe LTL (lexicographic optimisation)

B. Lacerda, F. Faruq, D. Parker, and N. Hawes. "Probabilistic planning with formal performance guarantees for mobile service robots". The International Journal of Robotics Research, 38(9), 2019.

Long-lived Mission Planning and Learning

- This approach has generated months of long-term behaviour
- Execution framework run for ~1 year, handling >23,000 tasks
- Evaluating the policy guarantees and effects of long-term adaptation is harder (and dependent on learning mechanisms, environment, people etc.)



Position Statement

Successful long-term robotic autonomy requires:

1. Data-driven model learning

2. Modelling and planning approaches that explicitly reason about the epistemic uncertainty inherent to models learnt from data

3. Incorporating rich specifications that go beyond typical reward maximisation in expectation

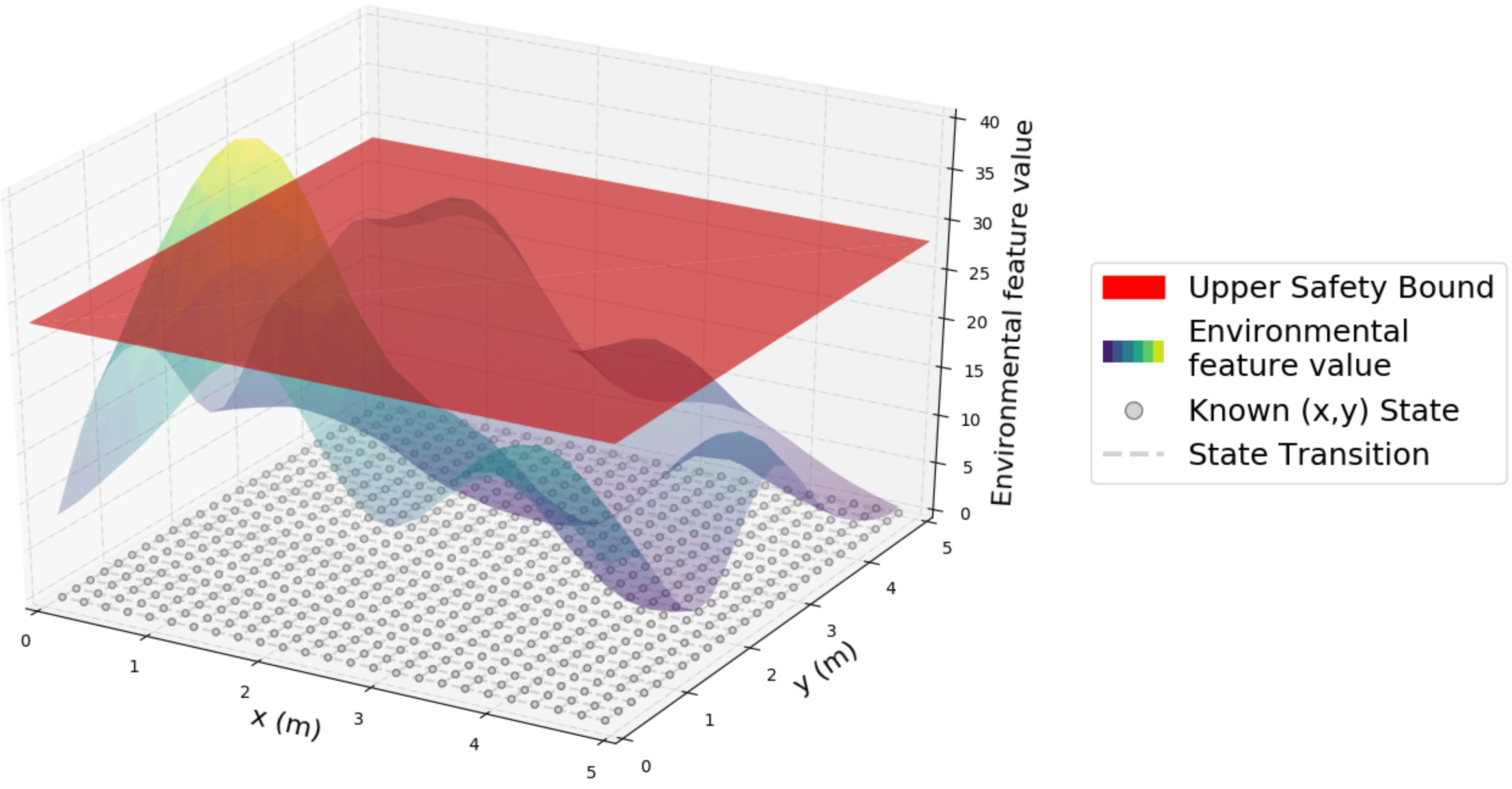
Explicitly modelling model uncertainty

Safe Exploration Overview

- Robot exploration with safety constraints over an environmental feature whose distribution is unknown a priori
  - Explore the environment whilst maintaining the level of radiation exposure under a bound
- We present a novel decision making under uncertainty model and show how it can be used for efficient exploration
  - **Markov decision processes + Gaussian processes**

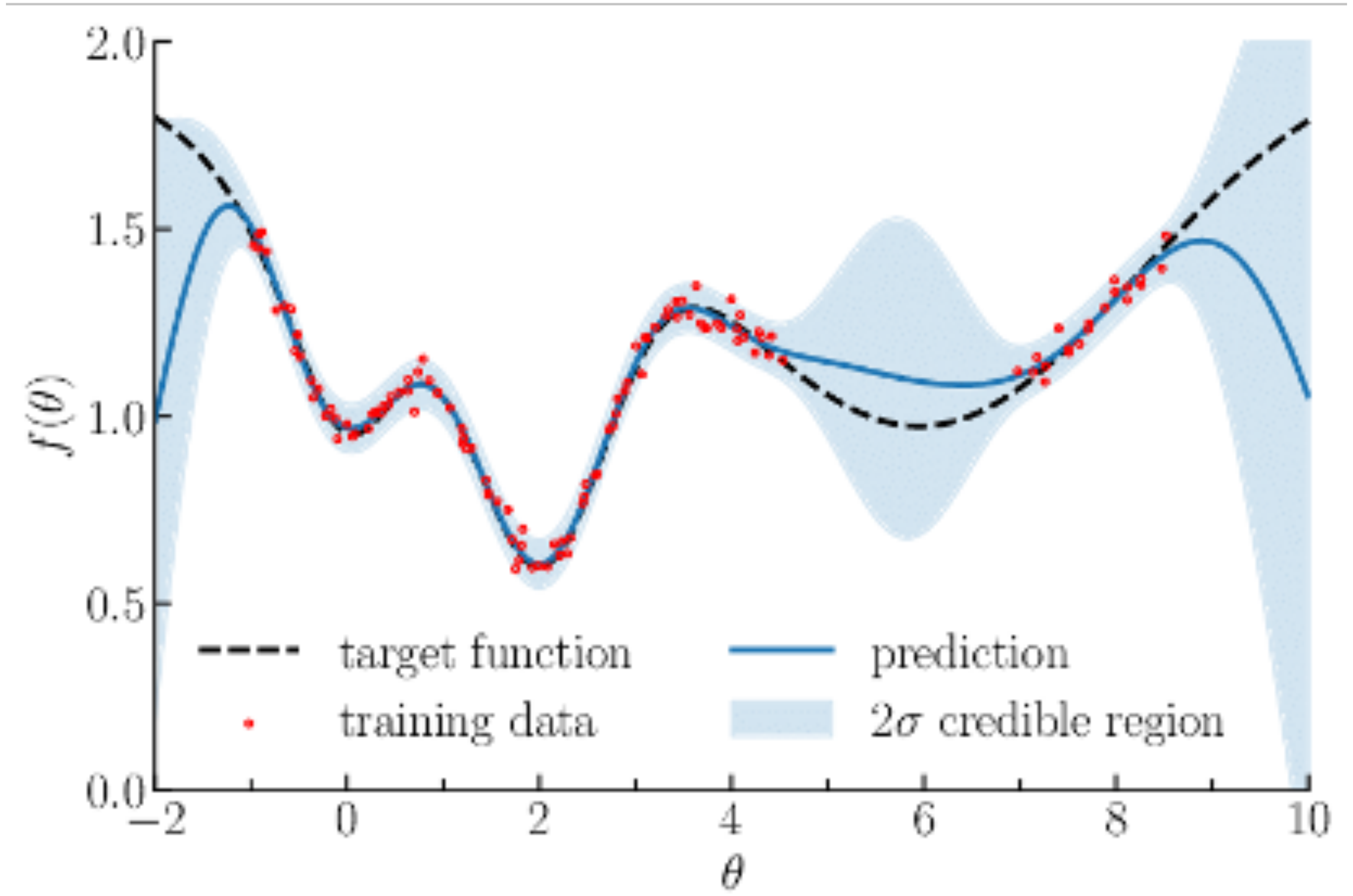
Problem Setup

- Underlying (known) MDP for navigation
- A priori unknown radiation - can be sensed at each location
- Bound on max radiation exposure at each location
- **Goal:** Estimate radiation across the whole environment whilst avoiding going over bound



Gaussian Processes

- Collection of random variables, any finite number of which have a joint Gaussian distribution
- Model is updated taking noisy observations at different locations
- Allows for prediction at unobserved locations



MDPs with Unknown Feature Values

$$S^O = V \times O$$

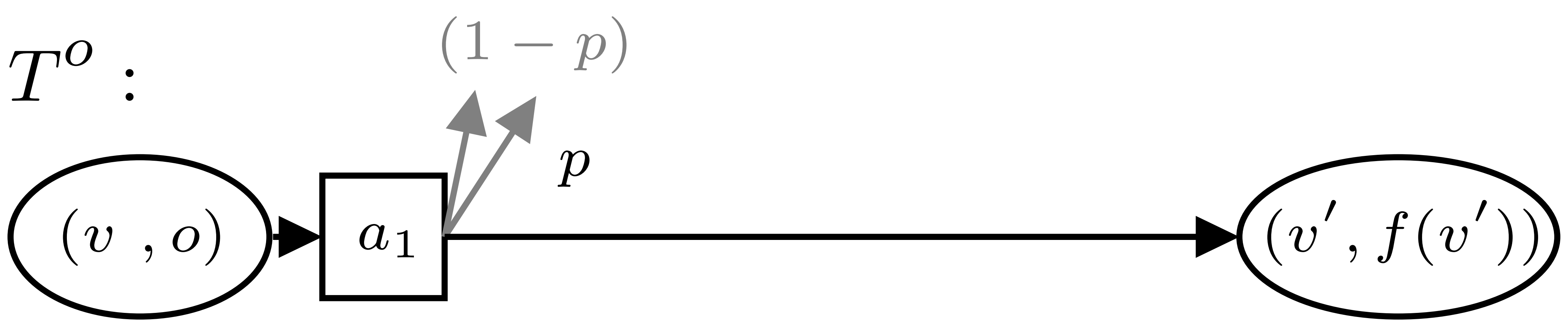
$$\downarrow$$

$$s^O = (v, f(v))$$

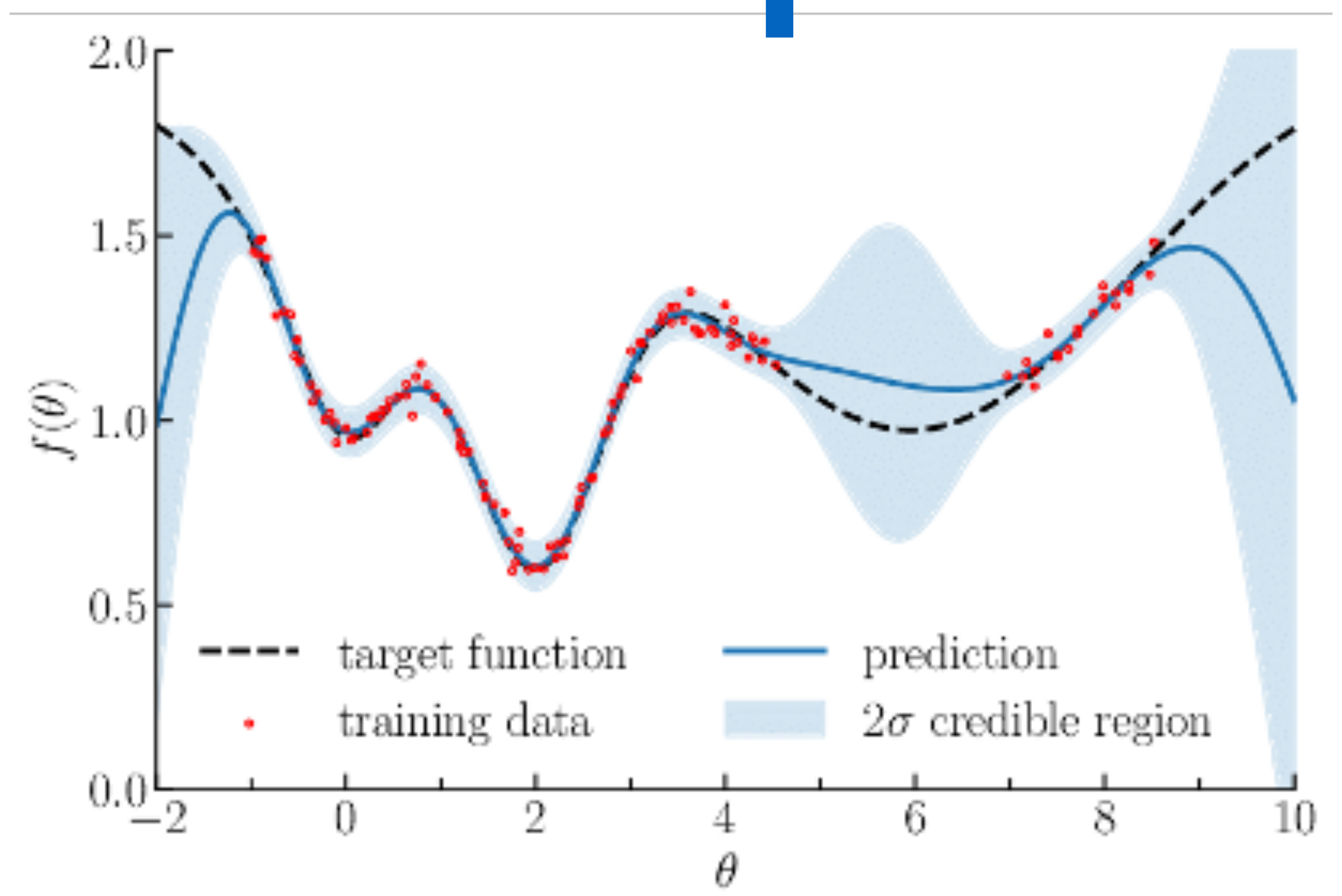
Radiation level function f is unknown a priori and will be approximated by a GP

$$T^{\mathcal{D}}((v, I), (v, v_g), (v', I')) = T^O((v, I), (v, v_g), v') P^{\mathcal{GP}}(f(v') \in I' \mid \mathcal{D})$$

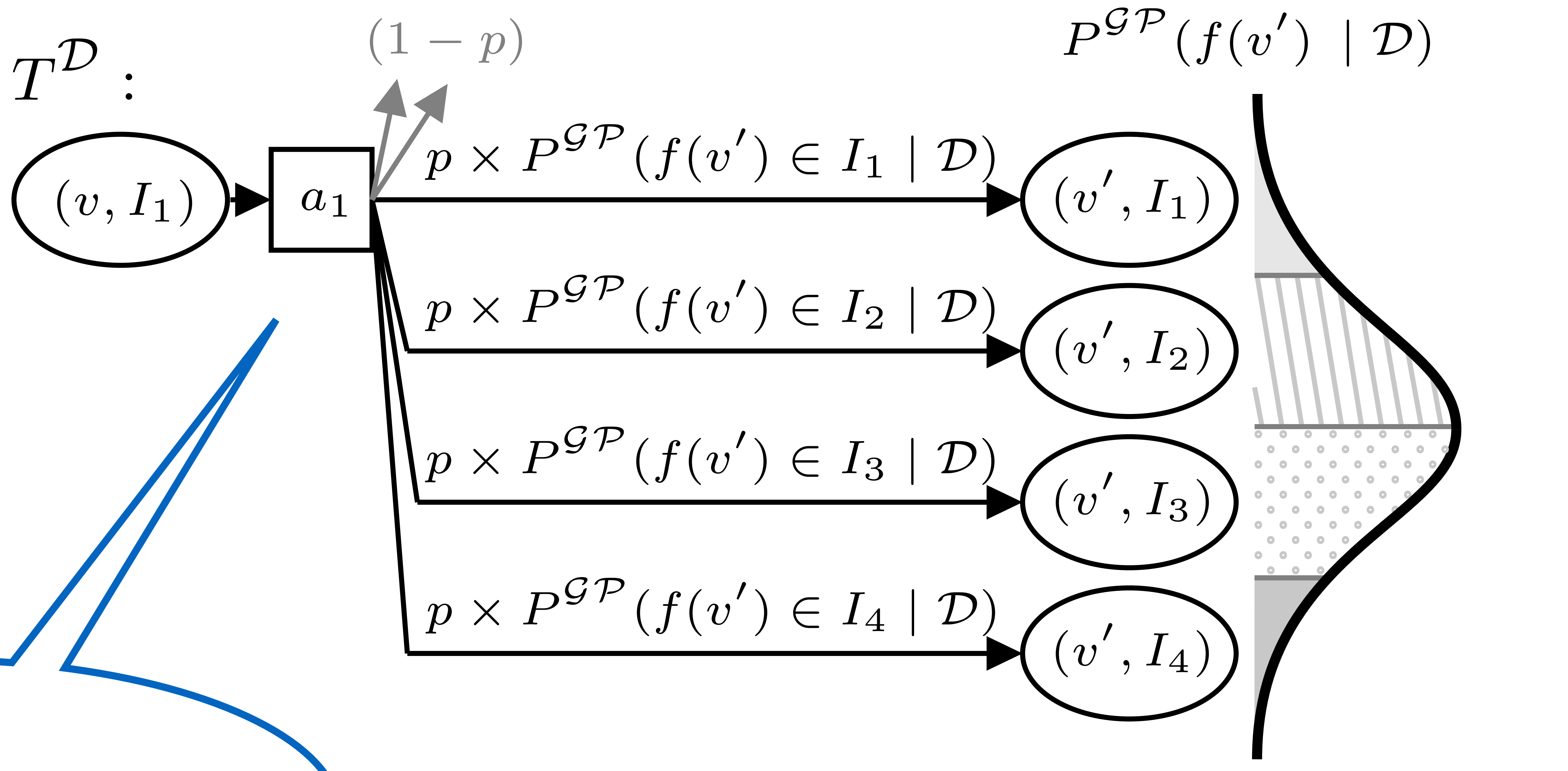
$$T^O : (V \times O) \times A^O \times V \rightarrow [0, 1]$$



+

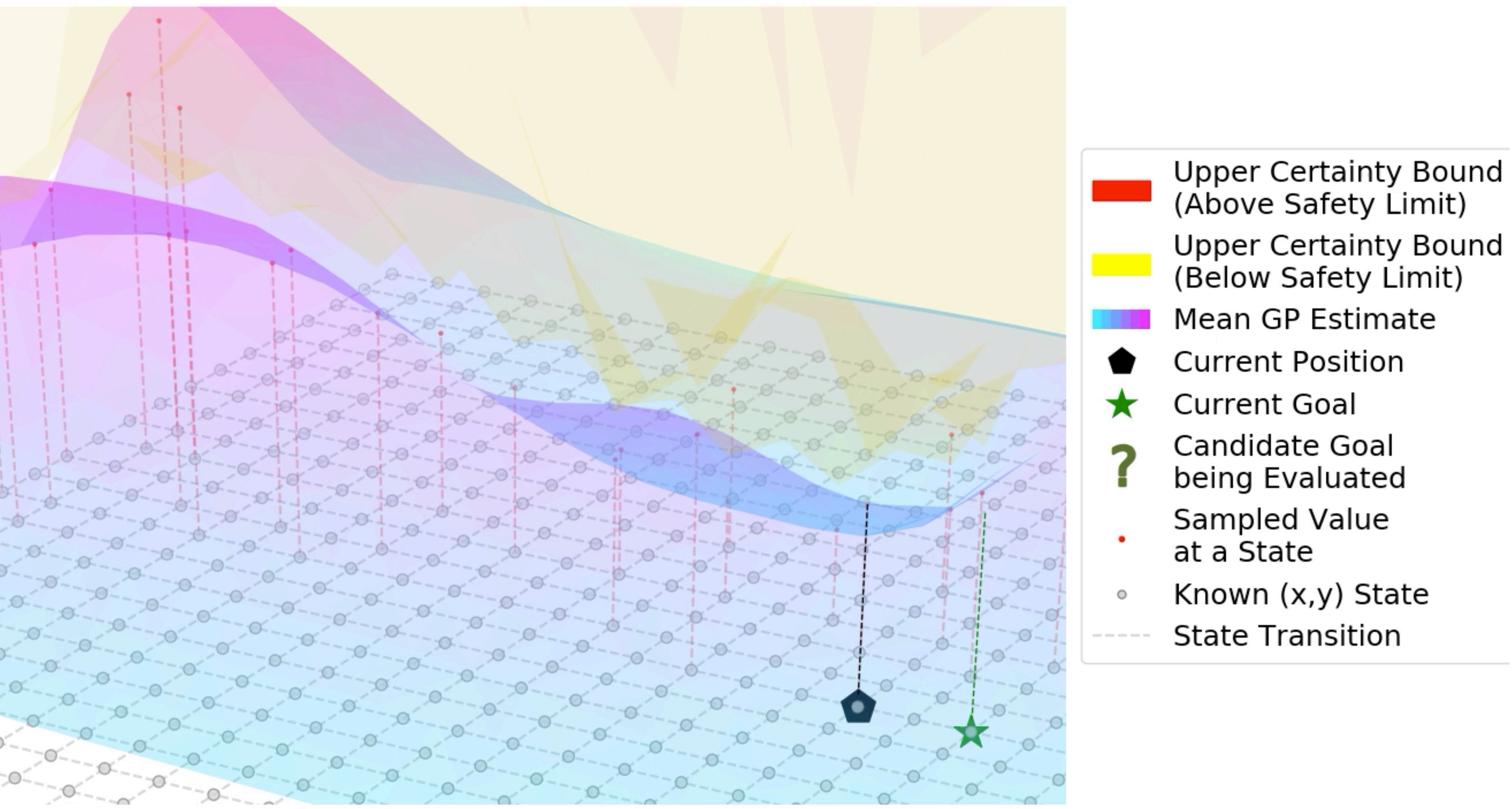


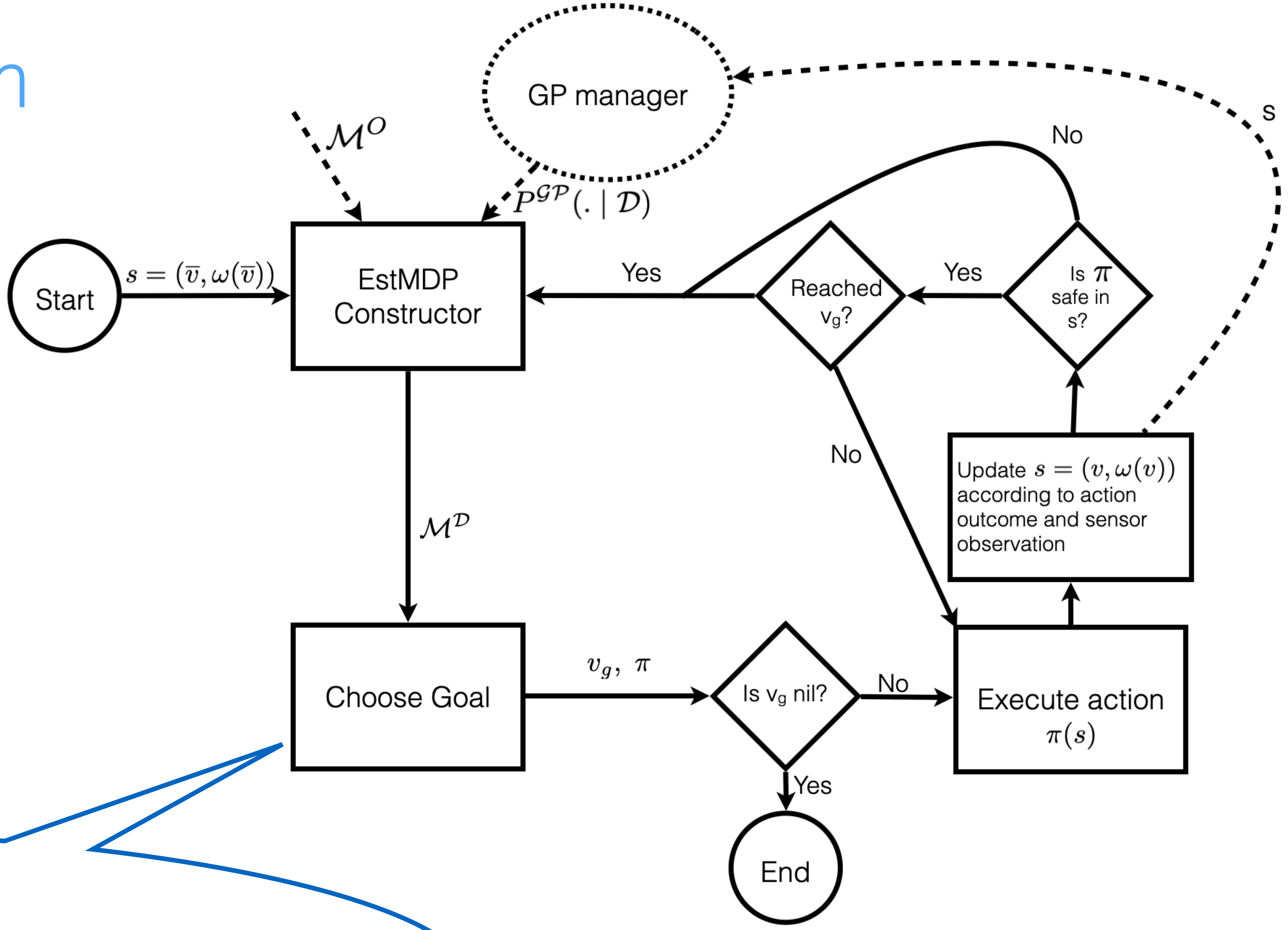
=



Estimated MDP

Exploration Algorithm



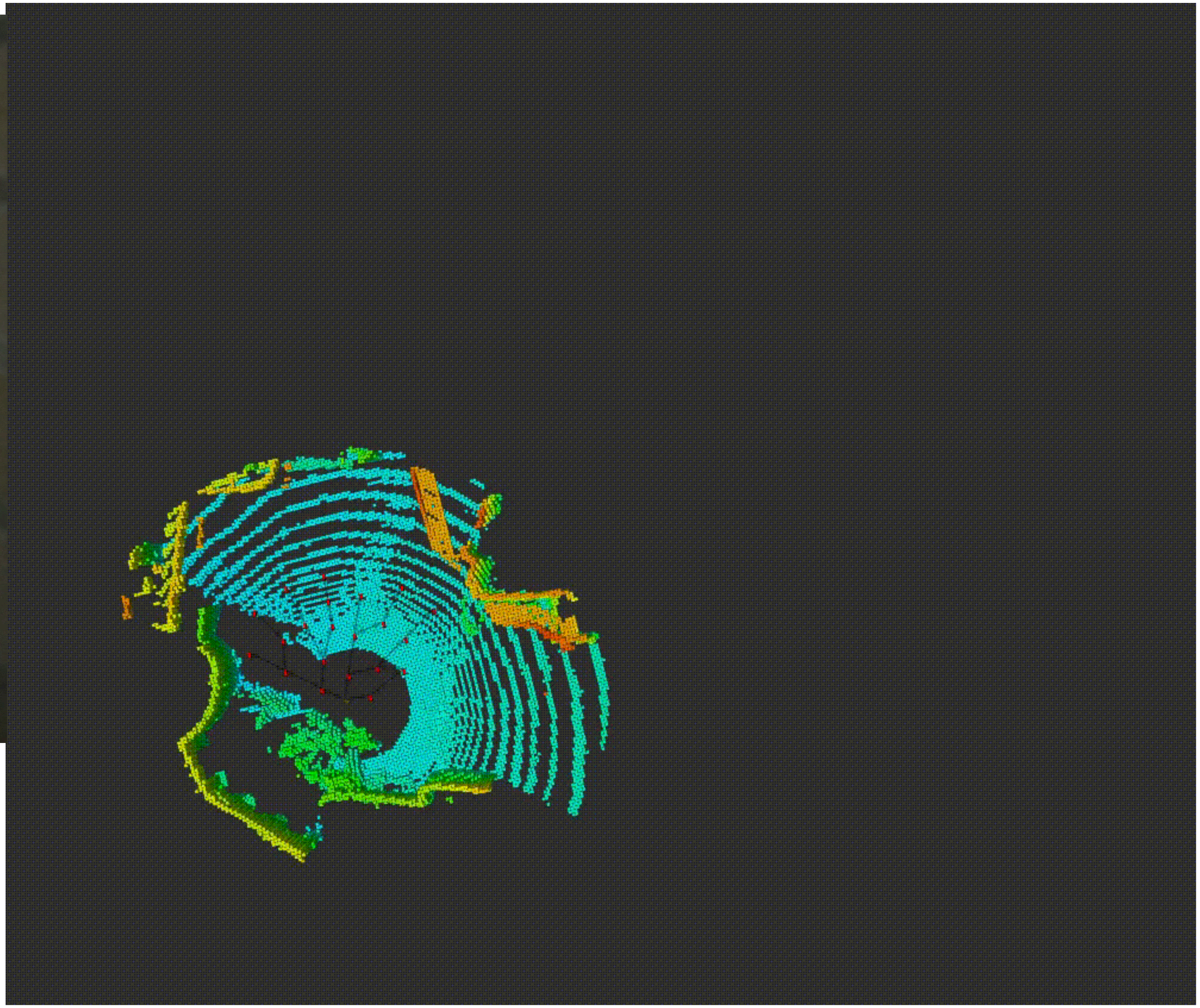


Maximise score based on:
1. Probability of safely reaching state (reach-avoid as partial satisfiability)
2. Expected time to reach state (reach-avoid as partial satisfiability)
3. How uncertain state is (GP variance)

M. Budd, B. Lacerda, P. Duckworth, A. West, B. Lennox, and N. Hawes, “Markov Decision Processes with Unknown State Feature Values for Safe Exploration using Gaussian Processes,” in IROS, 2020.

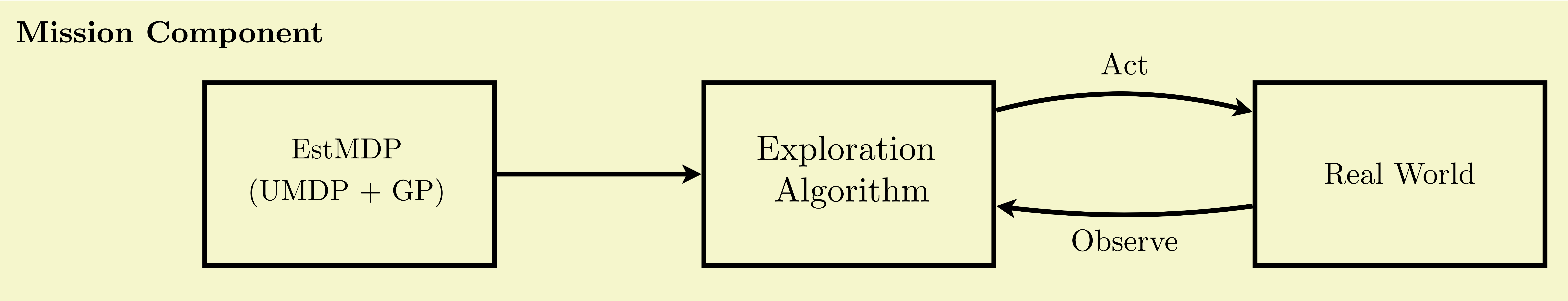
Unknown Map

Corsham Research Mine, Wiltshire, UK.



A. Stephens, M. Budd, M. Staniaszek, B. Casseau, P. Duckworth, M. Fallon, N. Hawes and B. Lacerda, “Decision-Making Under Uncertainty for Safe Exploration and Mapping,” under review at Autonomous Robots.

Safe Exploration

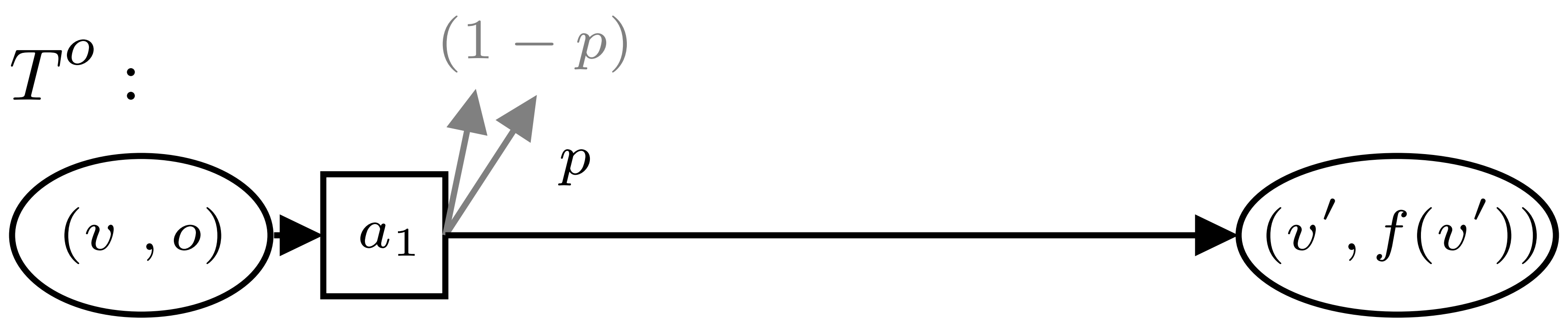


- **Data:** Online observations of unknown function
- **Model:** MDP + GP
- **Specification:** Safe exploration, sequence of reach-avoid problems

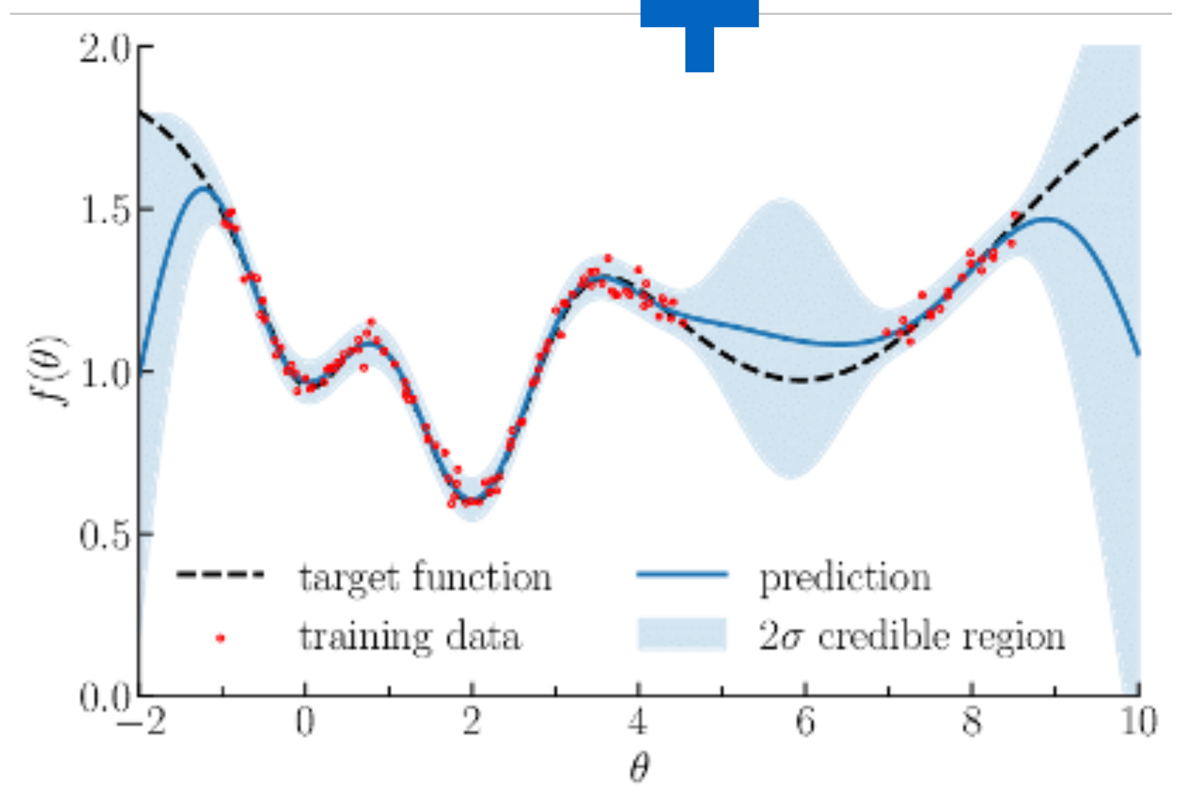
Mission Planning under Unknown Conditions

- UMDP + GP = BAMDP
  - The GP is encoding our belief over which is the true transition function
  - We can use BAMCP for planning in unknown environments with GP predictions

$$T^O \quad : \quad (\mathrm{V} \times O) \times A^O \times \mathrm{V} \rightarrow [0,1]$$



+



=

$$\text{BAMDP} \;\; \mathcal{M}^+ = \langle S^+, s_0^+, A, T^+, C^+, G^+ \rangle$$

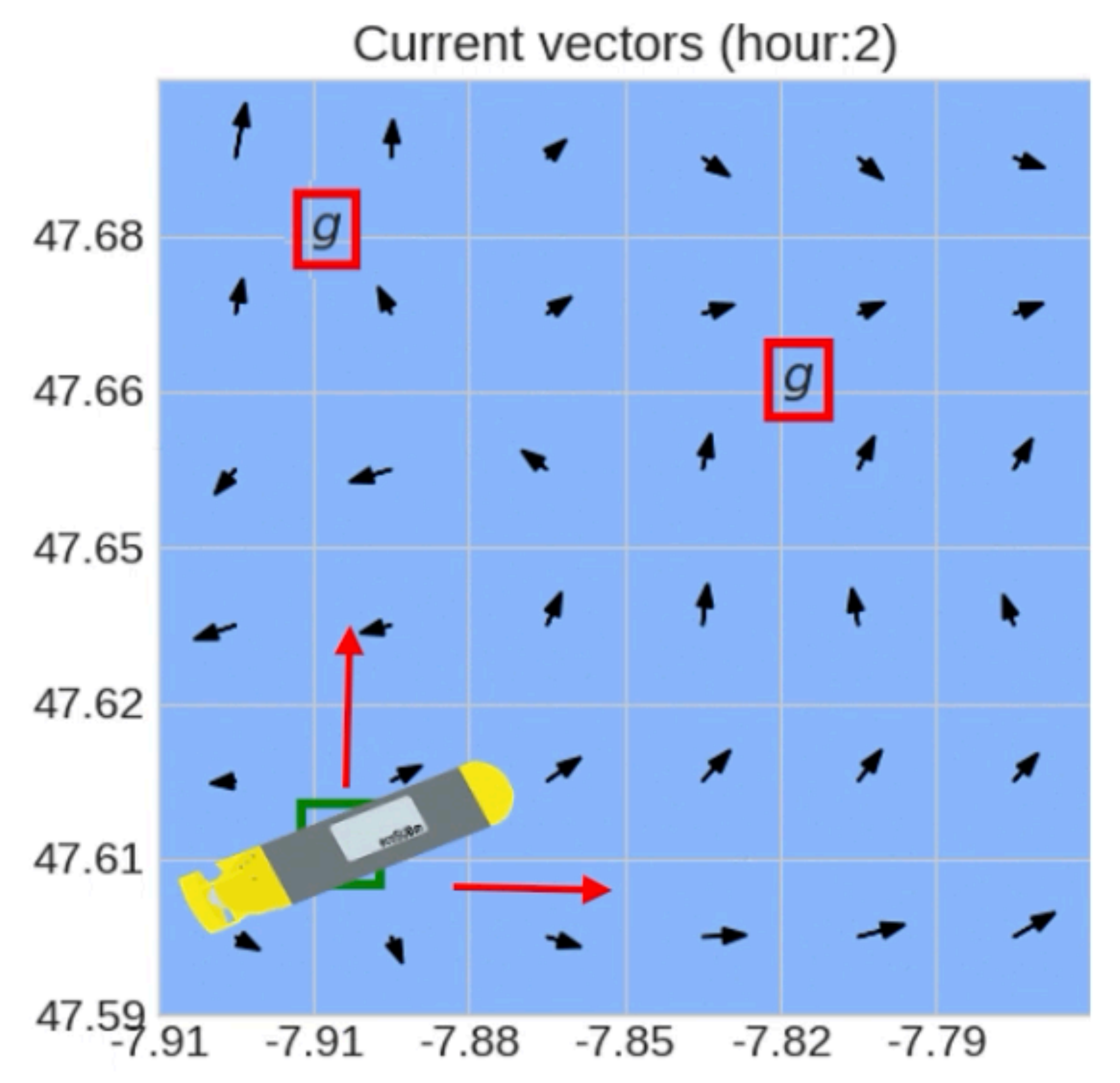
For $s = (v, o)$ and $s' = (v', o')$:

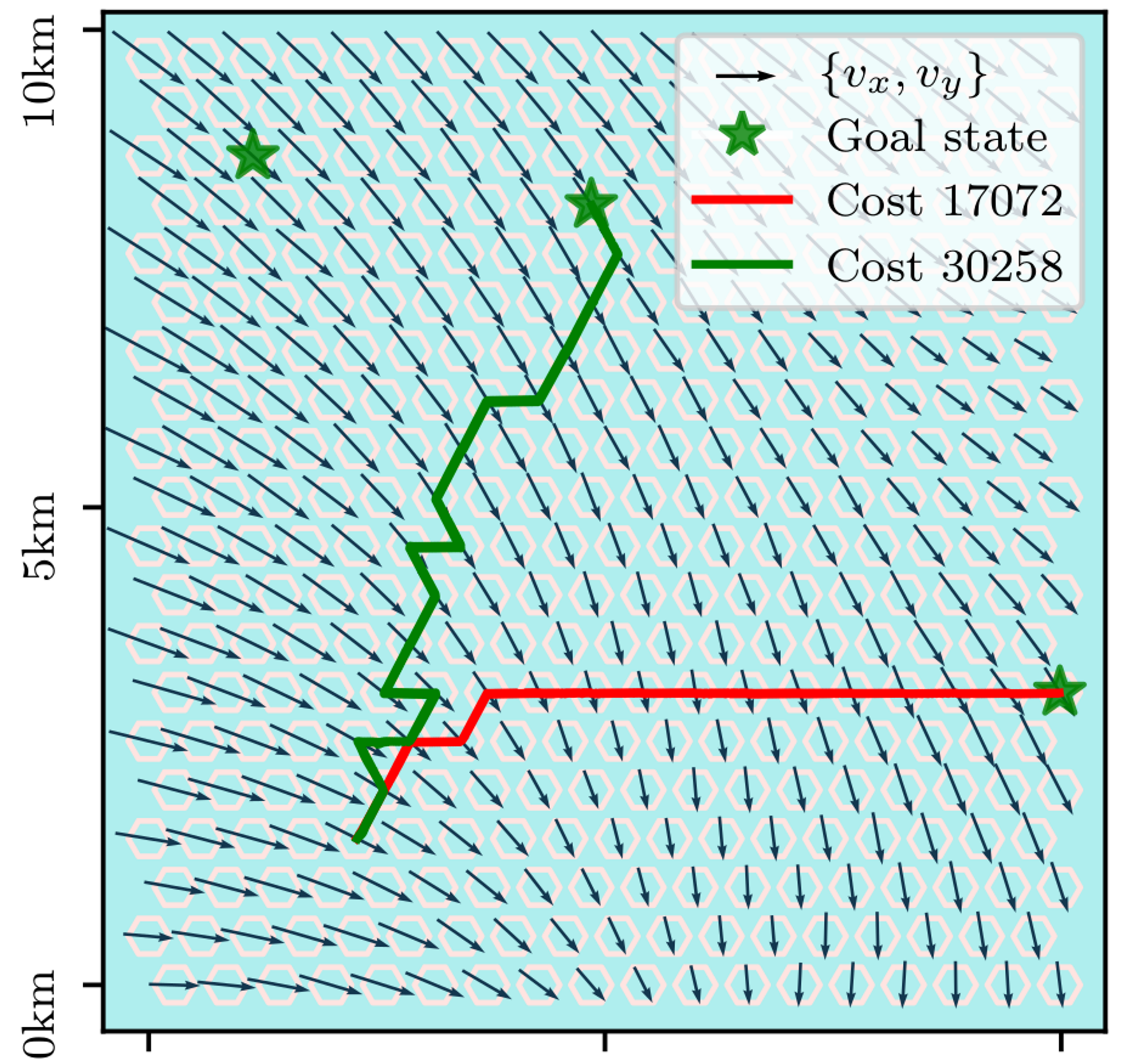
$$T^+((s,h), a, (s', has')) = \int_T T(s,a,s') p(T \mid h) dT$$

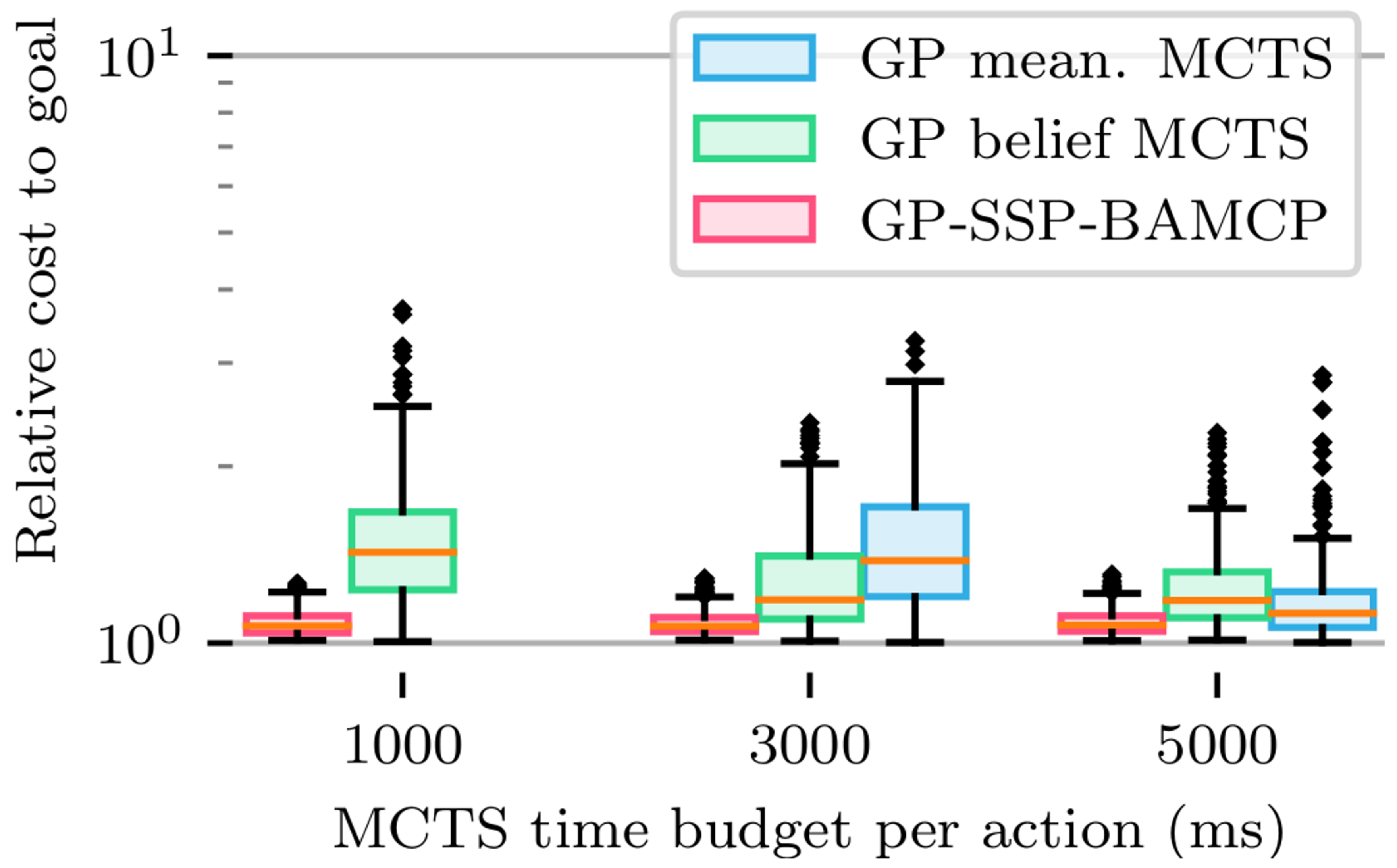
$$= T^O((v,o), a, v') P^{\mathcal{GP}}(f(v') = o' \mid \mathcal{D}_h)$$

Matthew Budd, Paul Duckworth, Nick Hawes, and Bruno Lacerda. “Bayesian Reinforcement Learning for Single-Episode Missions in Partially Unknown Environments”. In CoRL 2022.

Mission Planning under Unknown Conditions

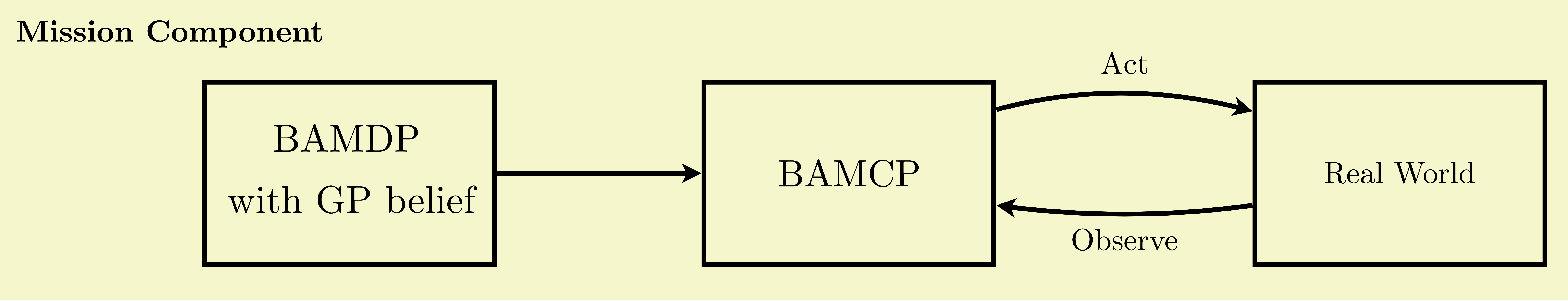






Matthew Budd, Paul Duckworth, Nick Hawes, and Bruno Lacerda. “Bayesian Reinforcement Learning for Single-Episode Missions in Partially Unknown Environments”. In CoRL 2022.

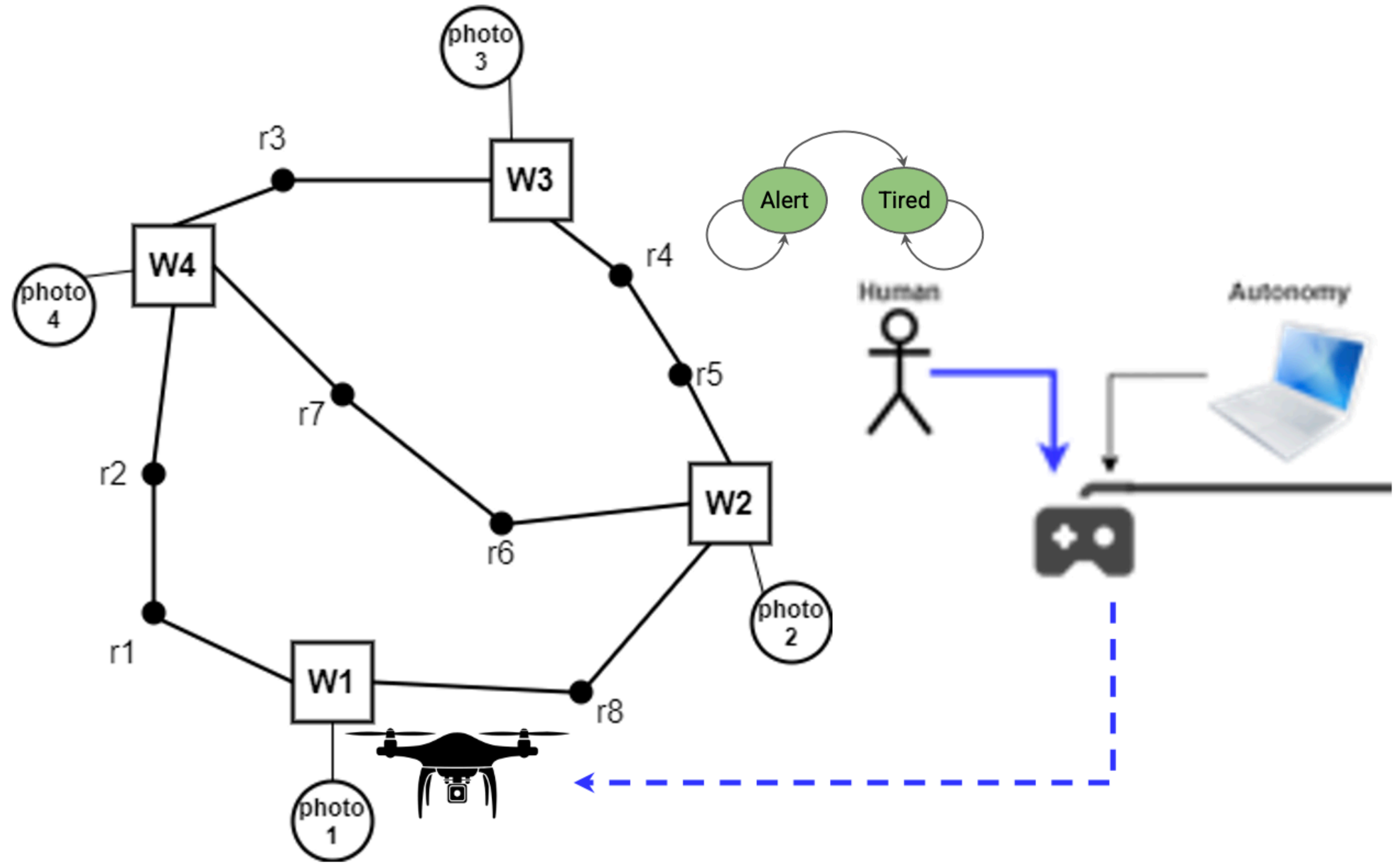
Mission Planning under Unknown Conditions

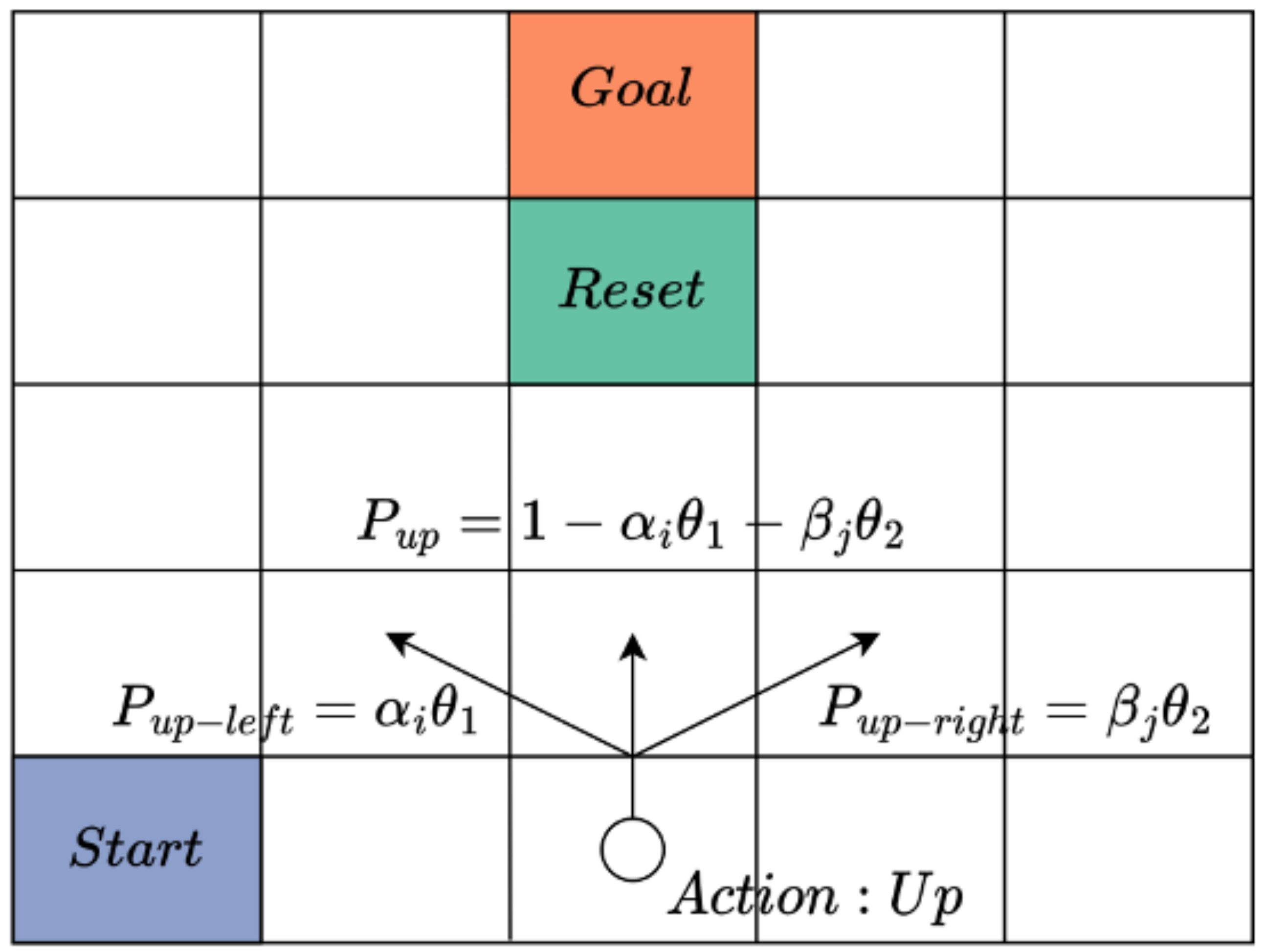


- **Data:** Online observations of unknown function; historical current data
- **Model:** BAMDP with GP belief
- **Specification:** Stochastic Shortest Path

Shared Autonomy Systems

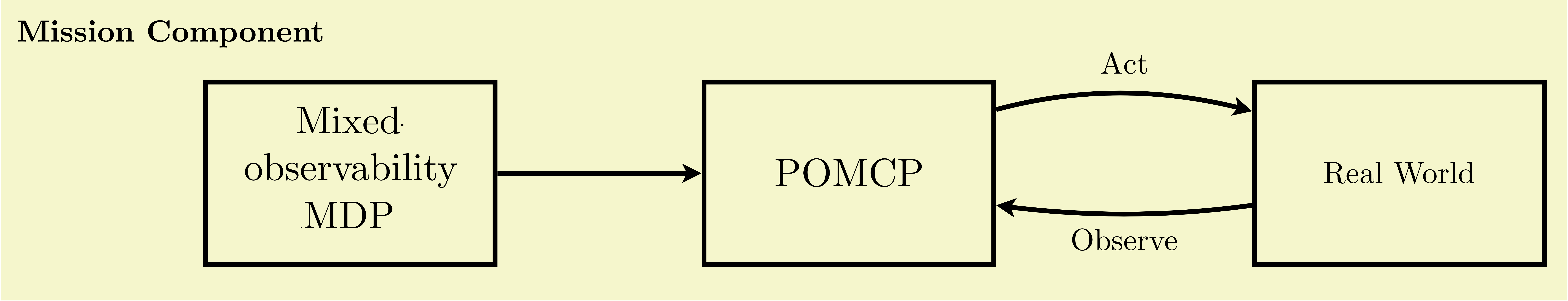
- **Goal:** Decide who takes control of the robot at each timestep
  - Human state is uncertain and time-varying
  - Modelled as a set of *n* possible performance profiles (Markov chains)
- Planning MDP plus human models yield a mixed-observability MDP
  - Maintain belief over current state of the human
- Novel hidden-parameter polynomial MDPs generalise to continuous spaces of human performance
  - Loses the time-varying aspect though :(

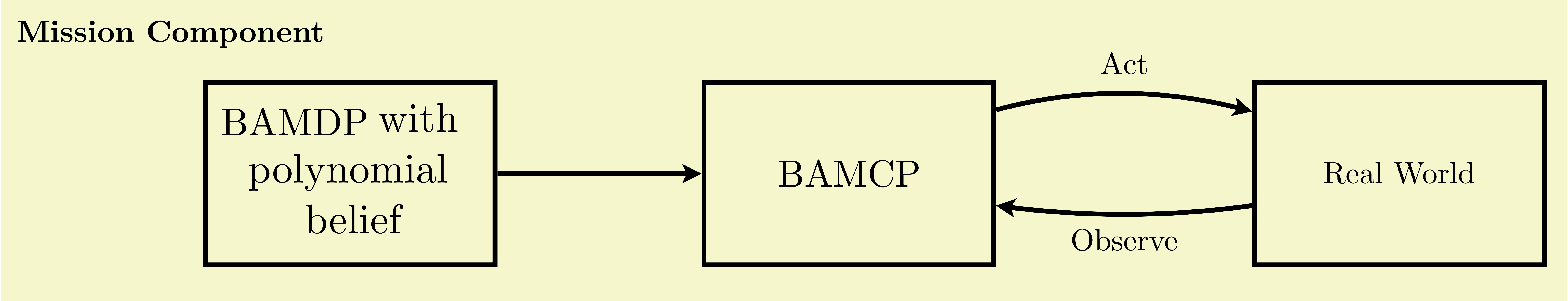




C. Costen, M. Rigter, B. Lacerda, N. Hawes. “Shared autonomy systems with stochastic operator models”. In IJCAI 2022.
C. Costen, M. Rigter, B. Lacerda, N. Hawes. “Planning with hidden parameter polynomial MDPs”. In AAAI 2023.

Shared Autonomy





- **Data:** Historical data of human performance
- **Model:** BAMDP/MOMDP
- **Specification:** Expected reward maximisation

Position Statement

Successful long-term robotic autonomy requires:

1. Data-driven model learning

2. Modelling and planning approaches that explicitly reason about the epistemic uncertainty inherent to models learnt from data

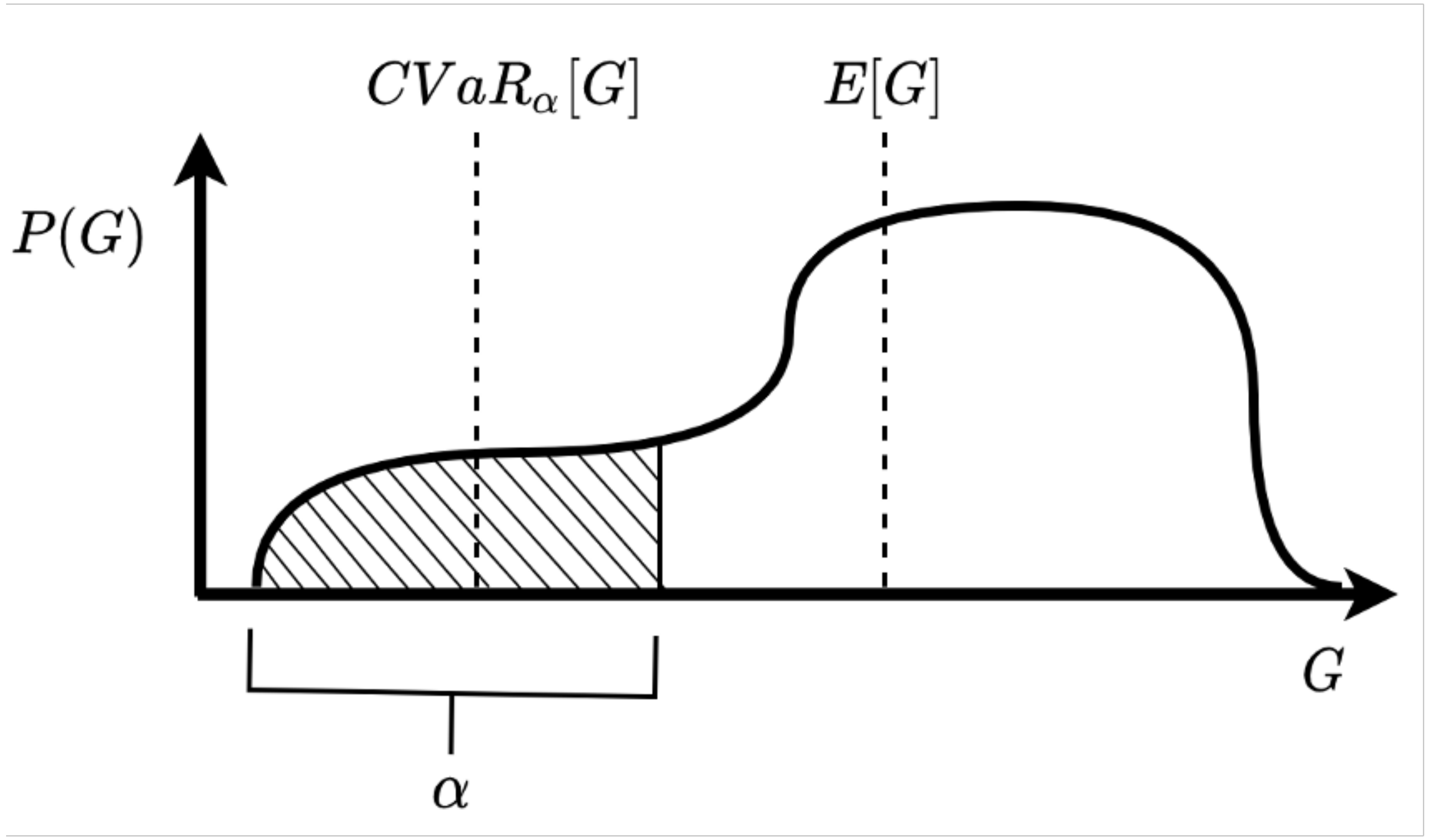
3. Incorporating rich specifications that go beyond typical reward maximisation in expectation

Robustness to model uncertainty

Risk Aversion

- When we can quantify uncertainty over models, we can consider a notion of risk
- We will consider conditional value at risk (CVaR)
  - Expected value of the alpha% worst cases

$$G = \sum_{t=0}^{t_H} r_t$$



$$CVaR_\alpha(G) = E[G \mid G \leq VaR_\alpha(G)]$$

- We will look into risk aversion for BAMDPs

Risk Aversion in BAMDPs as a Game

$$\max_{\pi} CVaR_{\alpha}(G^{+}) = \max_{\pi} \min_{\xi \in \Xi} E_{\xi}[G^{+}]$$

ξ is an adversarial perturbation to the transition probabilities in the BAMDP

- Pose problem as a stochastic game:
  1. Agent takes in action in the BAMDP to maximise the expected reward
  2. Adversary perturbs the transition probabilities (subject to budget) in the BAMDP to minimise the expected reward

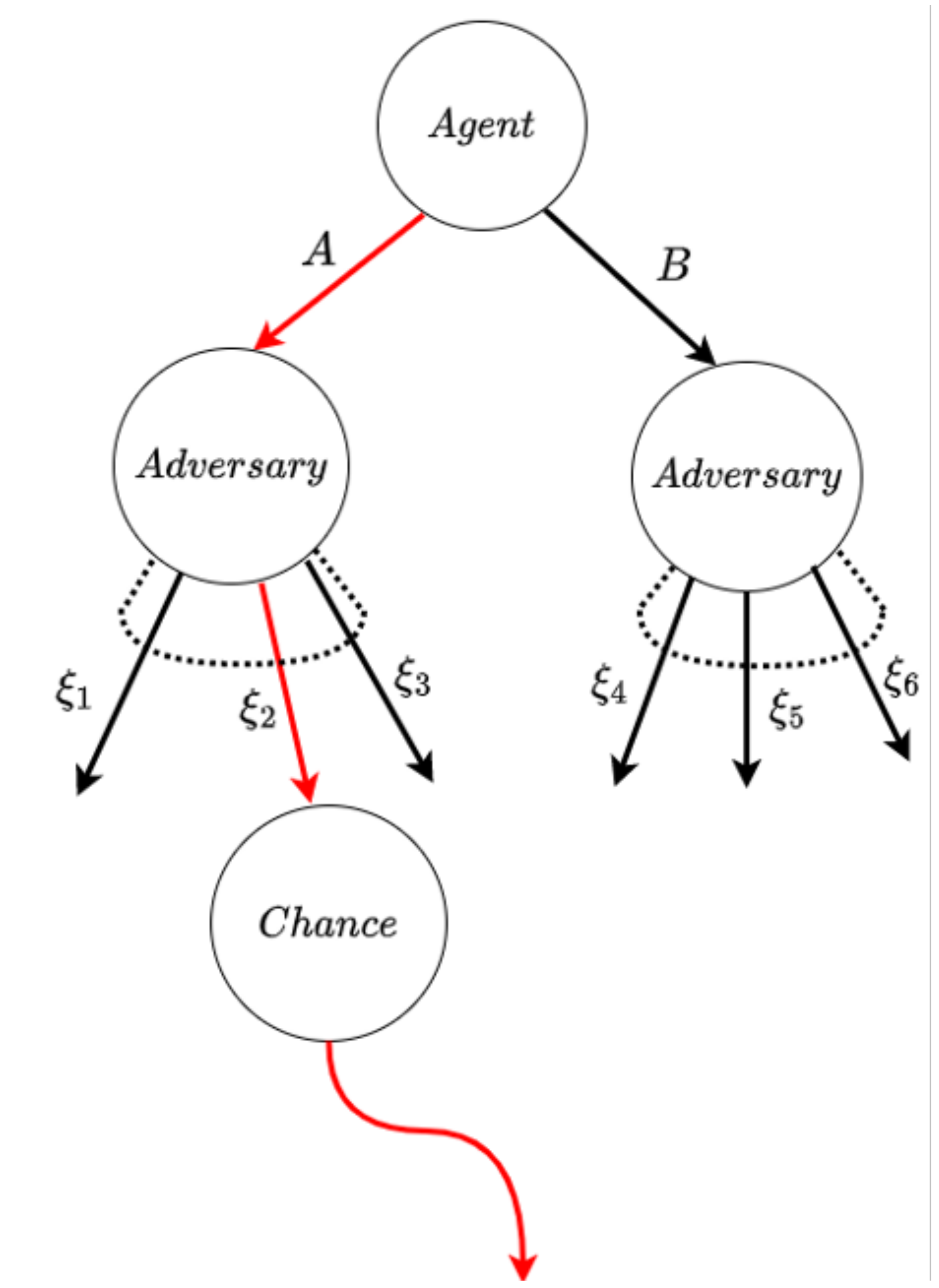
- Perturbing BAMDP transition probabilities can mean two things:
  - Perturbation to the prior over the true MDP - epistemic uncertainty
  - Perturbation to the transition probabilities in all possible MDPs - aleatoric uncertainty

M. Rigter, B. Lacerda, and N. Hawes, “Risk-Averse Bayes-Adaptive Reinforcement Learning,” in NeurIPS 2021.

Solution Method

- Difficult to solve exactly: BAMDP state space is large and adversary actions are continuous
- Solution: Two-player BAMCP
  - Progressive widening with Bayesian optimisation for continuous adversary action space

$$\max_{\pi} \min_{\xi \in \Xi} E_{\xi}\left[G^{+}\right]^{*}$$

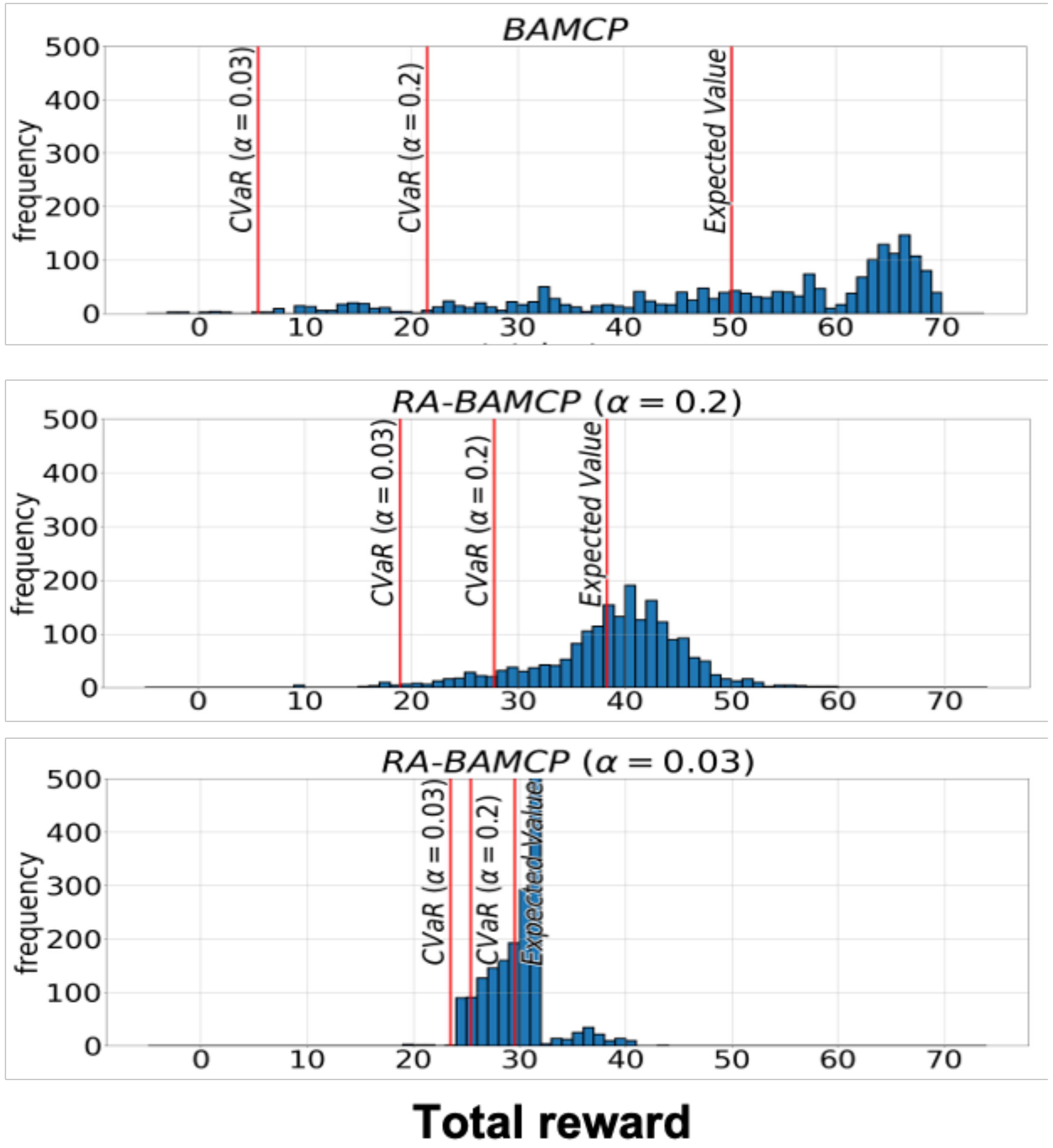


M. Rigter, B. Lacerda, and N. Hawes, “Risk-Averse Bayes-Adaptive Reinforcement Learning,” in NeurIPS 2021.

Results

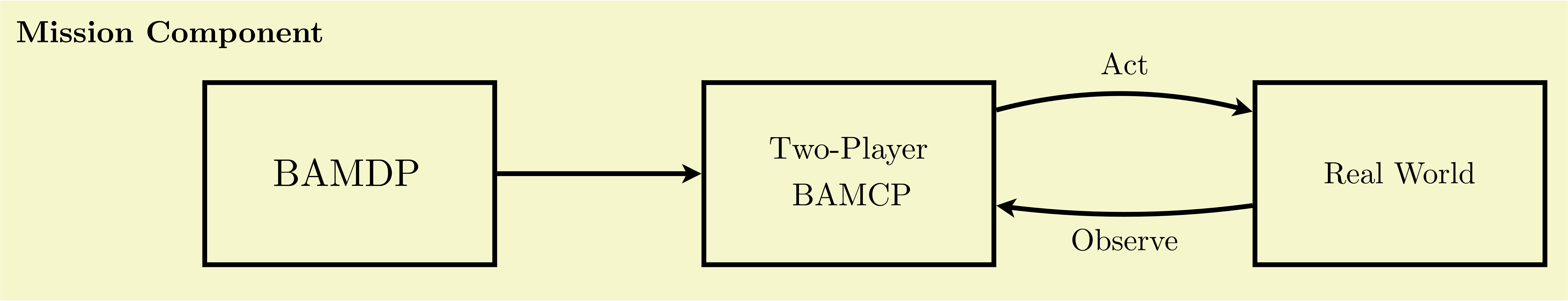
Risk-neutral baseline

Our approach



M. Rigter, B. Lacerda, and N. Hawes, "Risk-Averse Bayes-Adaptive Reinforcement Learning," in NeurIPS 2021.

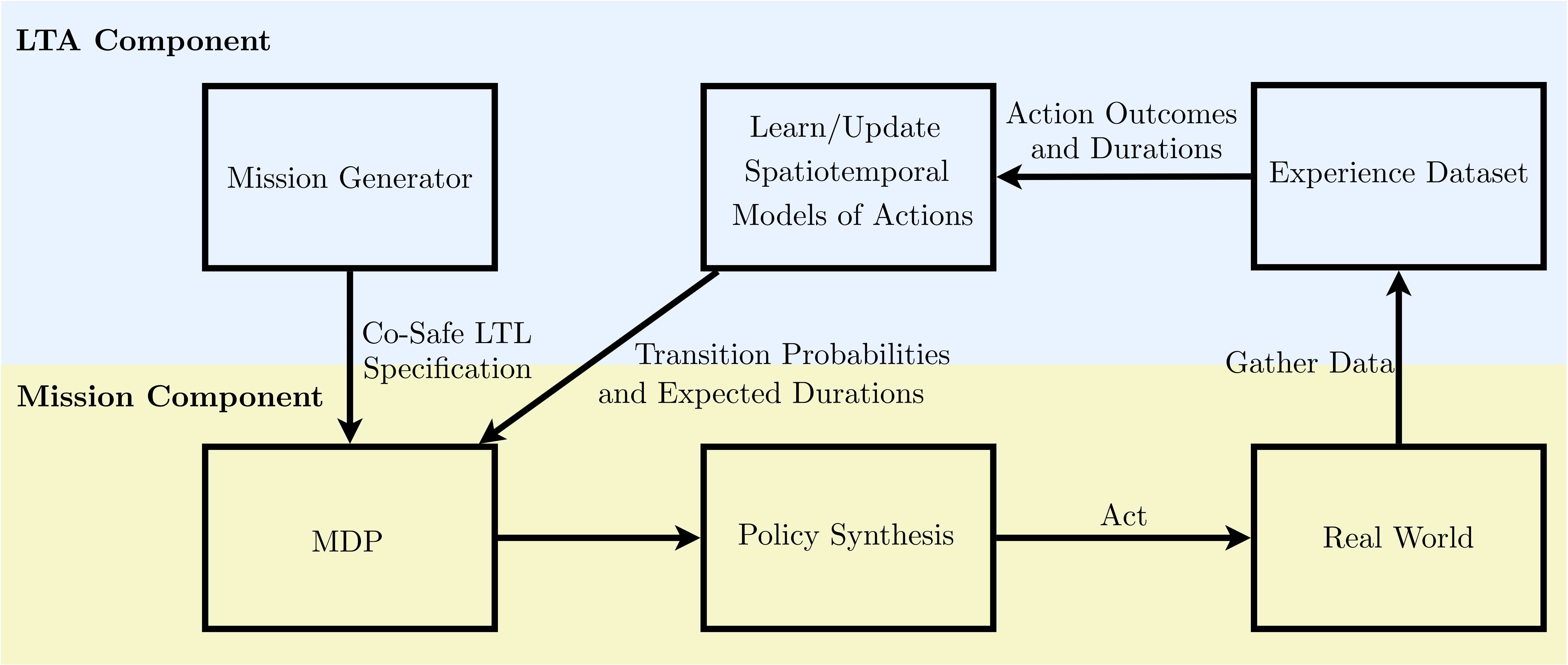
Risk-averse BAMDPs



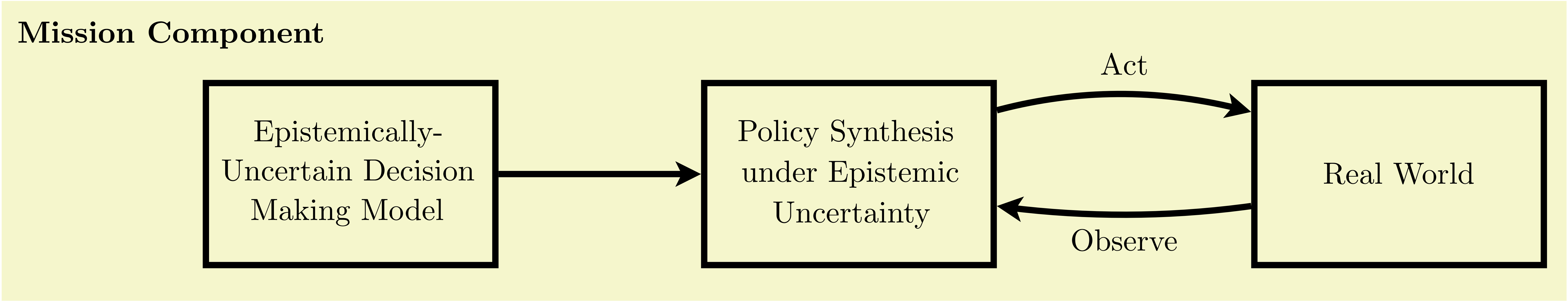
- **Data:** N/A
- **Model:** BAMDP
- **Specification:** Optimise for CVaR

Current Situation

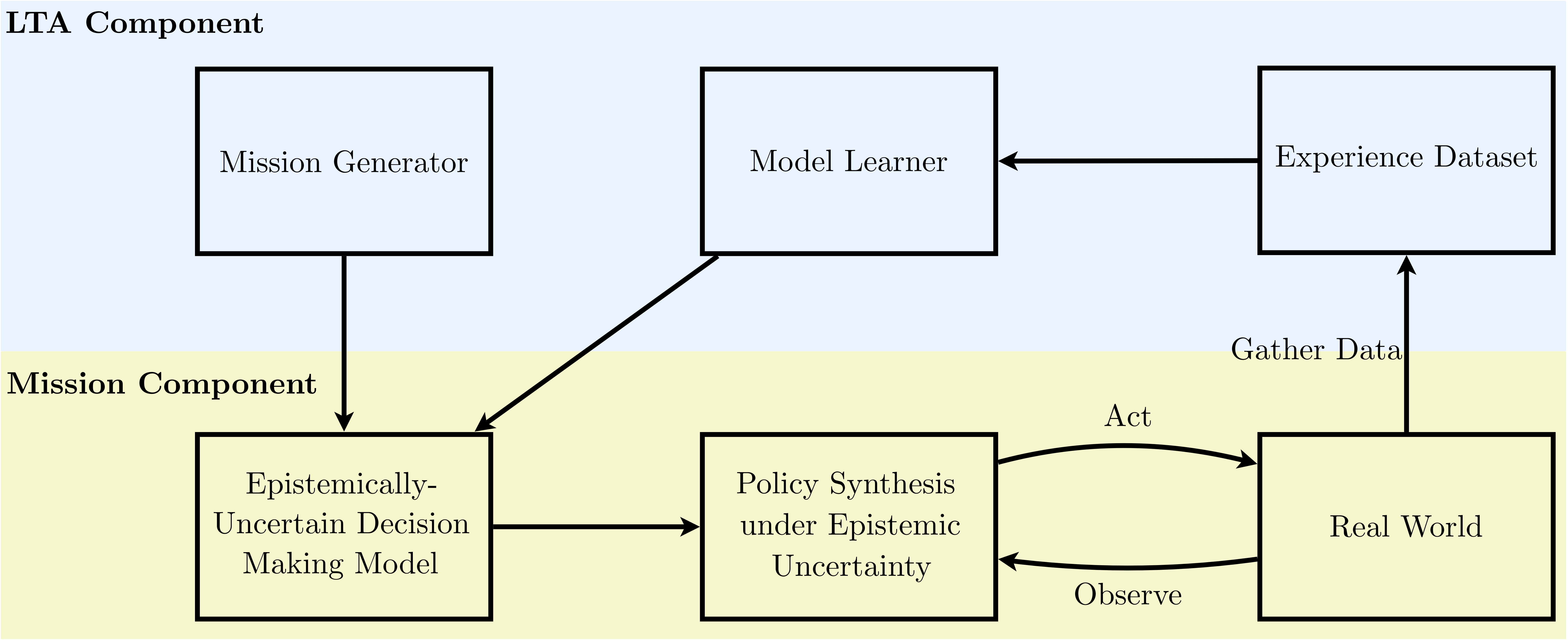
- Long-term autonomy
  - Epistemic uncertainty not considered
  - Assumes (single) model is correct when planning



- Epistemic uncertainty
  - Single mission
  - No offline learning from mission data



The Future



- How to use mission data to learn models that consider epistemic uncertainty?
- How to develop planning approaches that appropriately consider epistemic uncertainty when synthesising robot behaviour?
  - How to best represent and maintain the belief over the real model?
  - How to consider dynamic world models?

Summary

Successful long-term robotic autonomy requires:

1. Data-driven model learning
   - Transition probabilities, action duration, task request dynamics, battery dynamics, human behaviour, predictions from historical data

2. Modelling and planning approaches that explicitly reason about the epistemic uncertainty inherent to models learnt from data
   - MDPs with GP predictions, BAMDPs, polynomial MDPs, sample-based uncertain MDPs

3. Incorporating rich specifications that go beyond typical expected reward maximisation
   - Temporal logics, multi-objective, regret minimisation, risk-averse behaviour, chance constraints

Course contents

- Markov decision processes (MDPs) and stochastic games
  - MDPs: key concepts and algorithms
  - stochastic games: adding adversarial aspects
- Uncertain MDPs
  - MDPs + epistemic uncertainty, robust control, robust dynamic programming, interval MDPs, uncertainty set representation, challenges, tools
- Sampling-based uncertain MDPs
  - removing the transition independence assumption
- Bayes-adaptive MDPs
  - maintaining a distribution over the possible models
  - usage in mission planning for robots