Model-based Reinforcement Learning

Aidan Scannell
Finnish Center for Artificial Intelligence (FCAI)
Aalto University

Slides available here

17th July 2024

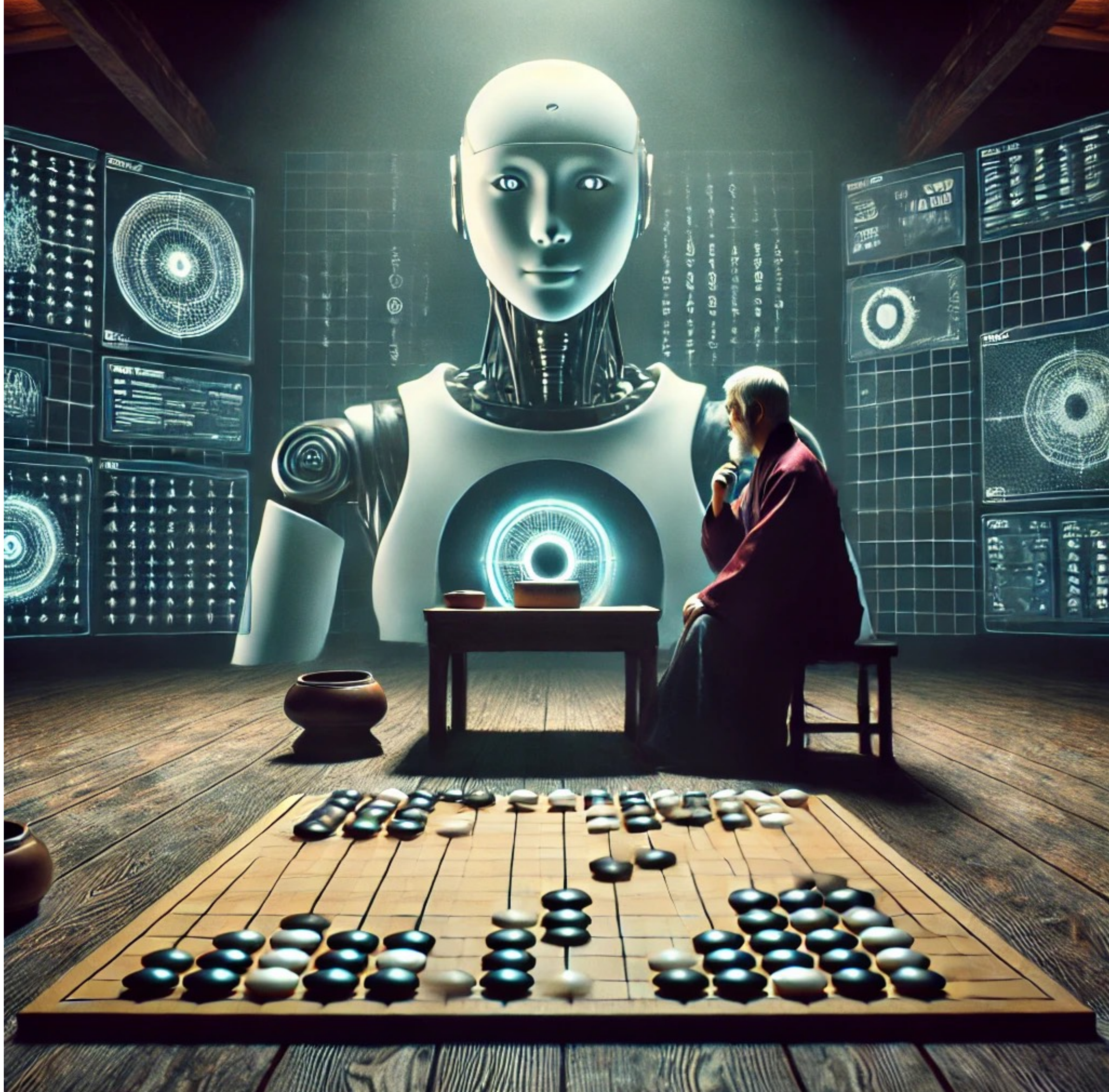
FCAI

AlphaGo

Model-based reasoning for games

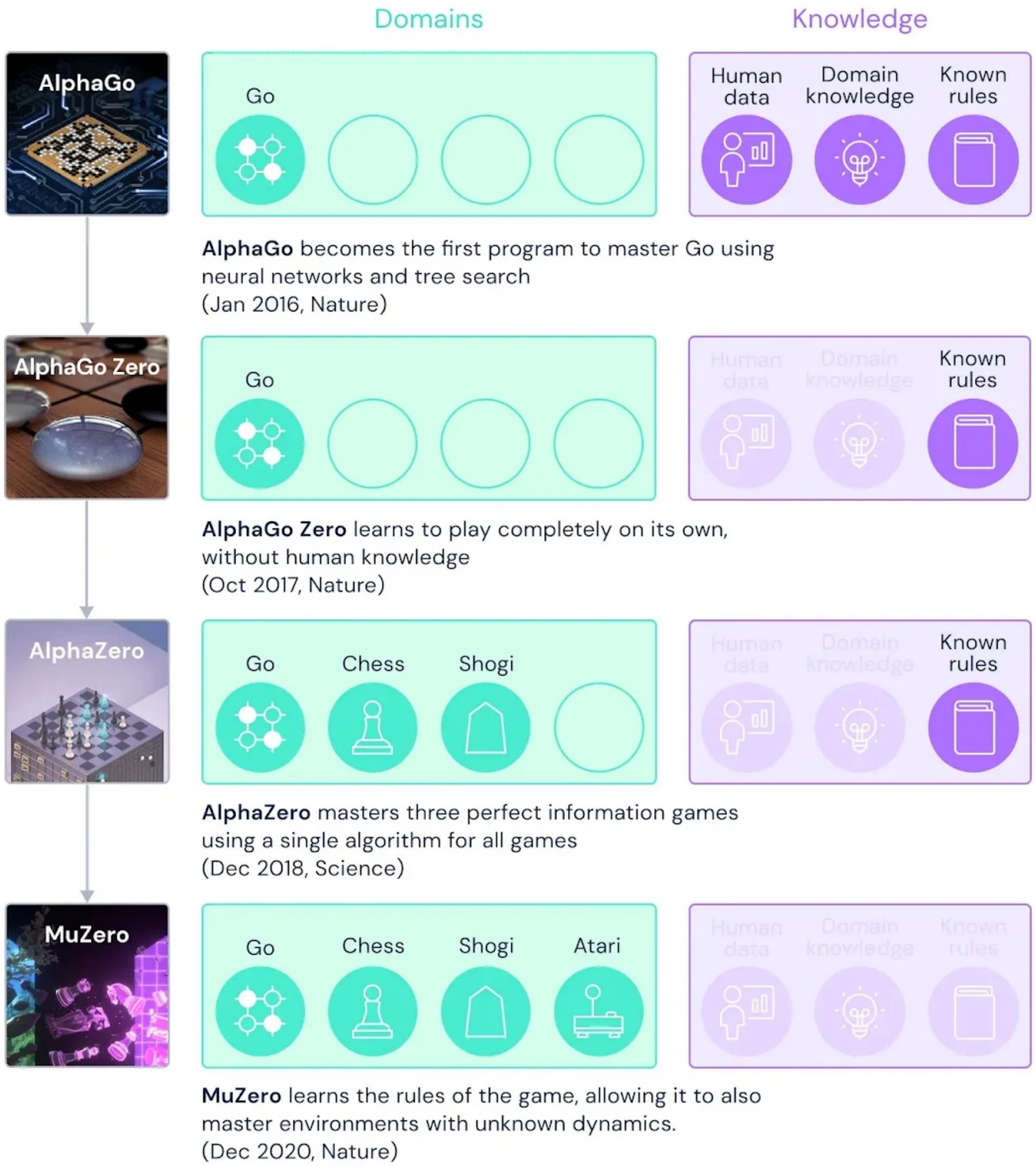
Silver et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484.

FCAI



AlphaGo

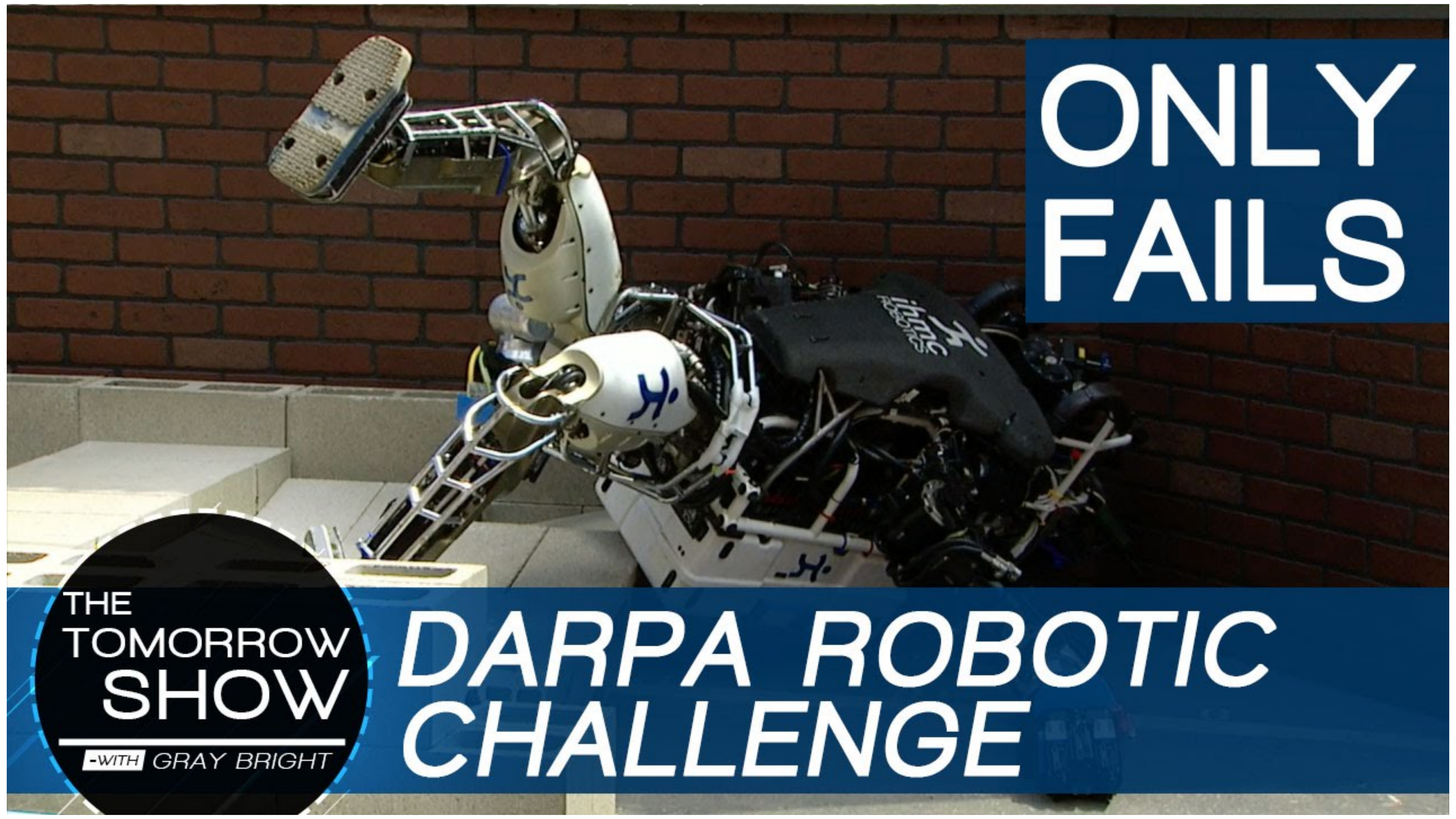
Model-based reasoning for games



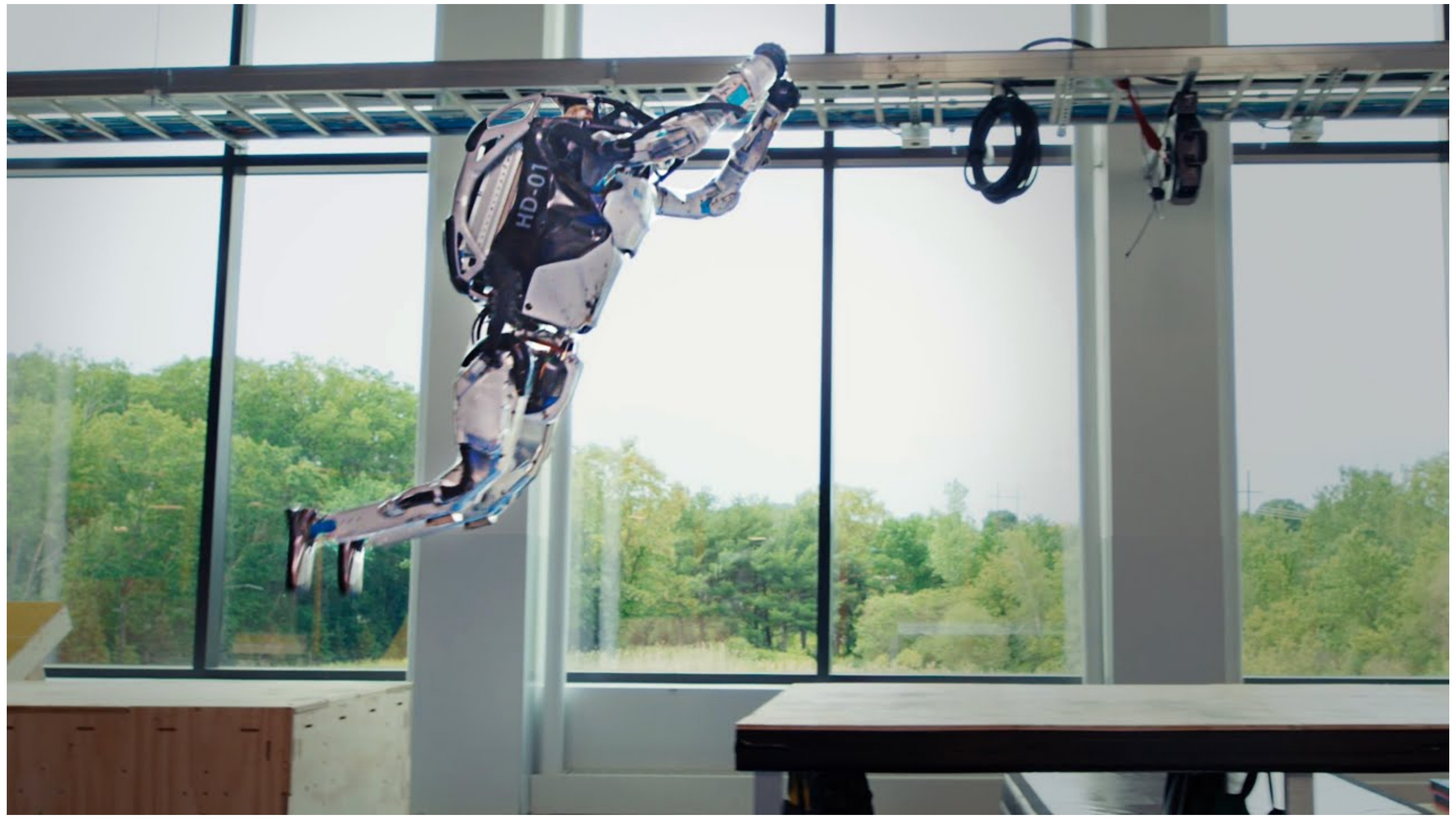
Silver et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484.

FCAI

Machine Learning for Robotics

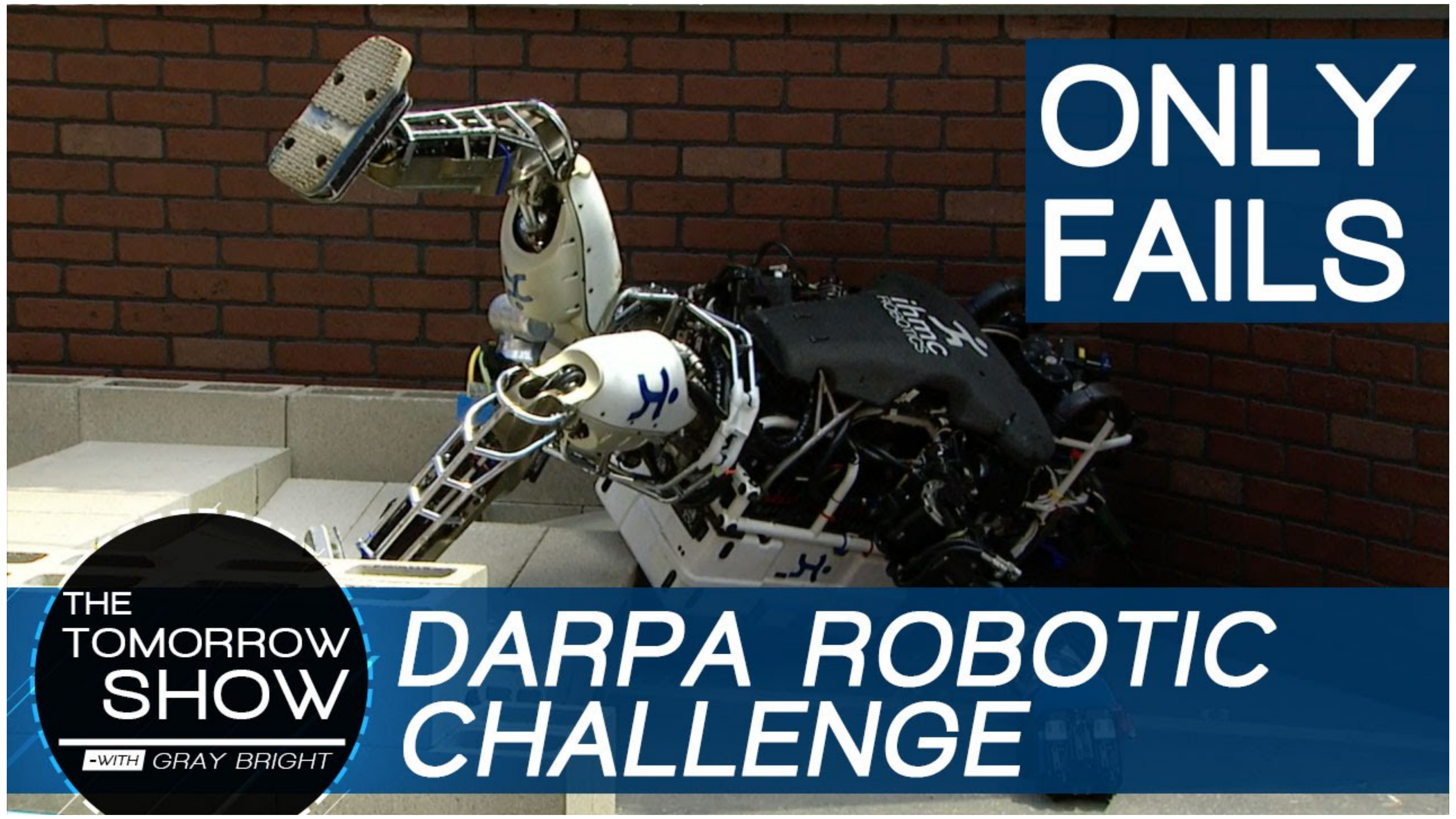


DARPA Robotics Challenge 2015

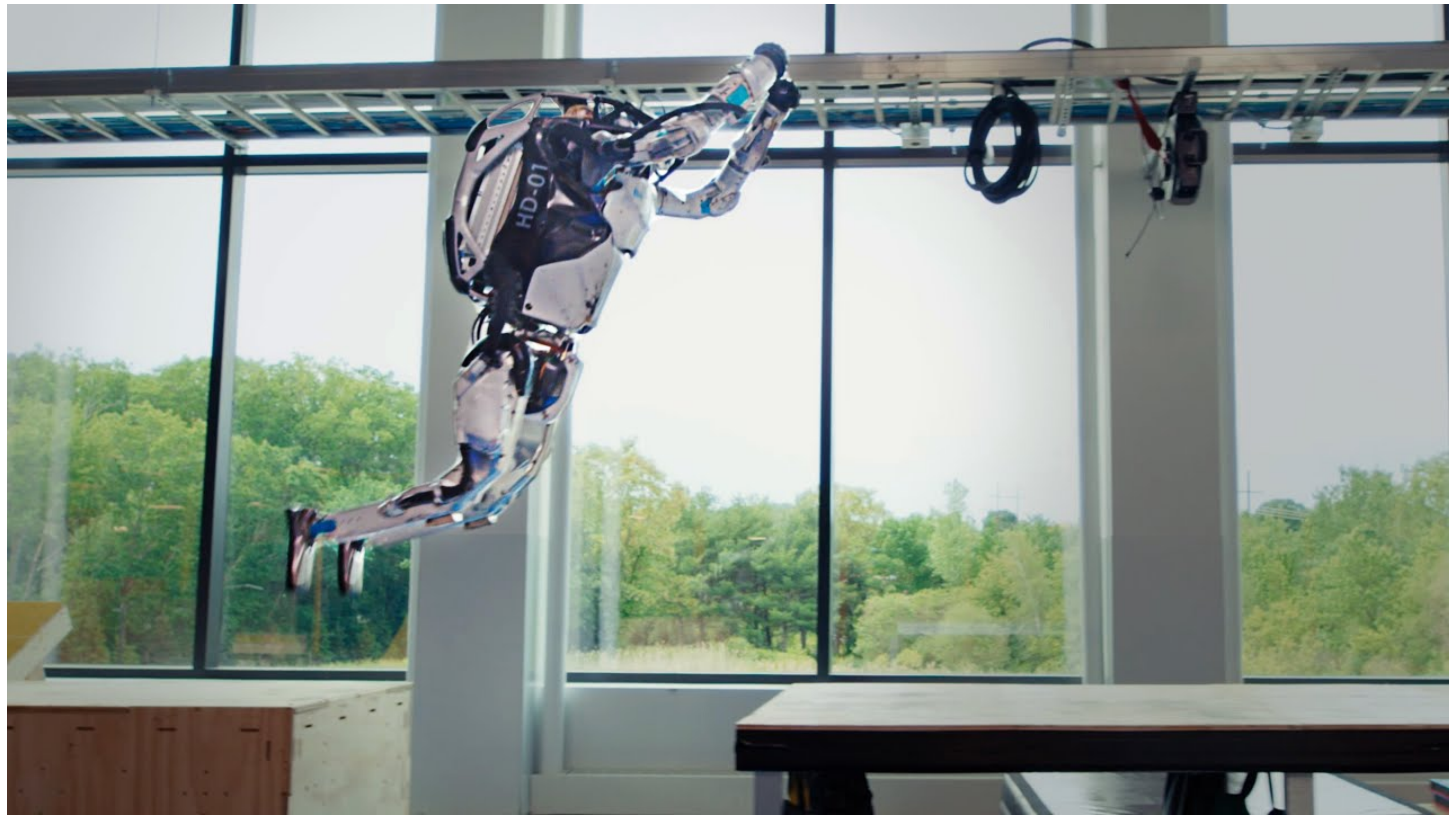


Boston Dynamics Atlas - Partners in Parkour

Machine Learning for Robotics

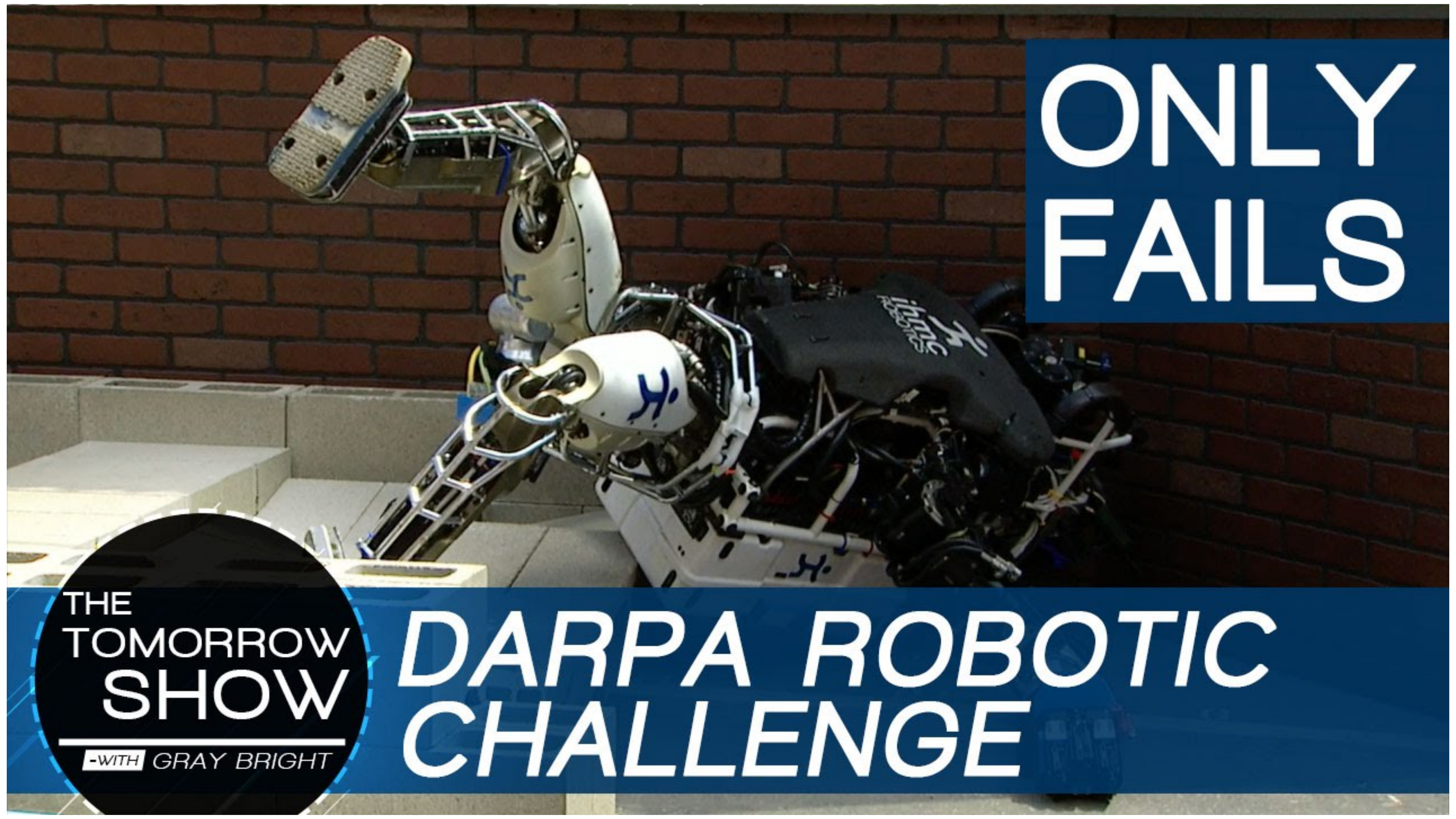


DARPA Robotics Challenge 2015

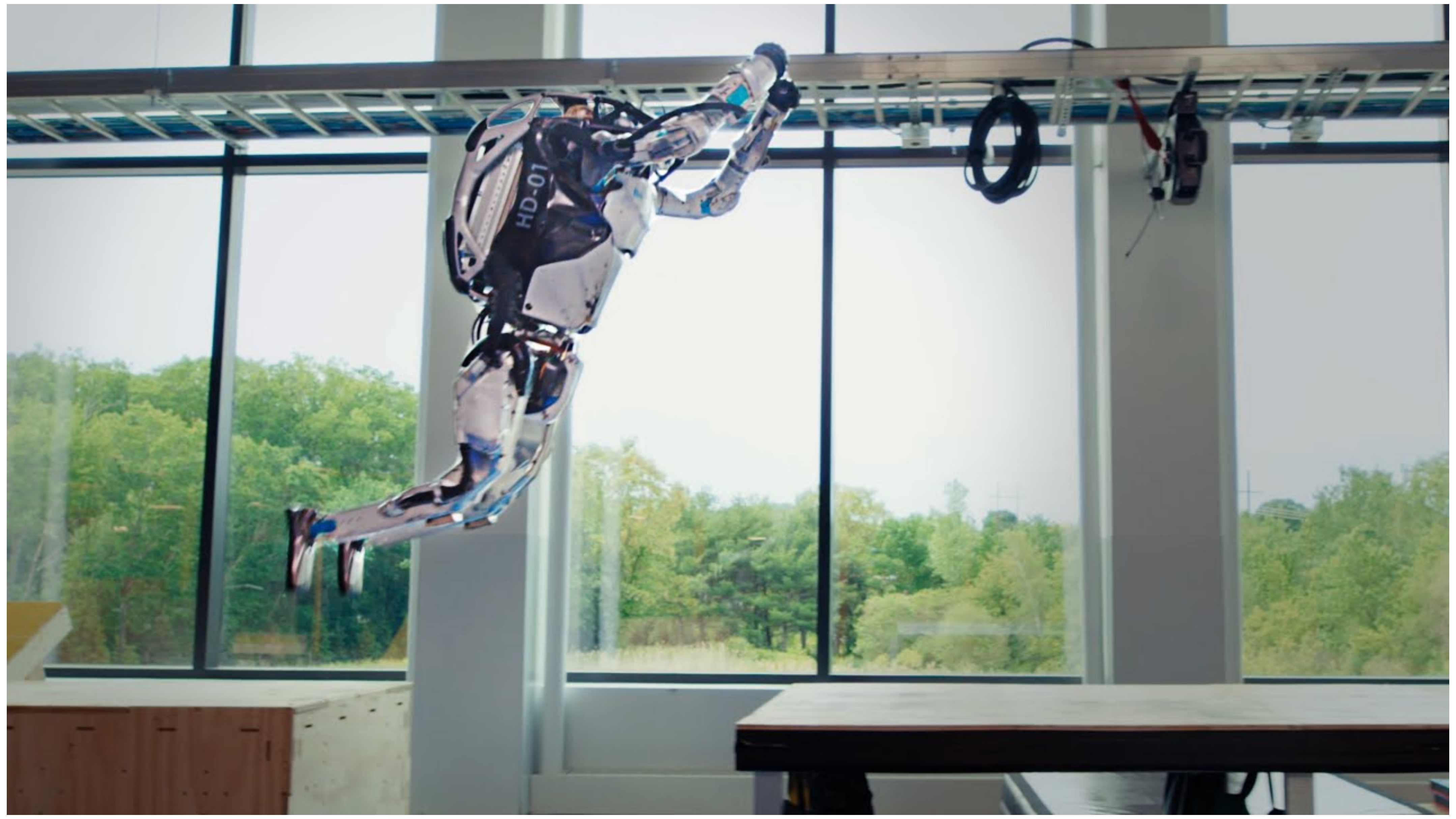


Boston Dynamics Atlas - Partners in Parkour

Machine Learning for Robotics



DARPA Robotics Challenge 2015


Boston Dynamics Atlas - Partners in Parkour

Learning Objectives

Learning Objectives

Understand

Learning Objectives

Understand

1. What a “model” is in model-based RL

Learning Objectives

Understand

1. What a “model” is in model-based RL
2. How a “model” can aid decision making

Learning Objectives

Understand

1. What a “model” is in model-based RL
2. How a “model” can aid decision making
3. Differences between background and decision-time planning

Learning Objectives

Understand

1. What a “model” is in model-based RL
2. How a “model” can aid decision making
3. Differences between background and decision-time planning
4. Decision-time planning strategies for continuous actions

Learning Objectives

Understand

1. What a “model” is in model-based RL
2. How a “model” can aid decision making
3. Differences between background and decision-time planning
4. Decision-time planning strategies for continuous actions
5. Sources of uncertainty in model-based RL

Learning Objectives

Understand

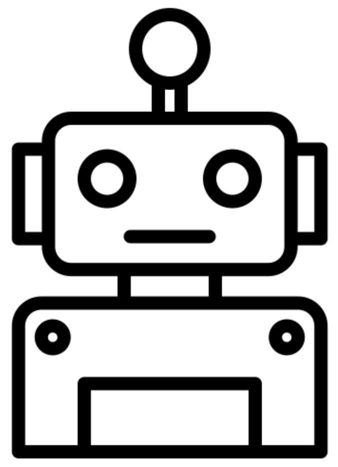
1. What a “model” is in model-based RL
2. How a “model” can aid decision making
3. Differences between background and decision-time planning
4. Decision-time planning strategies for continuous actions
5. Sources of uncertainty in model-based RL
6. Rationale and insights for decision-making under uncertainty

Reinforcement Learning

Markov Decision Process (MDP)

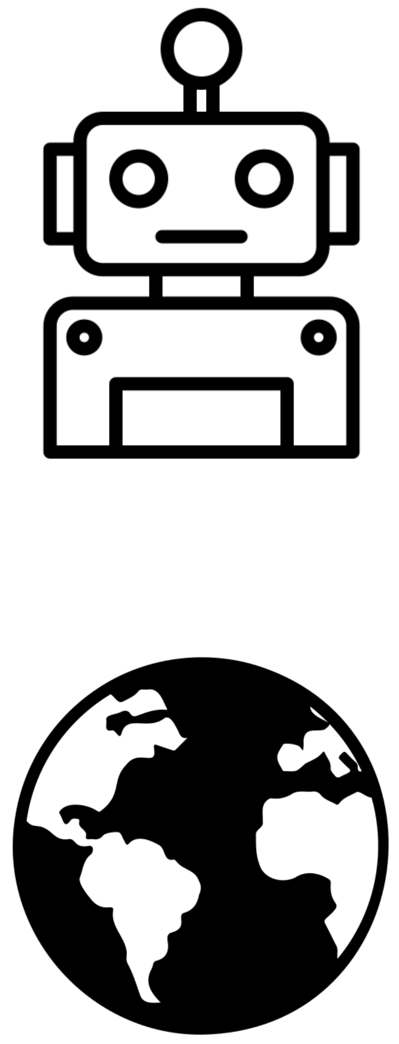
Reinforcement Learning

Markov Decision Process (MDP)



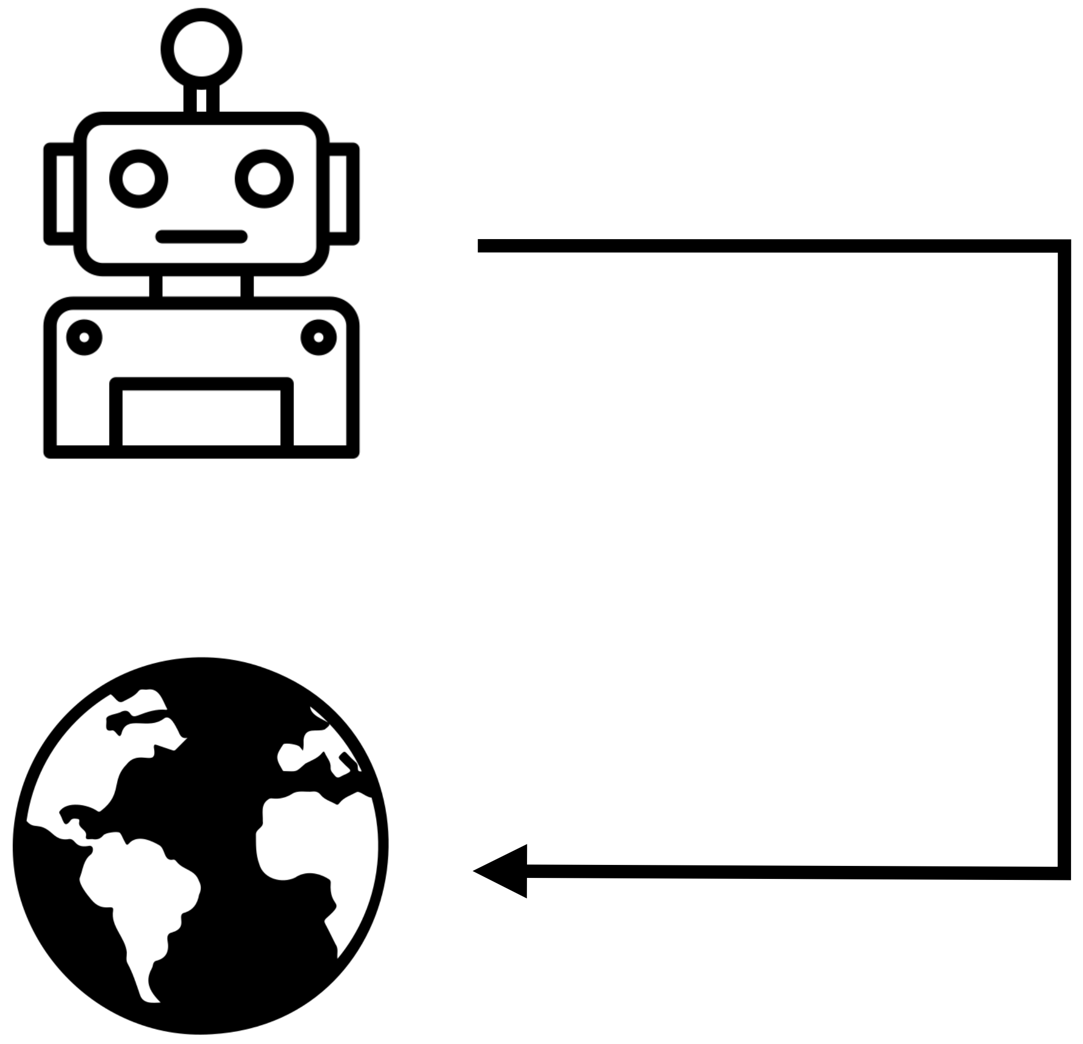
Reinforcement Learning

Markov Decision Process (MDP)



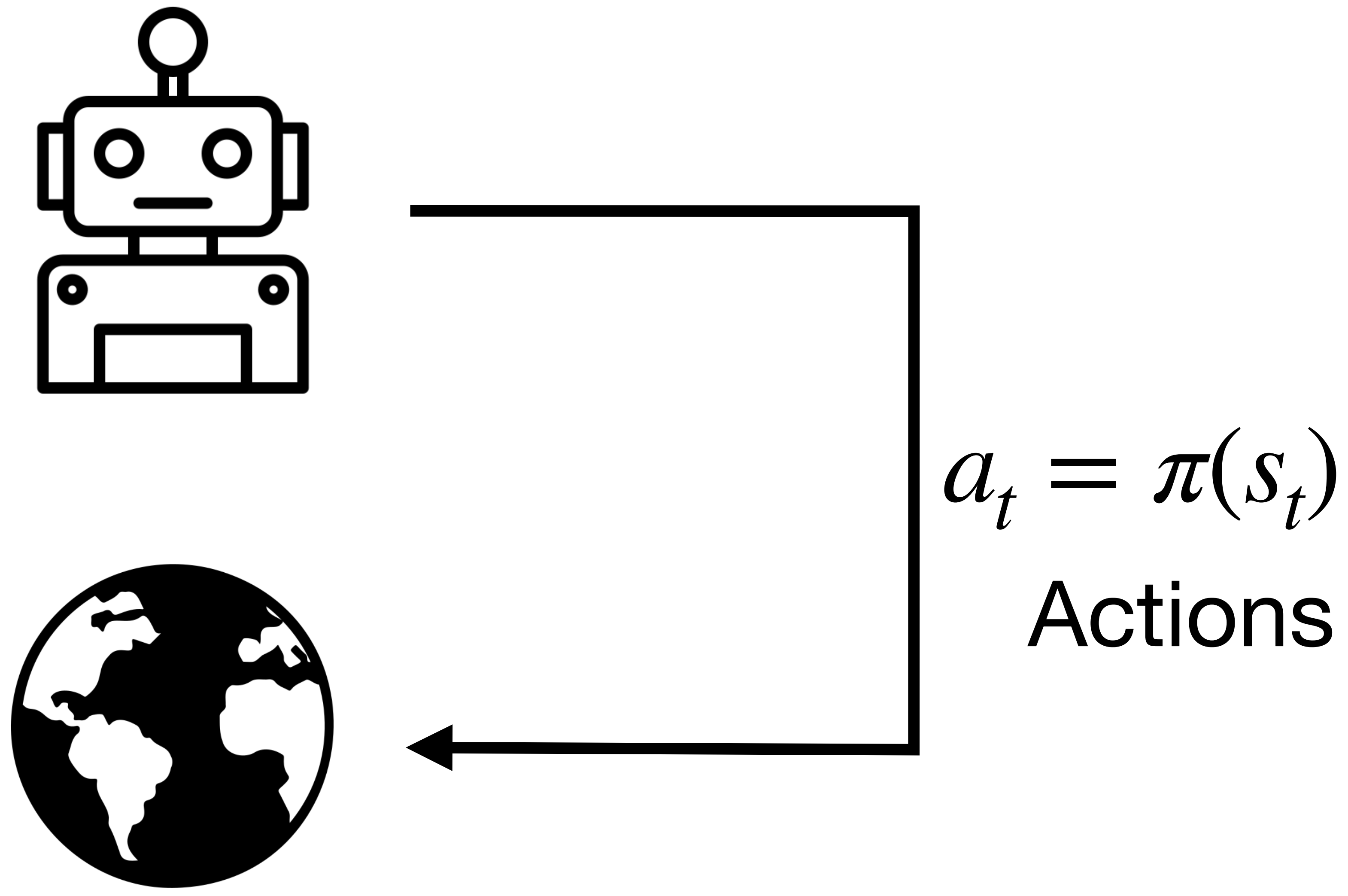
Reinforcement Learning

Markov Decision Process (MDP)



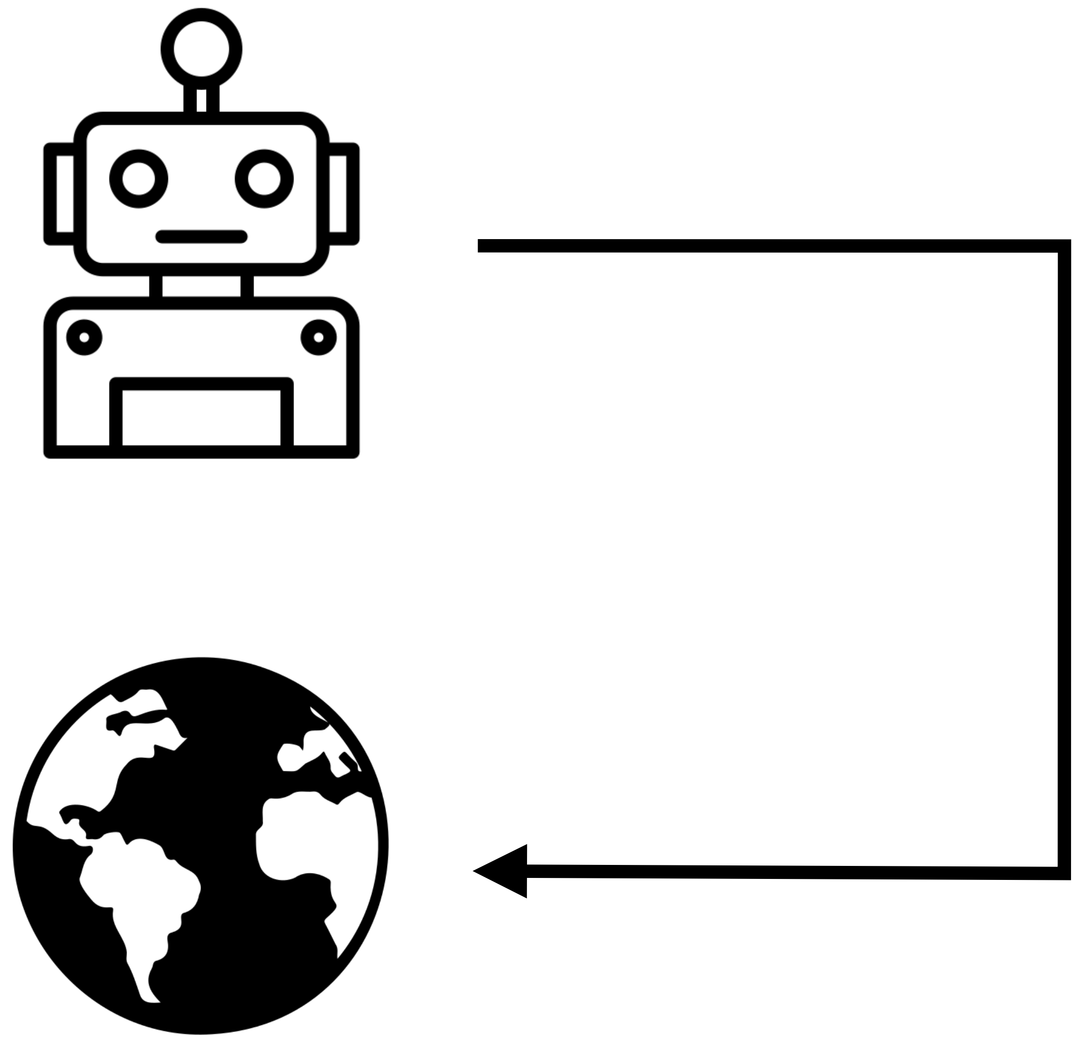
Reinforcement Learning

Markov Decision Process (MDP)



Reinforcement Learning

Markov Decision Process (MDP)



$a_t = \pi(s_t)$

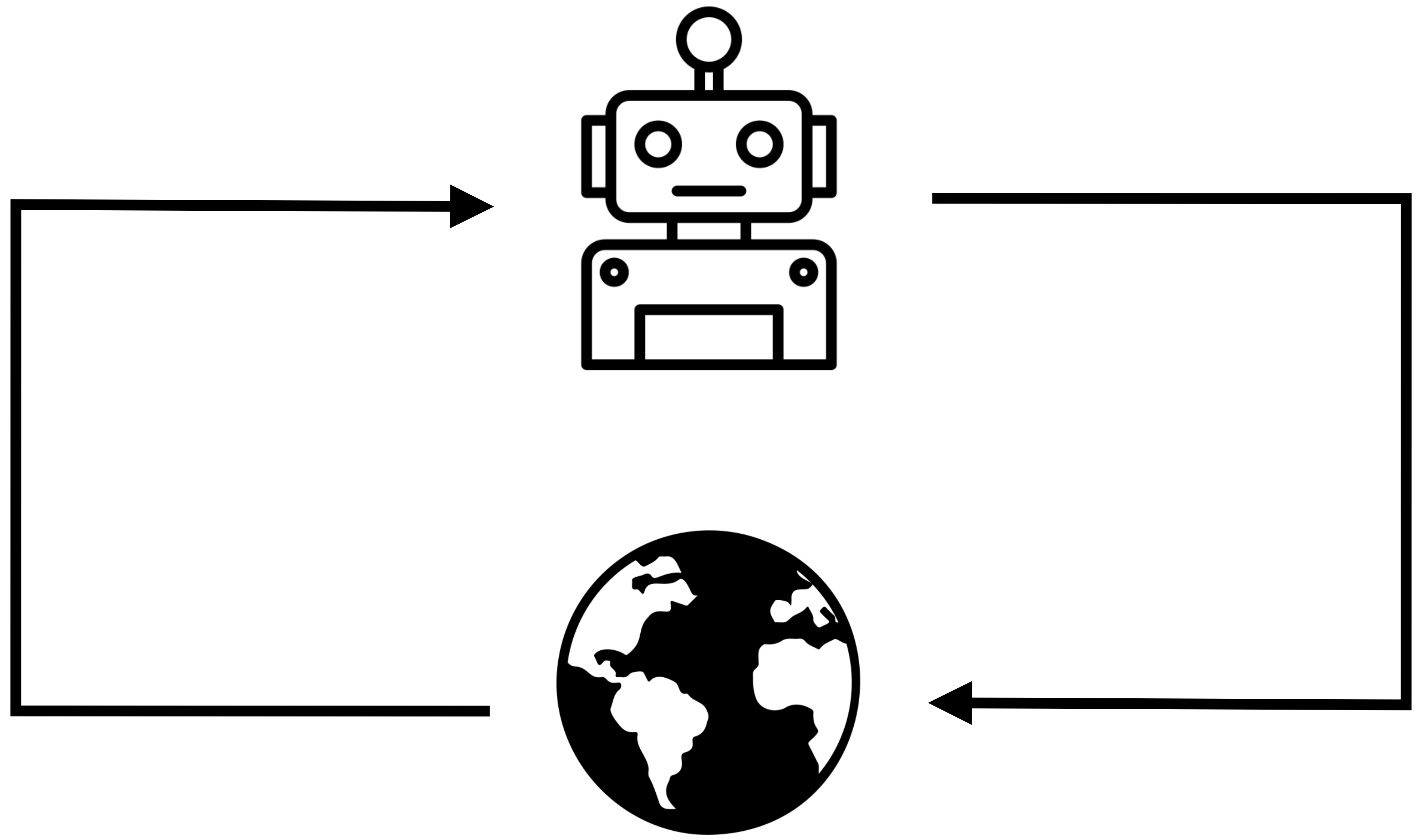
Actions

$s_{t+1} \sim P(\cdot \mid s_t, a_t)$

Transition function

Reinforcement Learning

Markov Decision Process (MDP)



$a_t = \pi(s_t)$

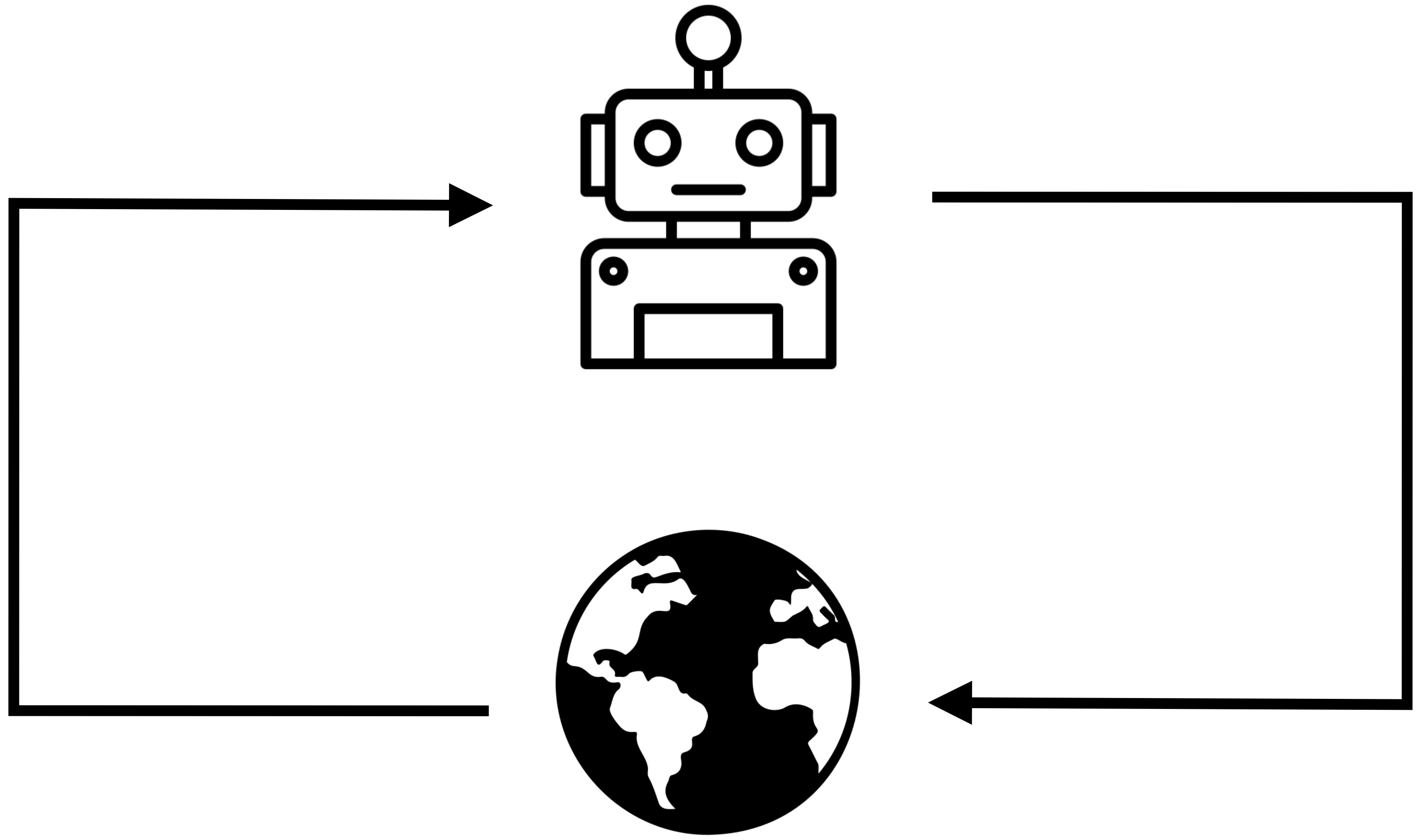
Actions

$s_{t+1} \sim P(\cdot \mid s_t, a_t)$

Transition function

Reinforcement Learning

Markov Decision Process (MDP)



$s_{t+1}, \quad r(s_t, a_t)$

State, Reward

$a_t = \pi(s_t)$

Actions

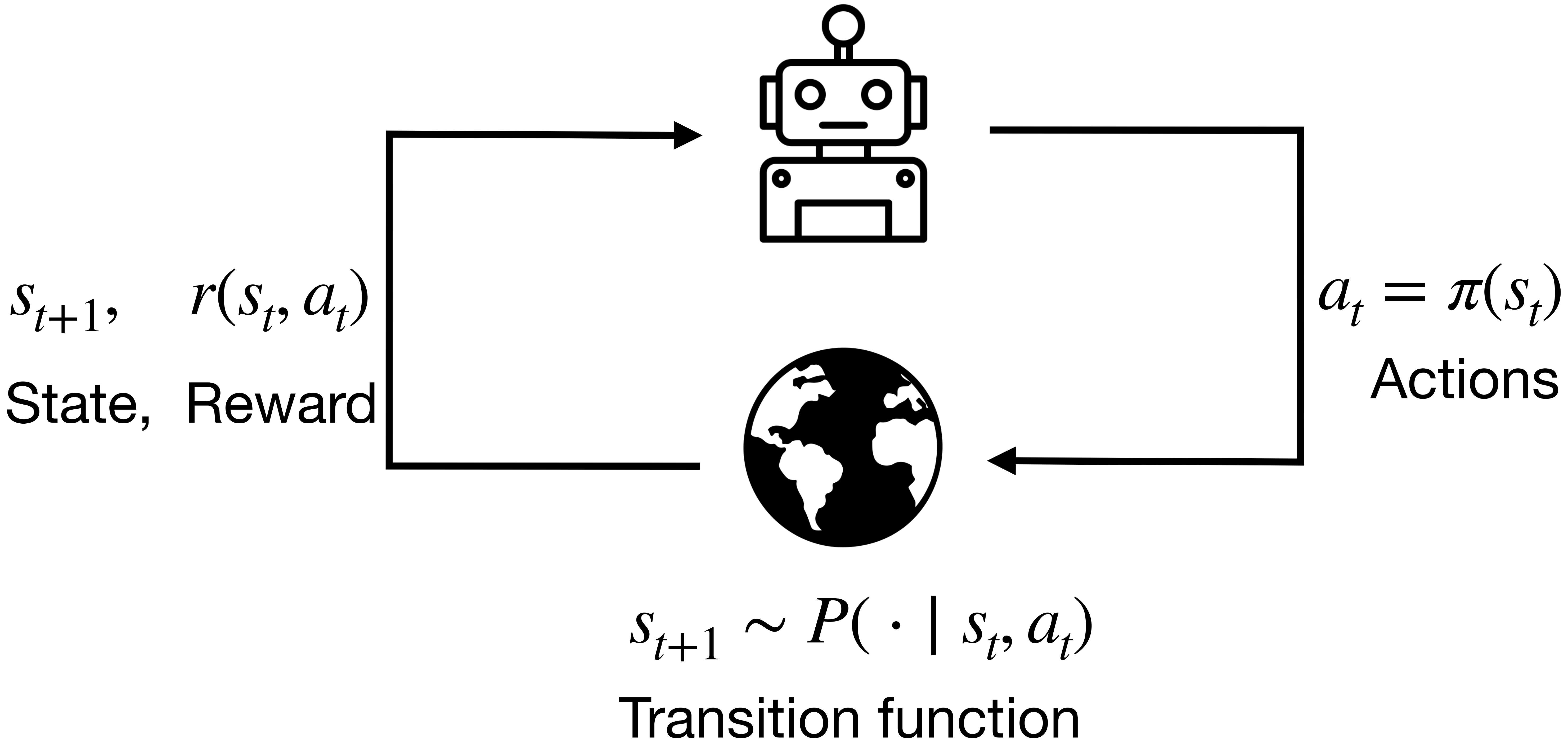
$s_{t+1} \sim P(\,\cdot \mid s_t, a_t)$

Transition function

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

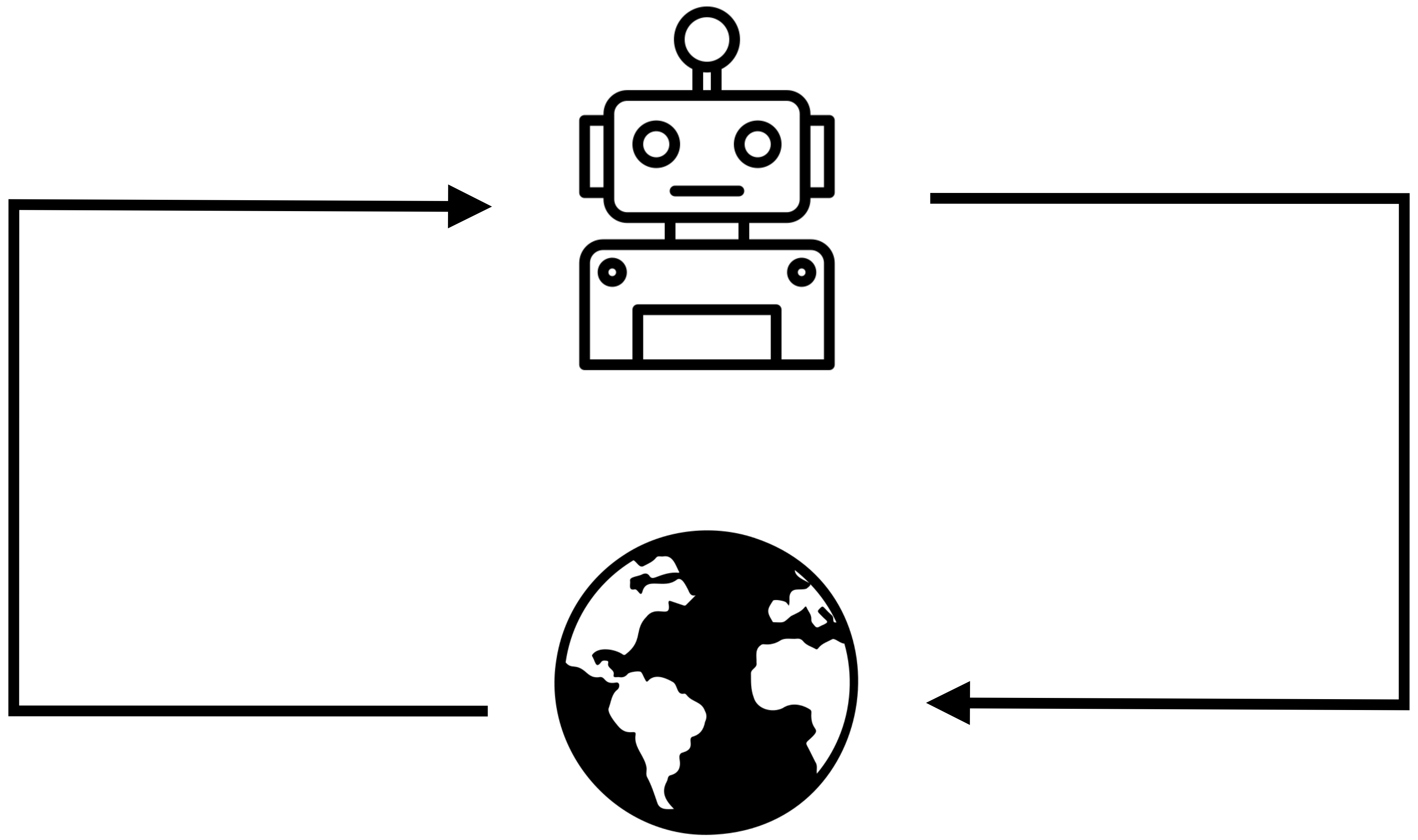


Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$



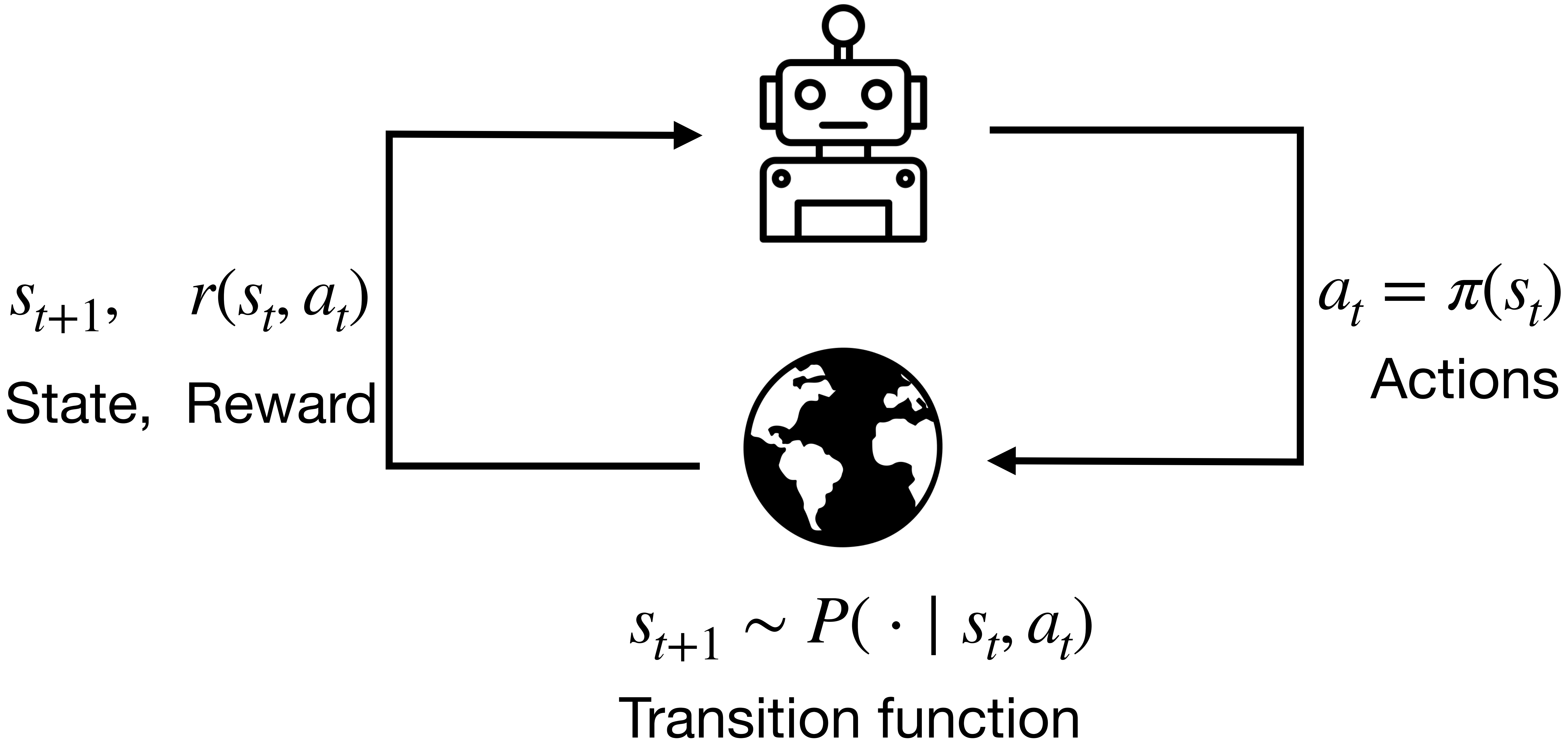
Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$



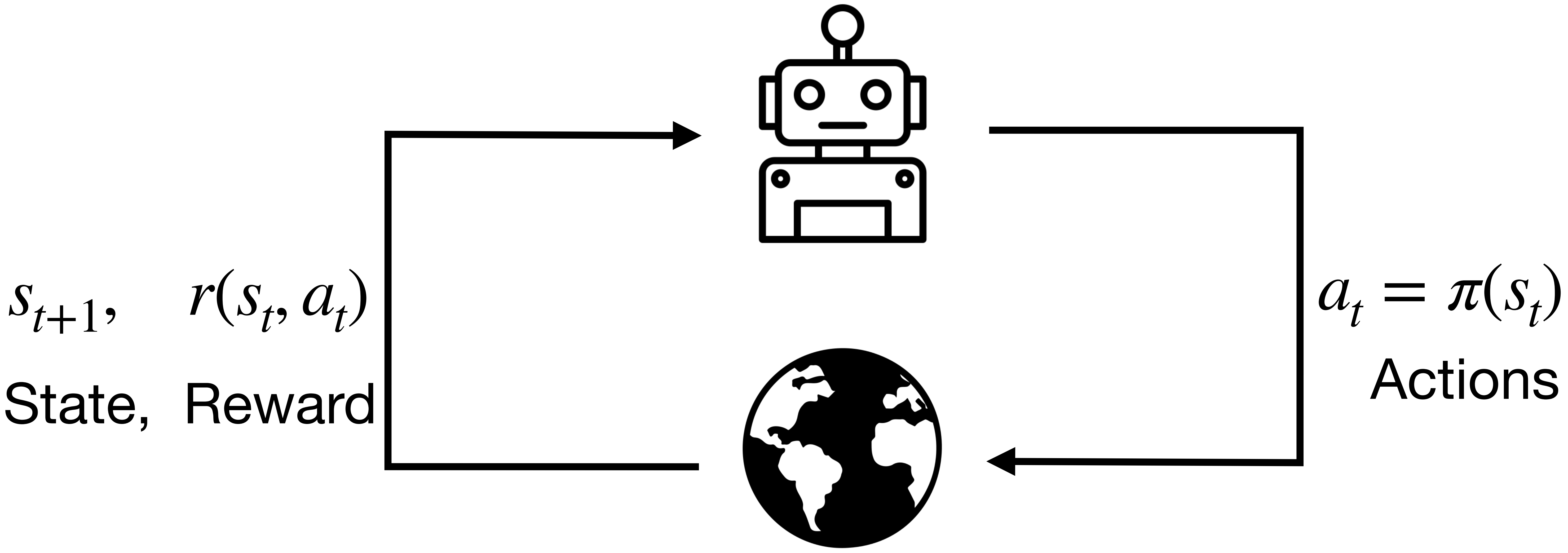
Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$



$s_{t+1} \sim P(\cdot \mid s_t, a_t)$

Transition function

$s_{t+1} = f(s_t, a_t) + \epsilon_t$

Reinforcement Learning

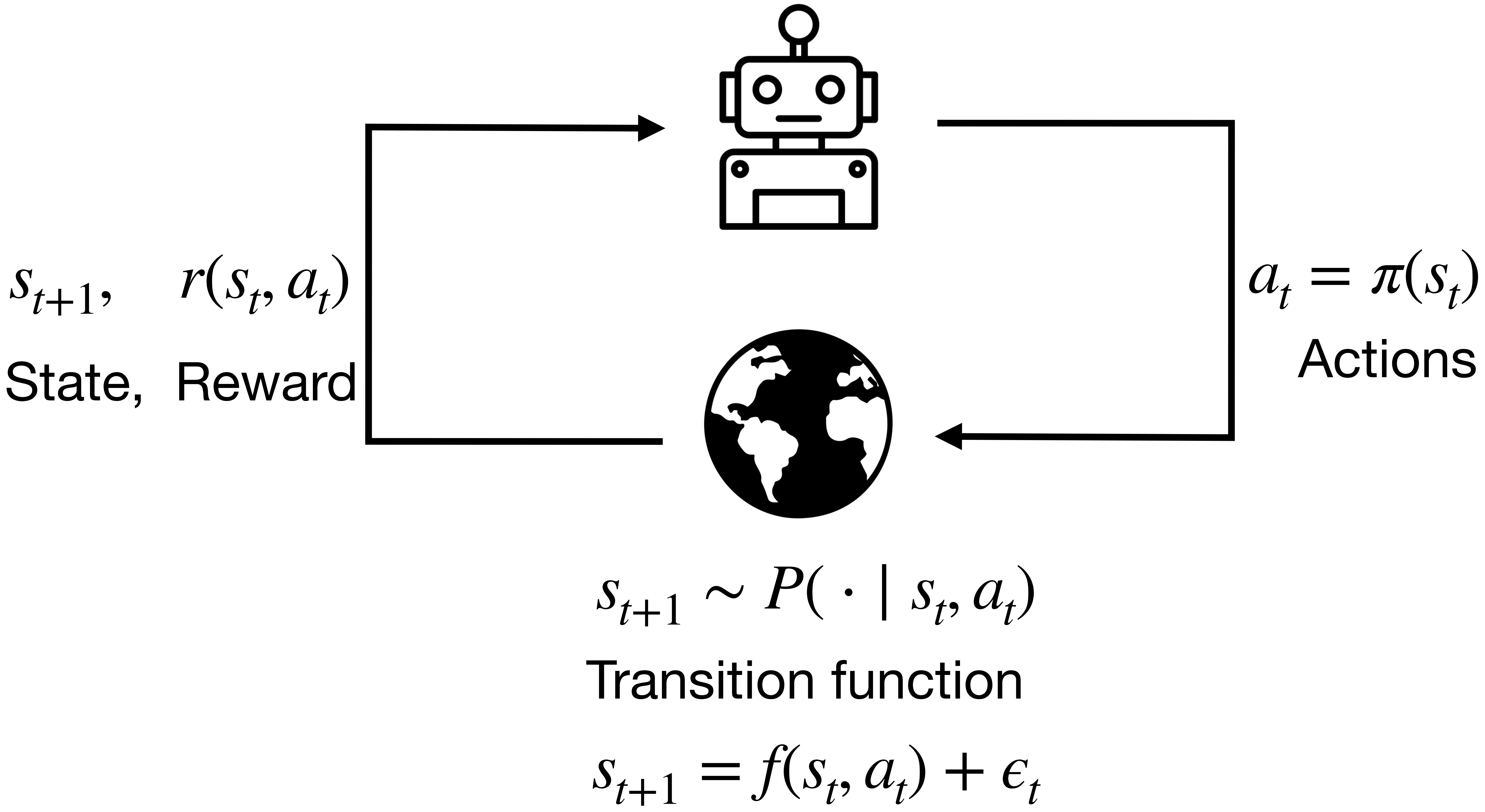
Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$



FCAI

fcai.fi

Reinforcement Learning

Markov Decision Process (MDP)

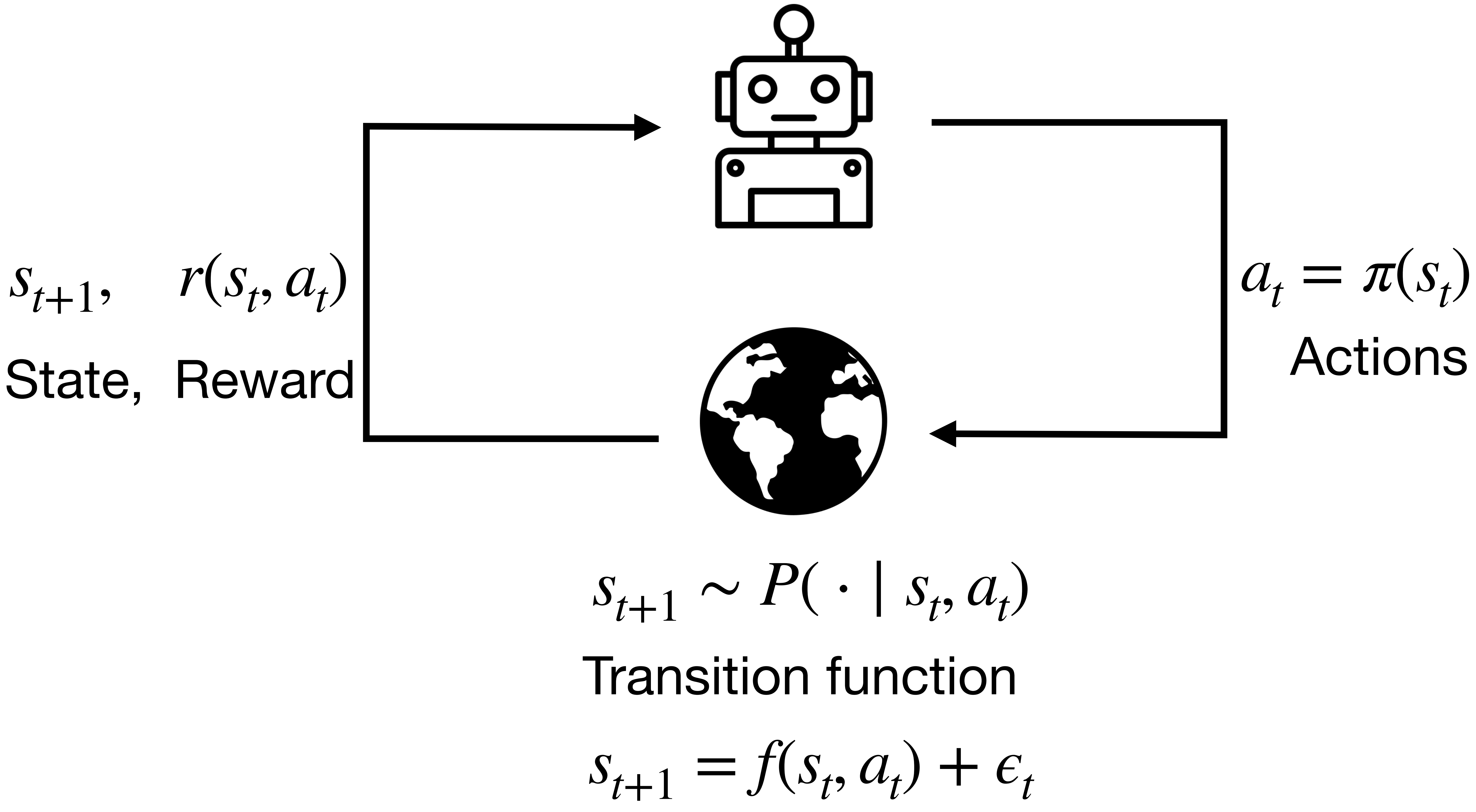
States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0



Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

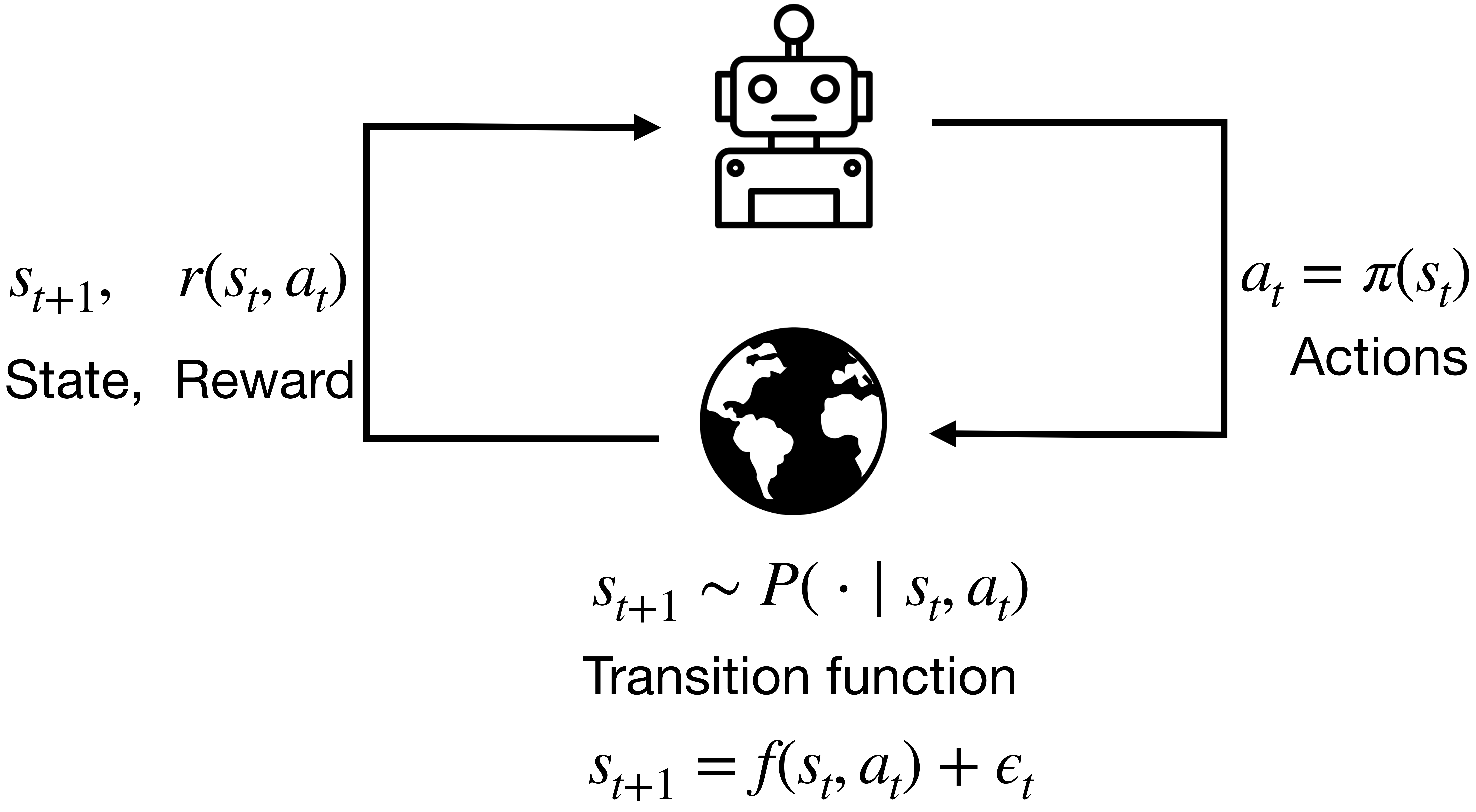
Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$



Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

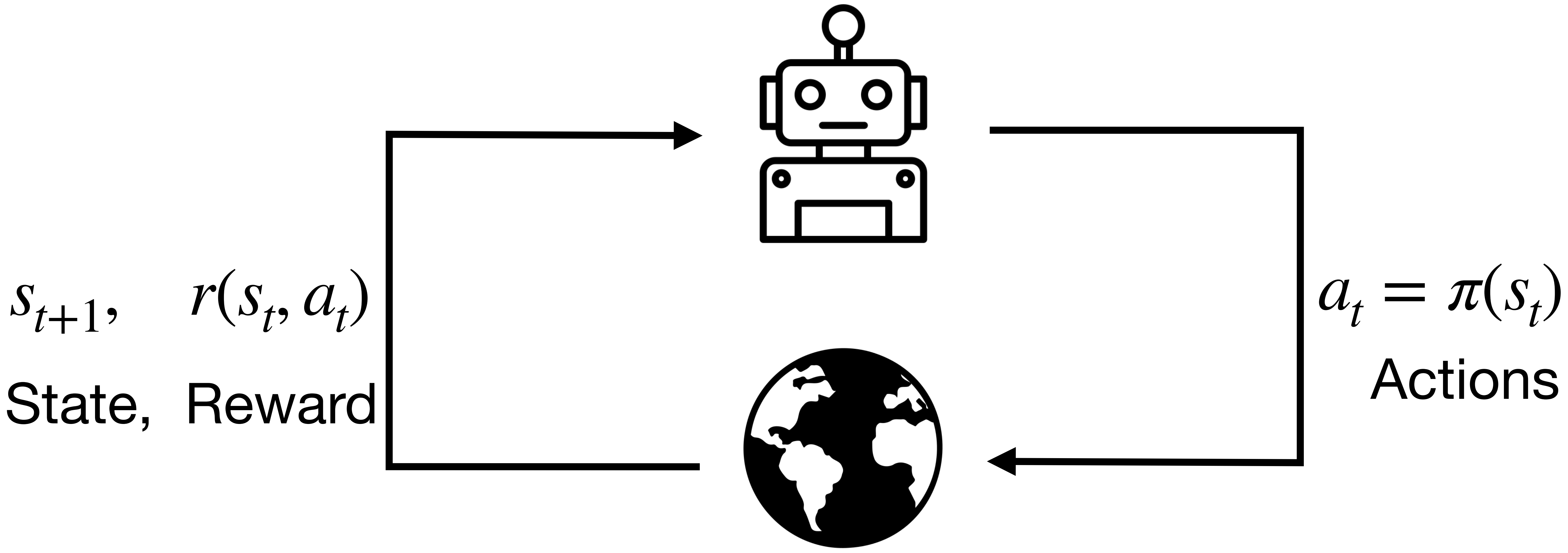
Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$



$s_{t+1} \sim P(\cdot \mid s_t, a_t)$

Transition function

$s_{t+1} = f(s_t, a_t) + \epsilon_t$

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right]$$

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right]$$

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right]$$

Value function:

FCAI

fcai.fi

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

Value function:

$$V_\pi(s) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

Value function:

$$V_\pi(s) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

FCAI

fcai.fi

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

Value function:

$$V_\pi(s) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\Big]$$

Action-value function (aka Q-function):

Reinforcement Learning

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

Transition function $P(s_{t+1} \mid s_t, a_t)$

Reward function $r_t = r(s_t, a_t)$

Start state s_0

Discount factor $\gamma \in [0,1]$

Policy $\pi : \mathcal{S} \to \mathcal{A}$

Goal:

$$\max_{\pi} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right]$$

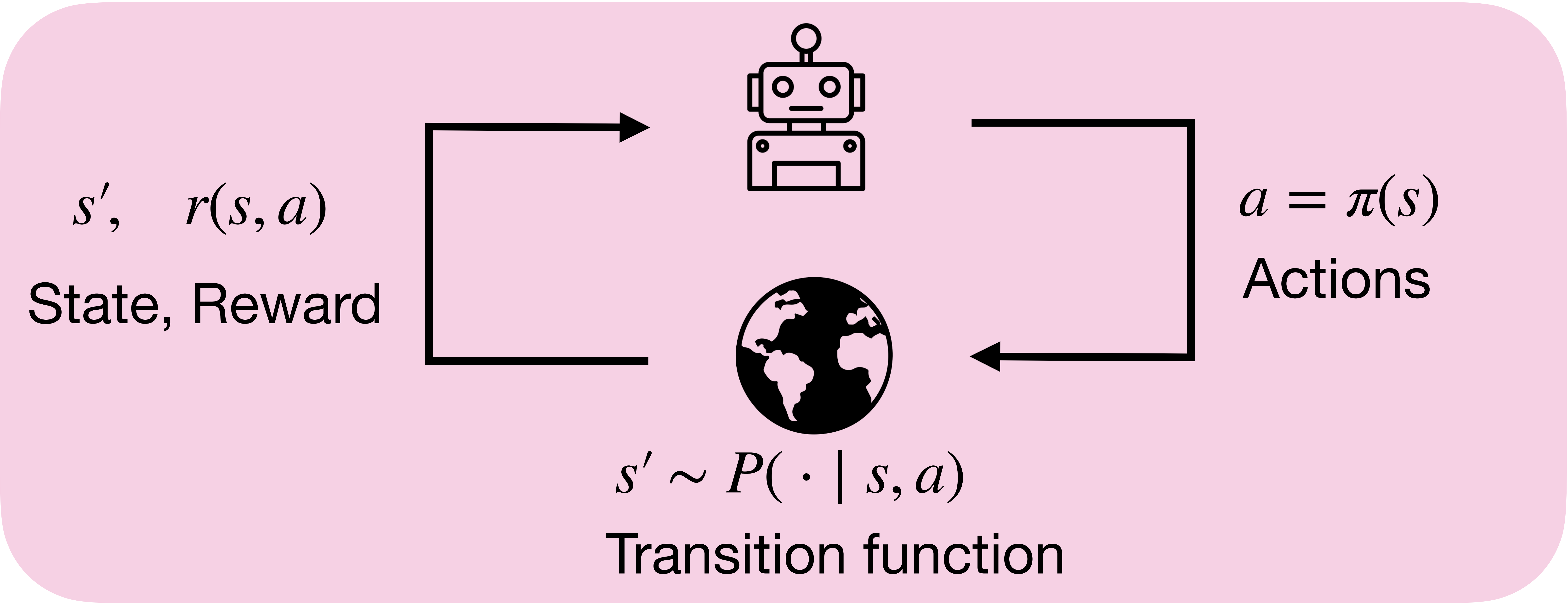
Value function:

$$V_\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, \pi\right]$$

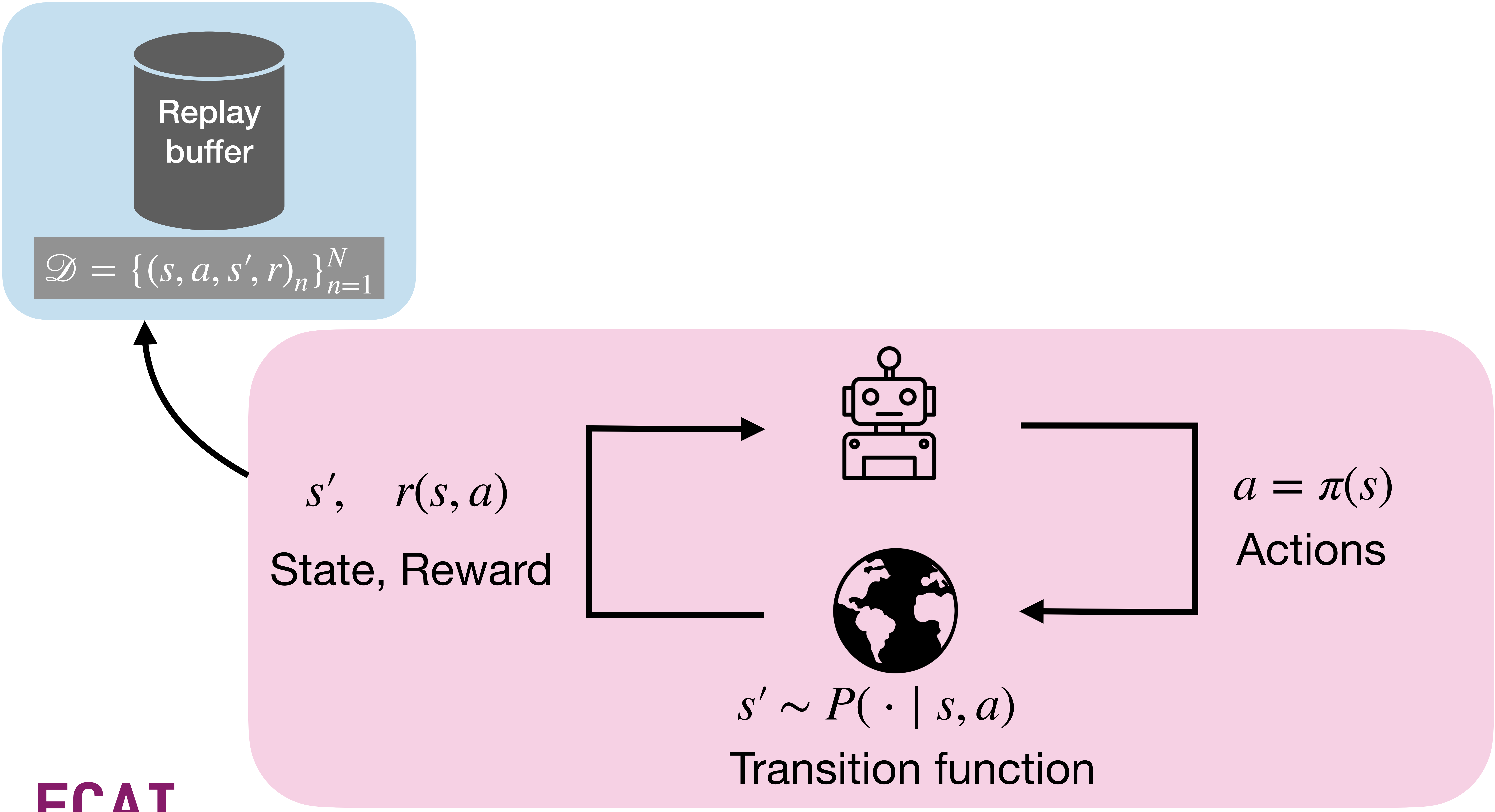
Action-value function (aka Q-function):

$$Q_\pi(s, a) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right]$$

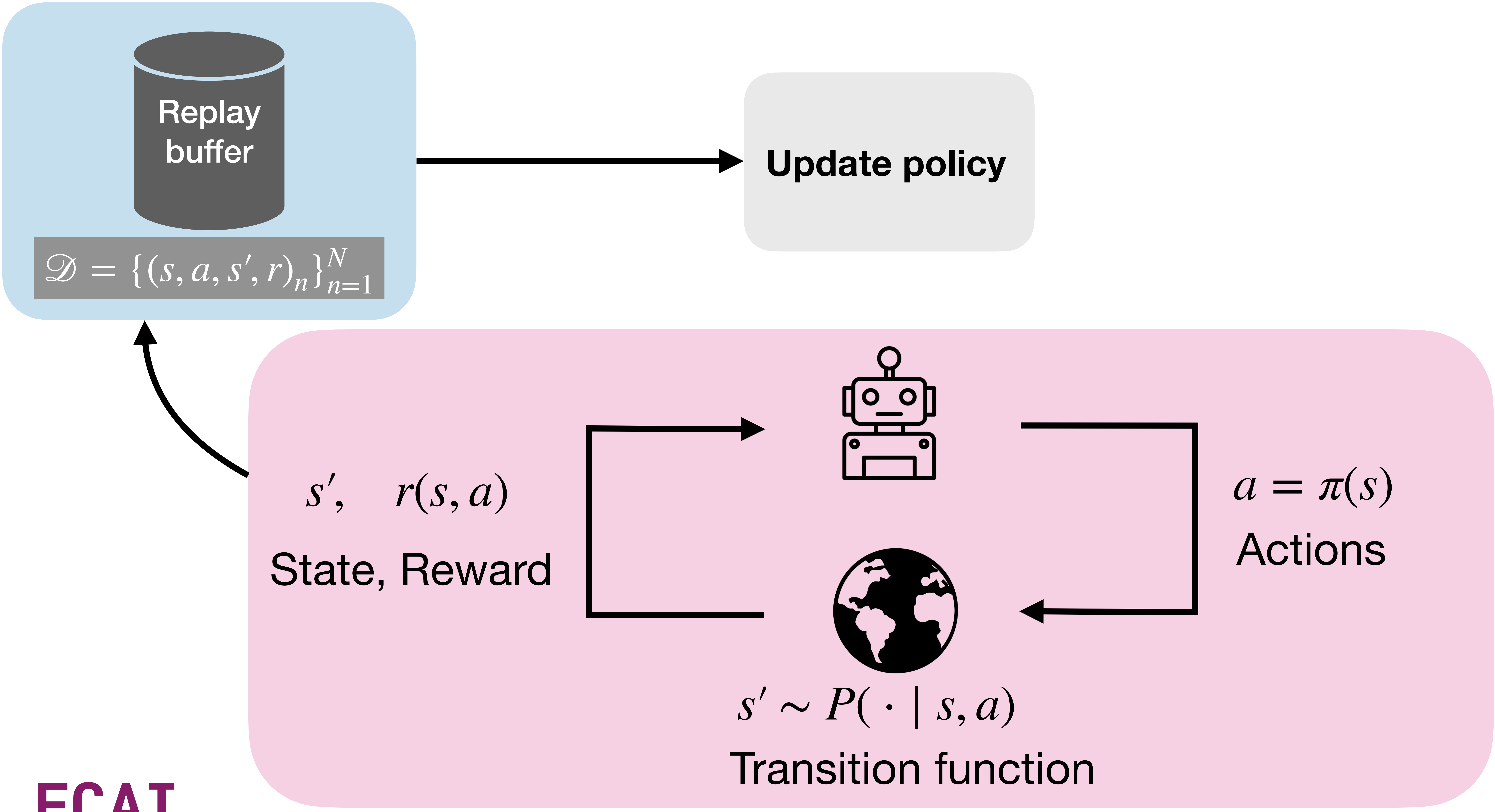
Reinforcement Learning



Reinforcement Learning



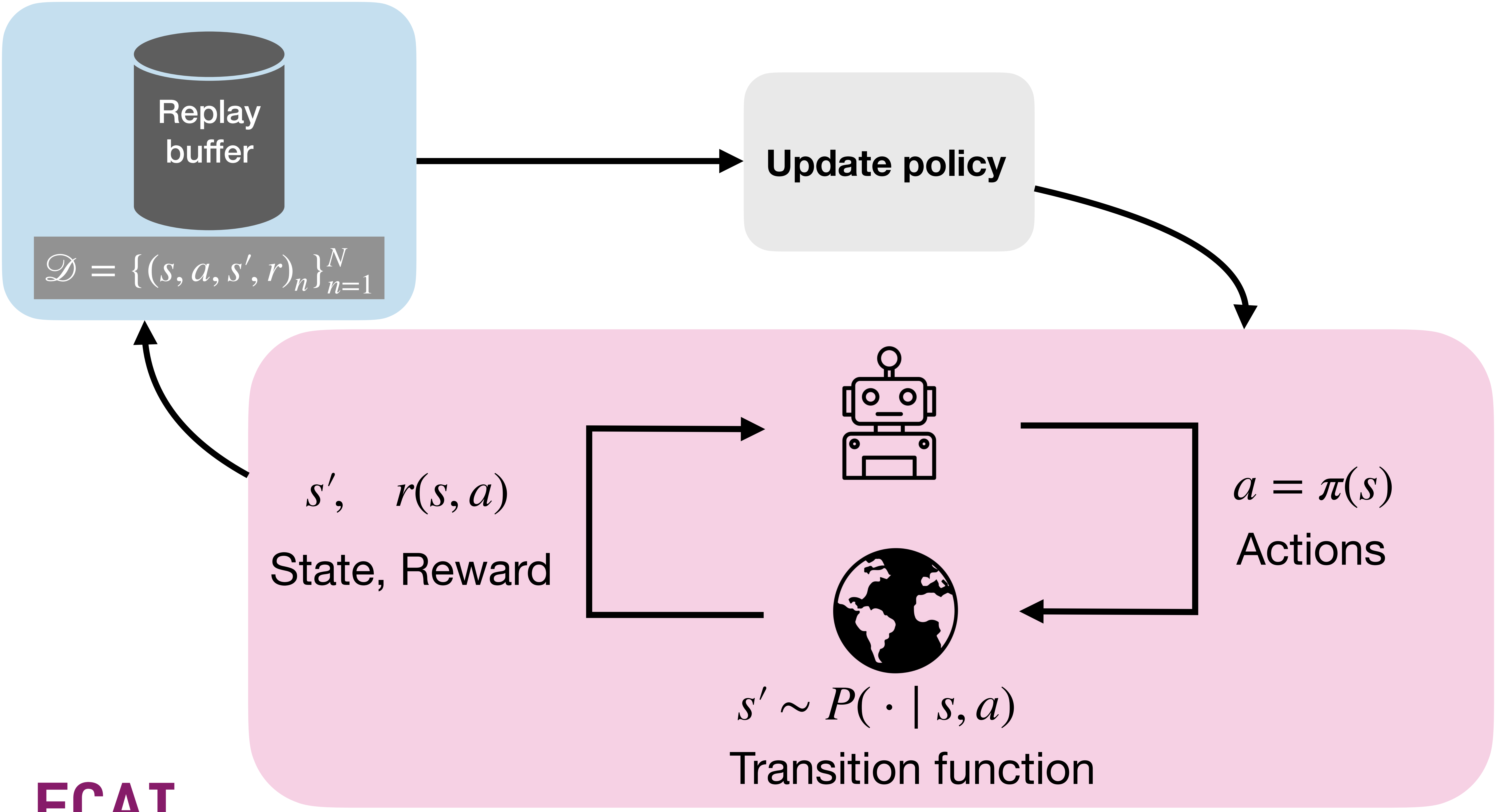
Reinforcement Learning



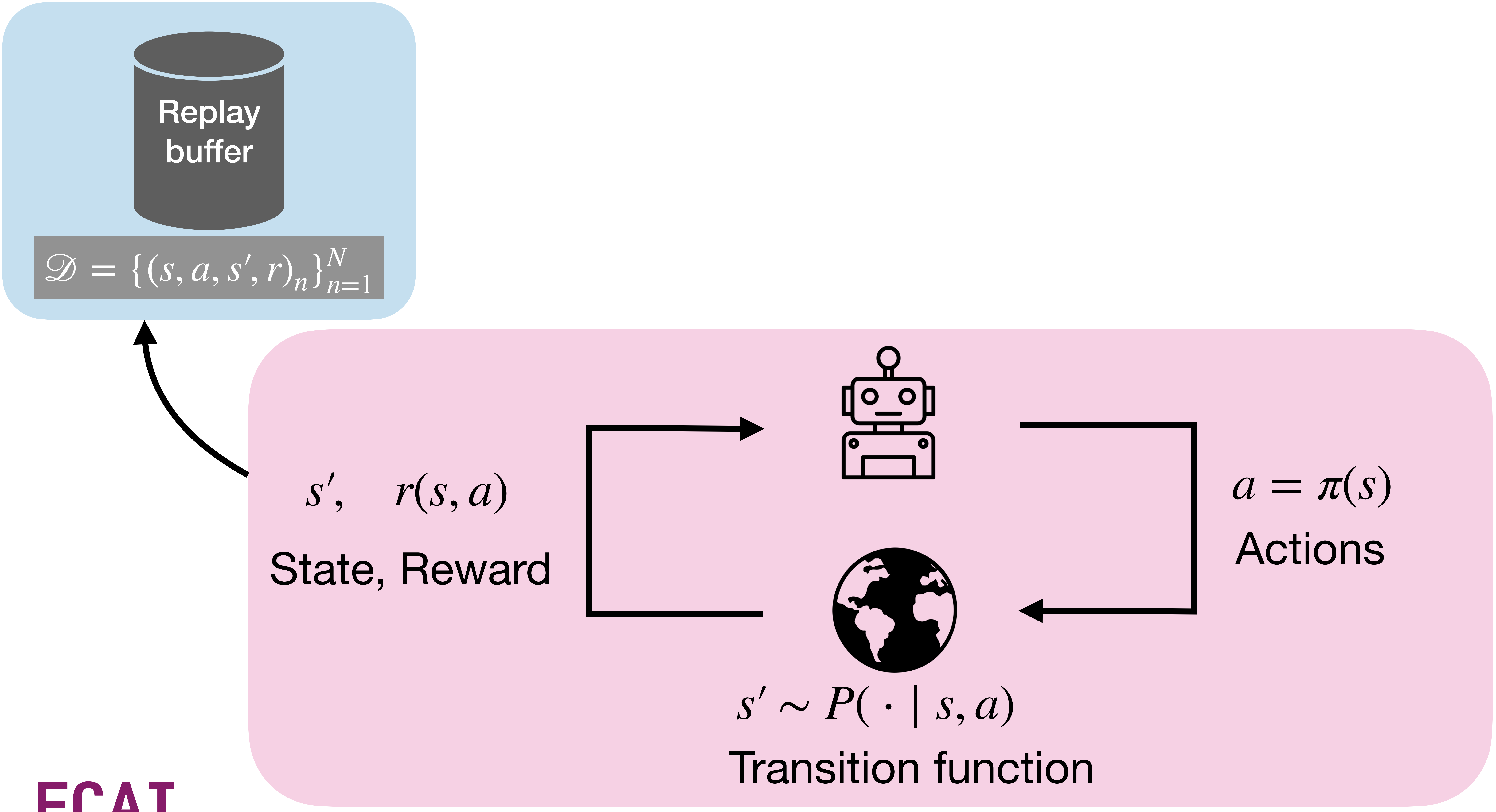
FCAI

fcai.fi

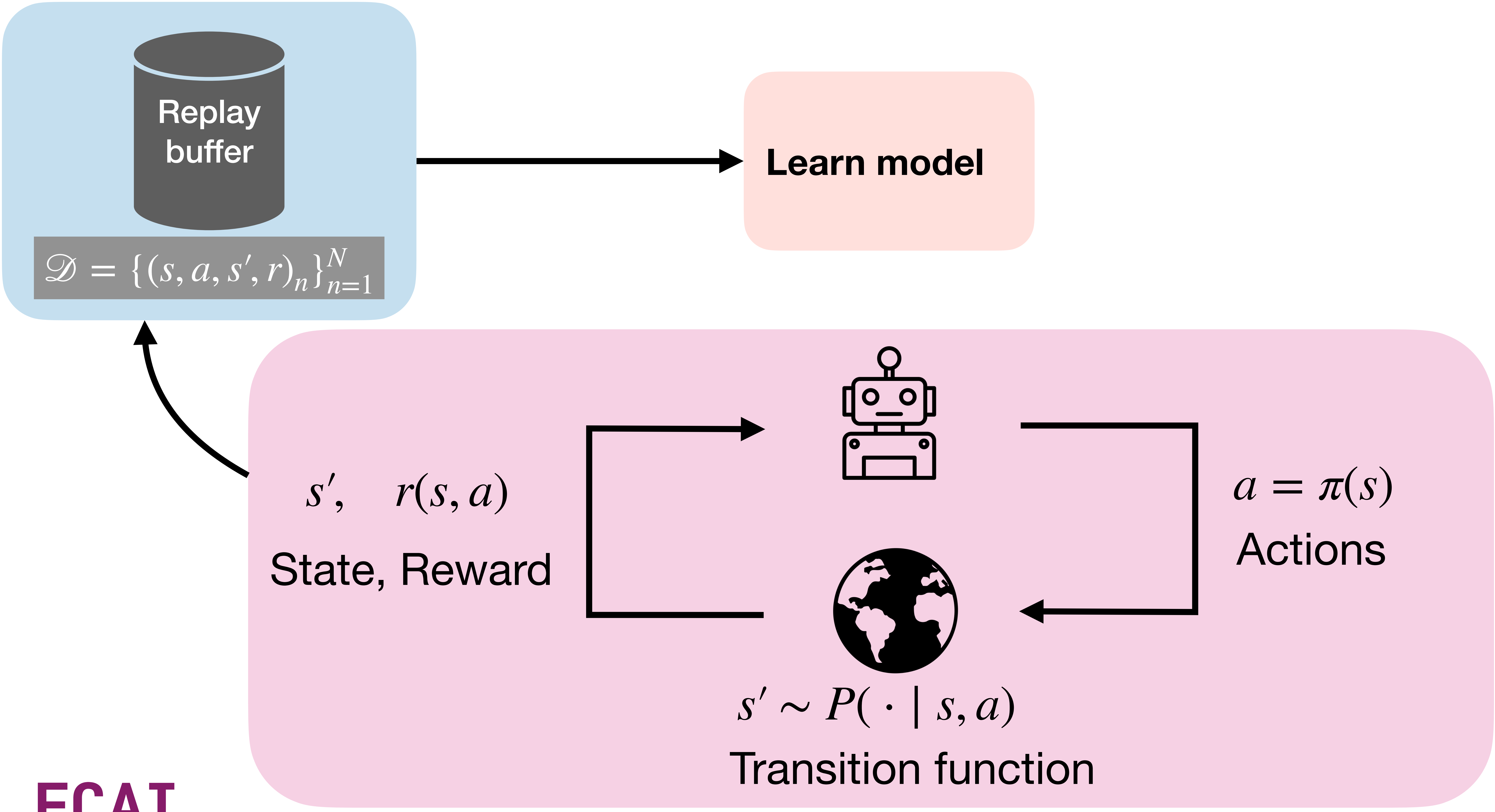
Reinforcement Learning



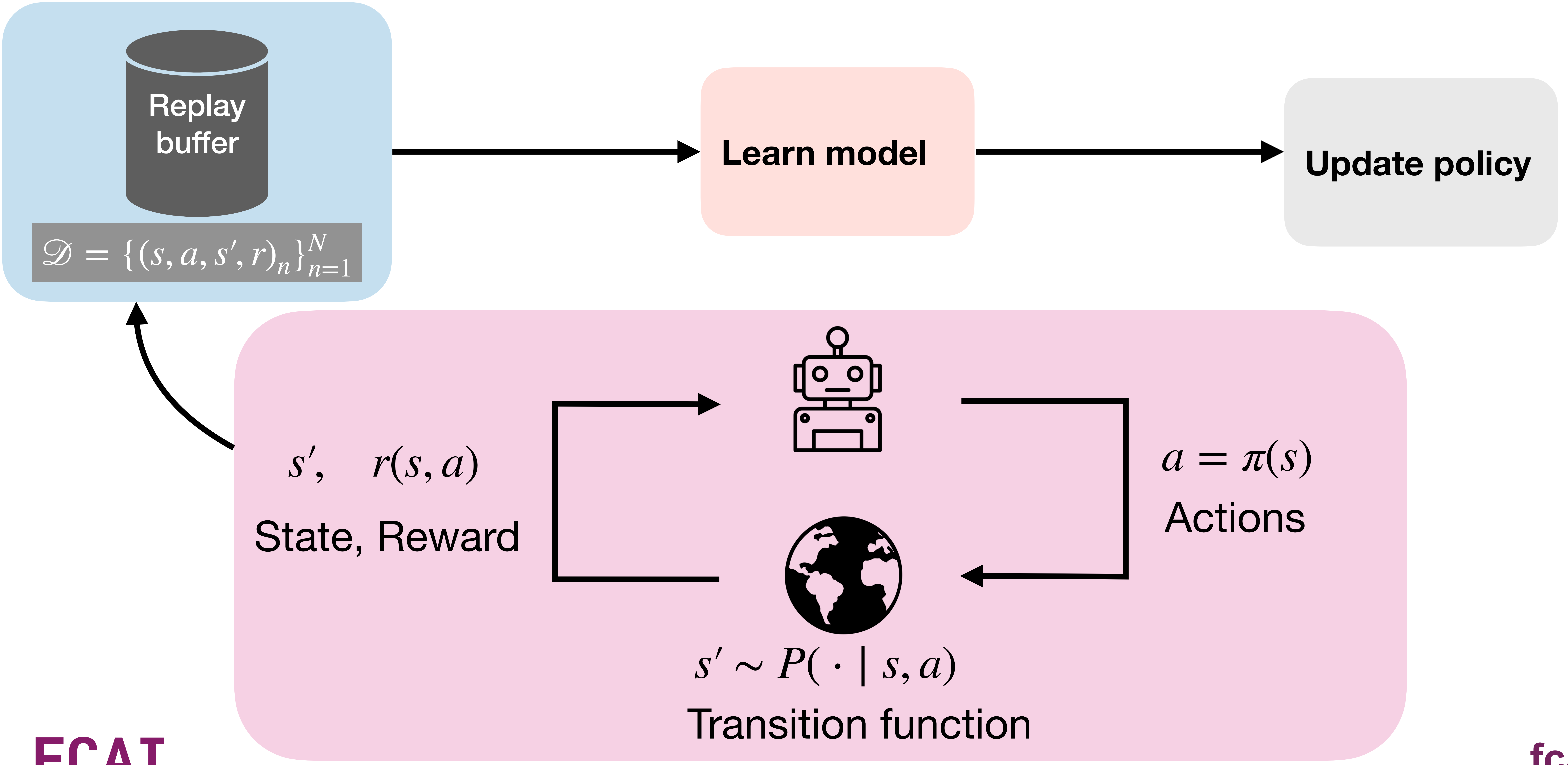
Model-based Reinforcement Learning



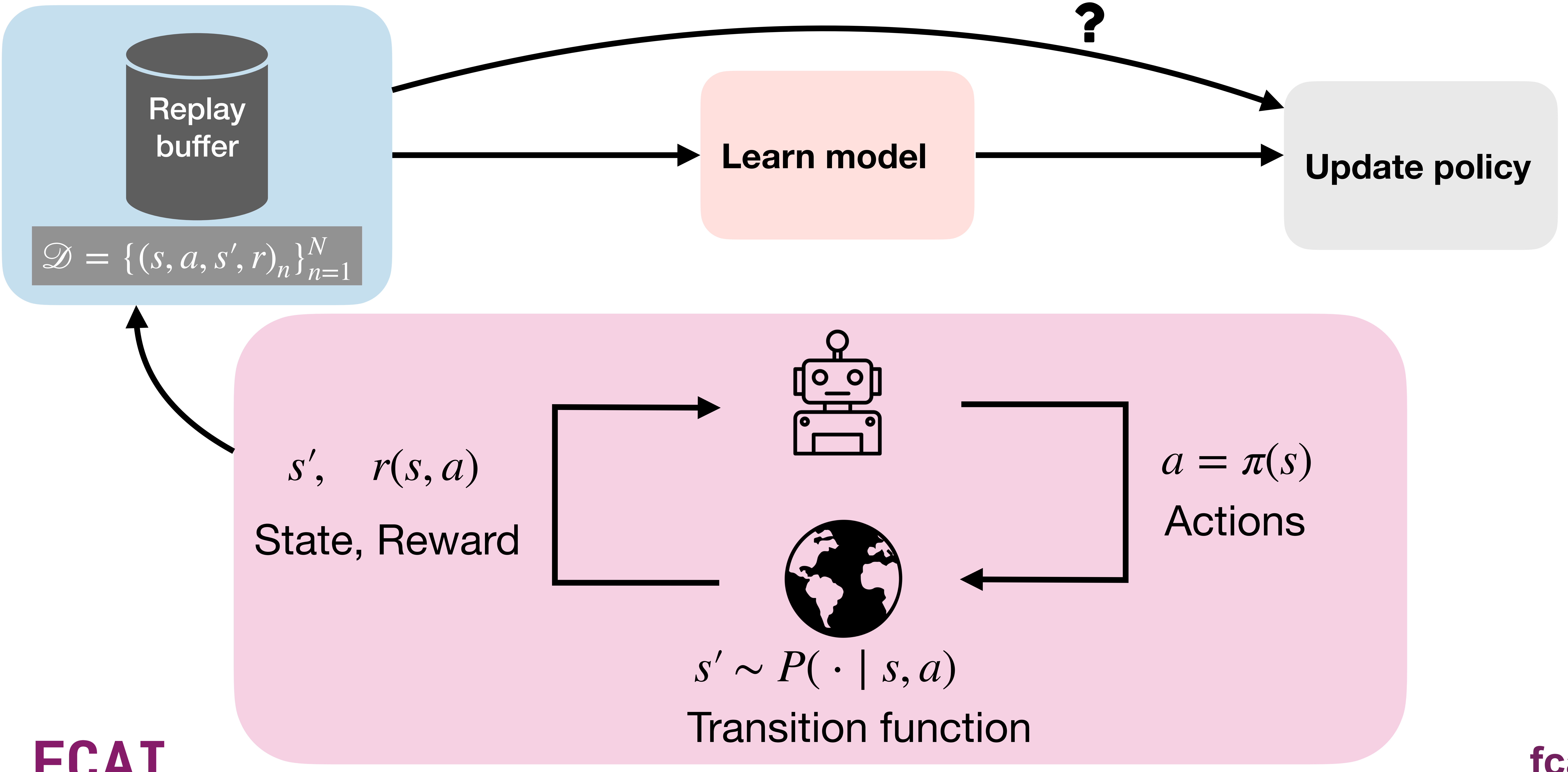
Model-based Reinforcement Learning



Model-based Reinforcement Learning



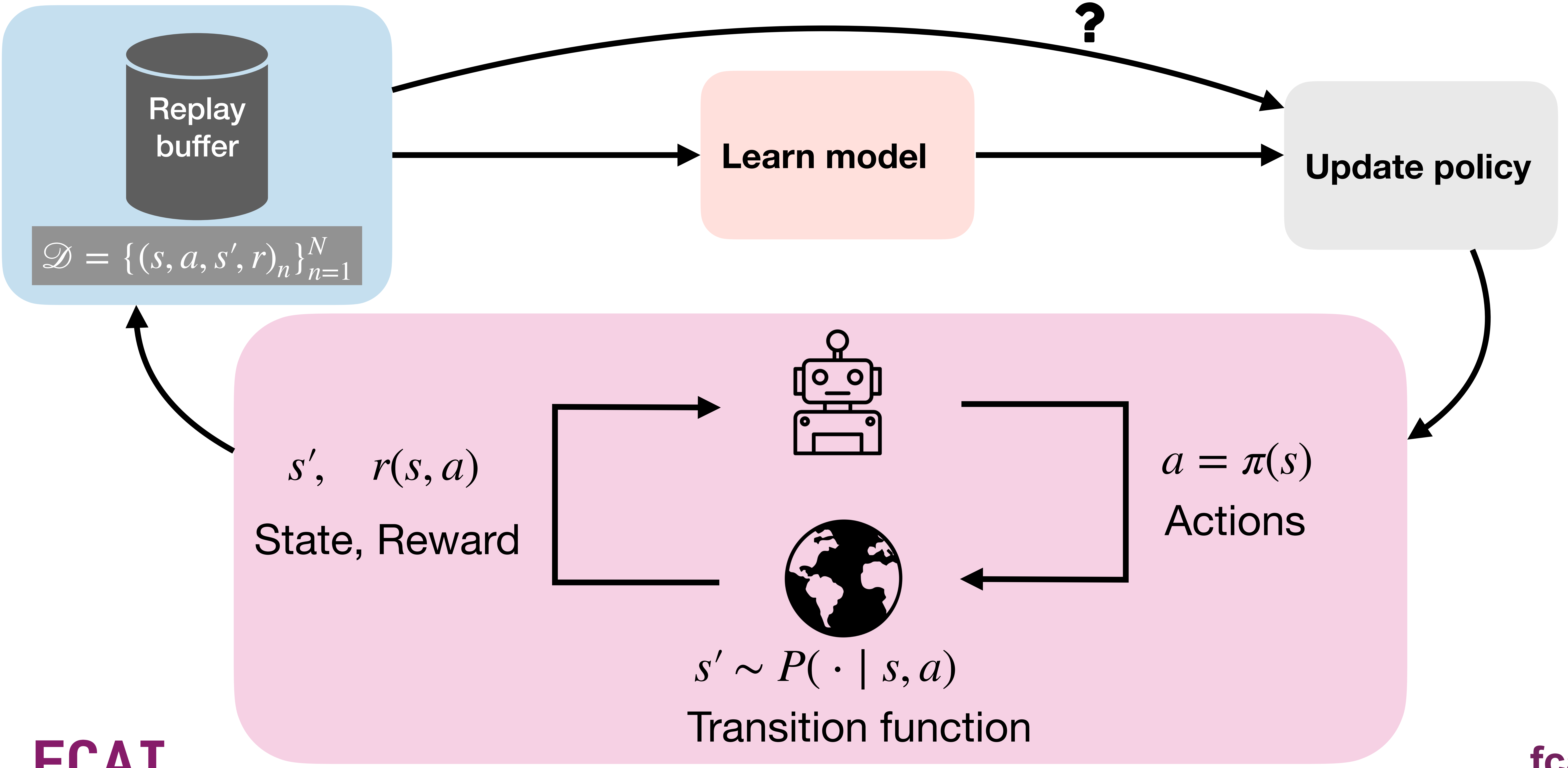
Model-based Reinforcement Learning



FCAI

fcai.fi

Model-based Reinforcement Learning



Reinforcement Learning Has Its Drawbacks

Reinforcement Learning Has Its Drawbacks

Reinforcement Learning Has Its Drawbacks

Reinforcement Learning Has Its Drawbacks

Reinforcement Learning Has Its Drawbacks

RL has a sample efficiency problem!

Model-free vs Model-based RL

Model-free vs Model-based RL

	Model-free	Model-based

Model-free vs Model-based RL

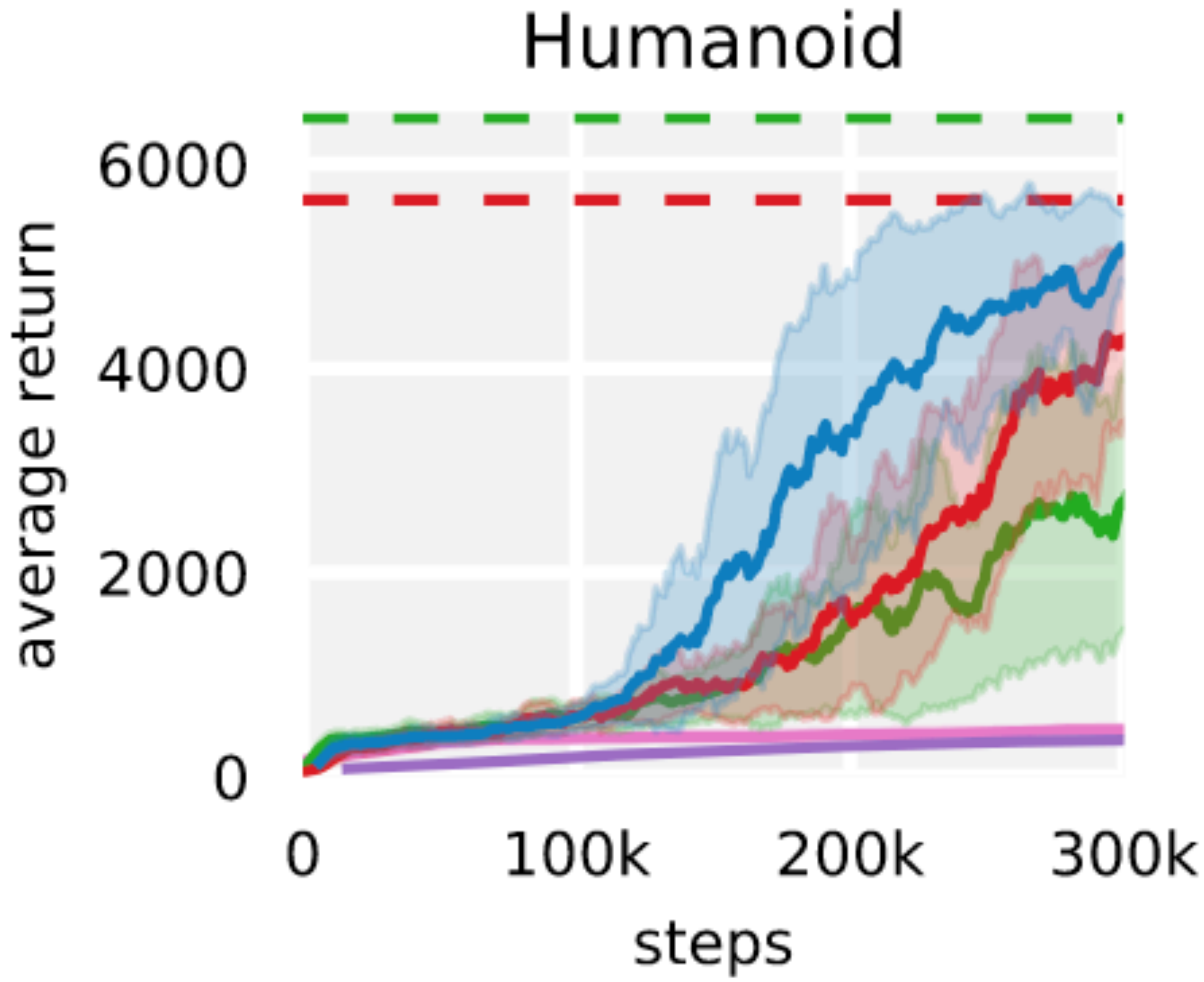
	Model-free	Model-based
Asymptotic performance	✅	Depends

Model-free vs Model-based RL

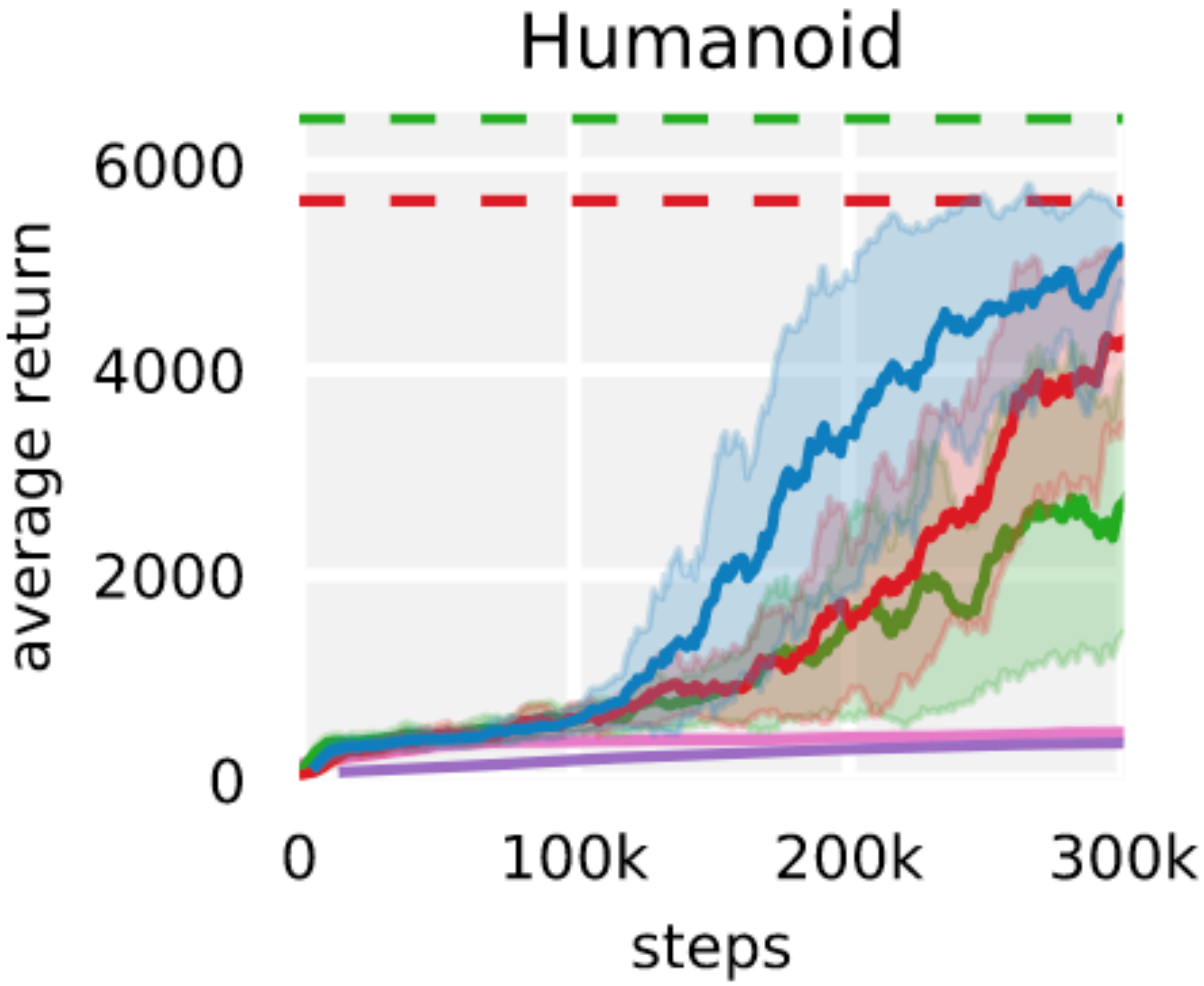
	Model-free	Model-based
Asymptotic performance	✅	Depends
Sample efficiency	❌	✅

Model-free vs Model-based RL

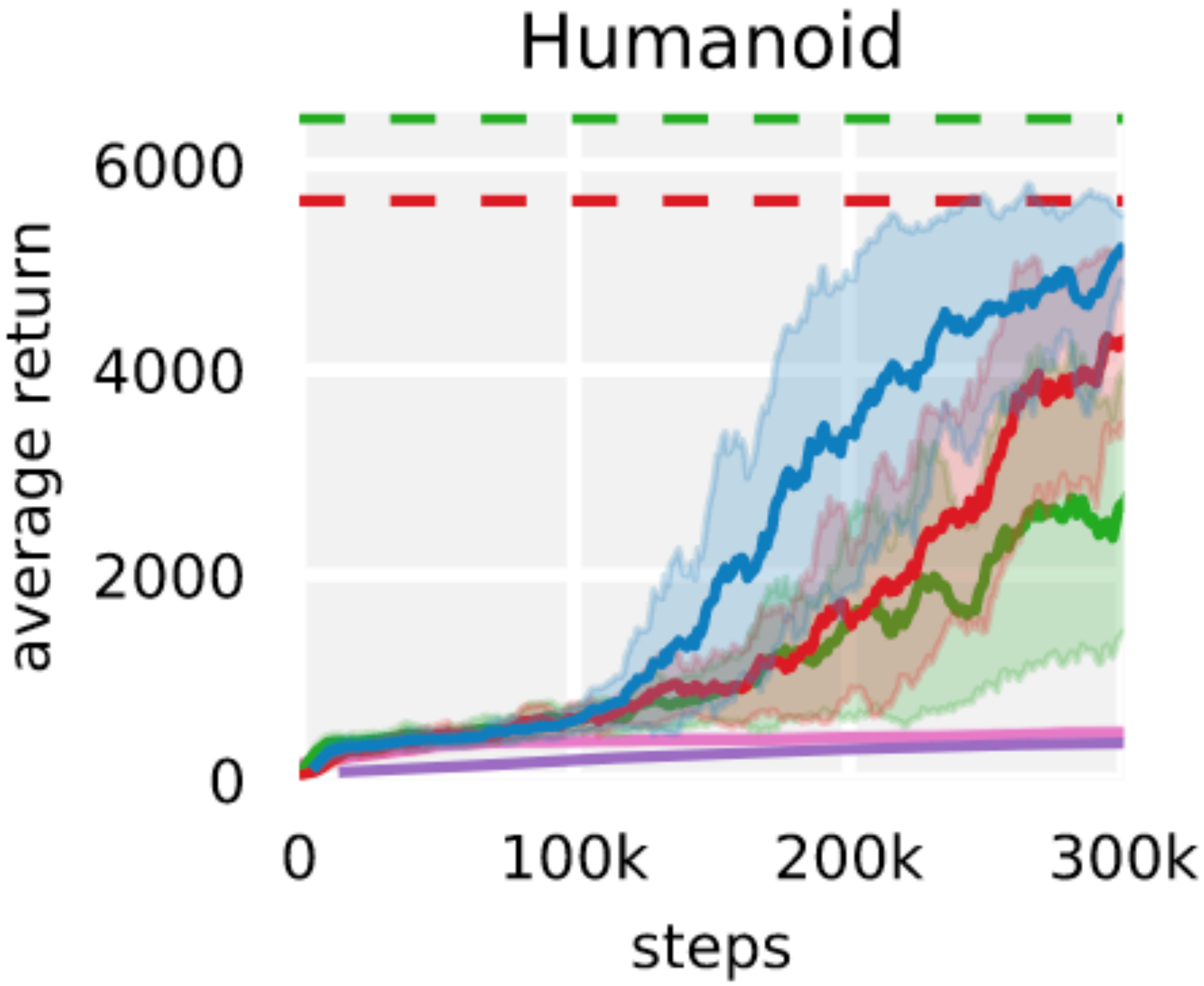
	Model-
Asymptotic performance	✅
Sample efficiency	❌



Model-free vs Model-based RL

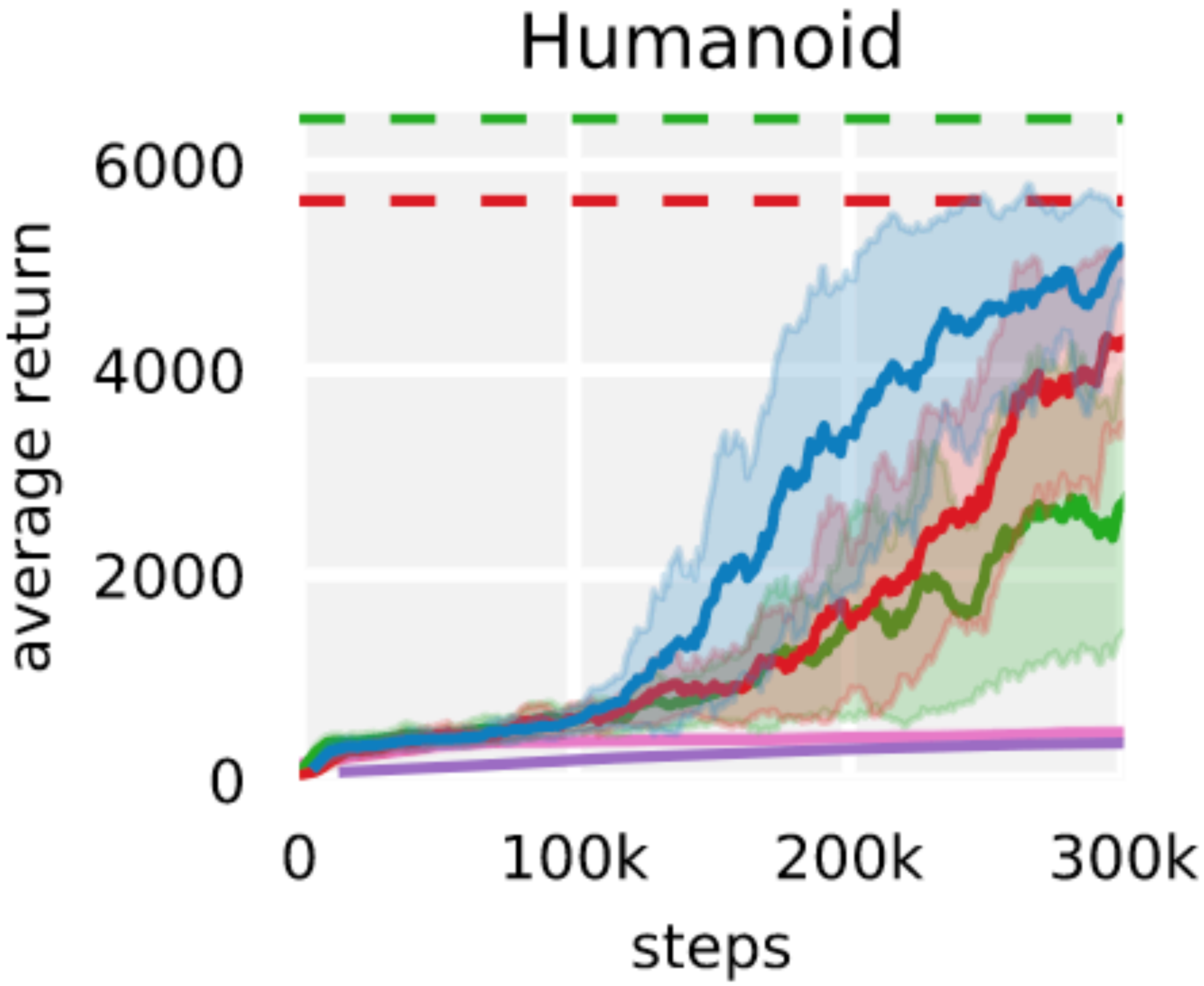


Model-free vs Model-based RL



Tutorial on Model-Based Methods in Reinforcement Learning @ ICML 2020 by Igor Mordatch and Jessica Hamrick

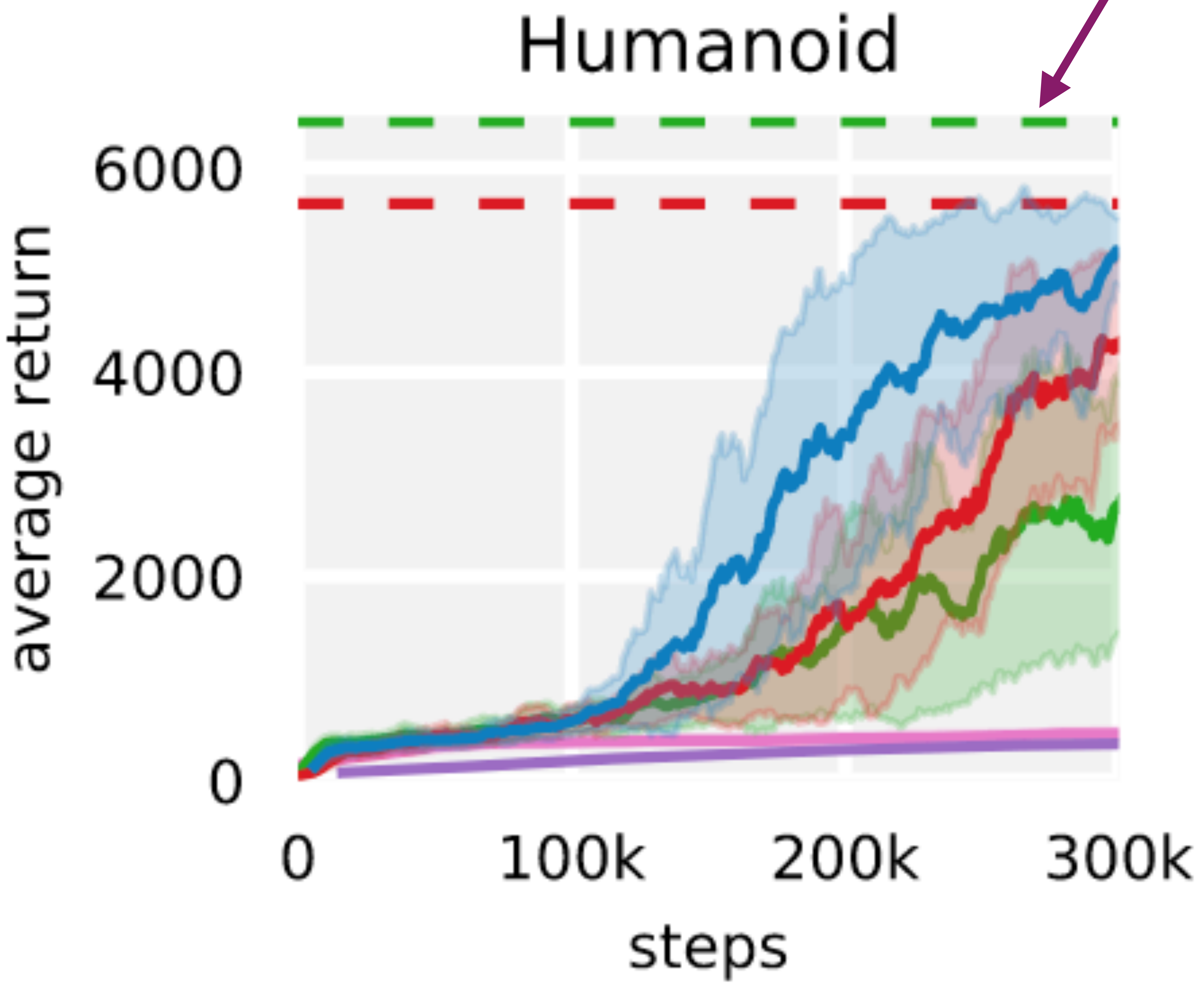
Model-free vs Model-based RL



Tutorial on Model-Based Methods in Reinforcement Learning @ ICML 2020 by Igor Mordatch and Jessica Hamrick

Model-free vs Model-based RL

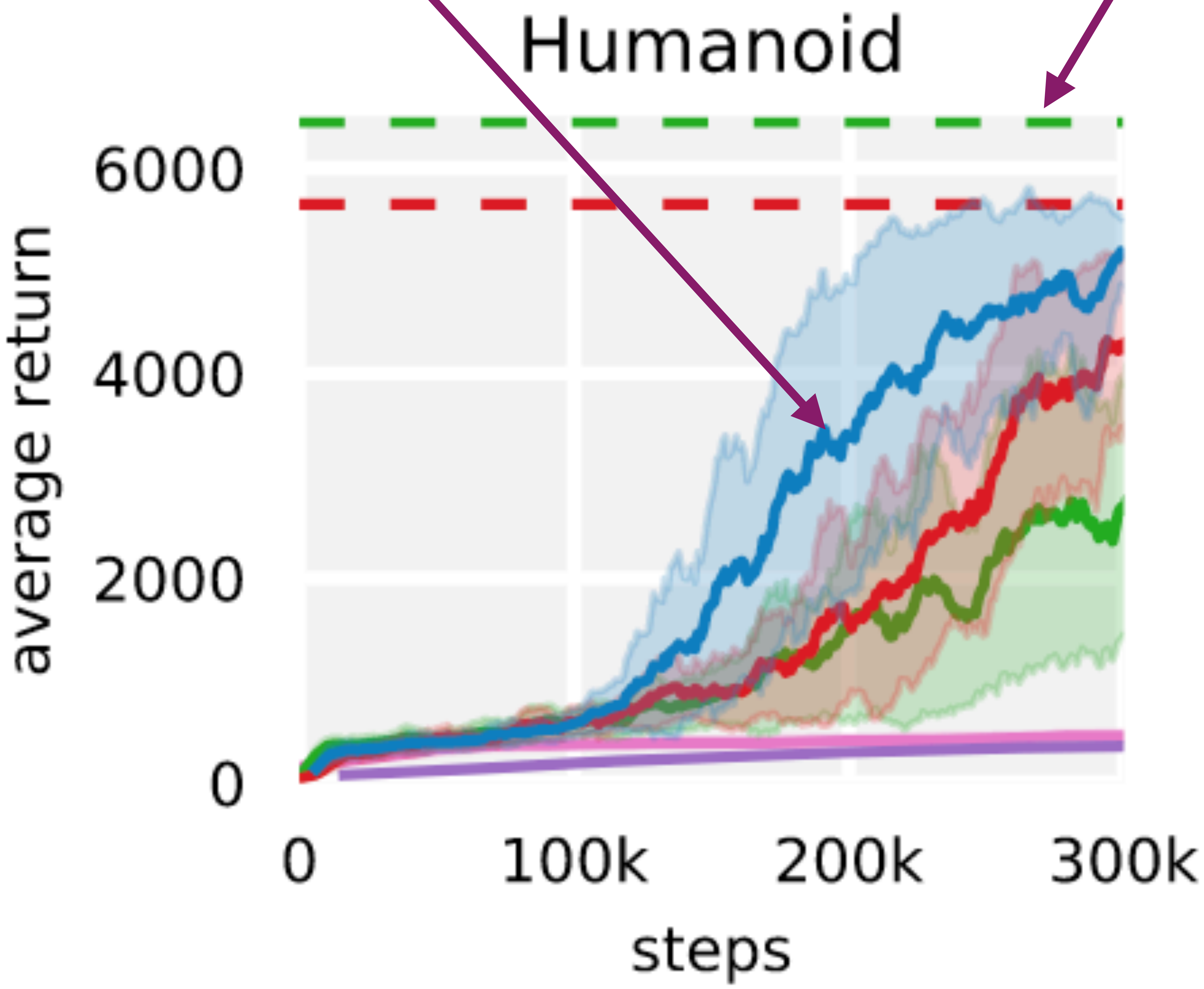
Best asymptotic performance



Model-free vs Model-based RL

Sample efficient

Best asymptotic performance



Model-free vs Model-based RL

	Model-free	Model-based
Asymptotic performance	✅	Depends
Sample efficiency	❌	✅

Model-free vs Model-based RL

	Model-free	Model-based
Asymptotic performance	✅	Depends
Sample efficiency	❌	✅
Computation at deployment	✅	❌/✅

Model-free vs Model-based RL

	Model-free	Model-based
Asymptotic performance	✅	Depends
Sample efficiency	❌	✅
Computation at deployment	✅	❌/✅
Adapting to new tasks	❌	✅

Model-free vs Model-based RL

	Model-free	Model-based
Asymptotic performance	✅	Depends
Sample efficiency	❌	✅
Computation at deployment	✅	❌/✅
Adapting to new tasks	❌	✅
Exploration	❌	✅

Tutorial on Model-Based Methods in Reinforcement Learning @ ICML 2020 by Igor Mordatch and Jessica Hamrick

What is a “Model”?

*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

What is a "Model"?

*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model $s_{t+1} = f(s_t, a_t)$

What is a “Model”?

Definition: a model is a representation that ***explicitly*** *encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model	$s_{t+1} = f(s_t, a_t)$
Reward model	$r_{t+1} = f(s_t, a_t)$

What is a “Model”?

*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model $s_{t+1} = f(s_t, a_t)$

Reward model $r_{t+1} = f(s_t, a_t)$

Inverse dynamics model $a_t = f^{-1}(s_t, s_{t+1})$

What is a “Model”?

Definition: a model is a representation that ***explicitly*** *encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model $s_{t+1} = f(s_t, a_t)$

Reward model $r_{t+1} = f(s_t, a_t)$

Inverse dynamics model $a_t = f^{-1}(s_t, s_{t+1})$

Model of distance $d_{ij} = f_d(s_i, s_j)$

What is a “Model”?

*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model	$s_{t+1} = f(s_t, a_t)$
Reward model	$r_{t+1} = f(s_t, a_t)$
Inverse dynamics model	$a_t = f^{-1}(s_t, s_{t+1})$
Model of distance	$d_{ij} = f_d(s_i, s_j)$
Model of future returns	$G_t = Q(s_t, a_t)$

What is a “Model”?

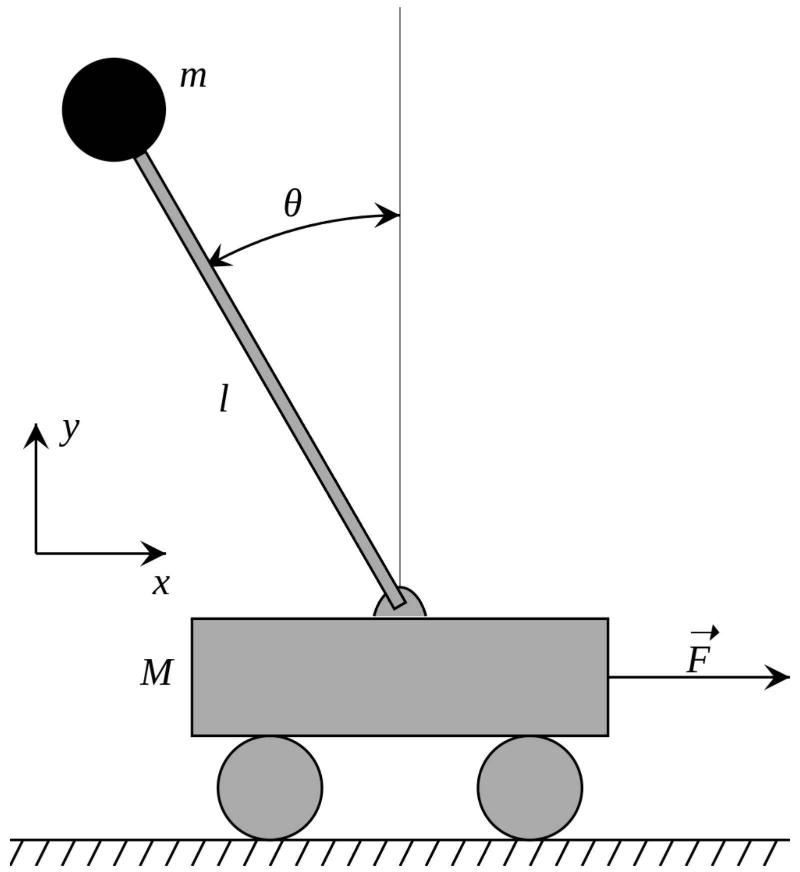
*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

Dynamics (transition) model	$s_{t+1} = f(s_t, a_t)$
Reward model	$r_{t+1} = f(s_t, a_t)$
Inverse dynamics model	$a_t = f^{-1}(s_t, s_{t+1})$
Model of distance	$d_{ij} = f_d(s_i, s_j)$
Model of future returns	$G_t = Q(s_t, a_t)$

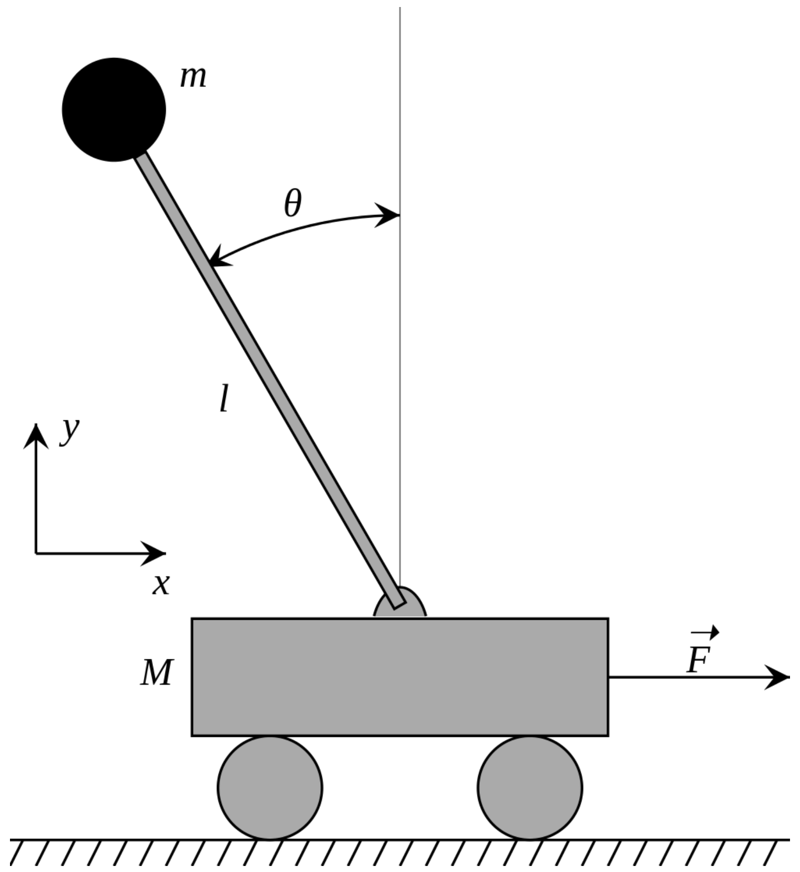
Typically this is what’s meant in model-based RL

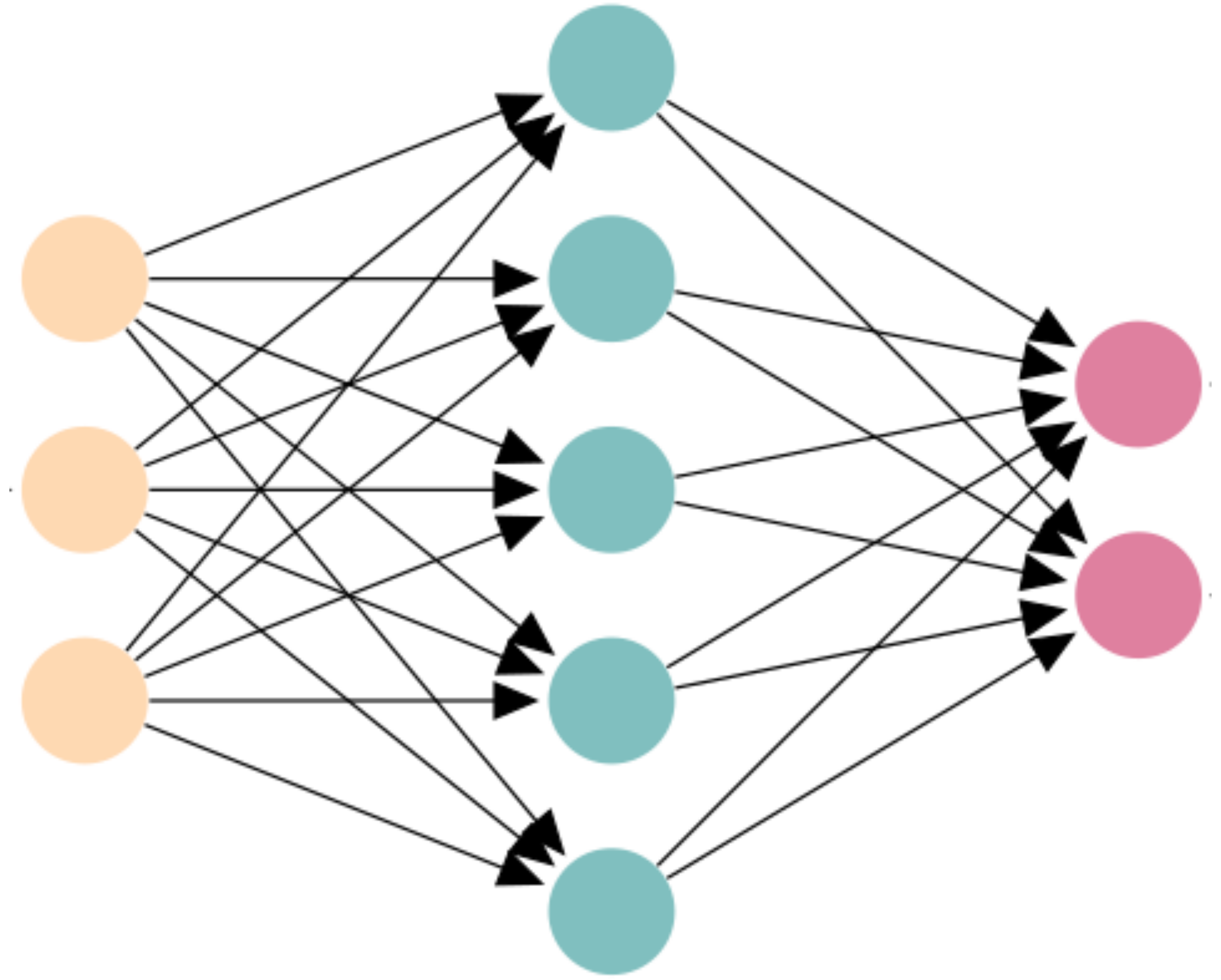
What is a “Model”?

What is a “Model”?

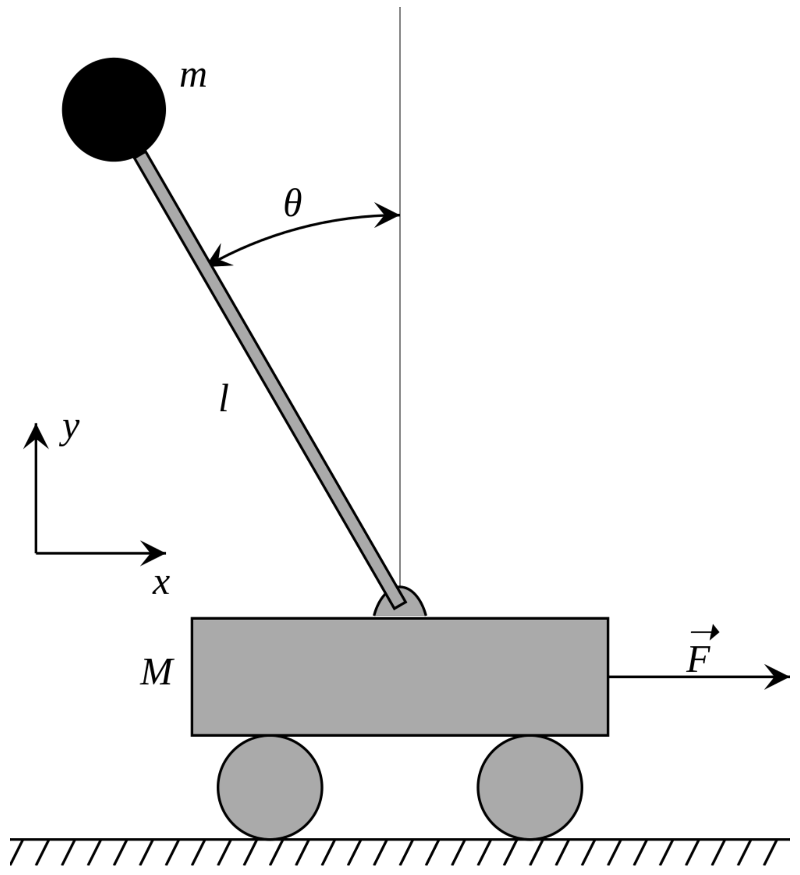


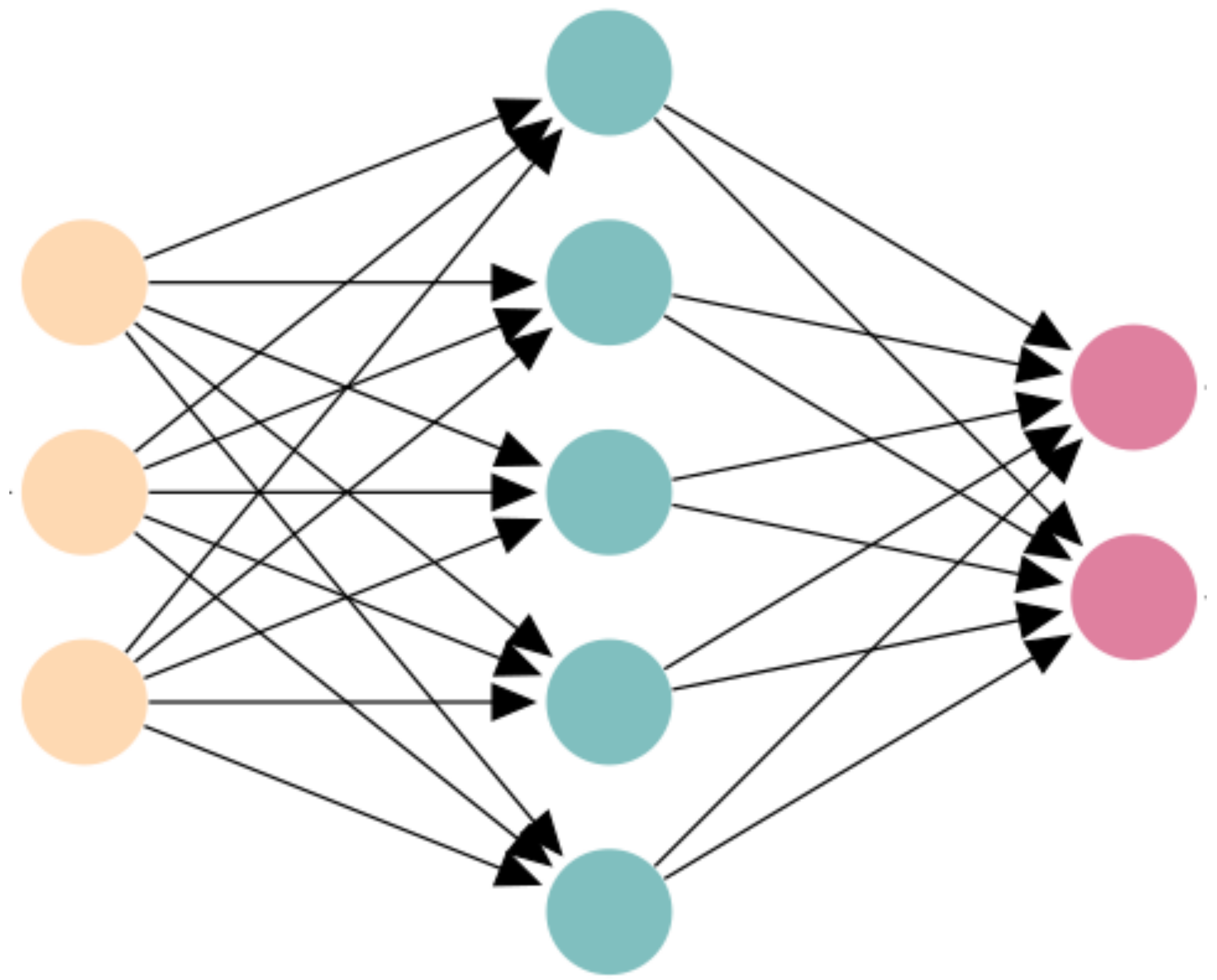
What is a "Model"?

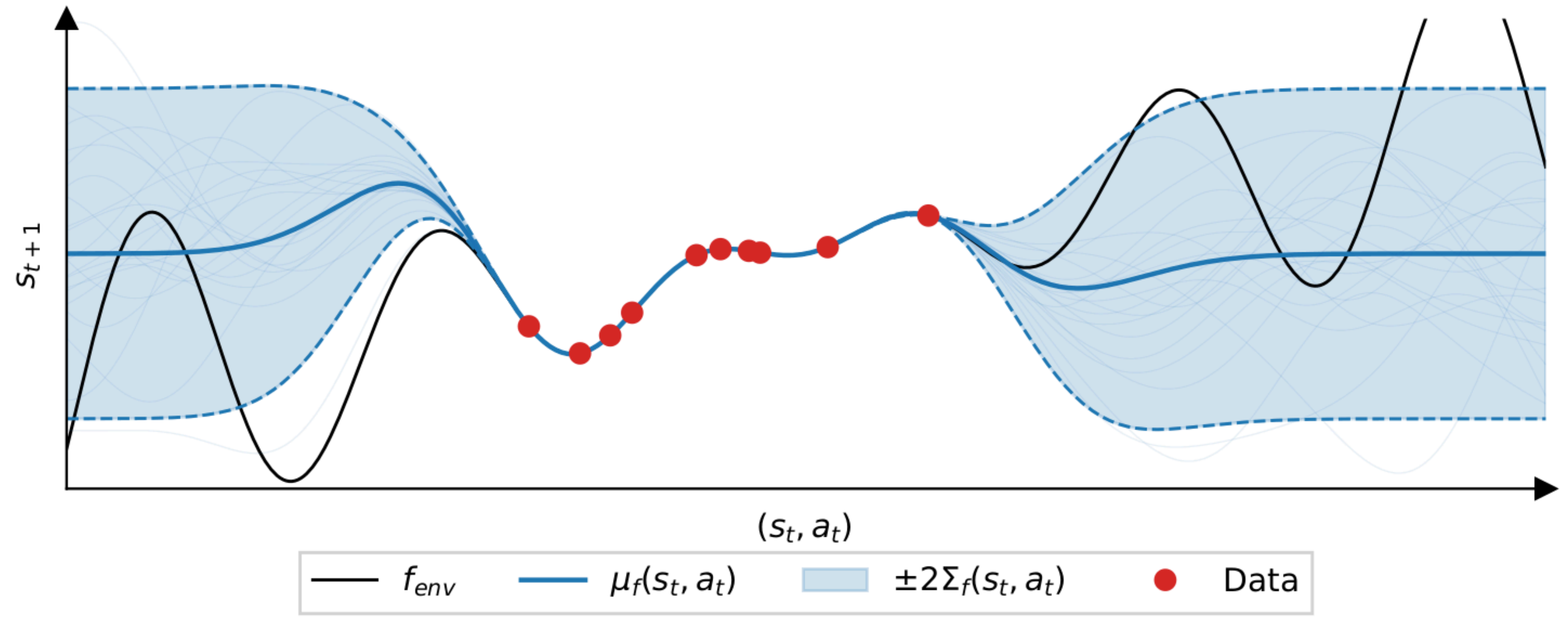




What is a “Model”?







Learning Objectives

Understand

1. ~~What a "model" is in model-based RL~~
2. How a "model" can aid decision making
3. Differences between background and decision-time planning
4. Decision-time planning strategies for continuous actions
5. Sources of uncertainty in model-based RL
6. Rationale and insights for decision-making under uncertainty

Planning

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of
Planning

How Do We Use The "Model"?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

- Find best action for current situation

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

- Find best action for current situation

Background Planning

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

- Find best action for current situation

Background Planning

- Learn (from past data) how to act in any situation

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

- Find best action for current situation

Continuous actions

Background Planning

- Learn (from past data) how to act in any situation

How Do We Use The "Model”?

Background vs Decision-time Planning

Time of Planning

Decision-time Planning

- Find best action for current situation

Background Planning

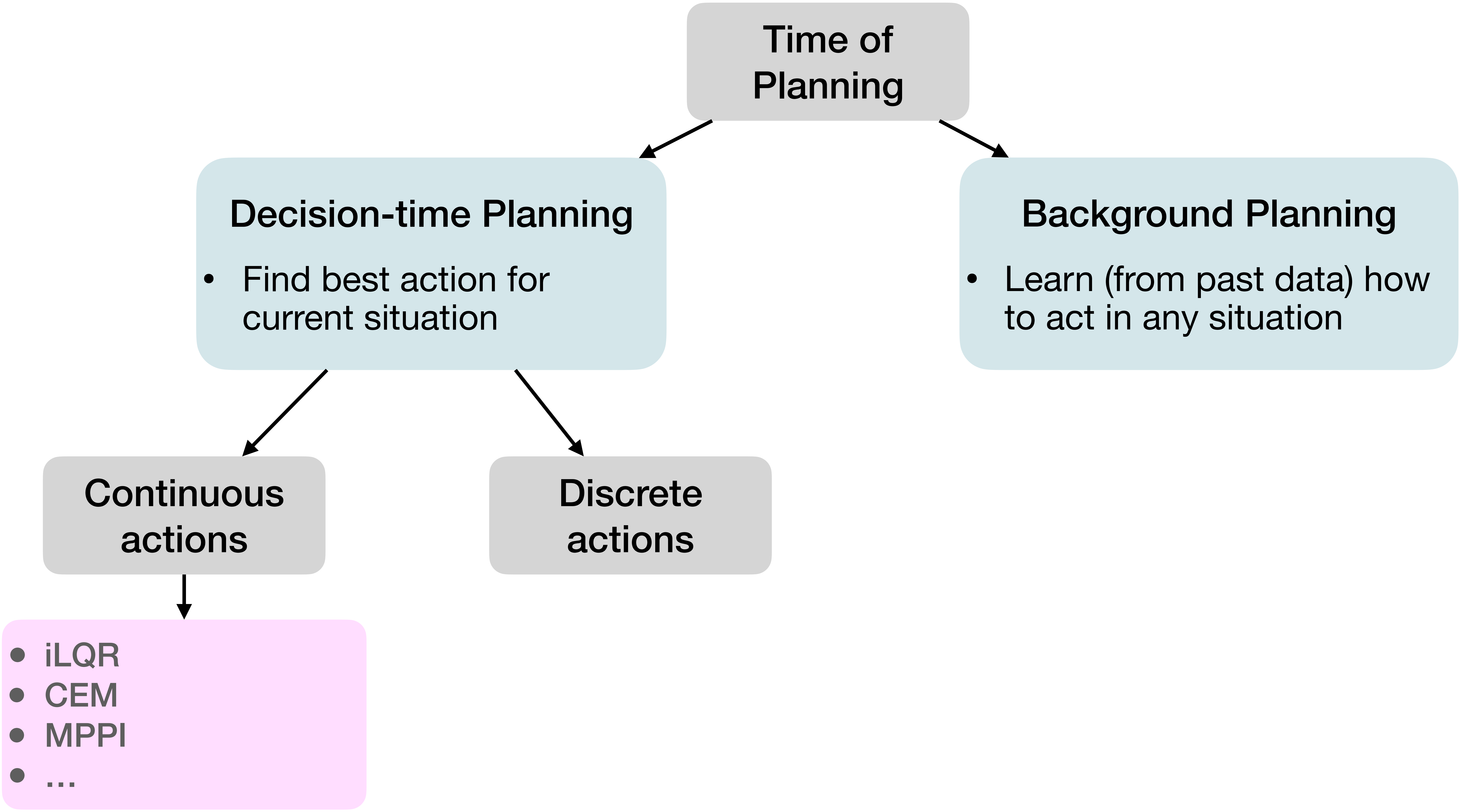
- Learn (from past data) how to act in any situation

Continuous actions

- iLQR
- CEM
- MPPI
- ...

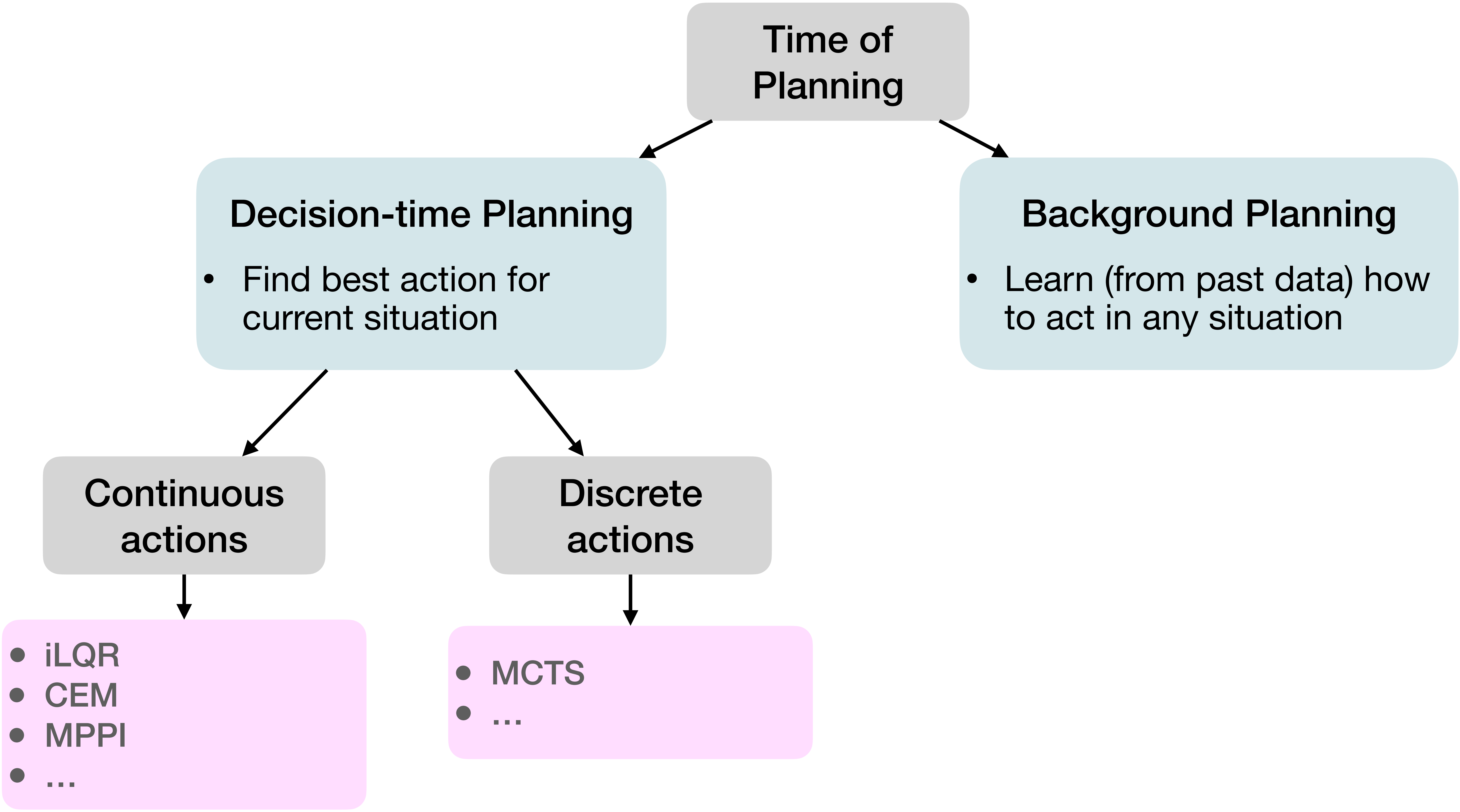
How Do We Use The "Model”?

Background vs Decision-time Planning



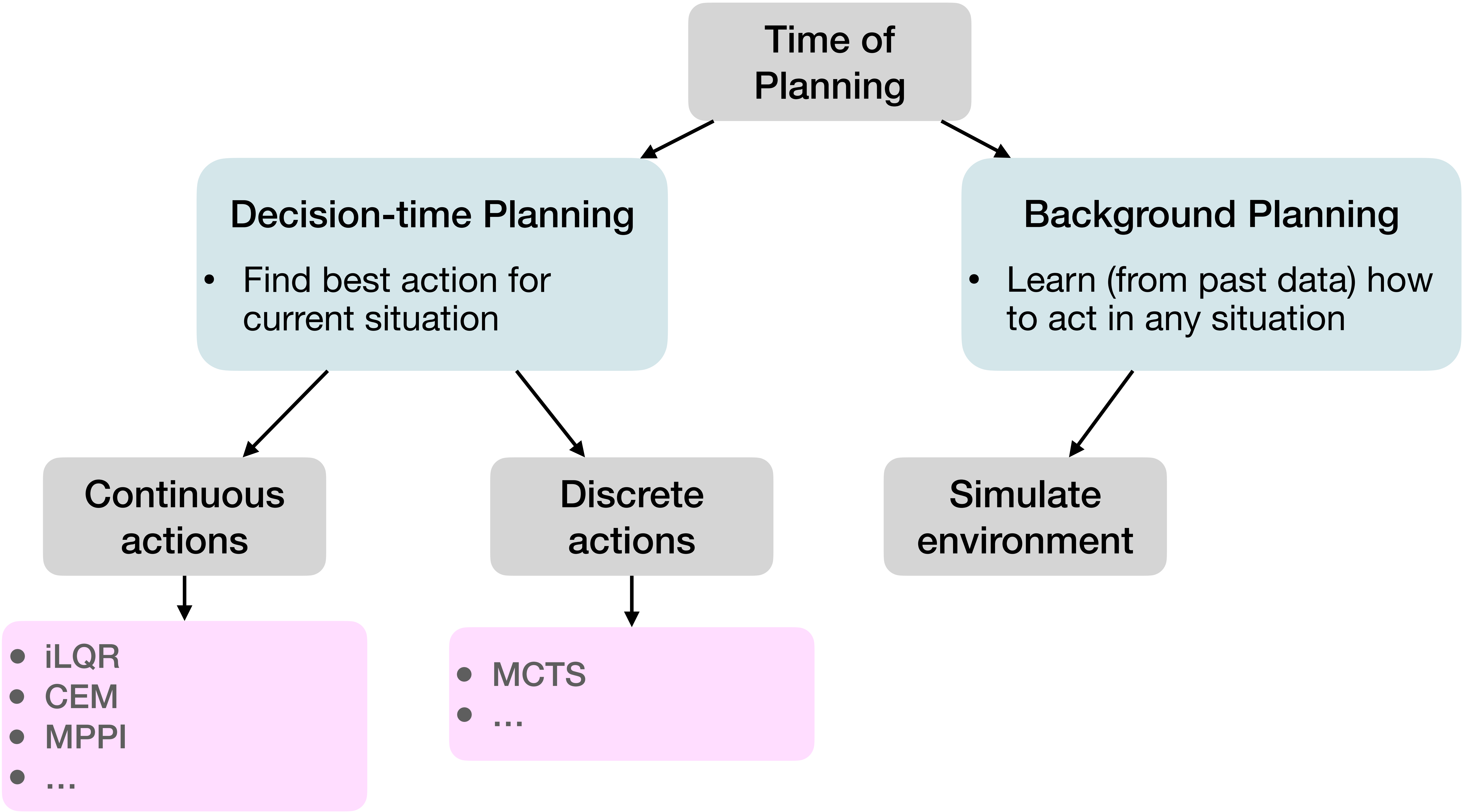
How Do We Use The "Model”?

Background vs Decision-time Planning



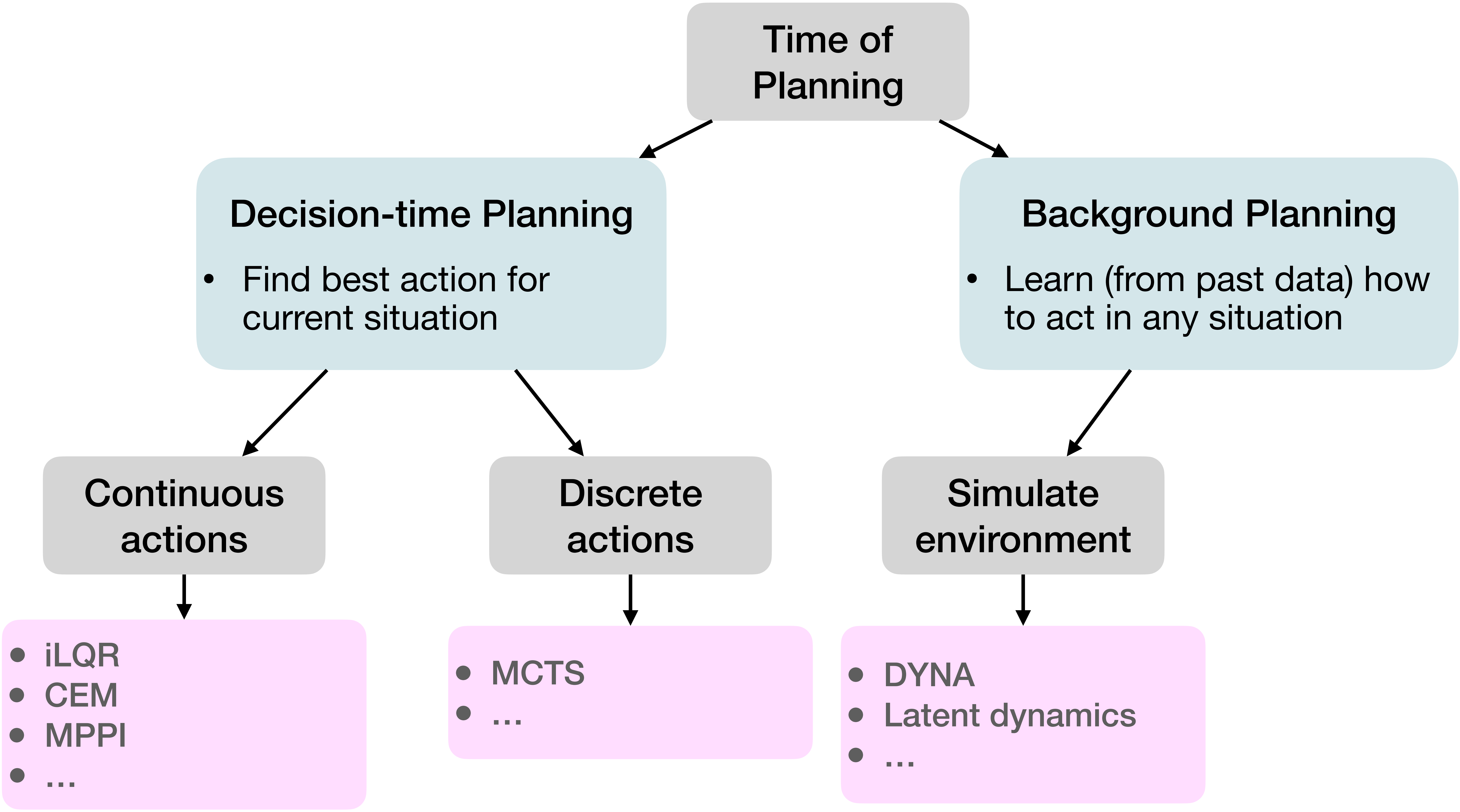
How Do We Use The "Model”?

Background vs Decision-time Planning



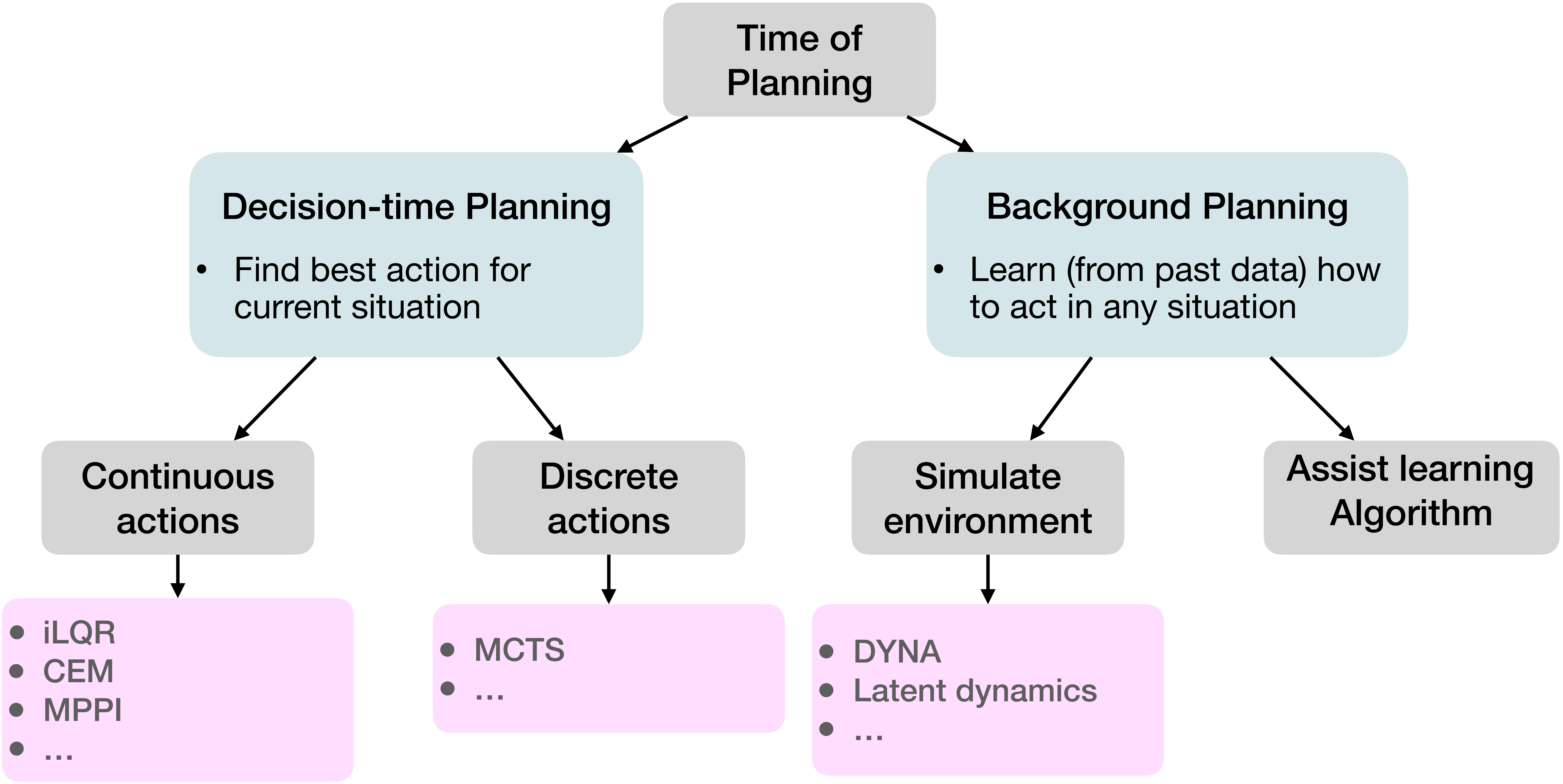
How Do We Use The "Model”?

Background vs Decision-time Planning



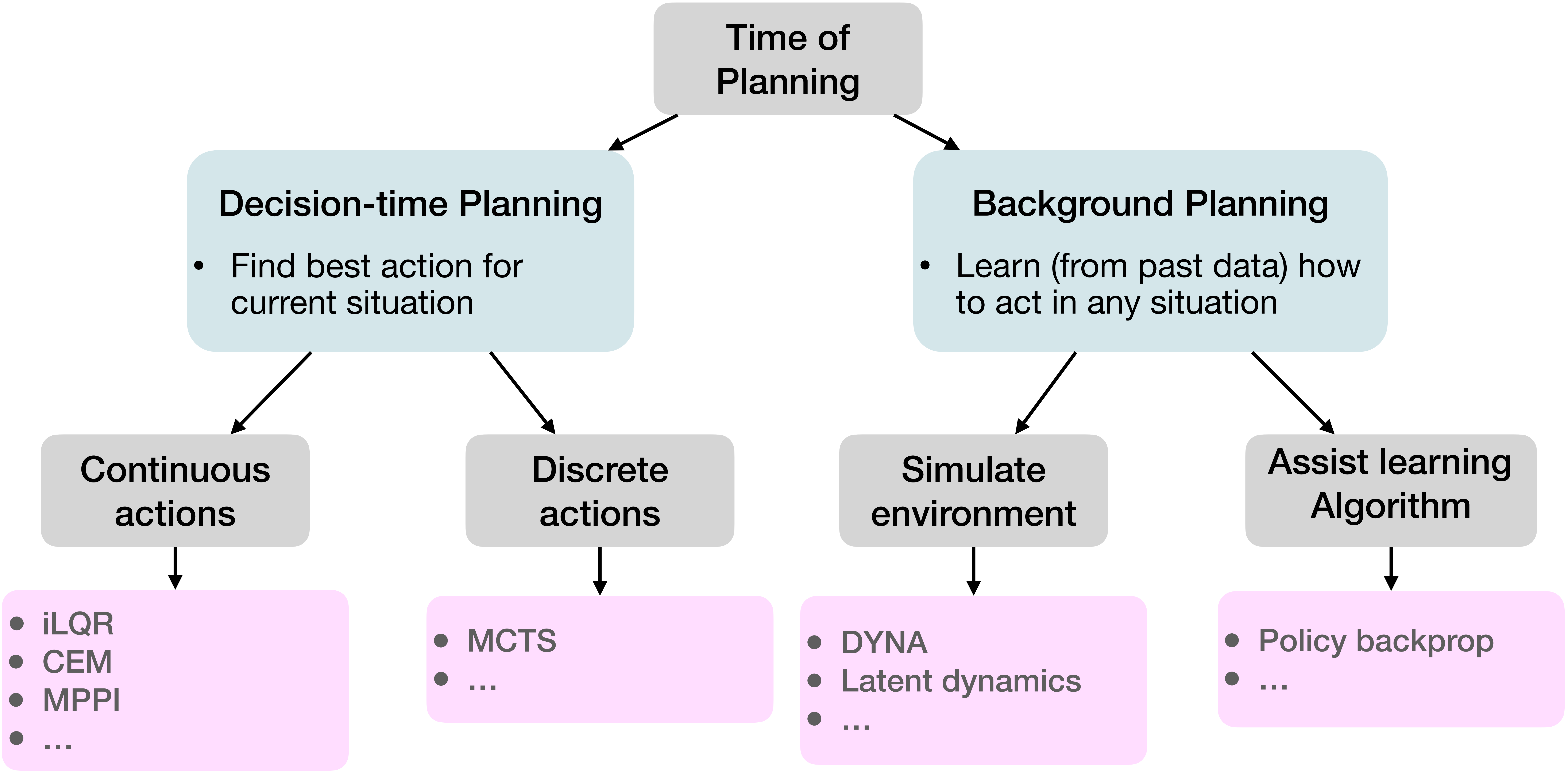
How Do We Use The "Model”?

Background vs Decision-time Planning



How Do We Use The "Model”?

Background vs Decision-time Planning



FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation

Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation

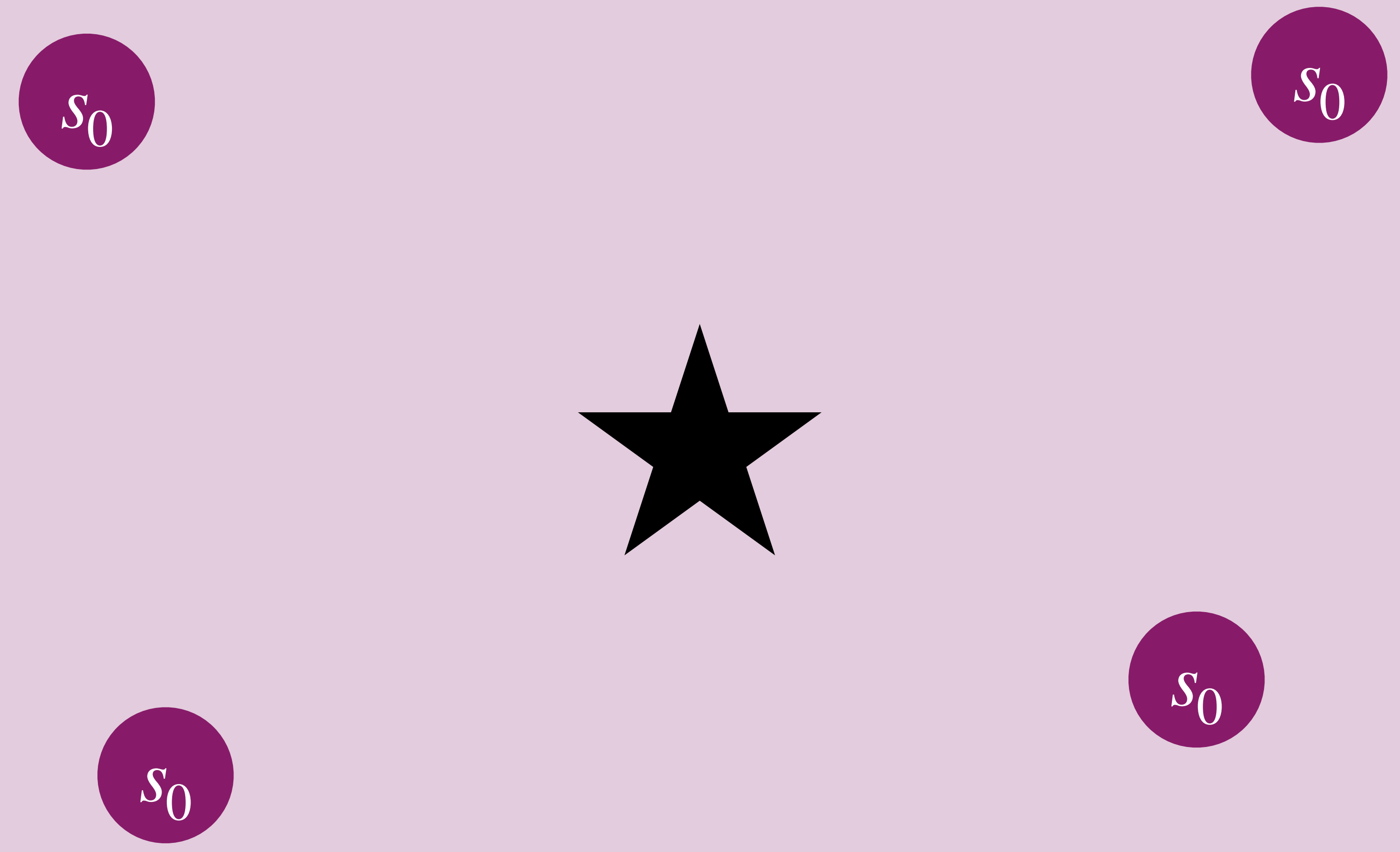
★

Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation

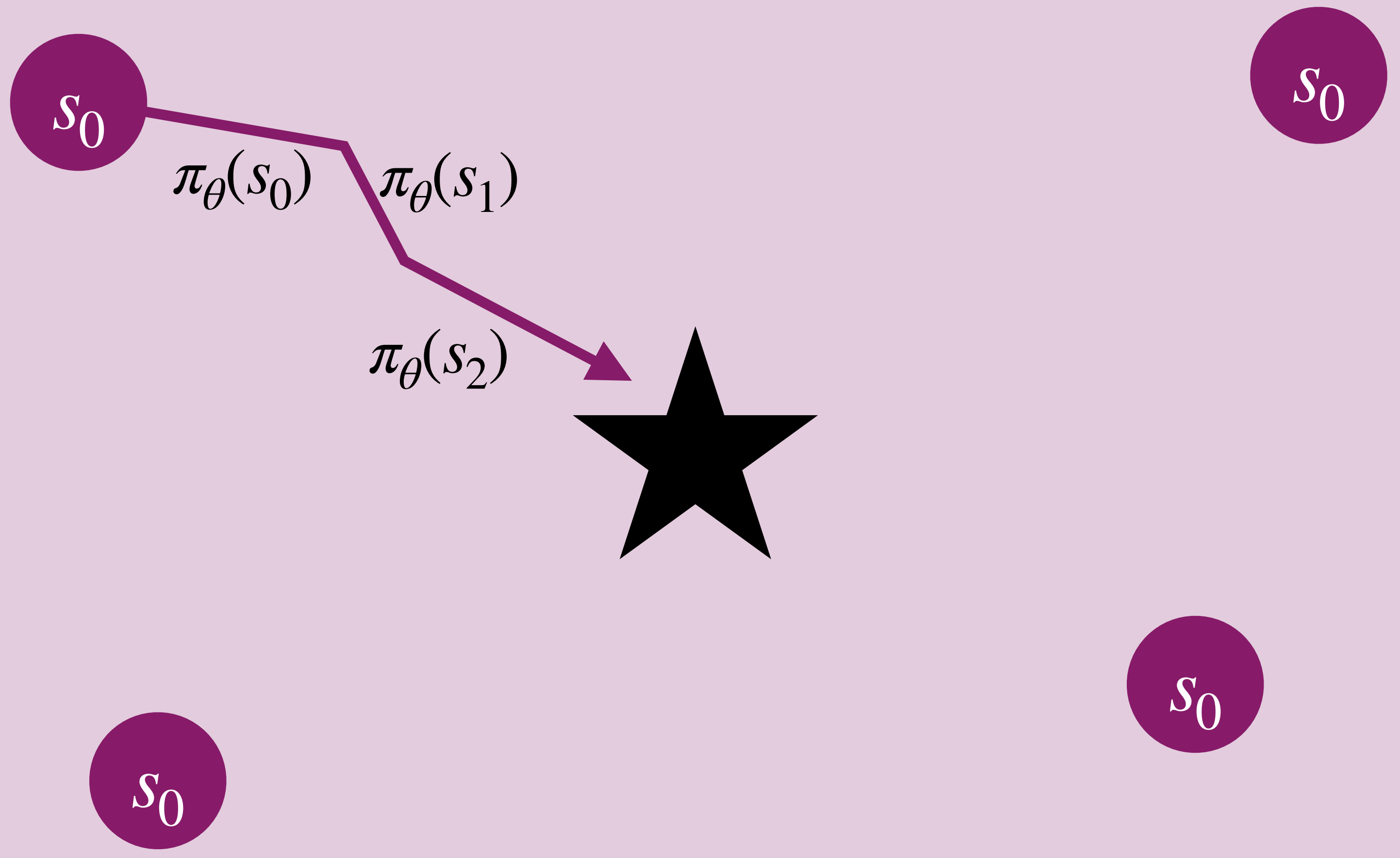


Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation

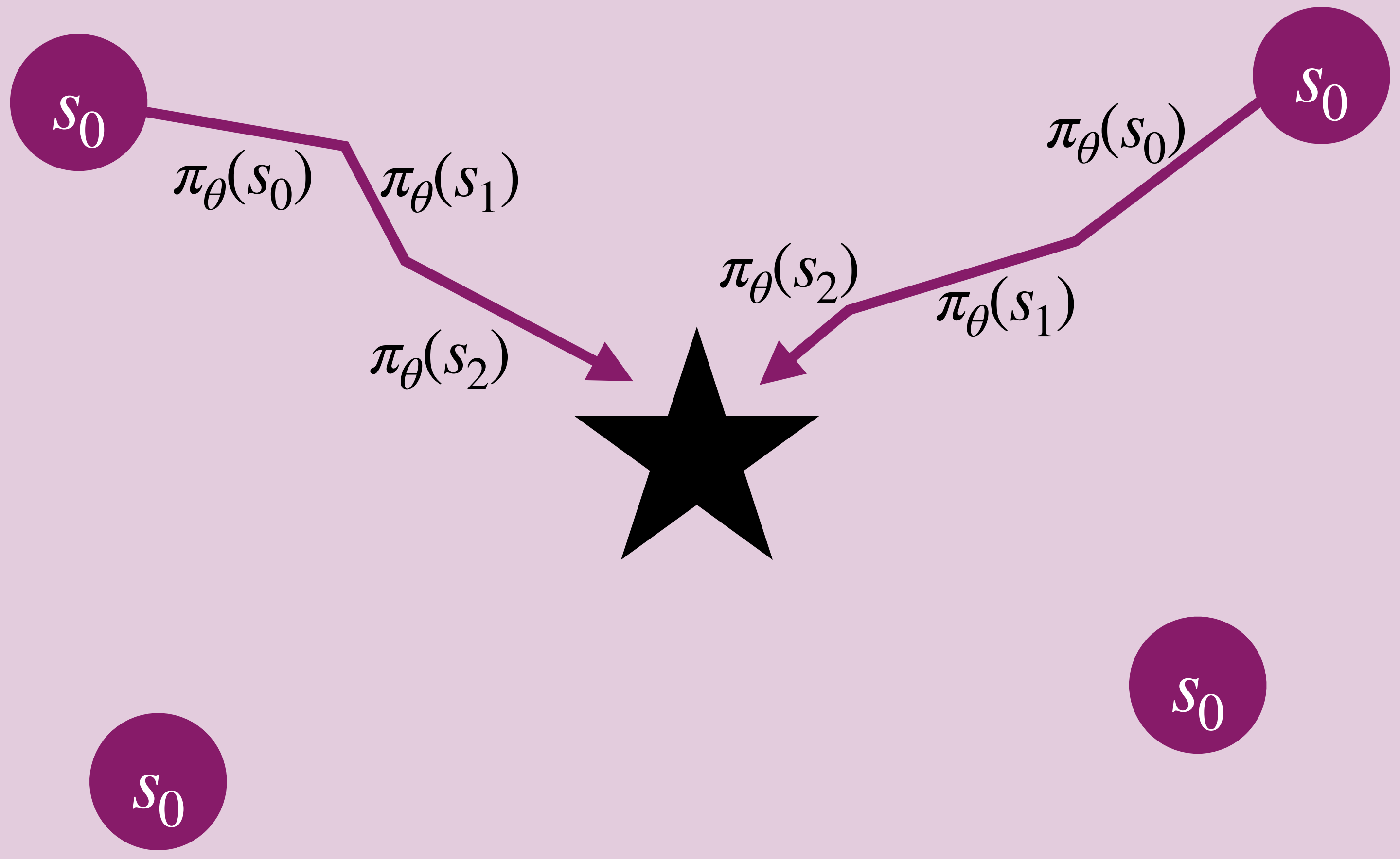


Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Decision-time planning

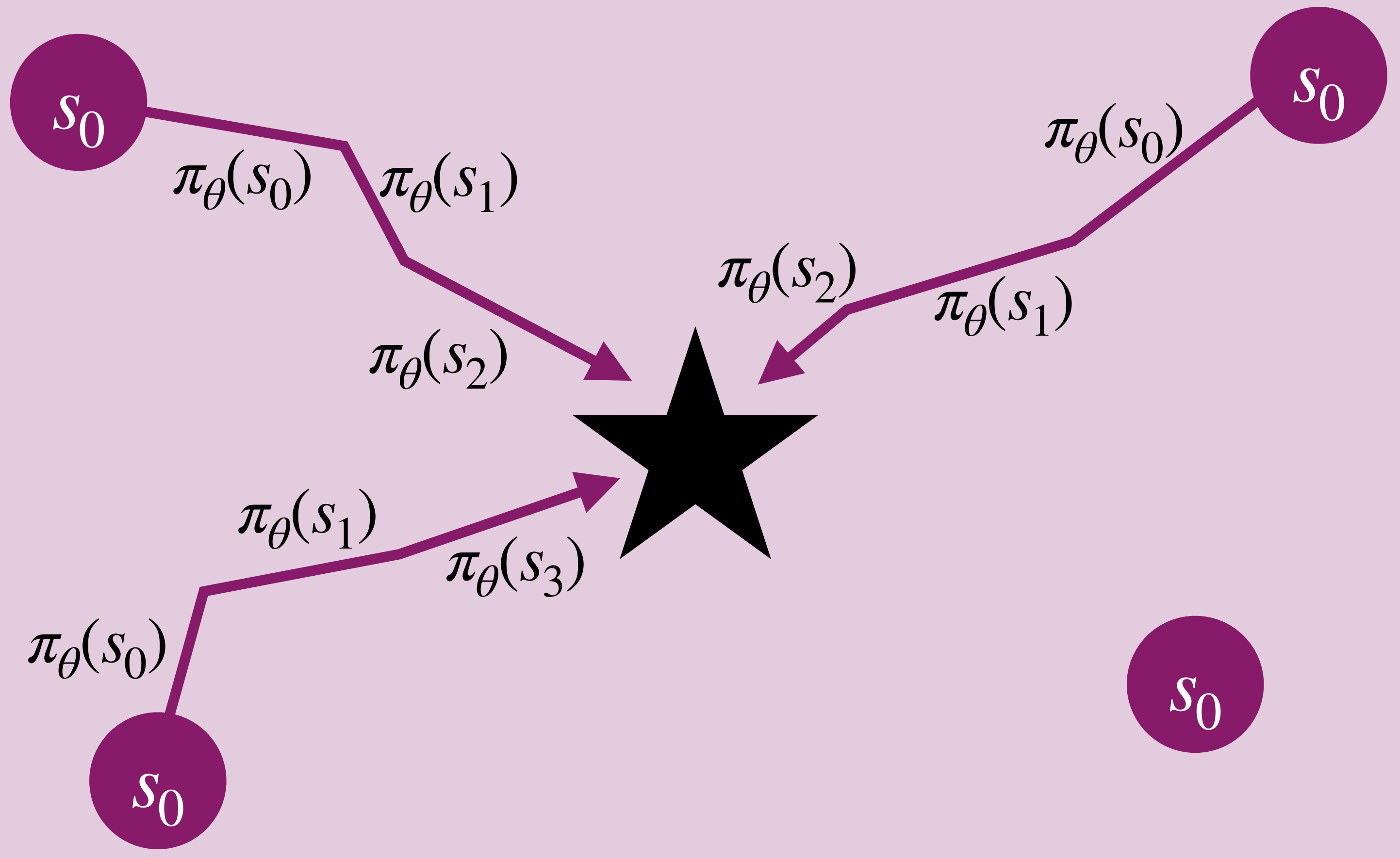
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Decision-time planning

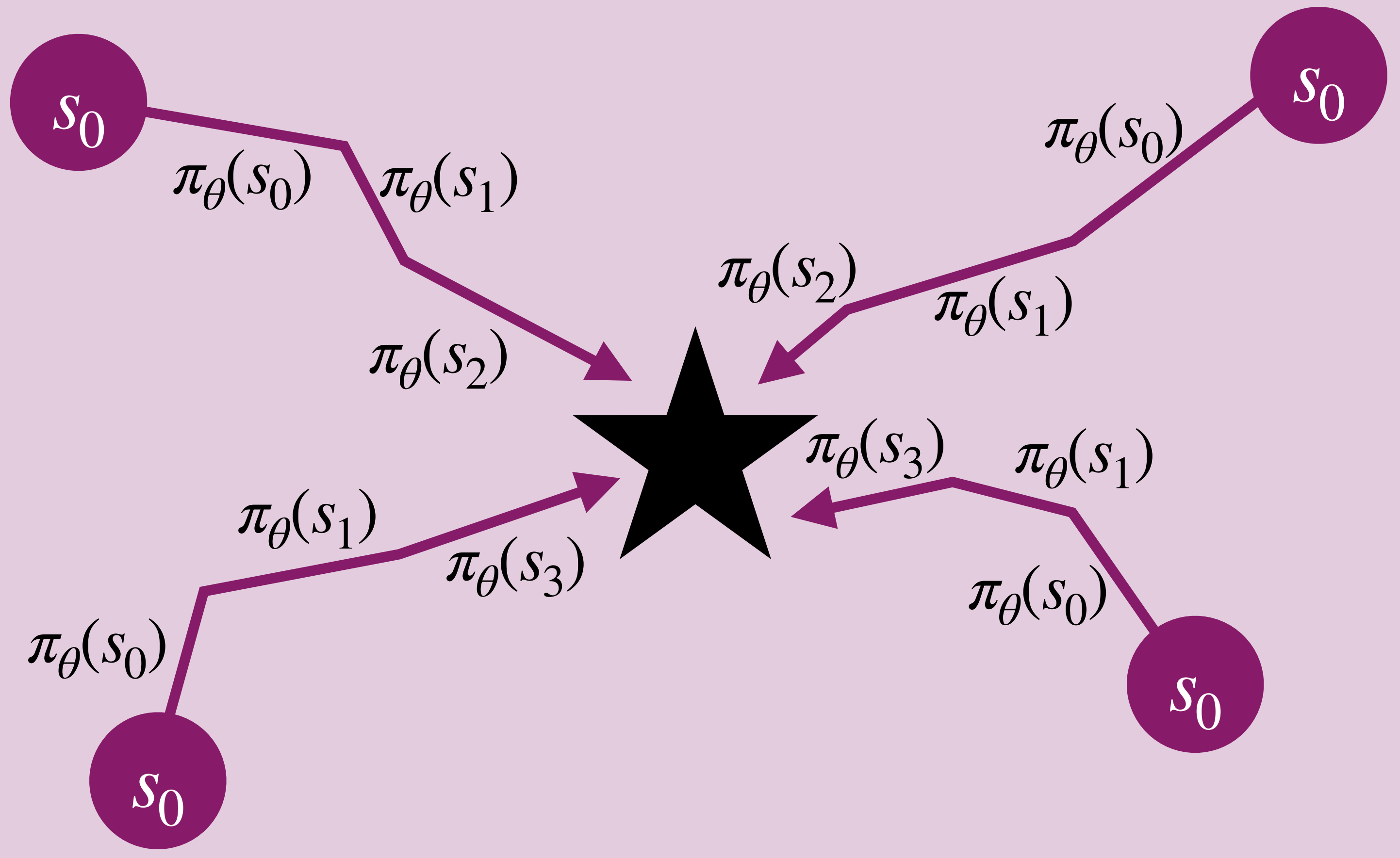
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Decision-time planning

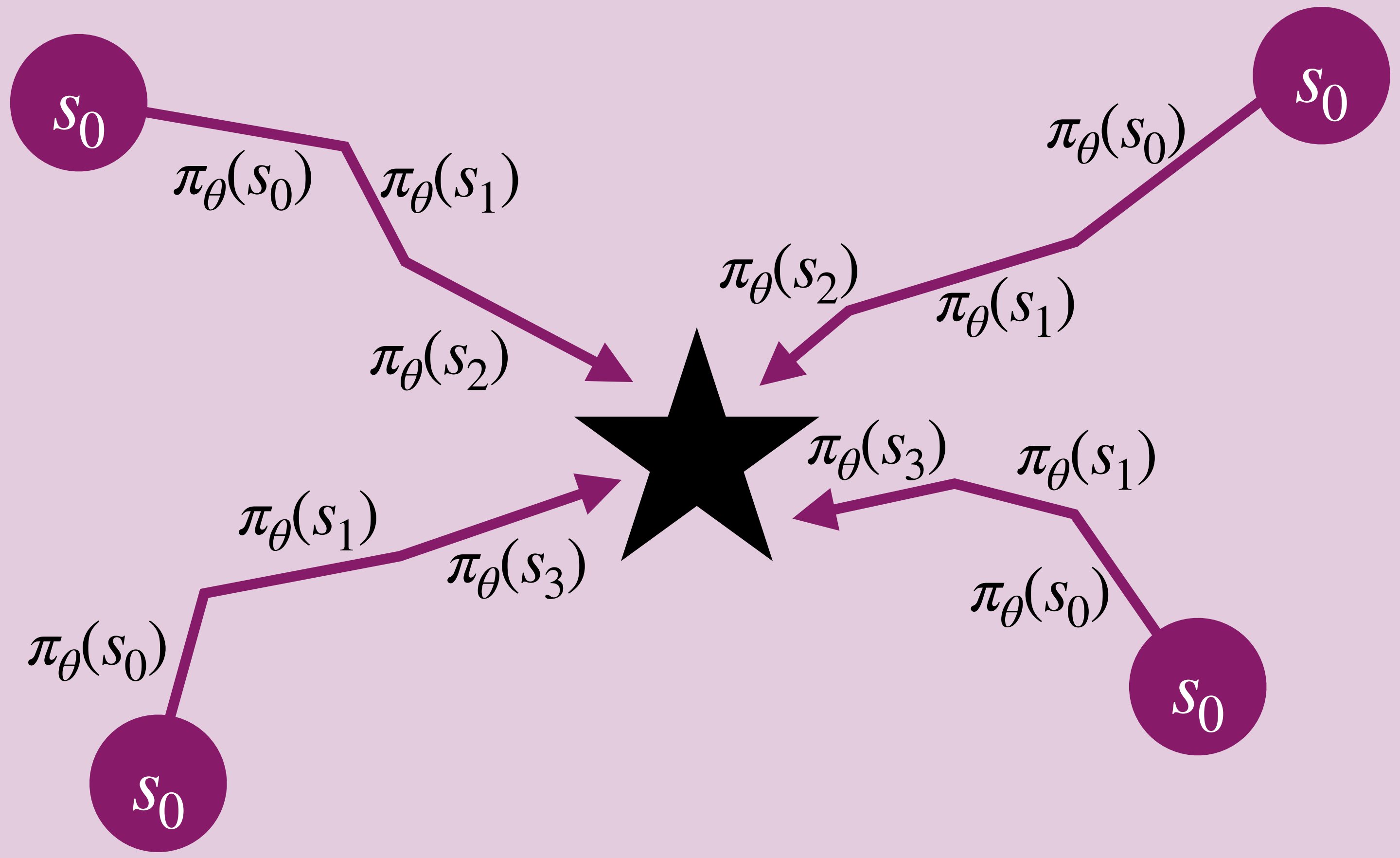
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



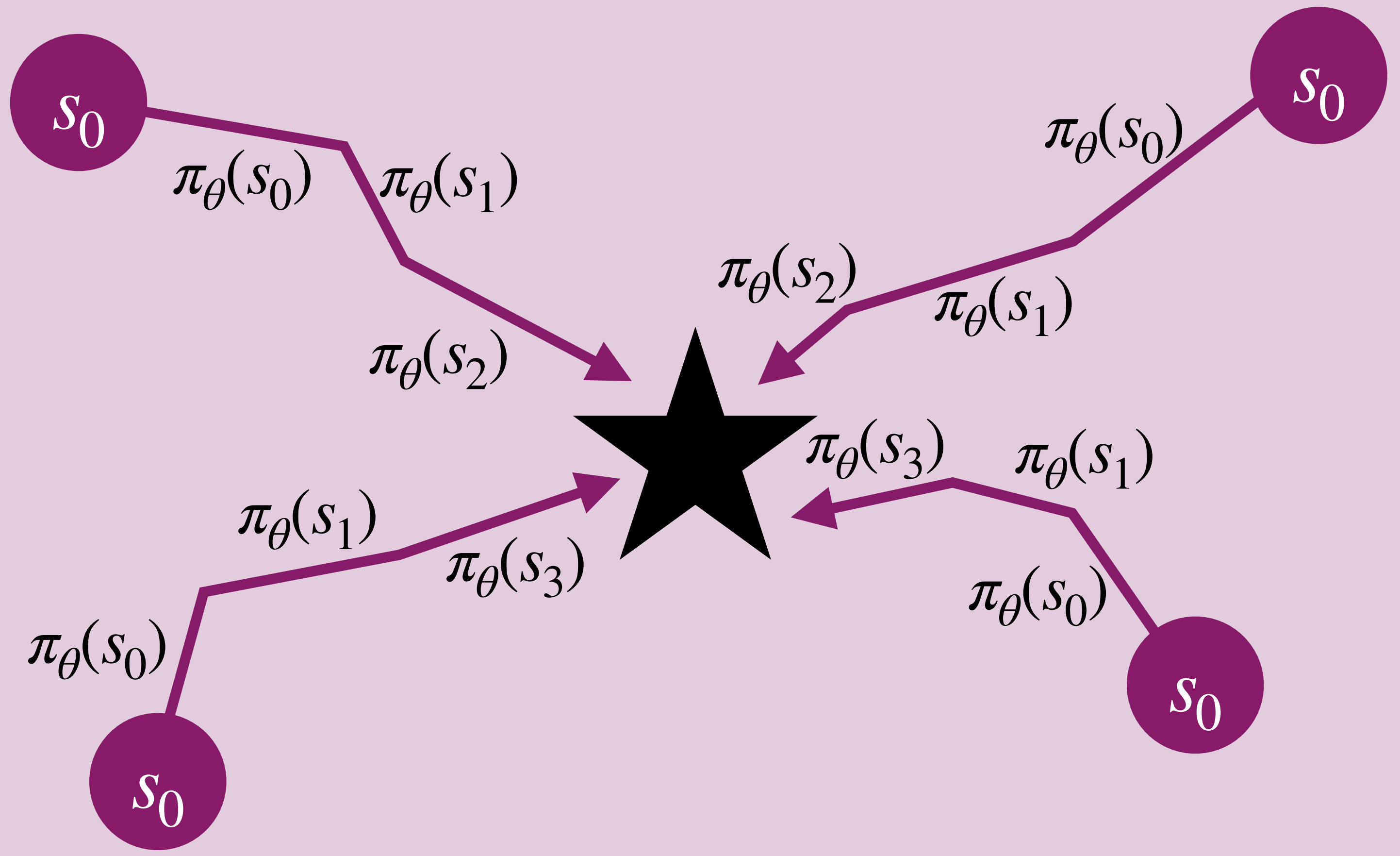
Optimisation variables: θ

Decision-time planning

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

Decision-time planning

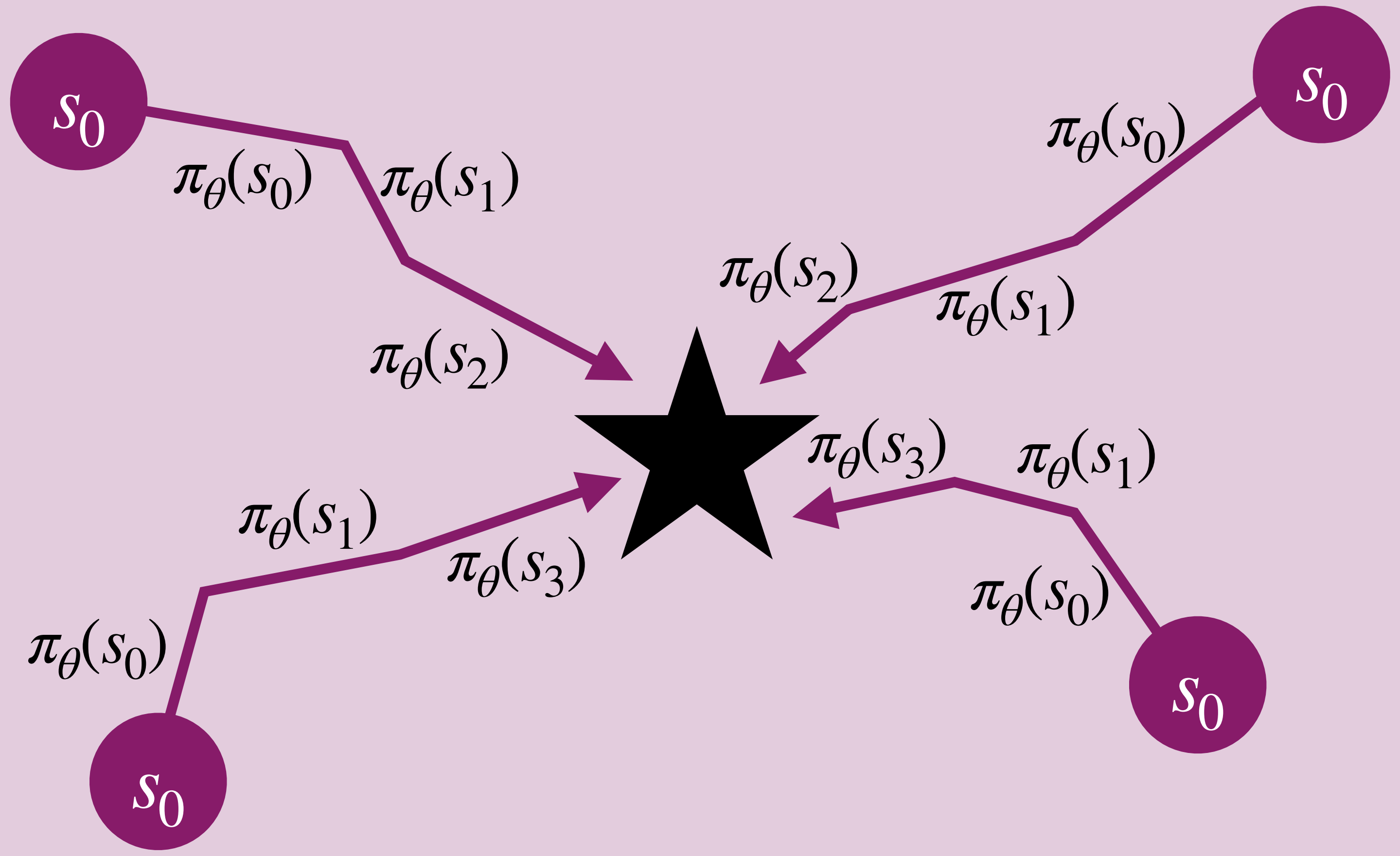
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\Big[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\Big]$$

Decision-time planning

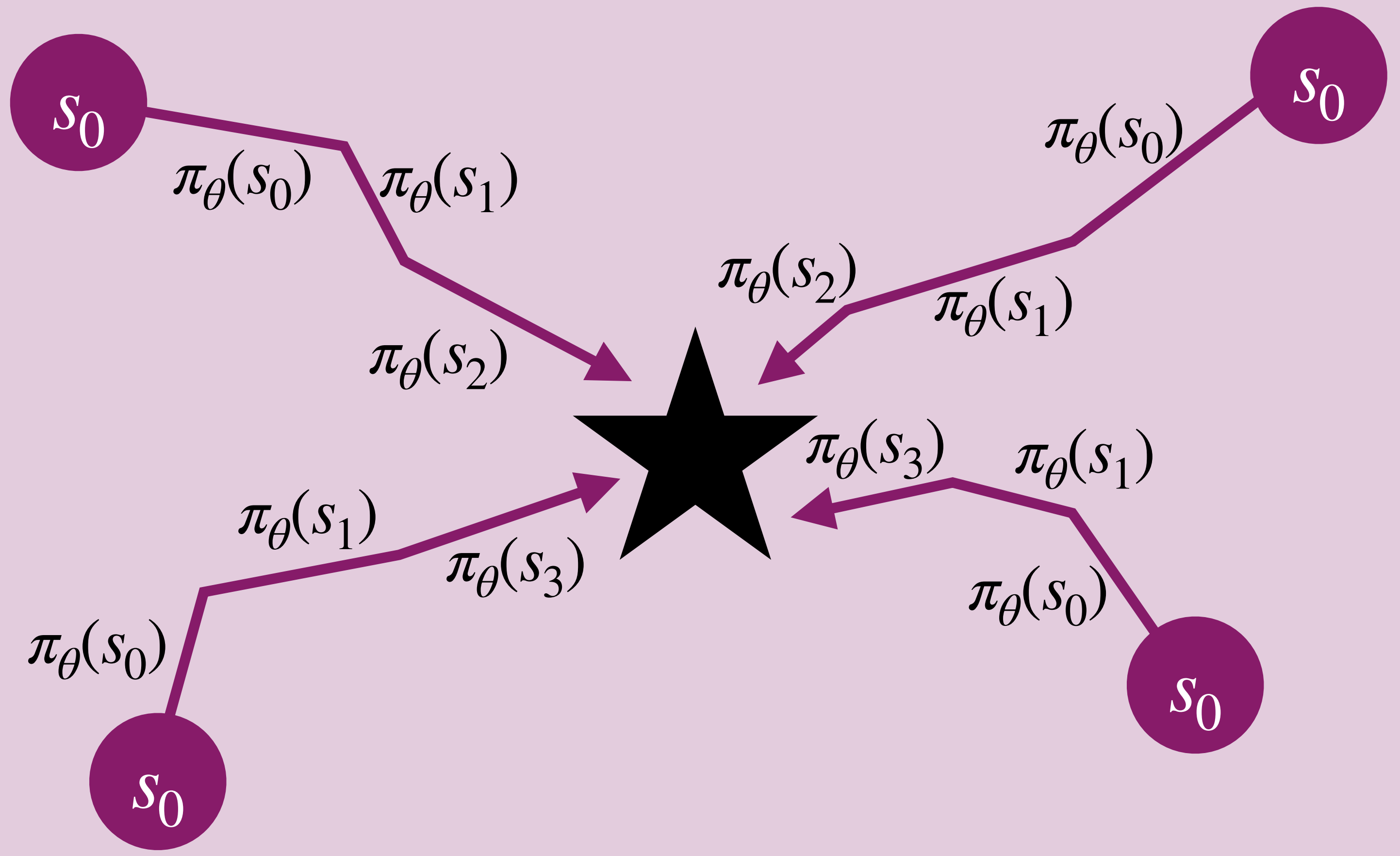
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\left[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\right]$$

Decision-time planning

Find best action for current situation

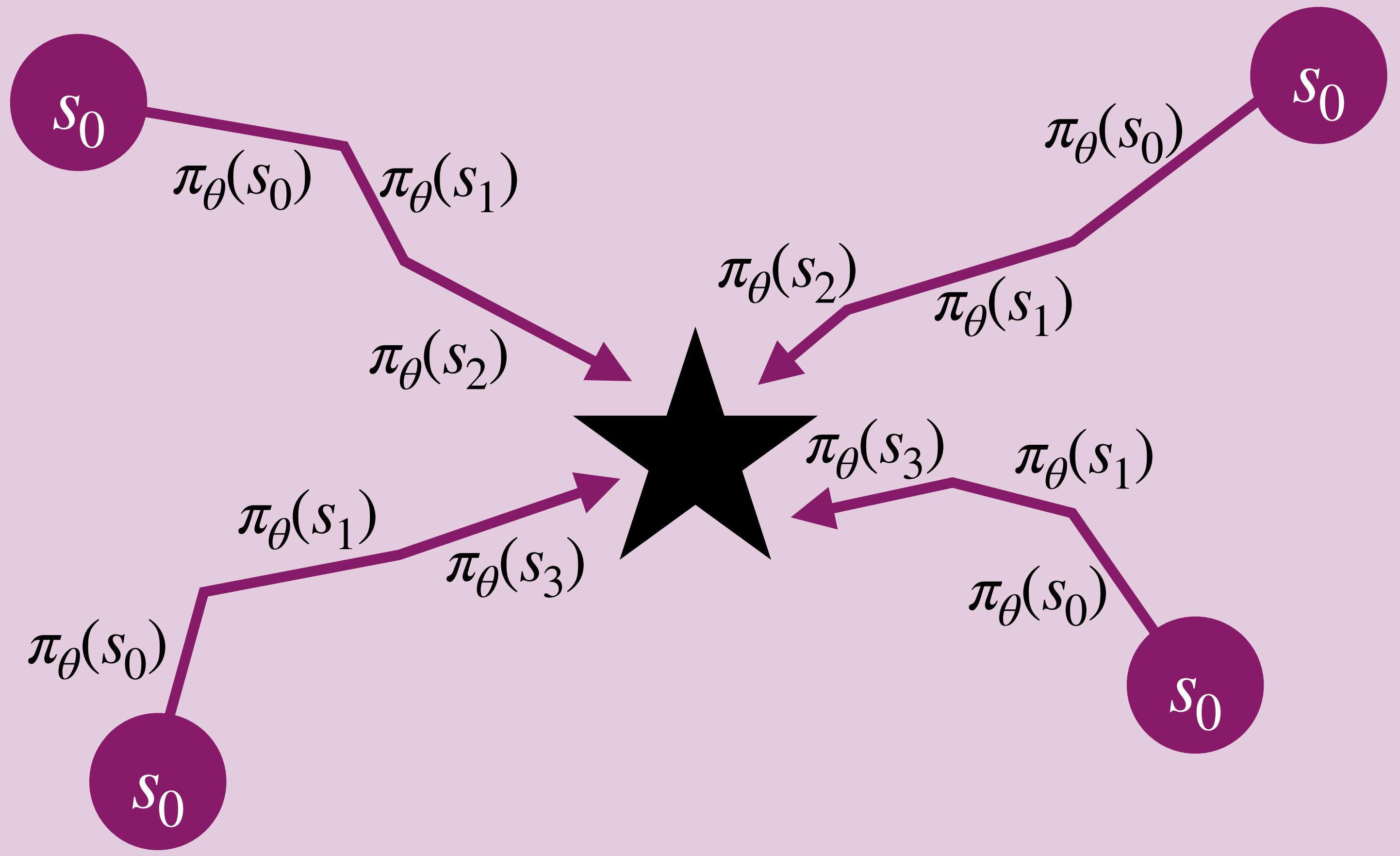
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



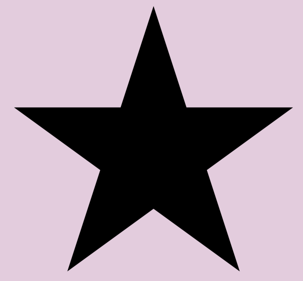
Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\left[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\right]$$

Decision-time planning

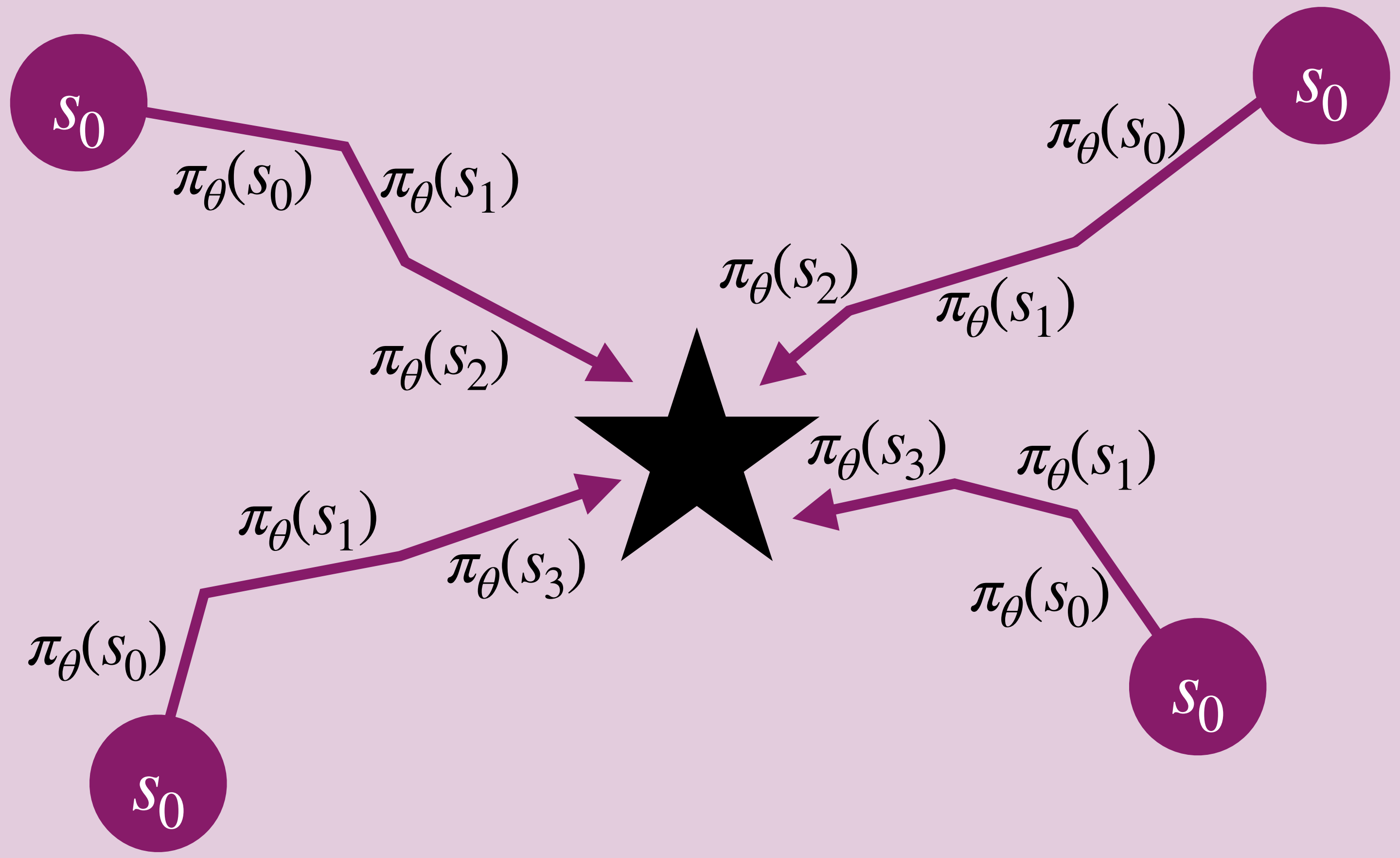
Find best action for current situation



Background vs Decision-time Planning

Background planning

Learn how to act in any situation



Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s,a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\Big[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\Big]$$

Decision-time planning

Find best action for current situation

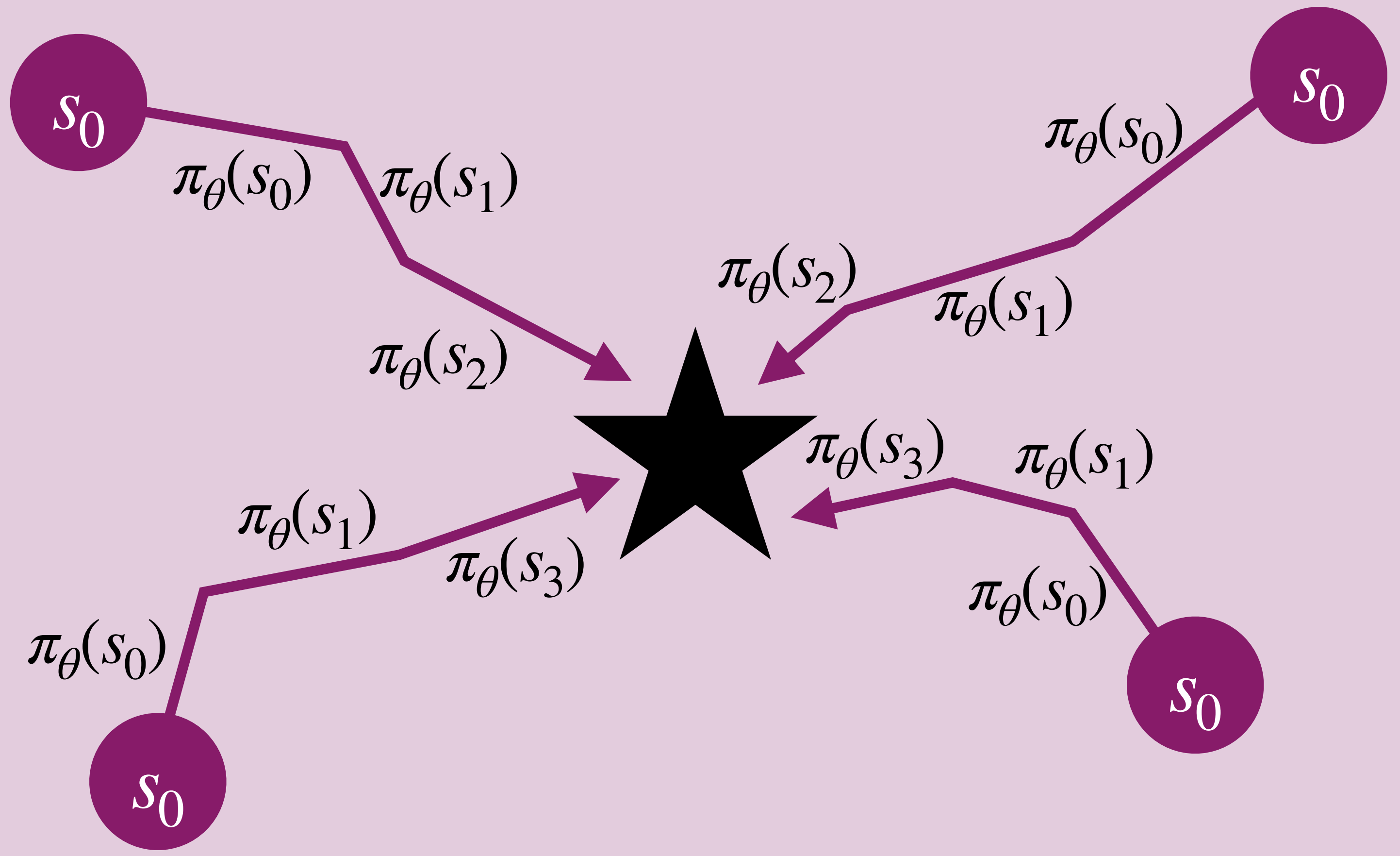
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



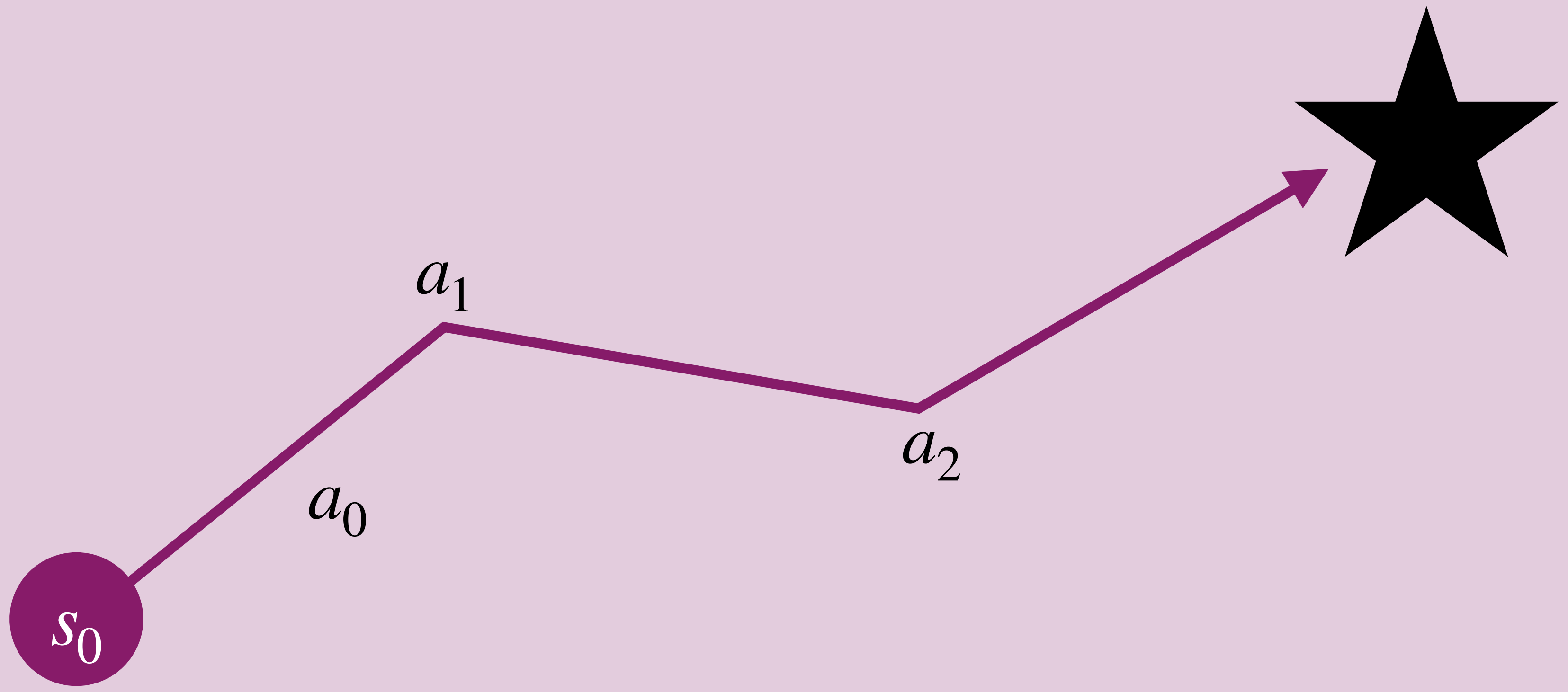
Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\Big[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\Big]$$

Decision-time planning

Find best action for current situation



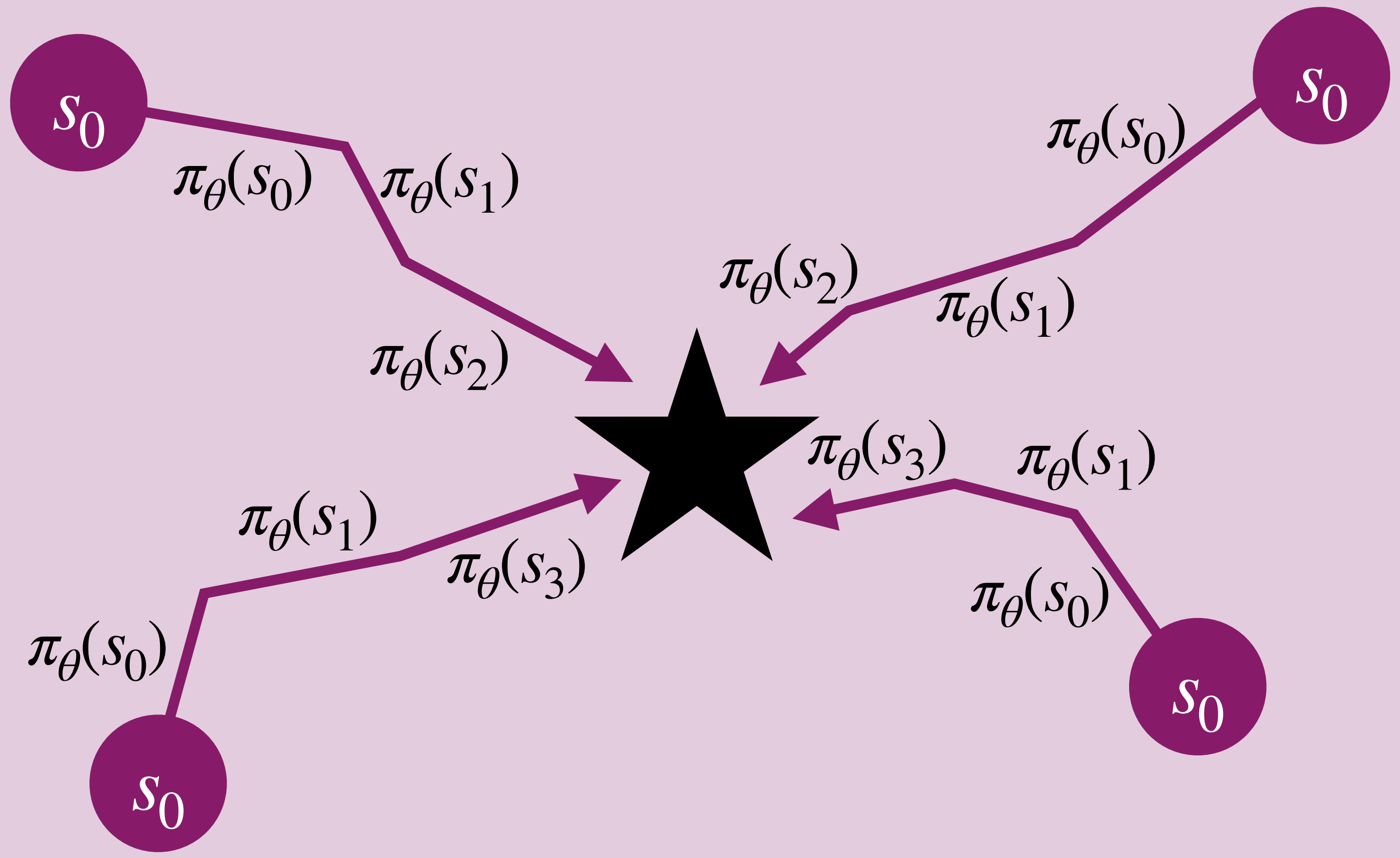
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



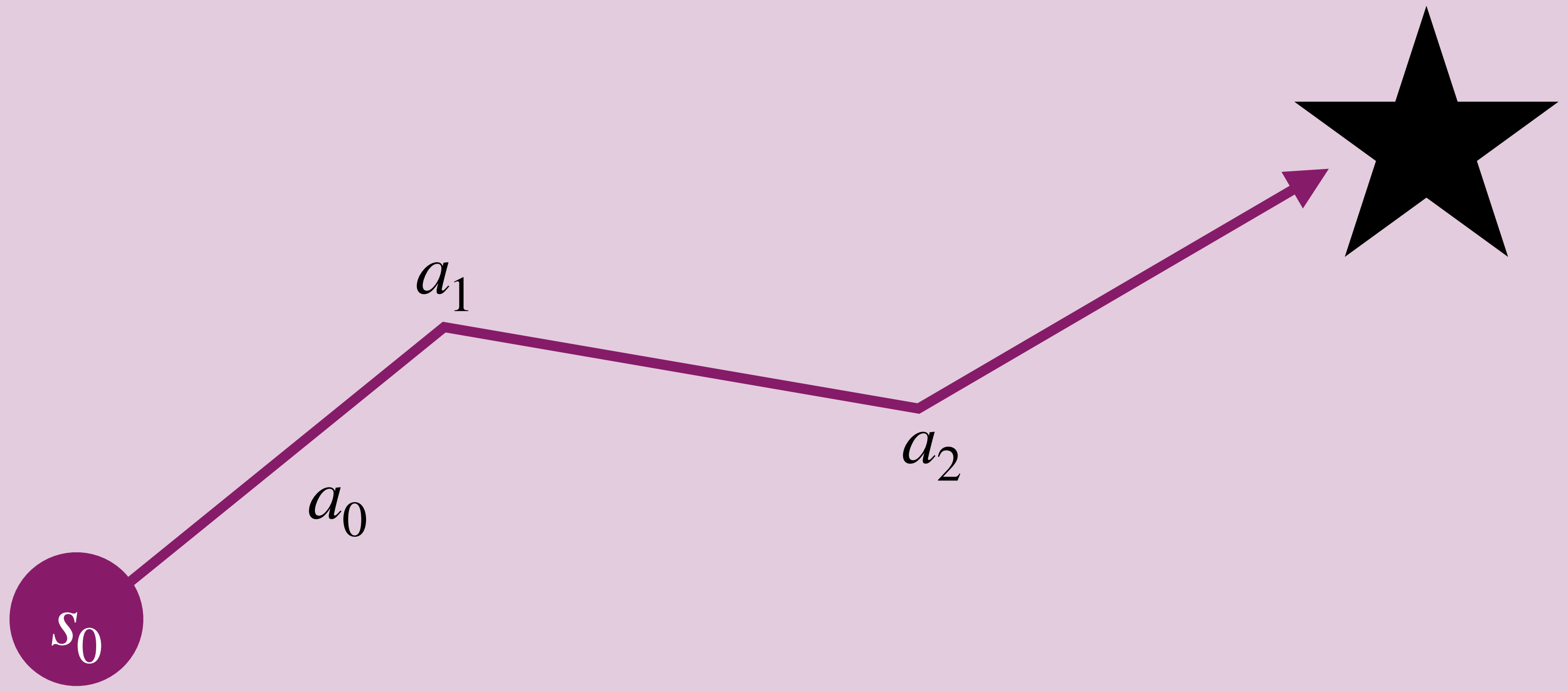
Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\Big[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\Big]$$

Decision-time planning

Find best action for current situation



Optimisation variables: $a_0, \ldots, a_H$

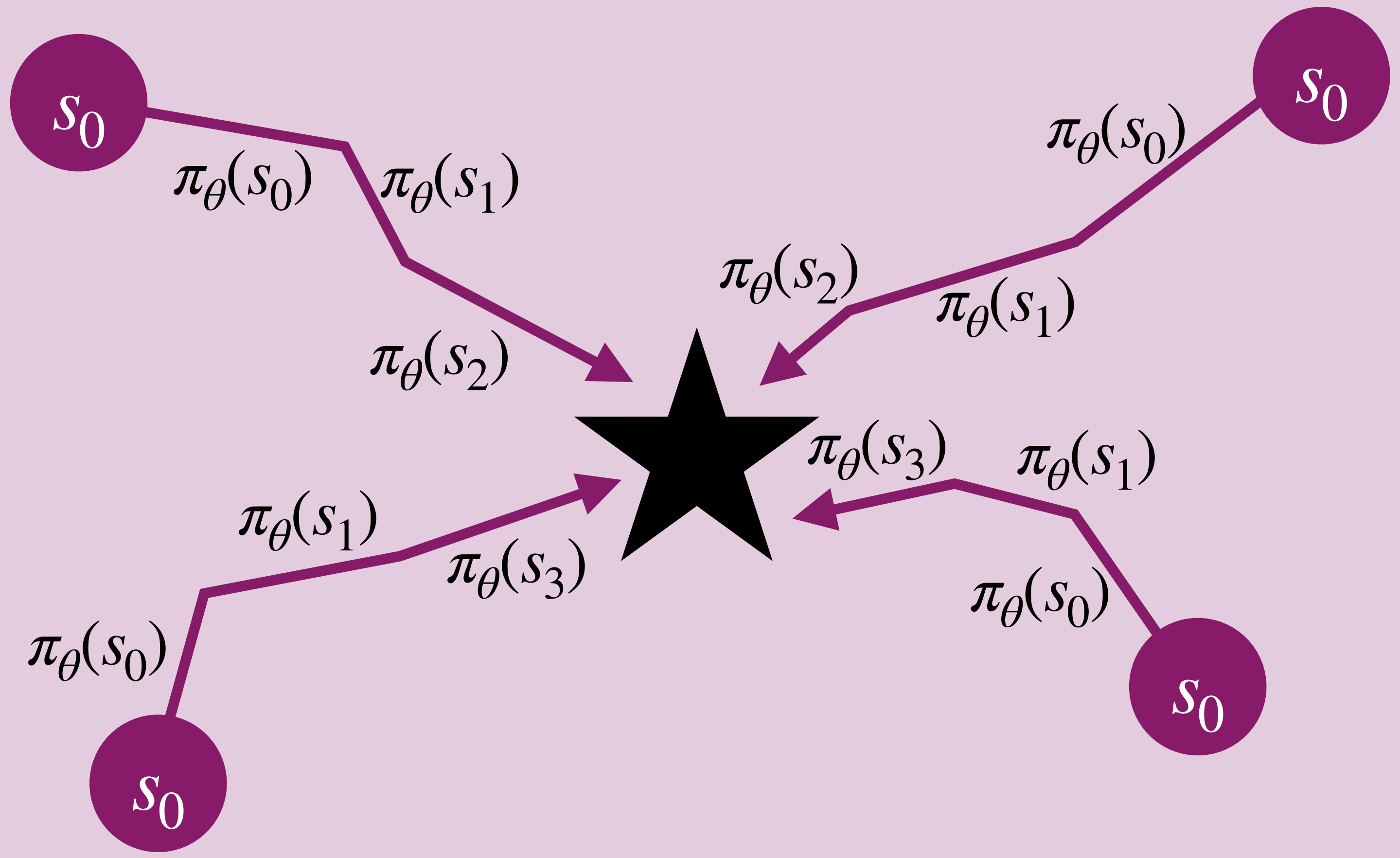
FCAI

fcai.fi

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



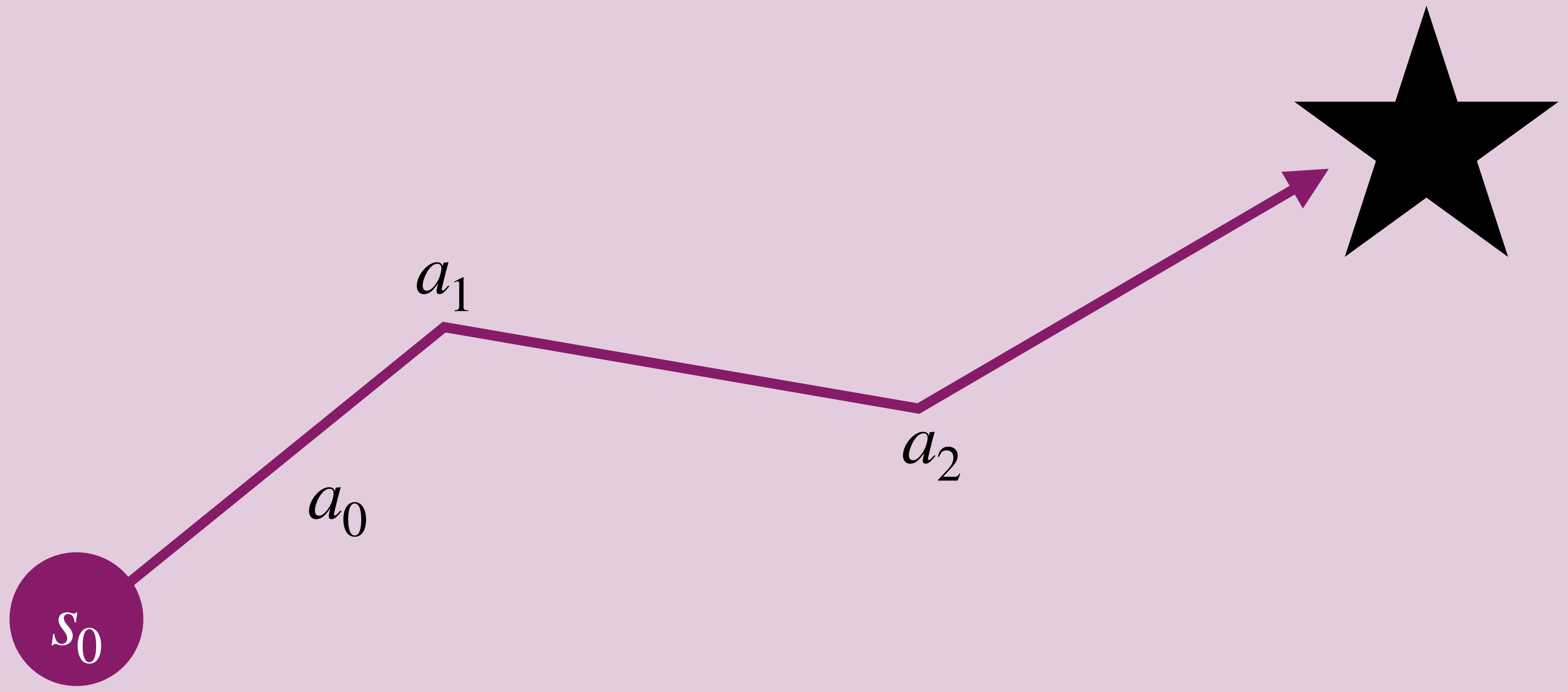
Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s,a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\Big[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\Big]$$

Decision-time planning

Find best action for current situation



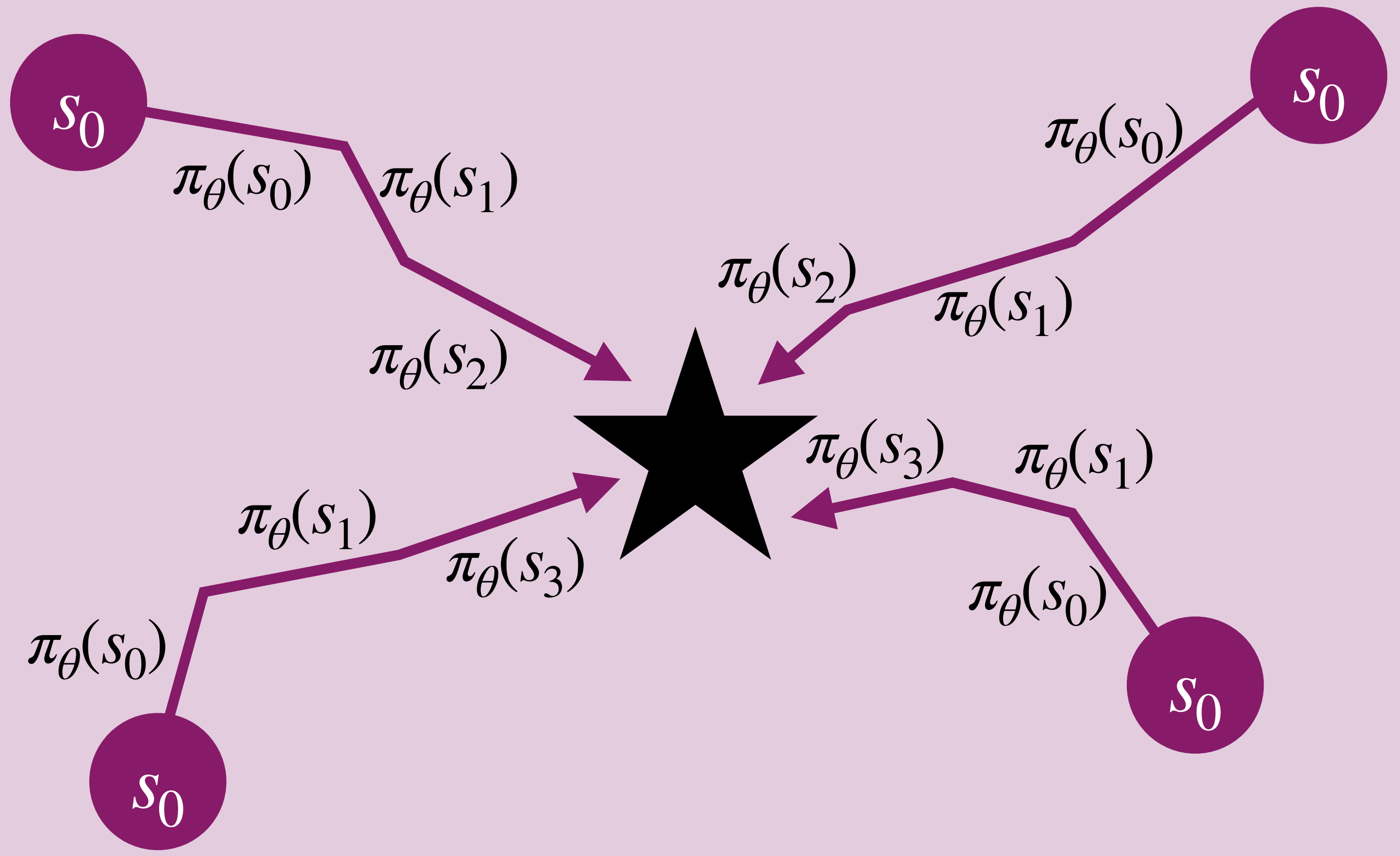
Optimisation variables: $a_0, \ldots, a_H$

Sequence of actions (and maybe also states)

Background vs Decision-time Planning

Background planning

Learn how to act in any situation



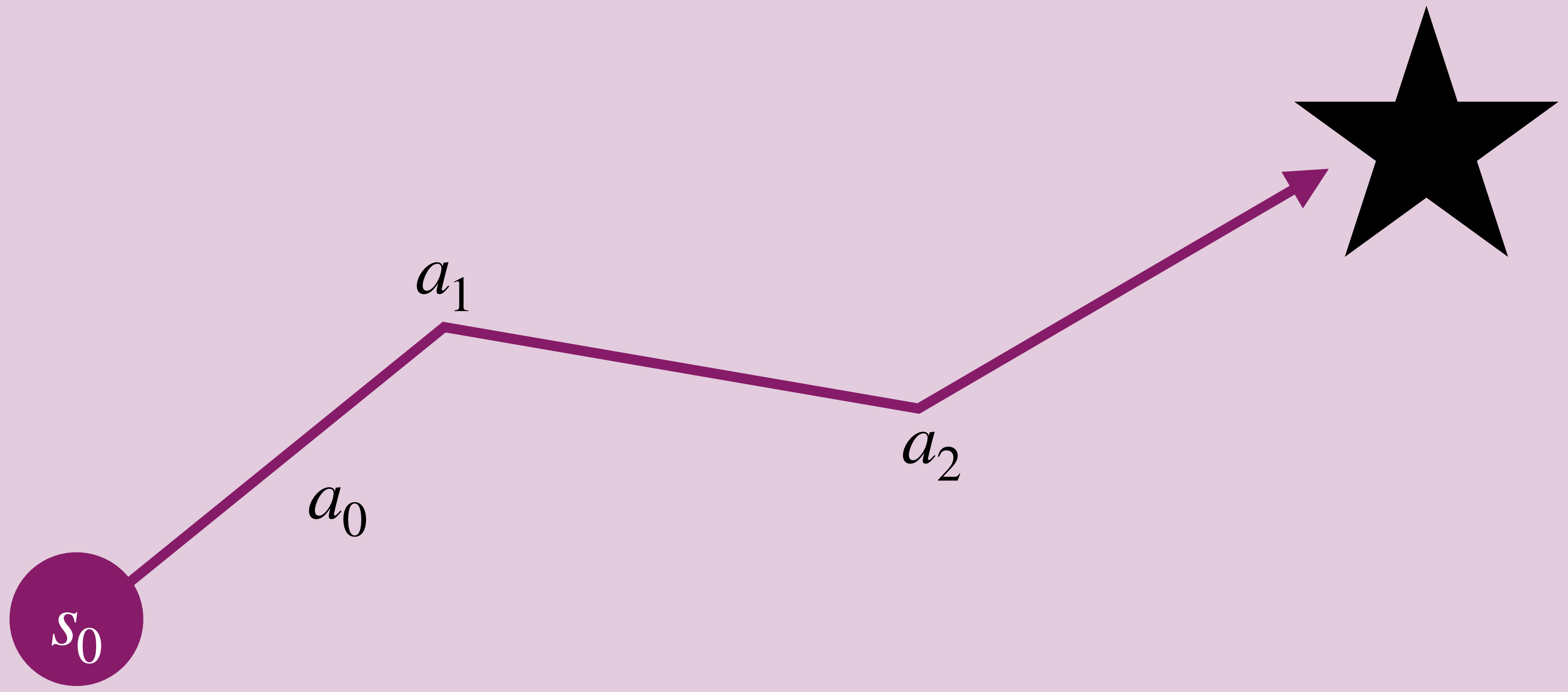
Optimisation variables: θ

Parameters of policy $\pi_\theta(s)$, value $Q_\theta(s, a)$, etc

$$J(\theta) = \mathbb{E}_{s_0}\left[\sum_{t=0}^{H} r(s_t, \pi_\theta(s_t))\right]$$

Decision-time planning

Find best action for current situation



Optimisation variables: $a_0, \ldots, a_H$

Sequence of actions (and maybe also states)

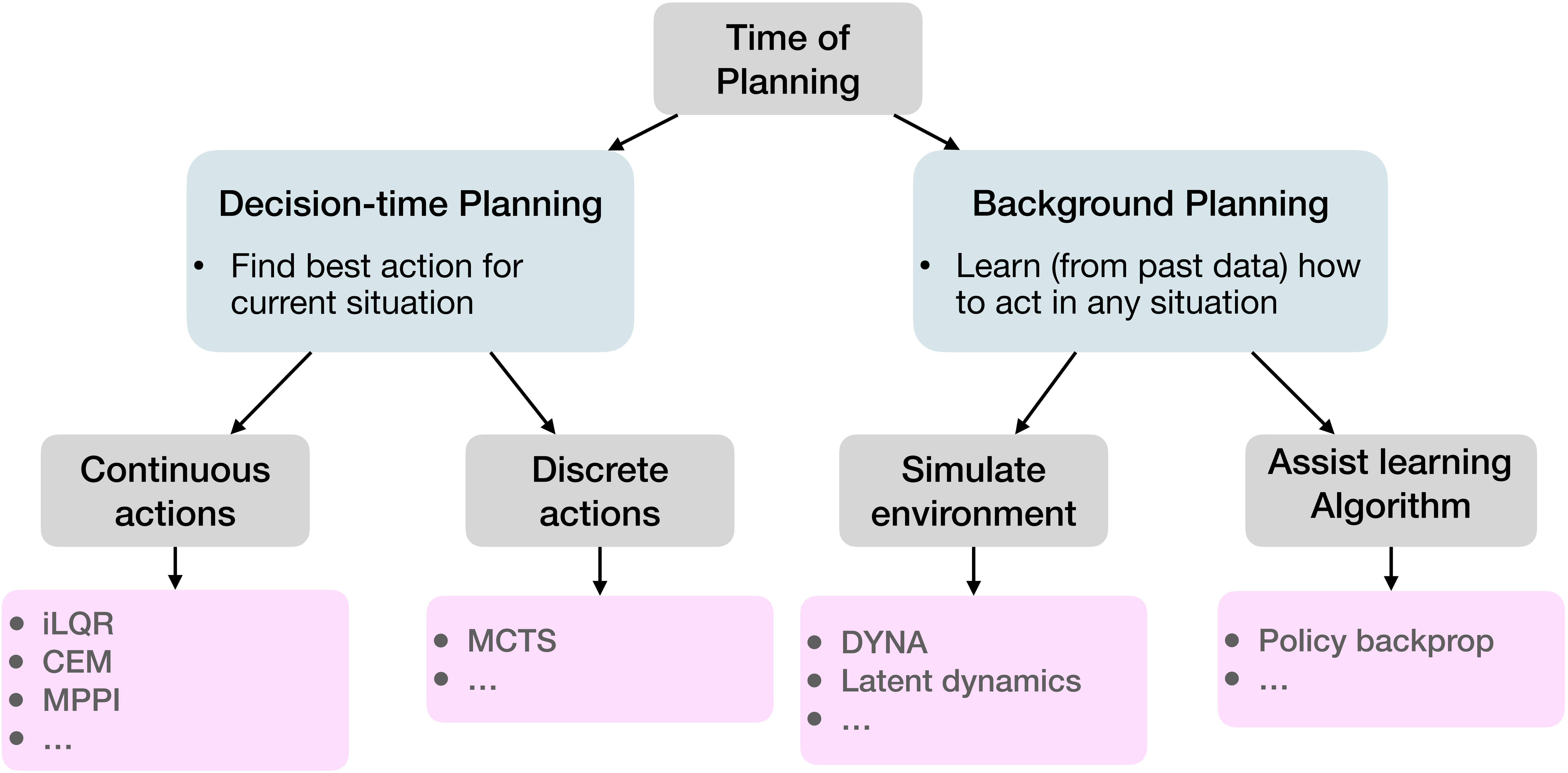
$$J(a_0, \ldots, a_H) = \sum_{t=0}^{H} r(s_t, a_t)$$

FCAI

fcai.fi

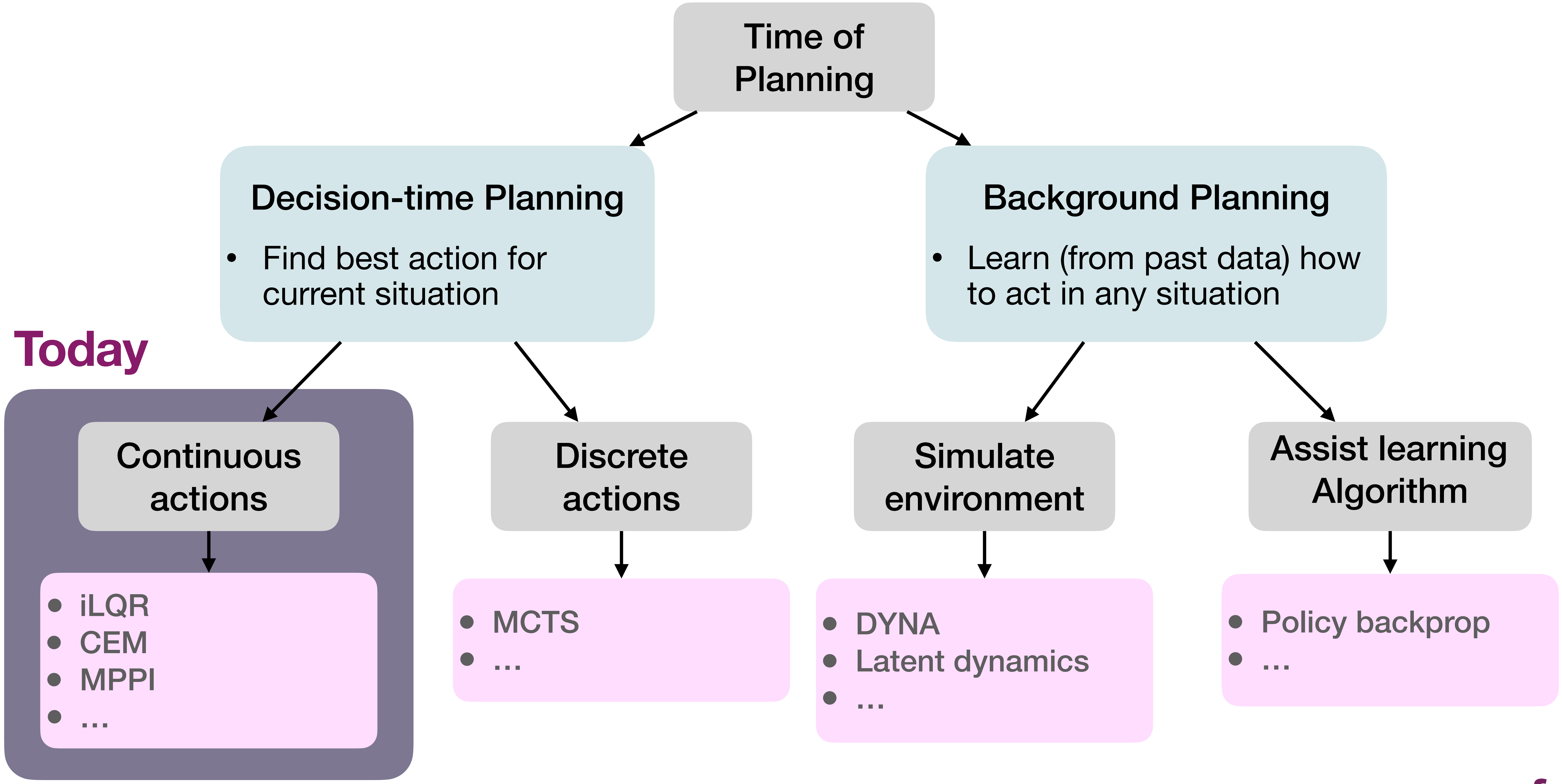
How Do We Use The "Model”?

Background vs Decision-time Planning



How Do We Use The "Model”?

Background vs Decision-time Planning



Decision-time Planning (Continuous Actions)

Decision-time Planning (Continuous Actions)

We’ll start by assuming known, deterministic dynamics

Decision-time Planning (Continuous Actions)

We’ll start by assuming known, deterministic dynamics

$$s_{t+1} = f(s_t, a_t)$$

Decision-time Planning

Trajectory optimisation

Decision-time Planning

Trajectory optimisation

Observe state s

Decision-time Planning

Trajectory optimisation

Observe state s

Plan $a_0, \ldots, a_H$ to maximise return $\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad$ s.t. $s_0 = s$

Decision-time Planning

Trajectory optimisation

Observe state s

Plan $a_0, \ldots, a_H$ to maximise return $\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad$ s.t. $s_0 = s$

Execute each action

FCAI

fcai.fi

Trajectory Optimisation

Shooting methods

Collocation methods

Trajectory Optimisation

Shooting methods

Optimisation variables: $a_0, \ldots, a_H$

Actions

$$J(a_{0:H}) = \sum_{t=0}^{H} r(s_t, a_t)$$

Collocation methods

Trajectory Optimisation

Shooting methods

Optimisation variables: $a_0, \ldots, a_H$

Actions

$$J(a_{0:H}) = \sum_{t=0}^{H} r(s_t, a_t)$$

Collocation methods

Optimisation variables: $a_0, s_0, \ldots, a_H, s_H$

Actions and states

$$J(a_{0:H}, s_{0:H}) = \sum_{t=0}^{H} r(s_t, a_t)$$

$$\text{s.t. } \|s_{t+1} - f(s_t, a_t)\| = 0$$

Shooting Methods

Illustration

Shooting Methods

Illustration

$$J(a_{0:H}) = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

Optimising actions

Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

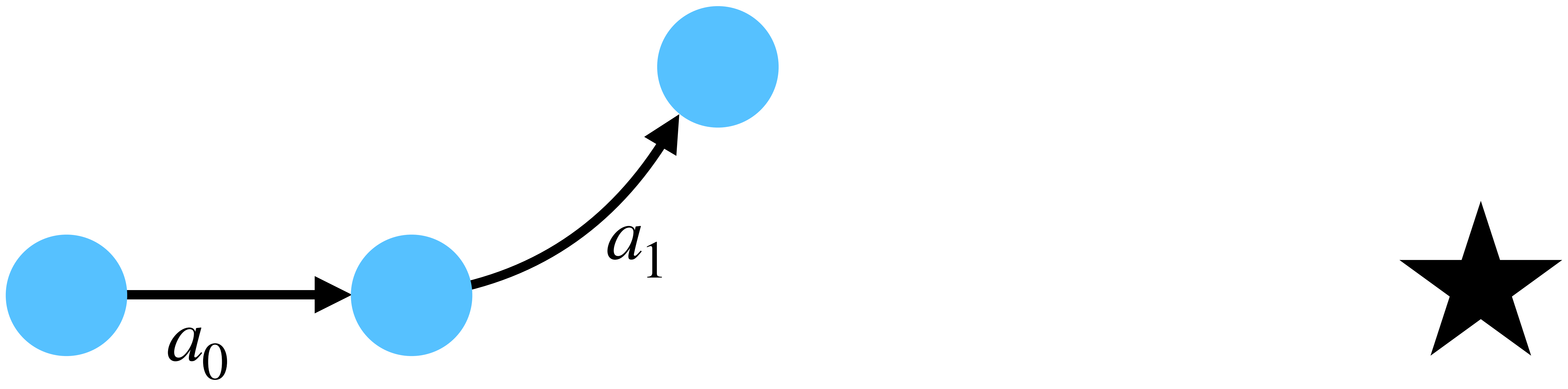
Optimising actions

Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

Optimising actions

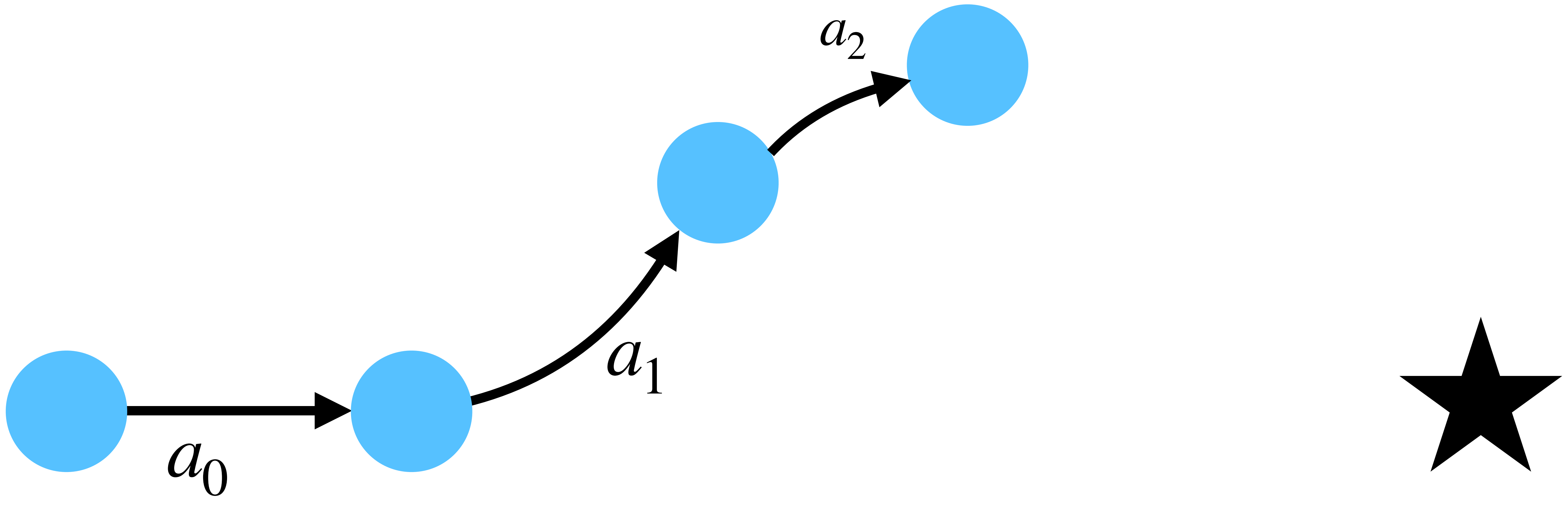


Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

Optimising actions

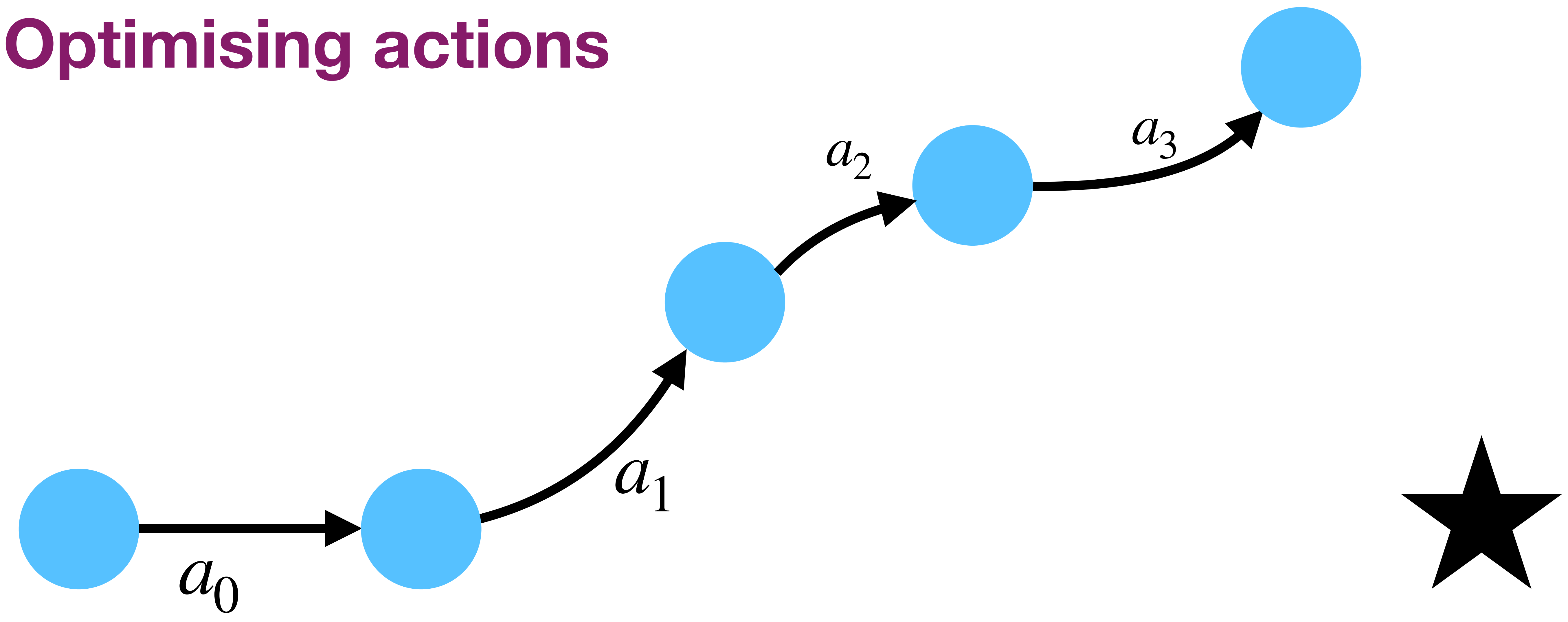


Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(f(f(\ldots), a_{H-1}), a_H)$$

Optimising actions



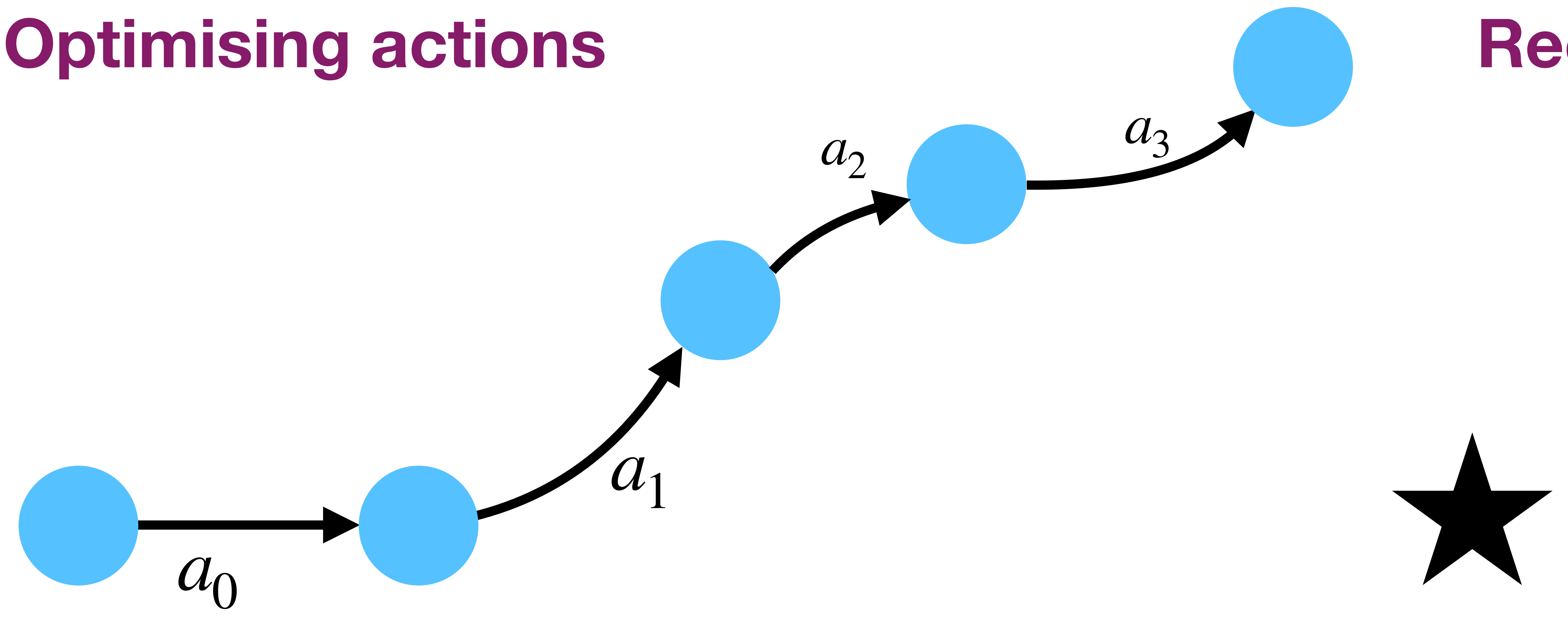
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics



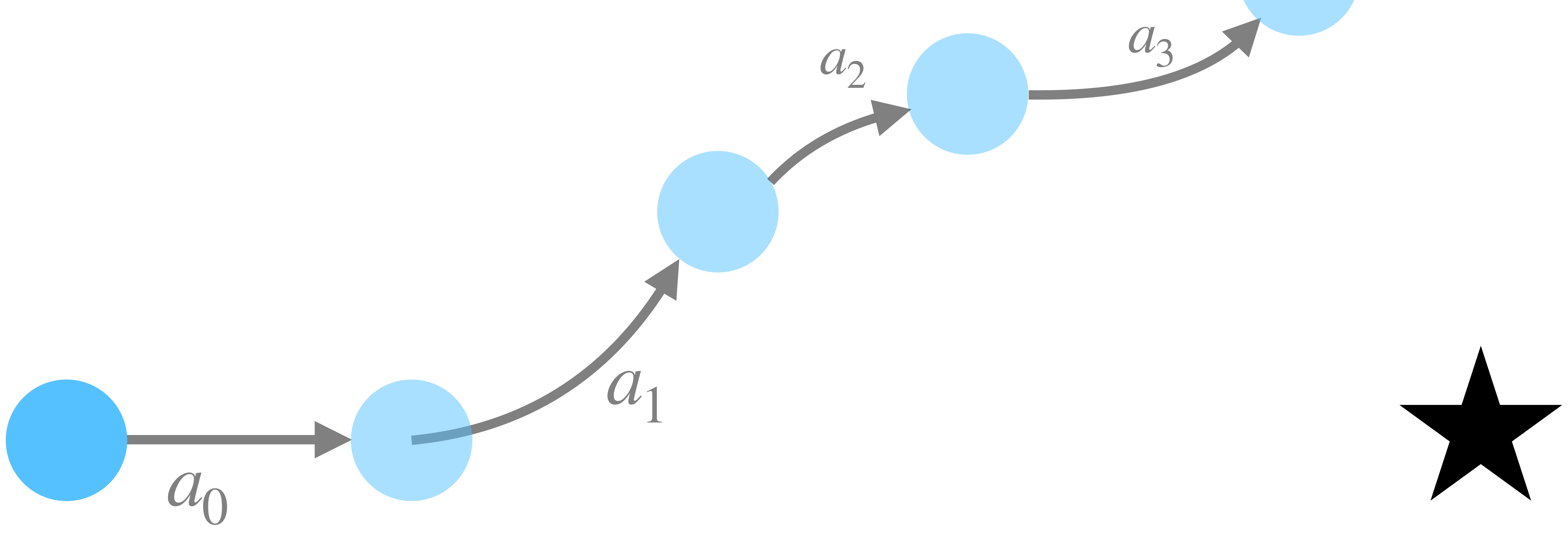
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics



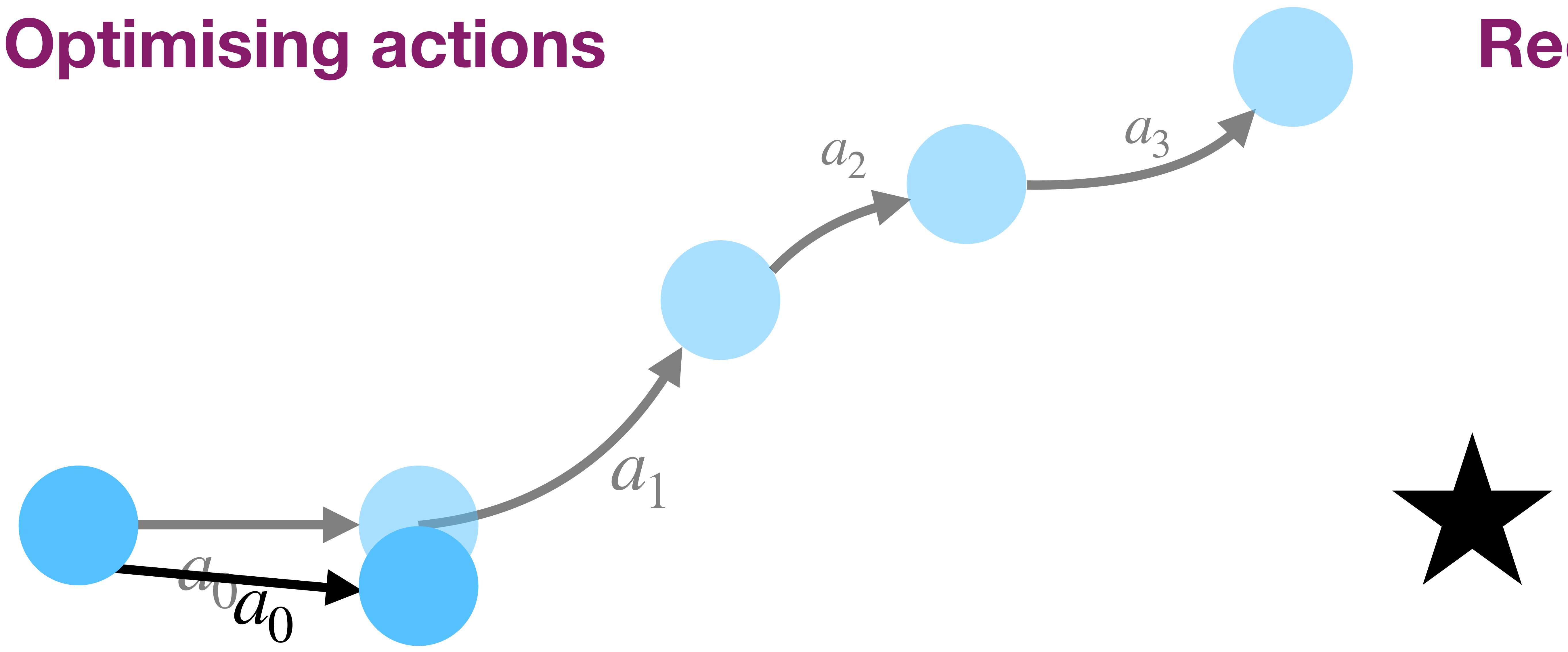
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



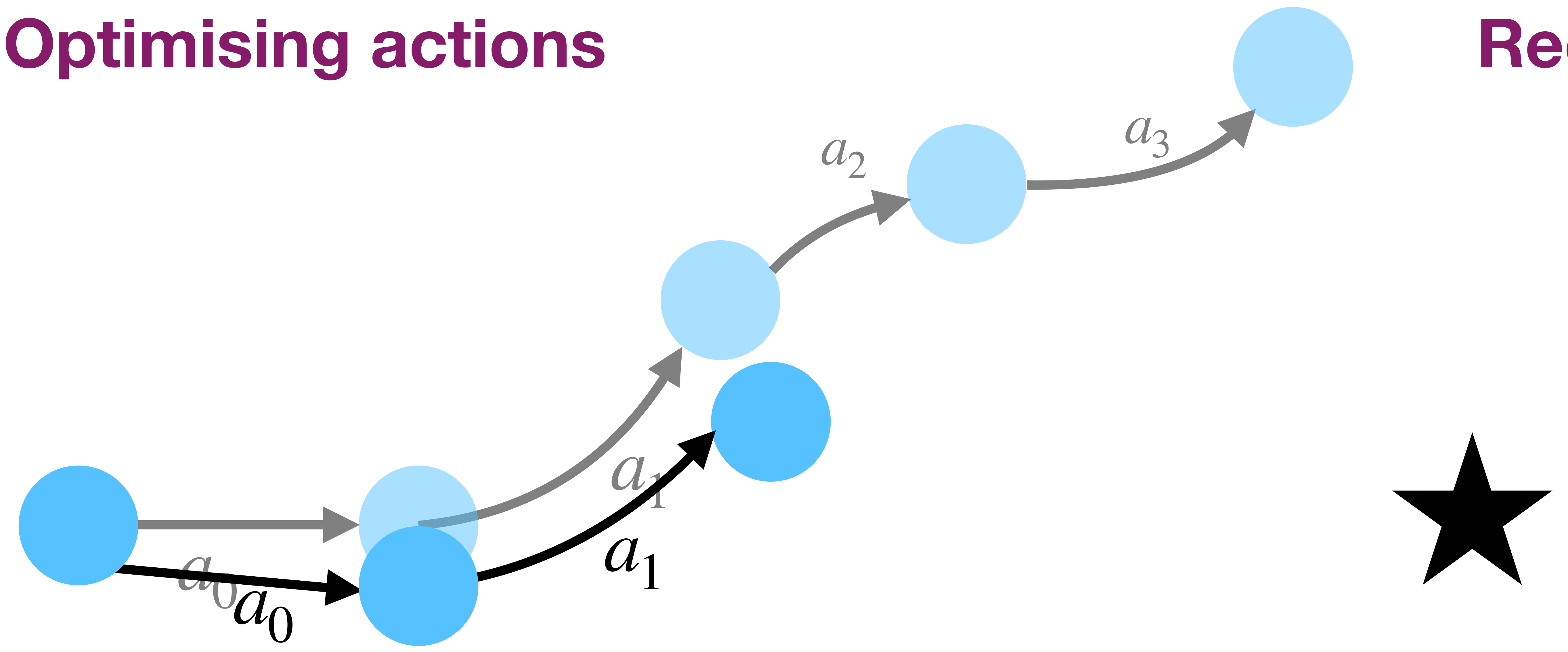
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r\boxed{(f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics



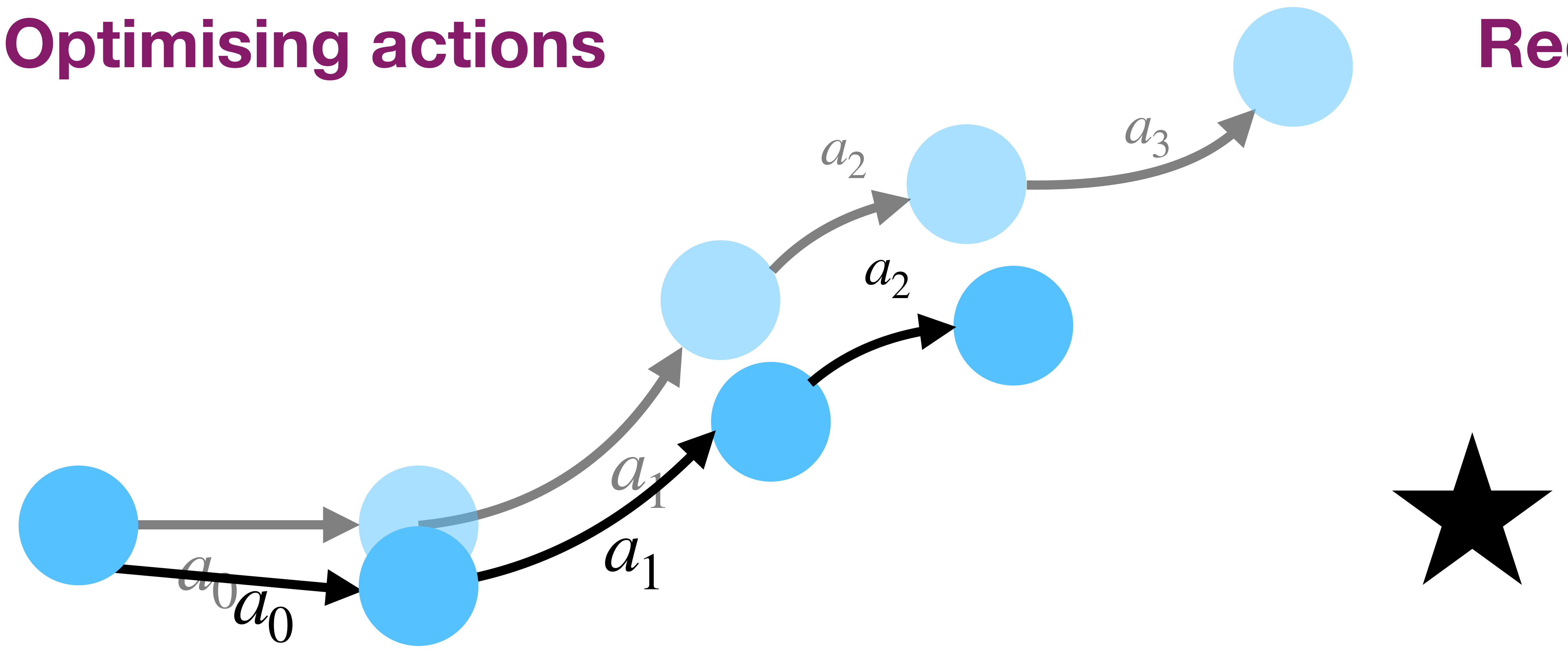
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics

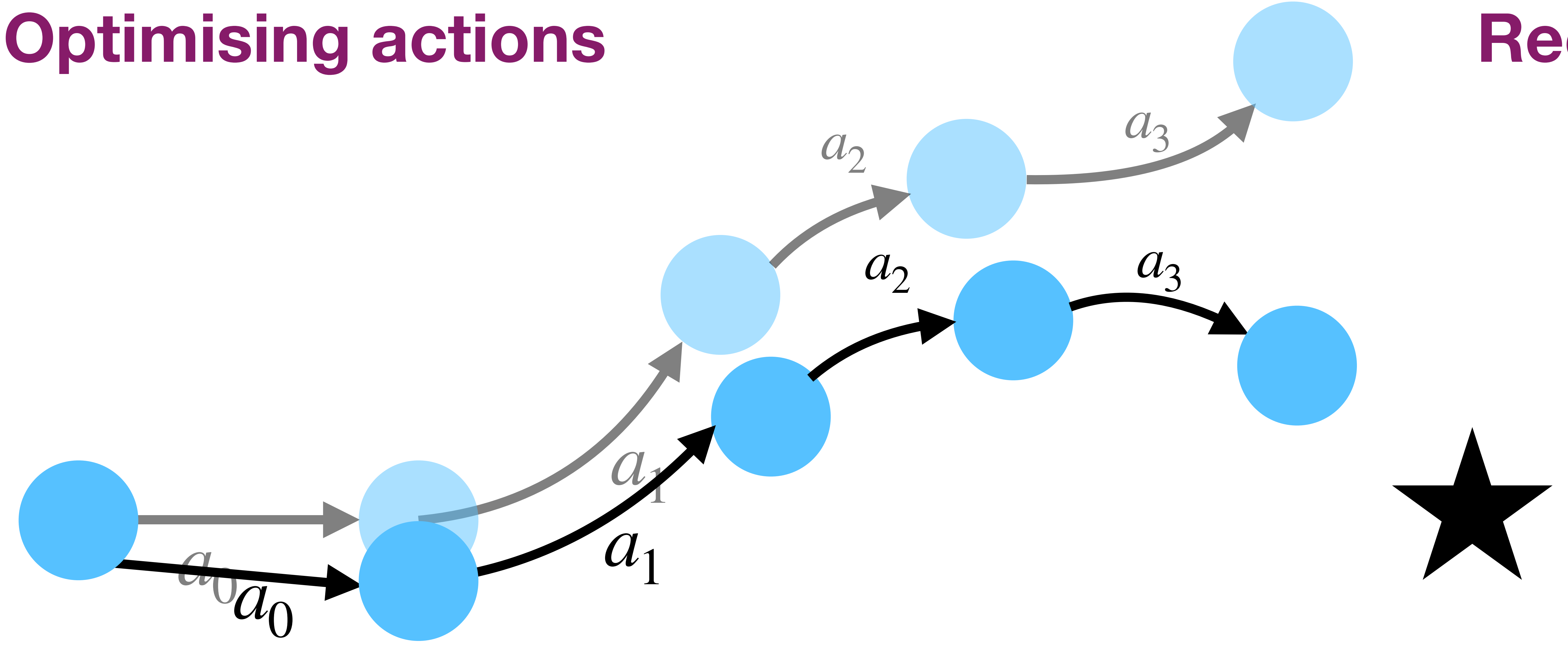


Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r\boxed{(f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions **Recursively evaluate dynamics**



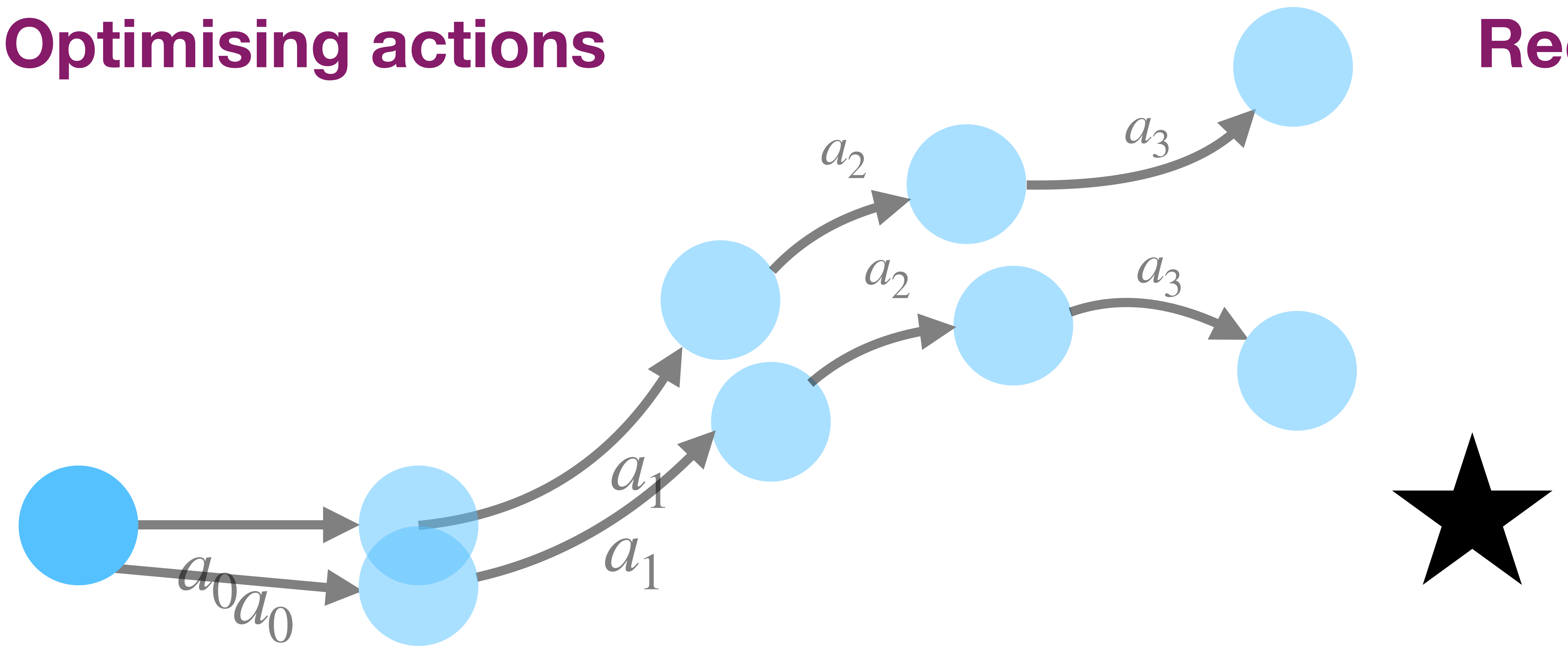
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



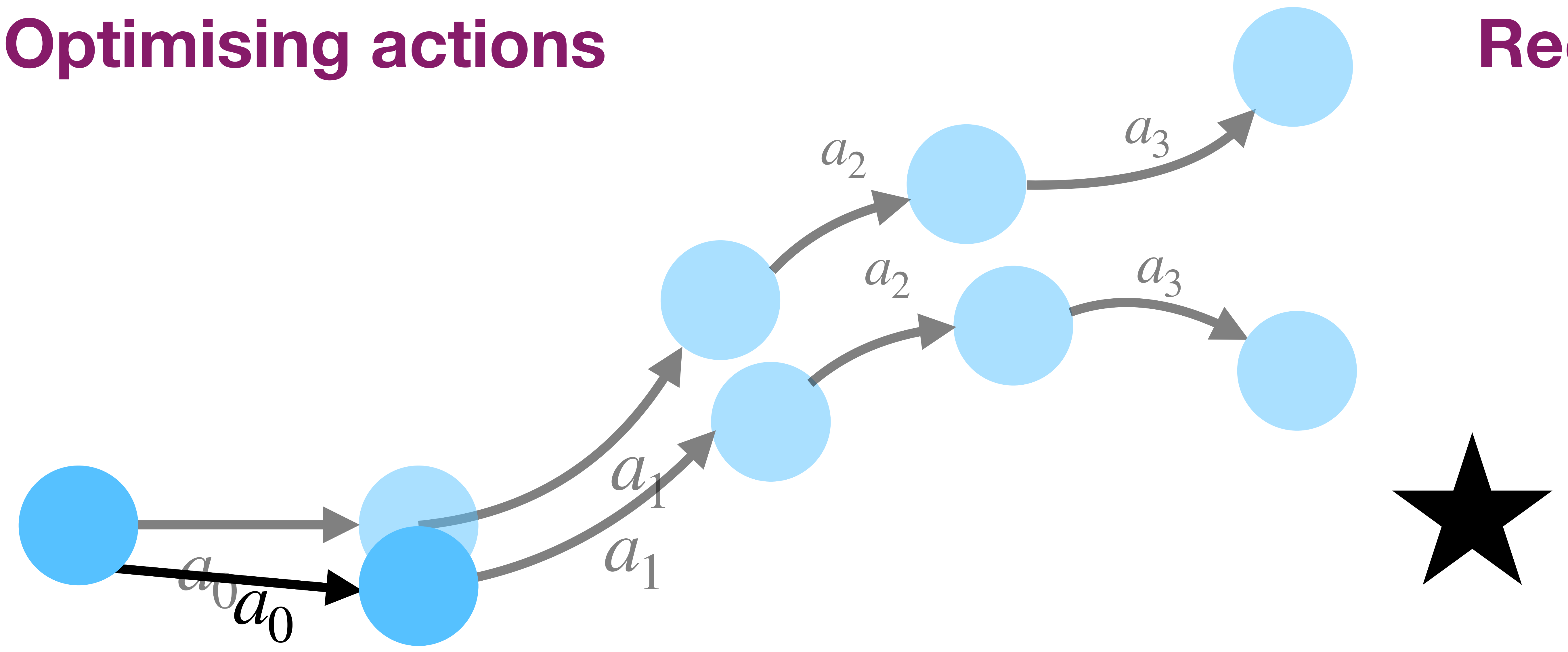
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



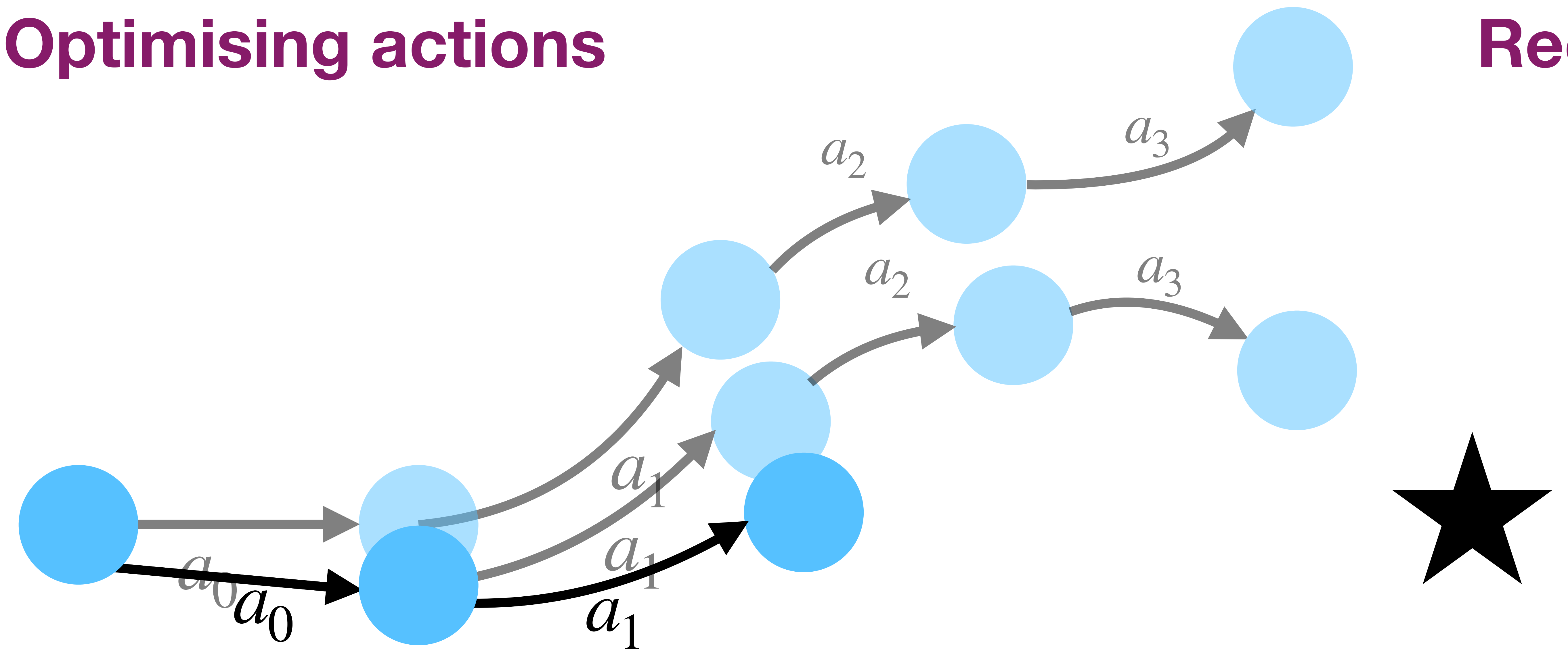
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



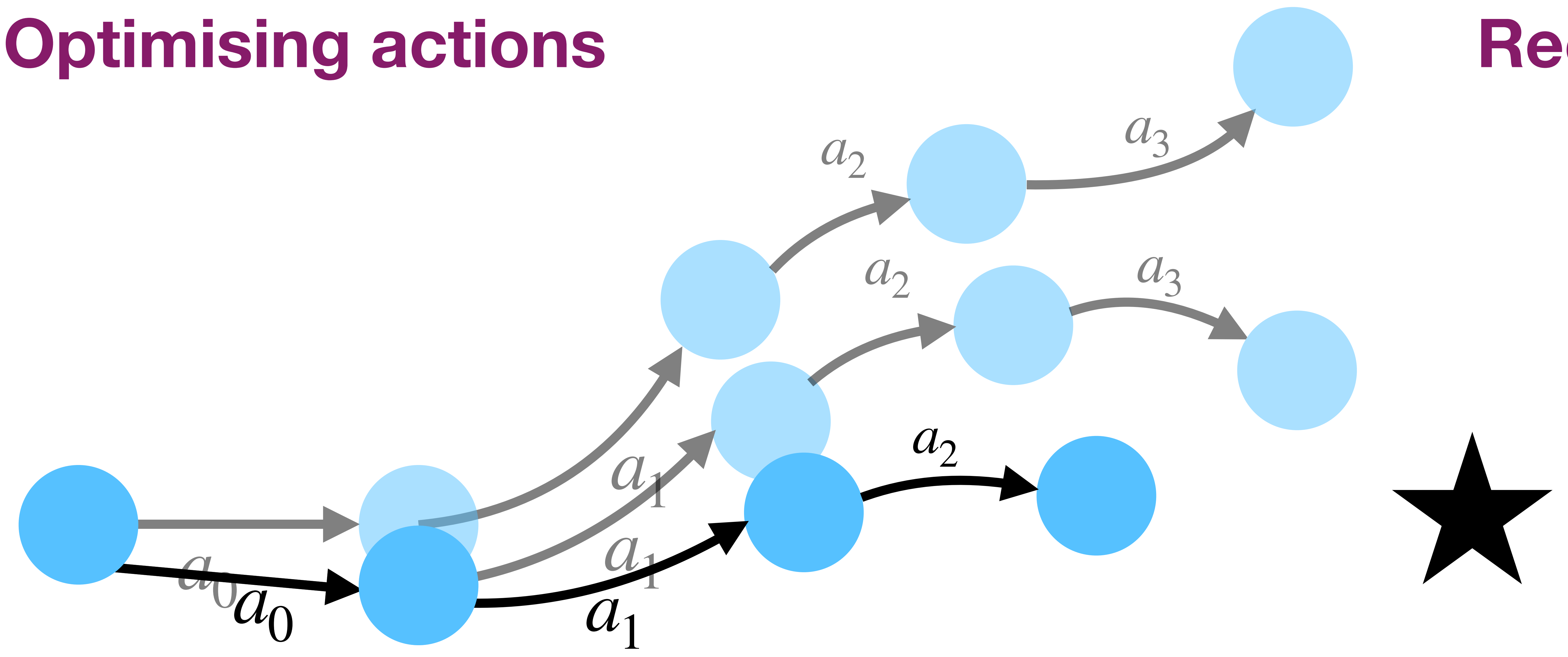
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics



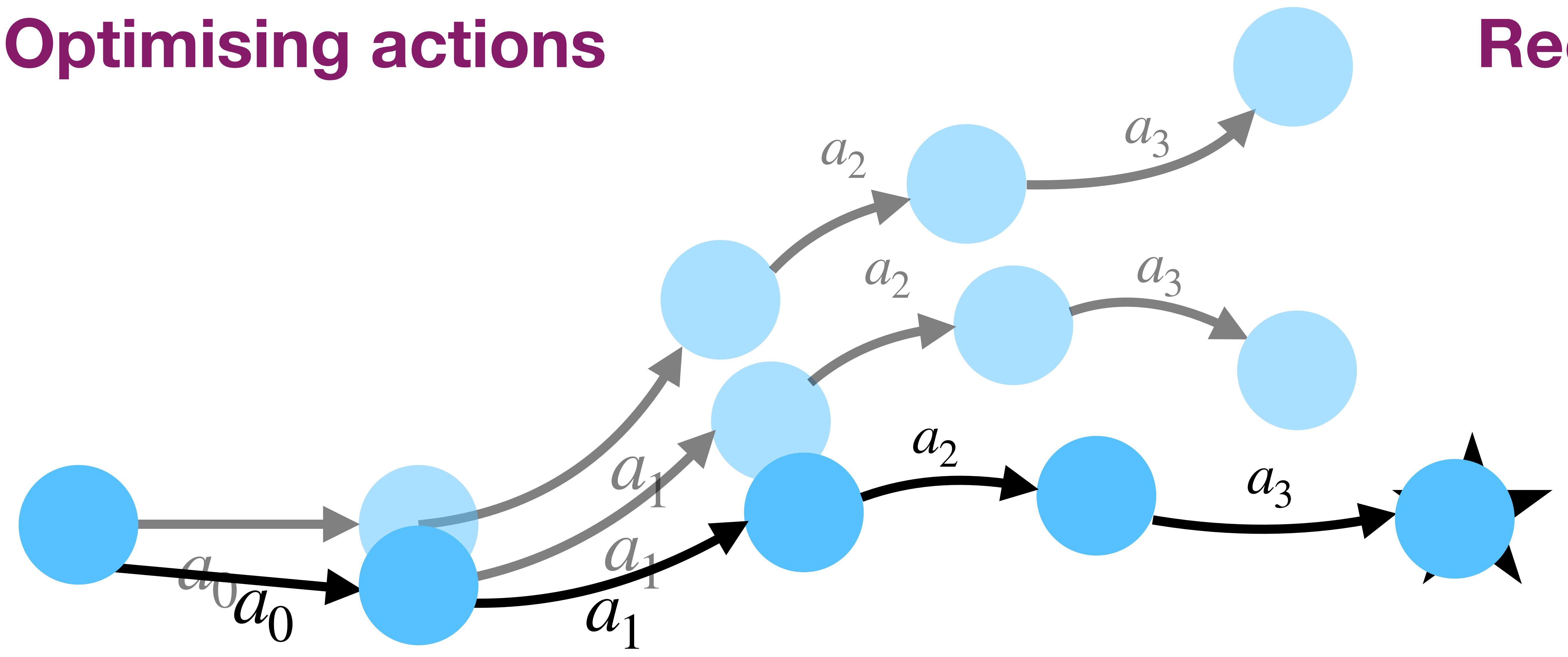
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



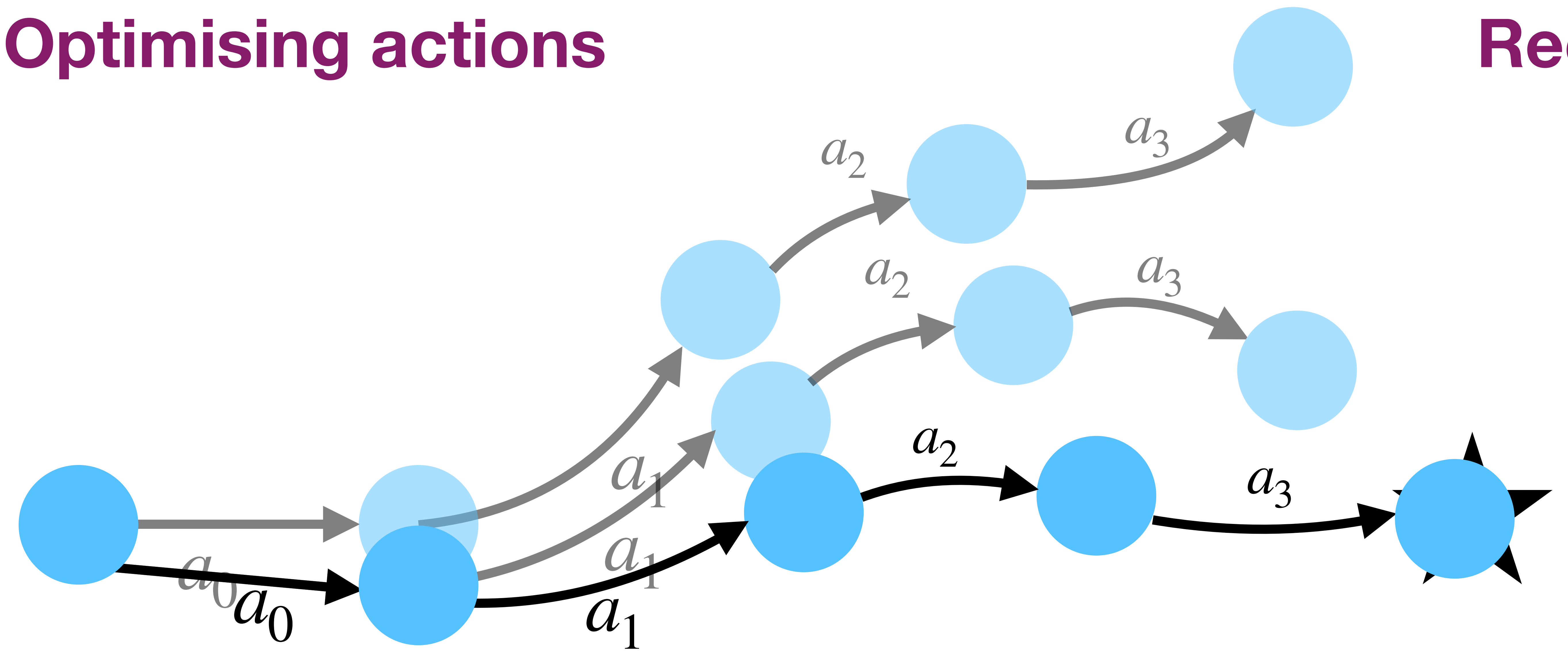
Shooting Methods

Illustration

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H})$$

Optimising actions

Recursively evaluate dynamics



Shooting Methods
Illustration

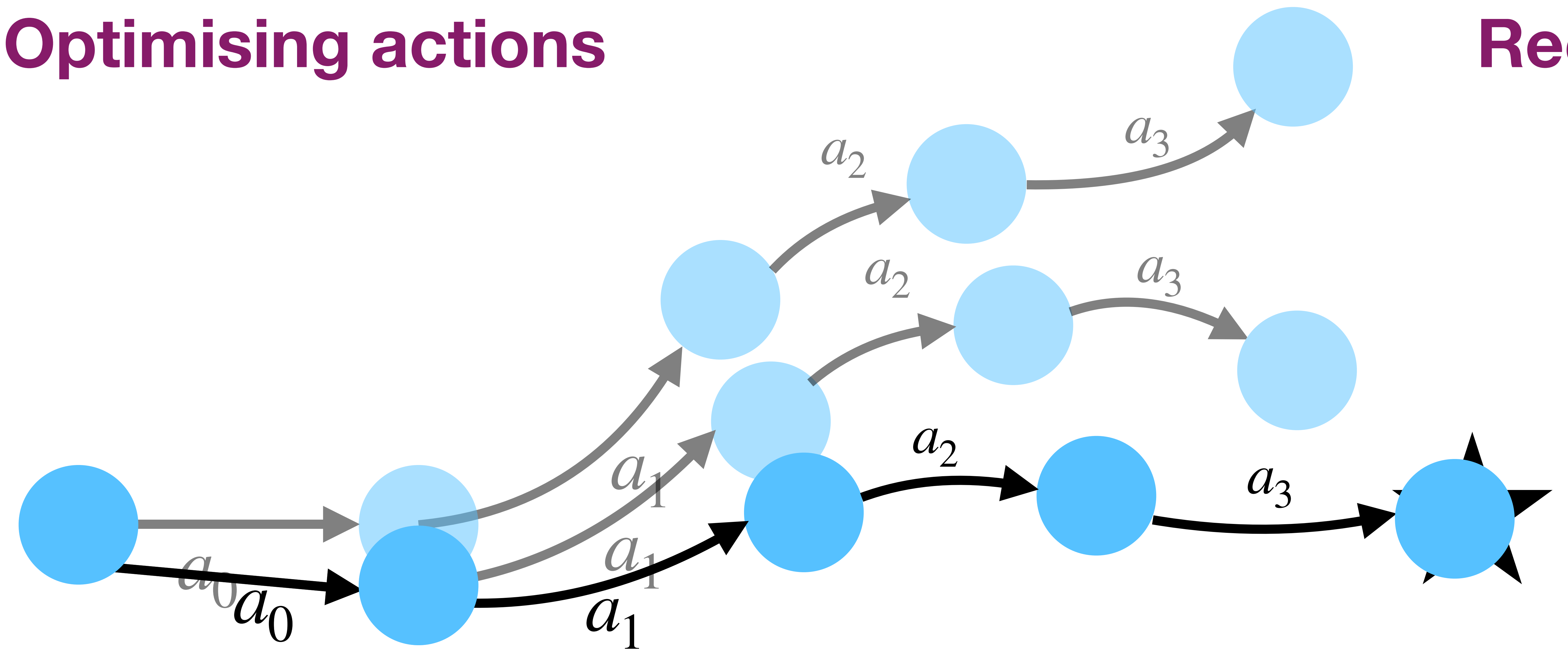
Gradient based approaches are fast

But local minima

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics



Shooting Methods

Illustration

Gradient based approaches are fast

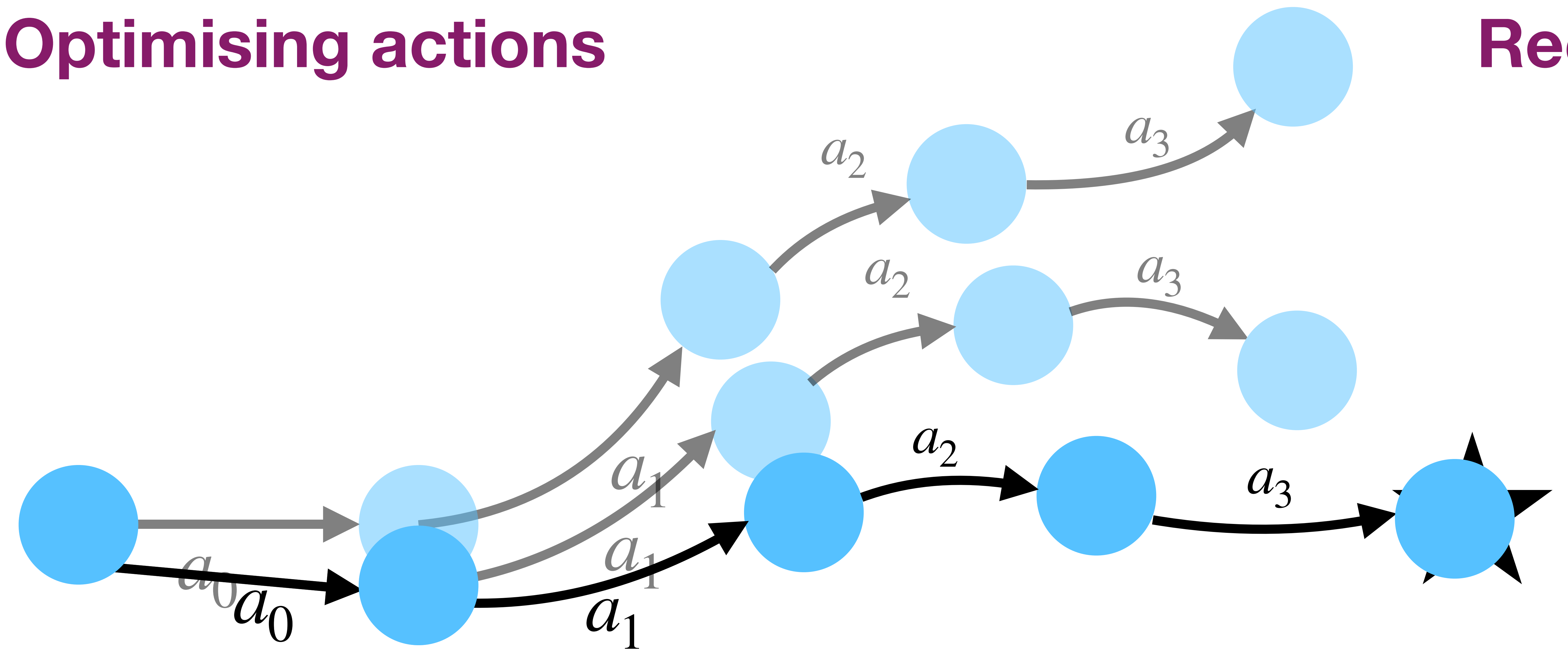
But local minima

And vanishing/exploding gradients

$$\boxed{J(a_{0:H})} = \gamma^0 r(s_0, a_0) + \gamma^1 r(f(s_0, a_0), a_1) + \ldots + \gamma^H r(\boxed{f(f(\ldots), a_{H-1}), a_H)}$$

Optimising actions

Recursively evaluate dynamics

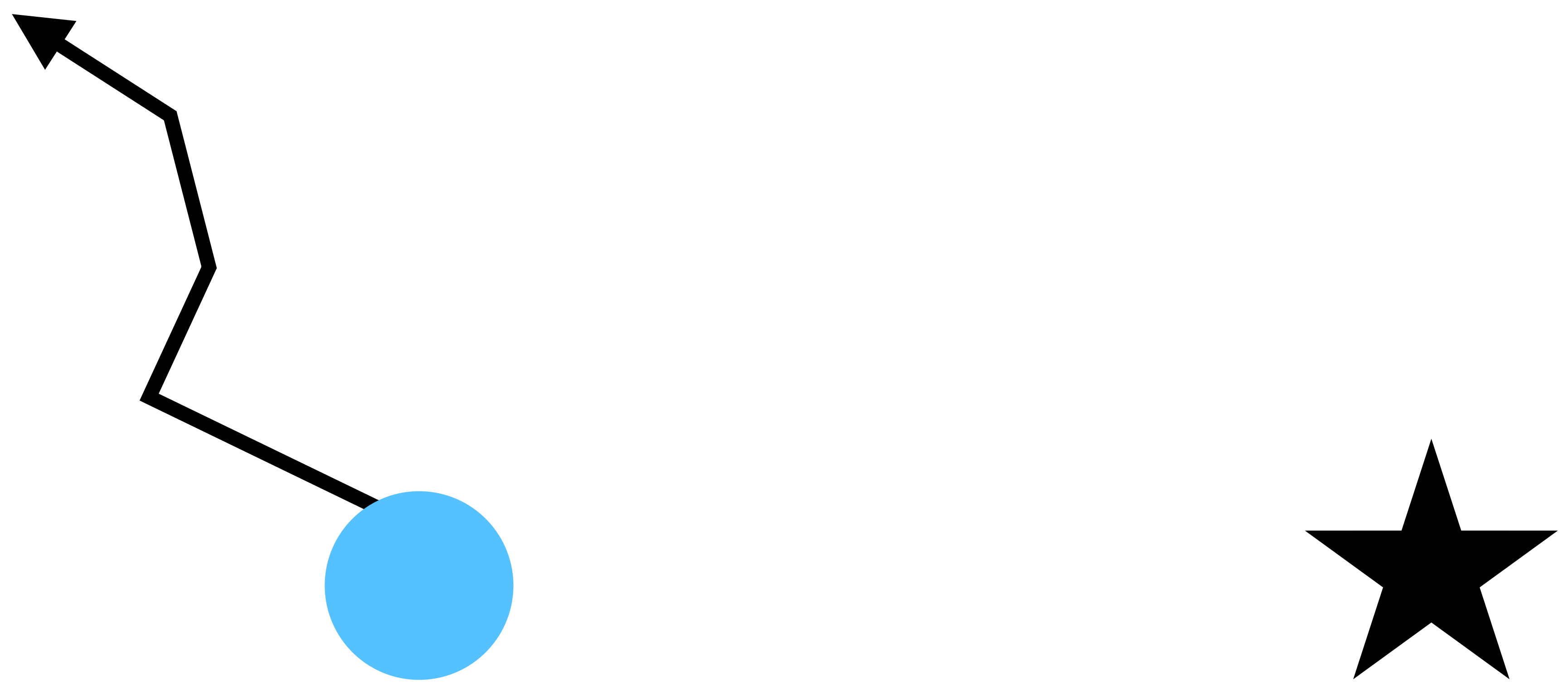


Shooting Methods

Random shooting

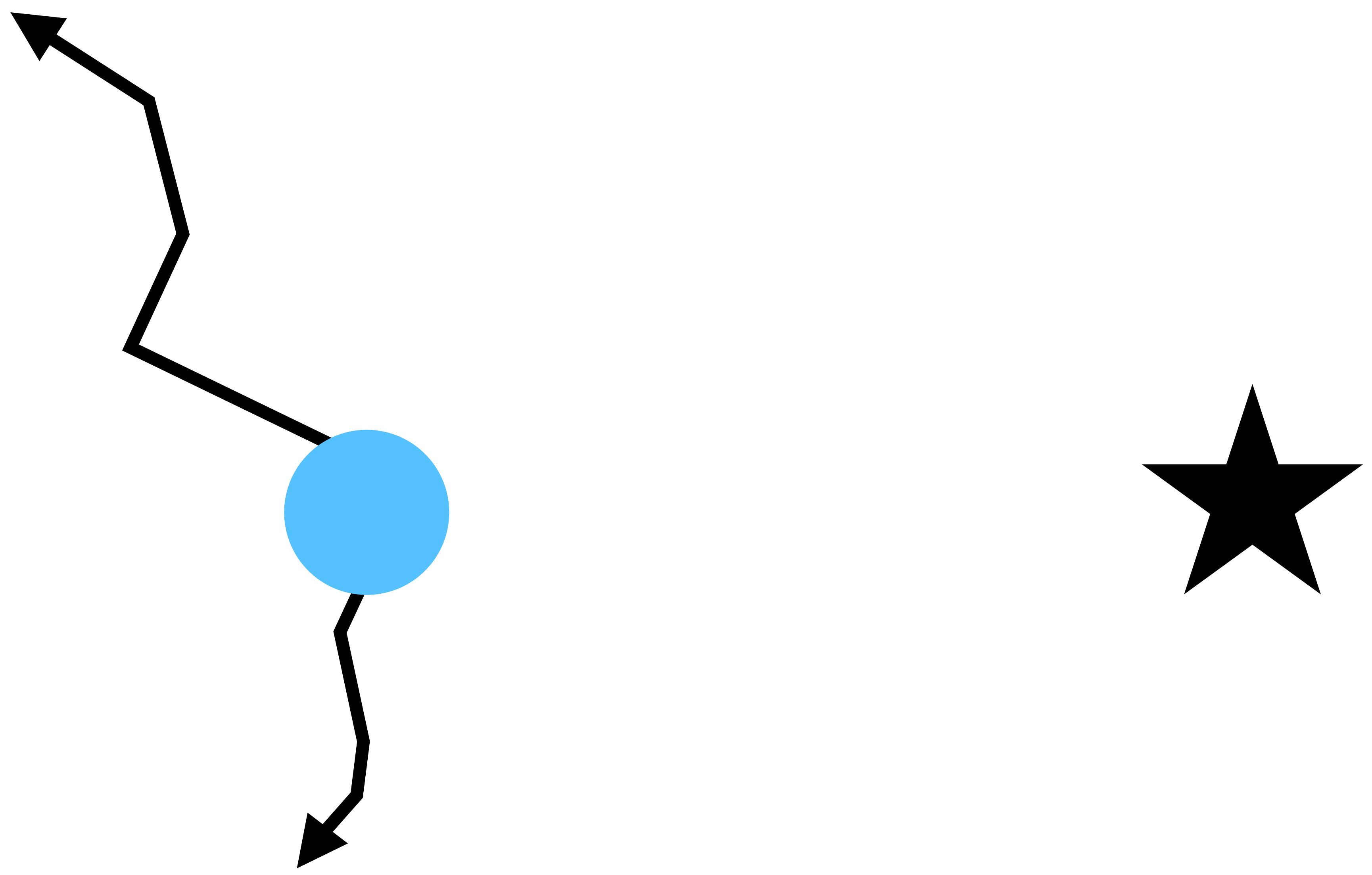
Shooting Methods

Random shooting



Shooting Methods

Random shooting

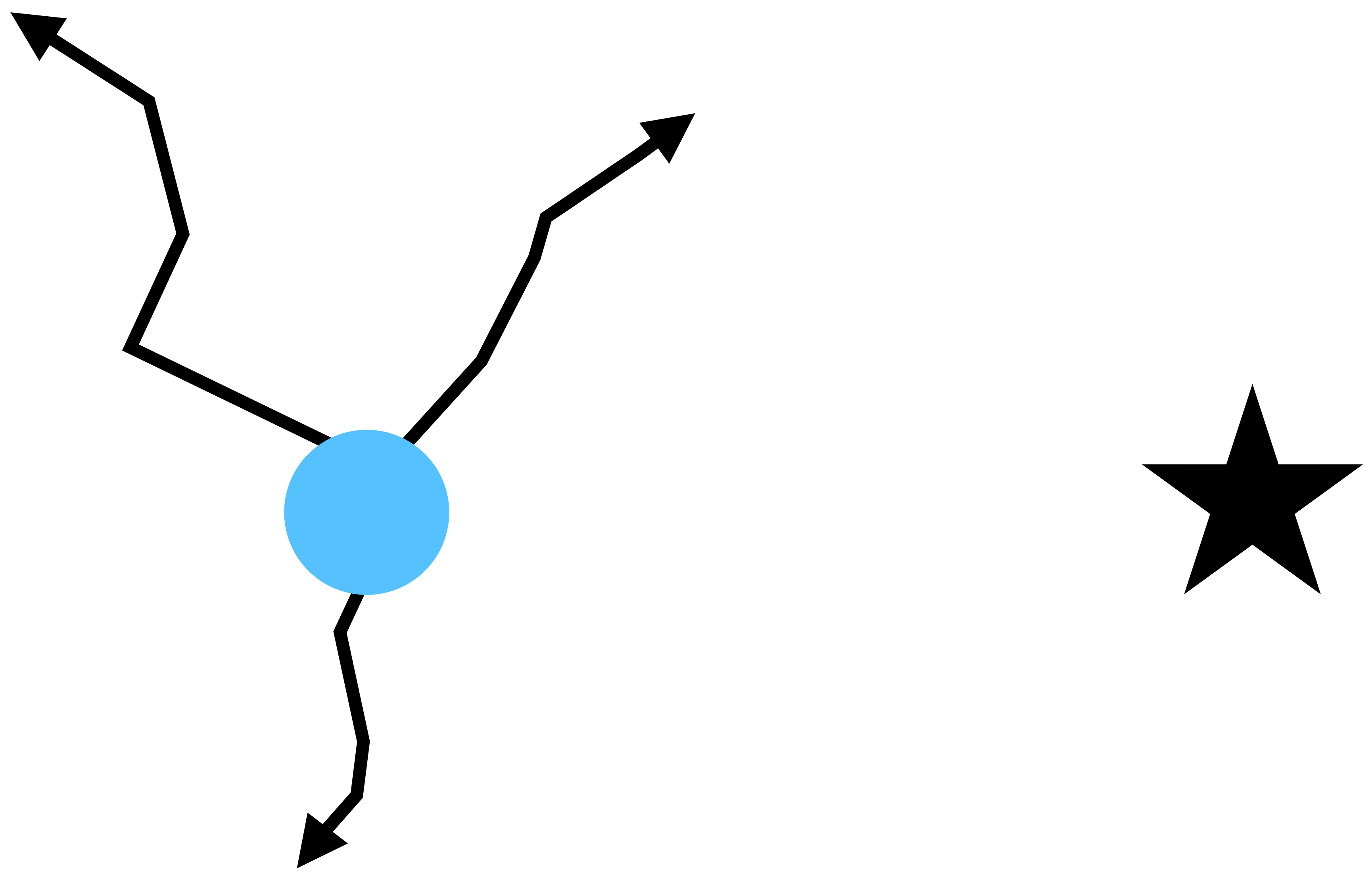


FCAI

fcai.fi

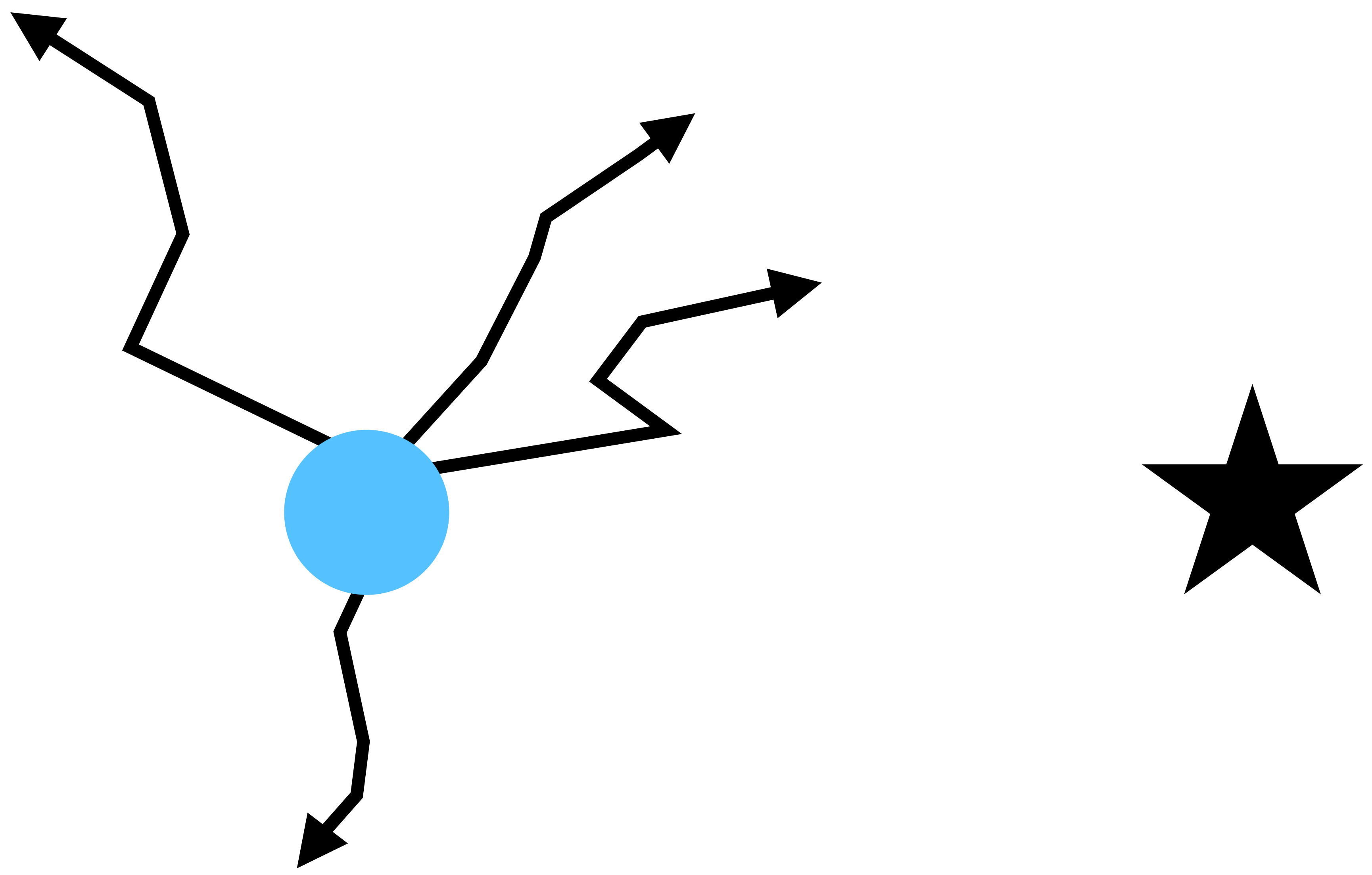
Shooting Methods

Random shooting



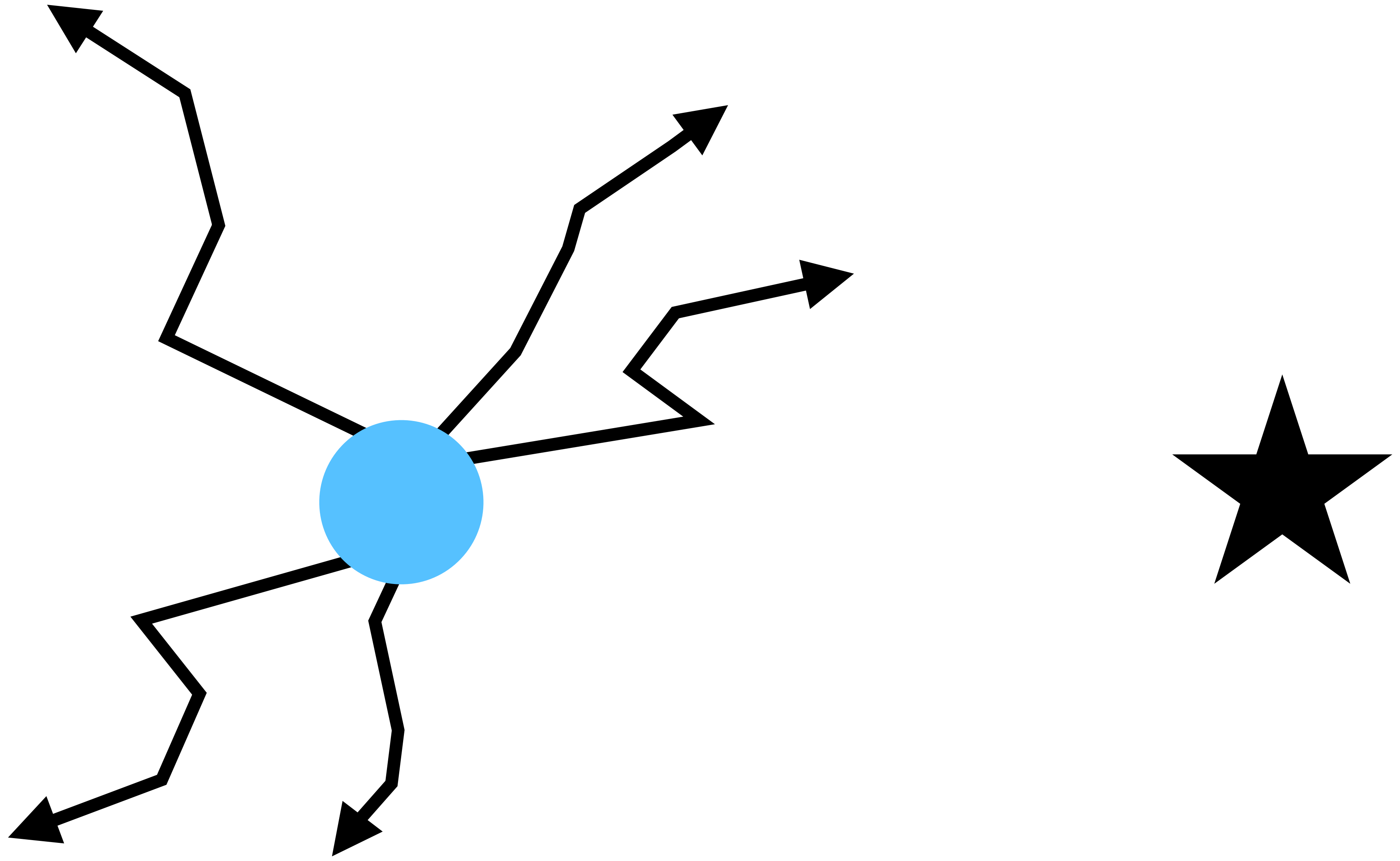
Shooting Methods

Random shooting



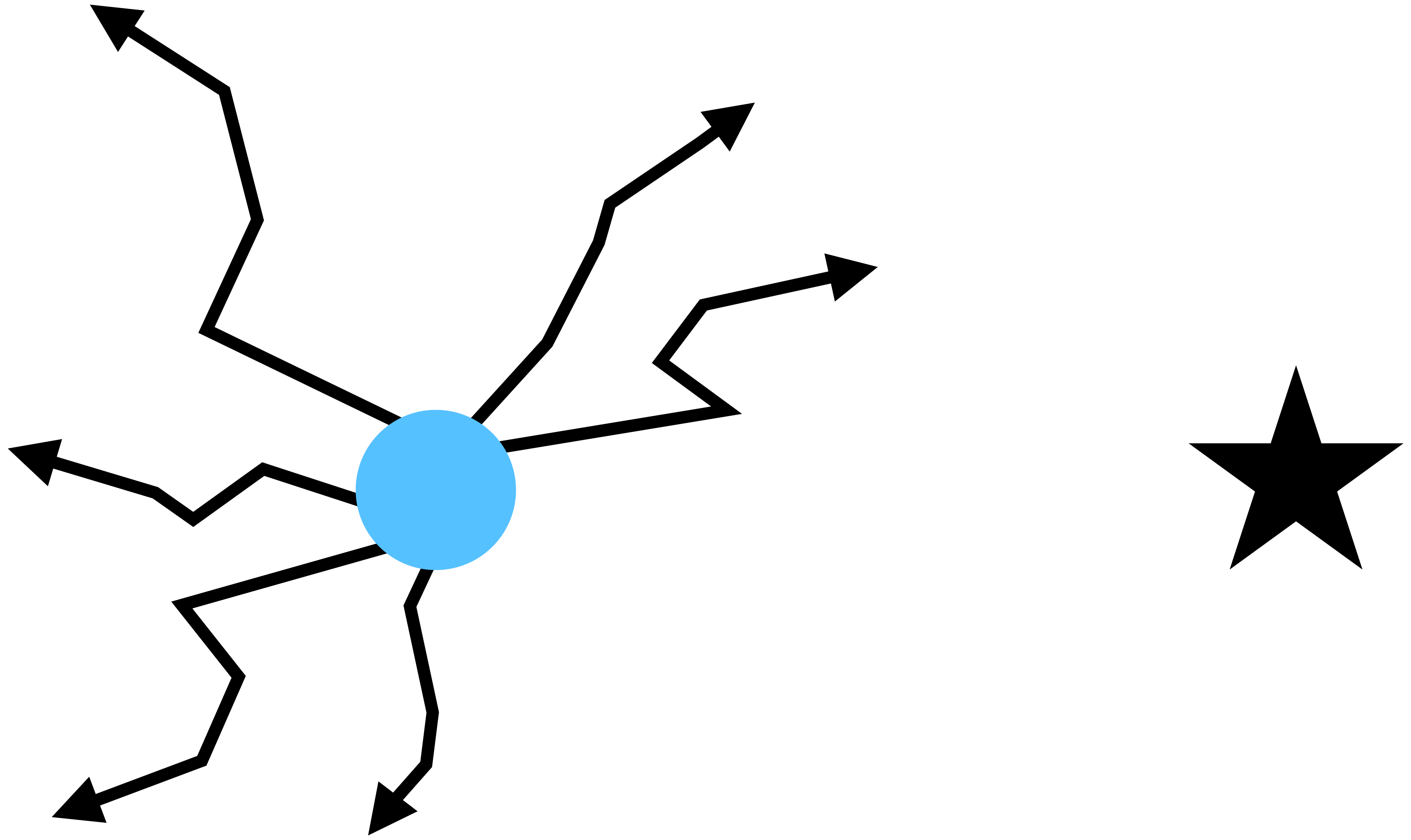
Shooting Methods

Random shooting



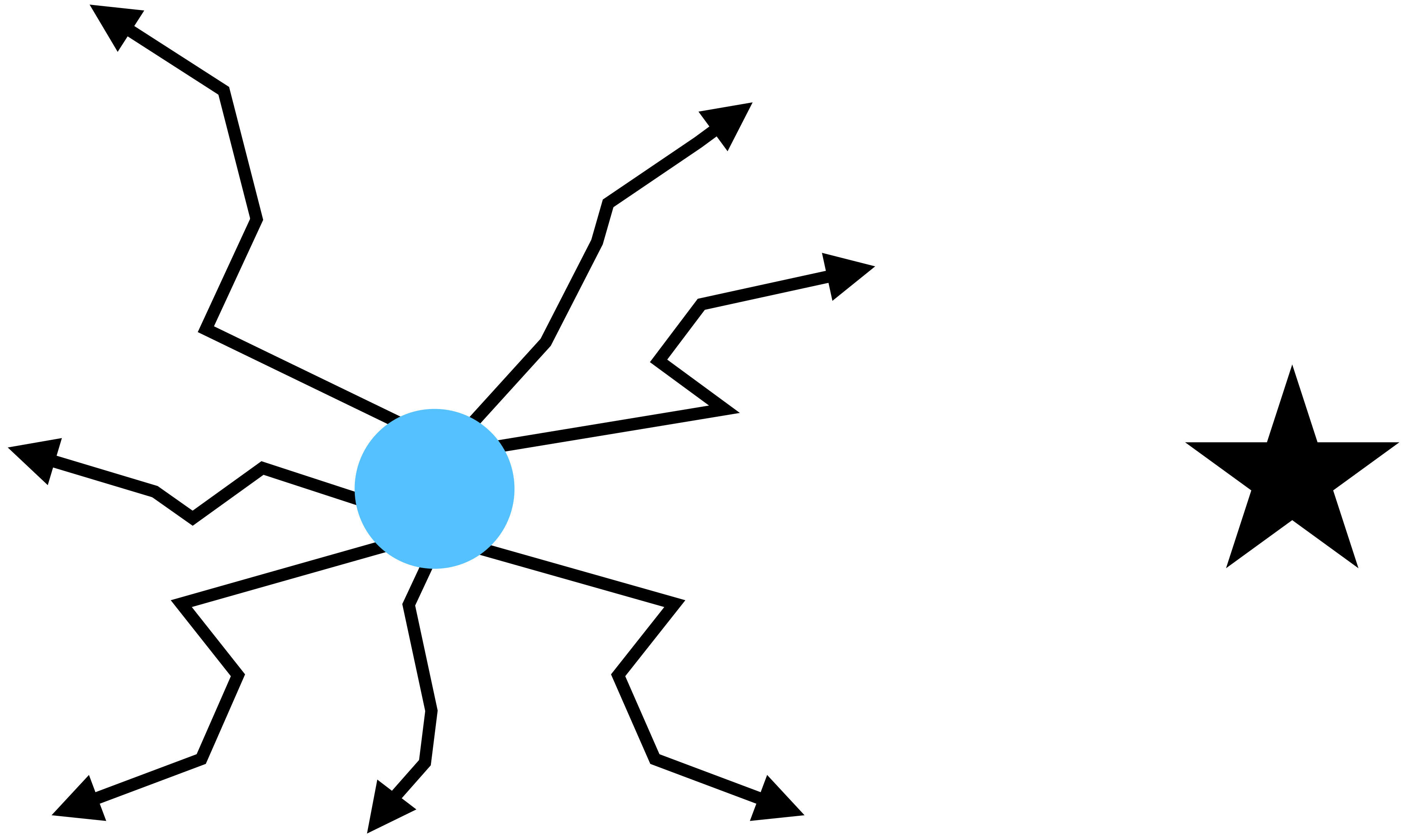
Shooting Methods

Random shooting



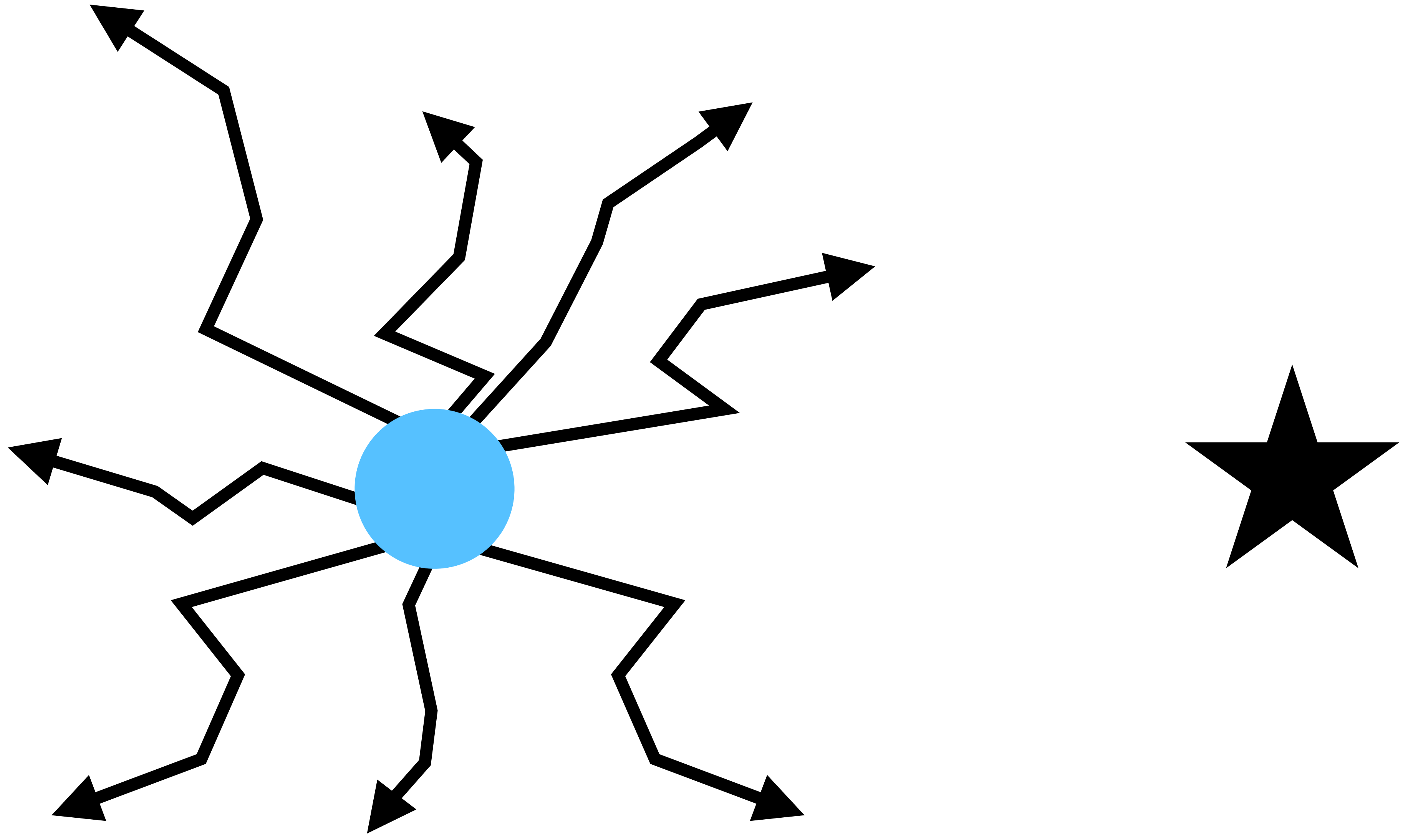
Shooting Methods

Random shooting



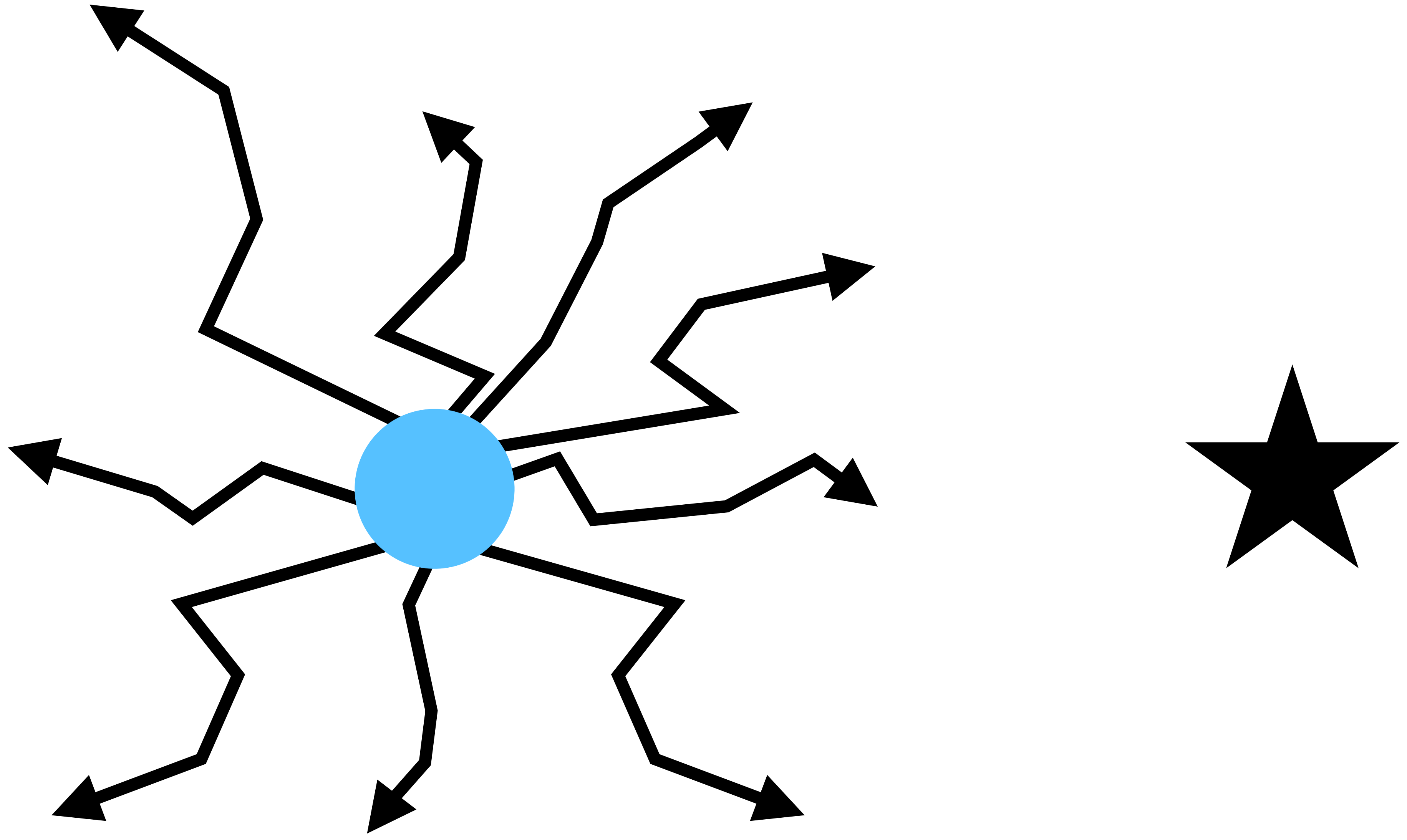
Shooting Methods

Random shooting

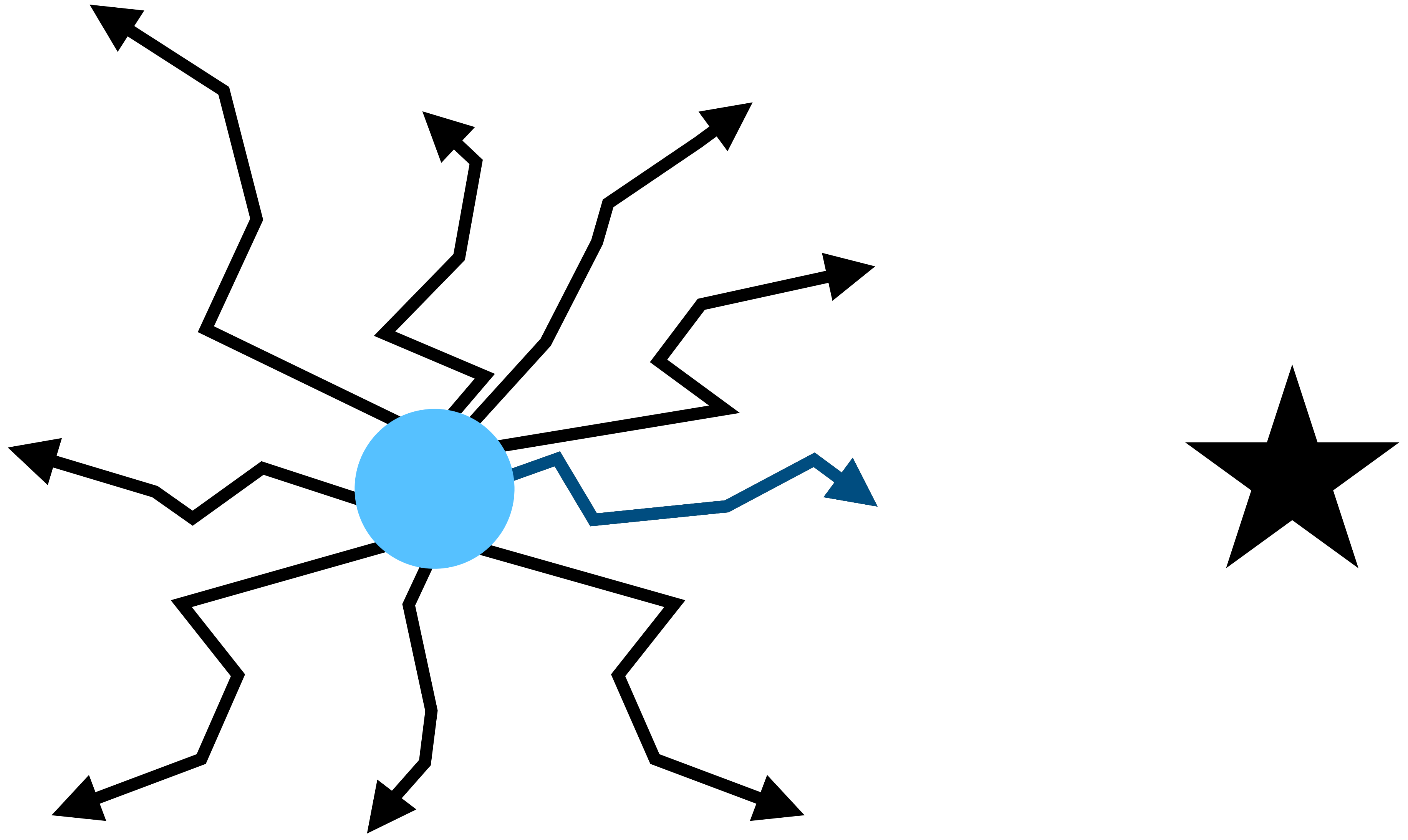


Shooting Methods

Random shooting



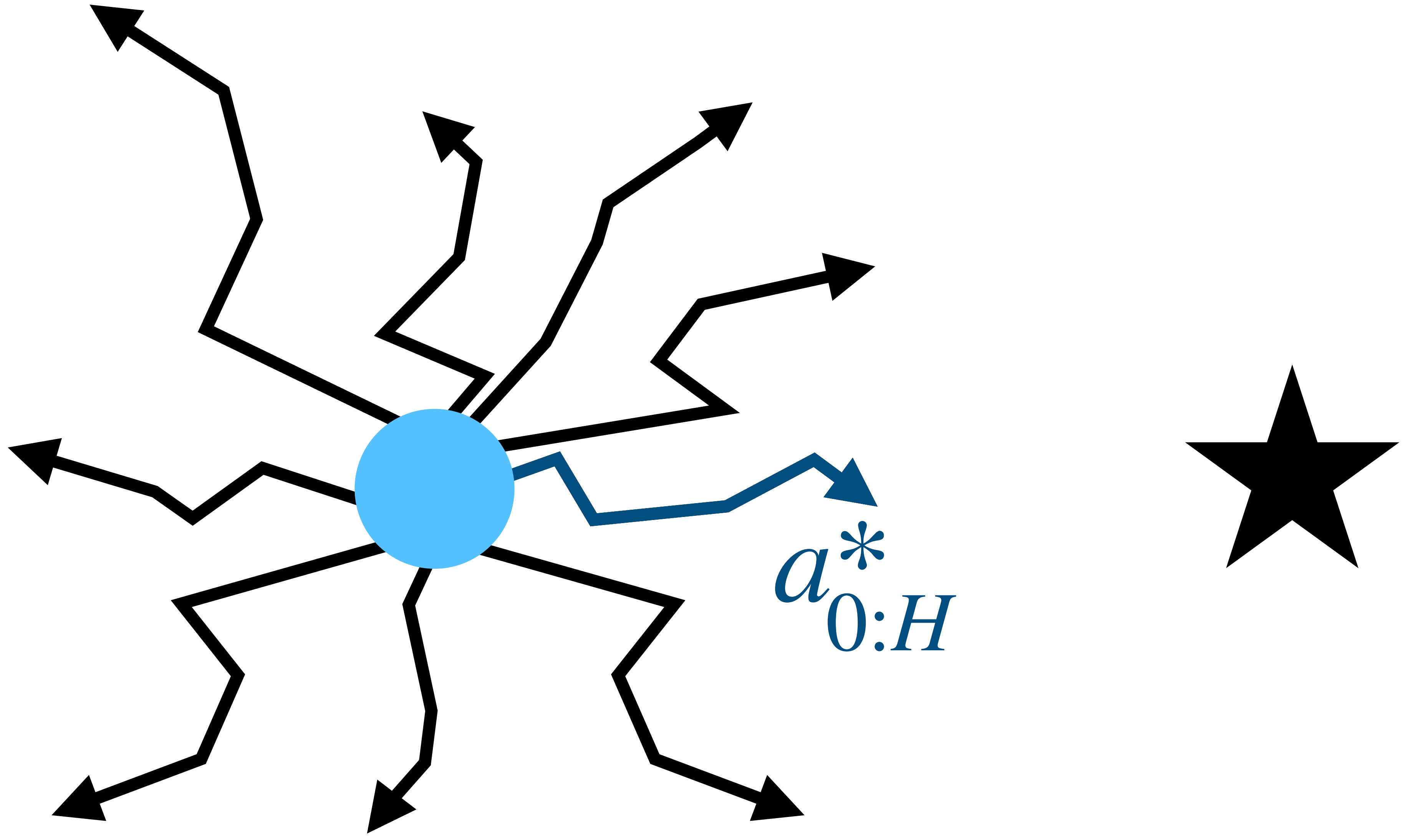
Shooting Methods
Random shooting



FCAI fcai.fi

Shooting Methods

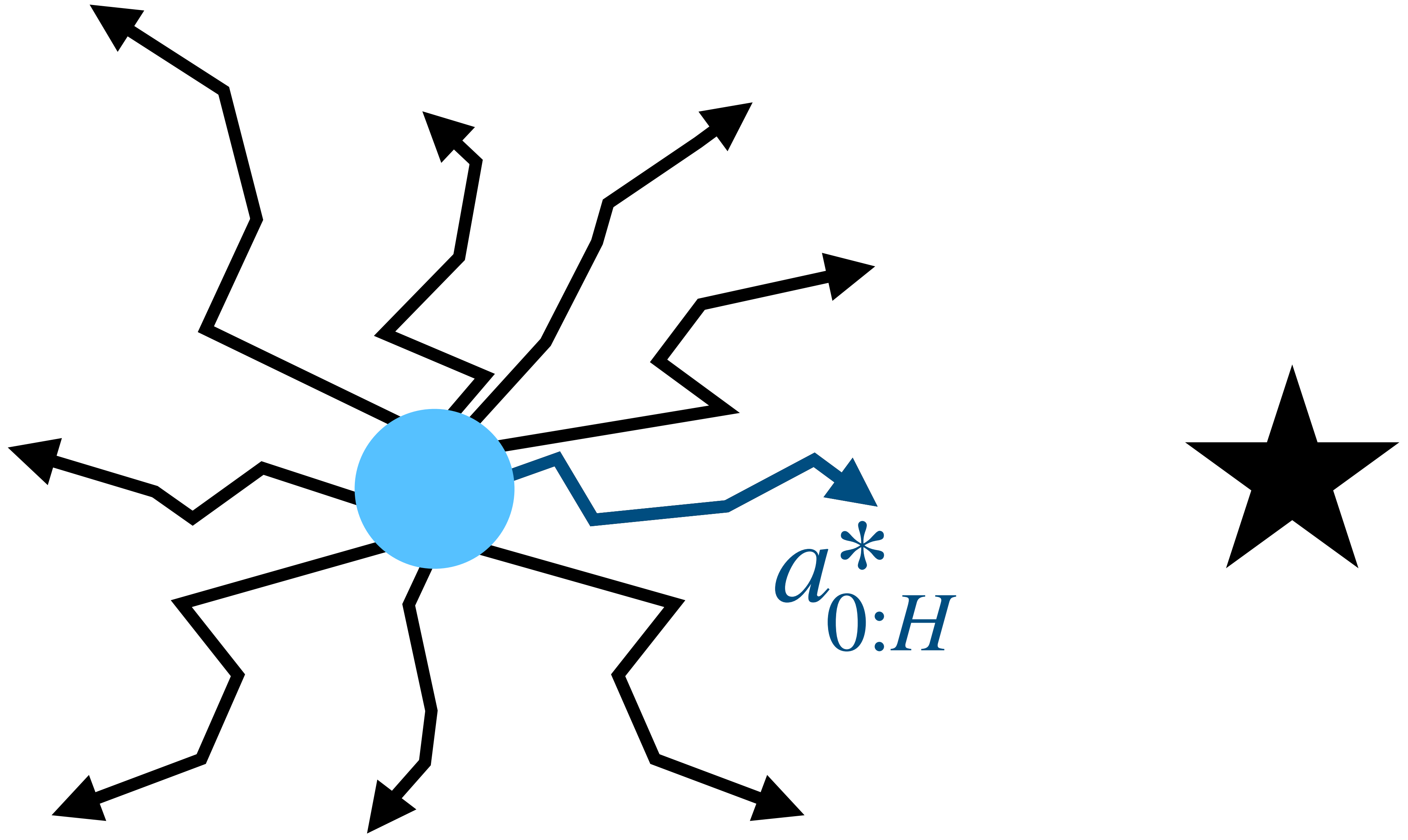
Random shooting



Shooting Methods

Random shooting

Simple

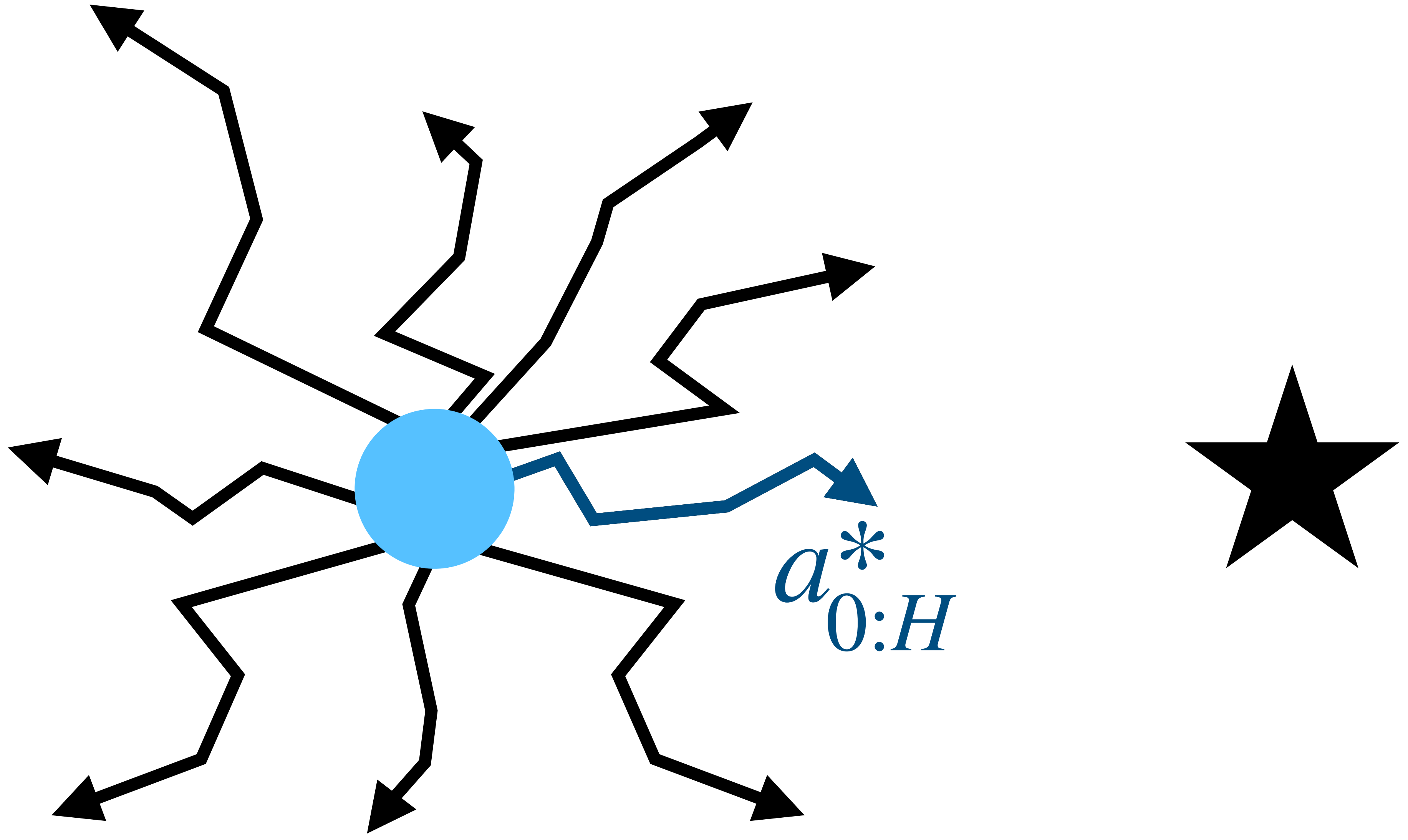


Shooting Methods

Random shooting

Simple

Parallelisable

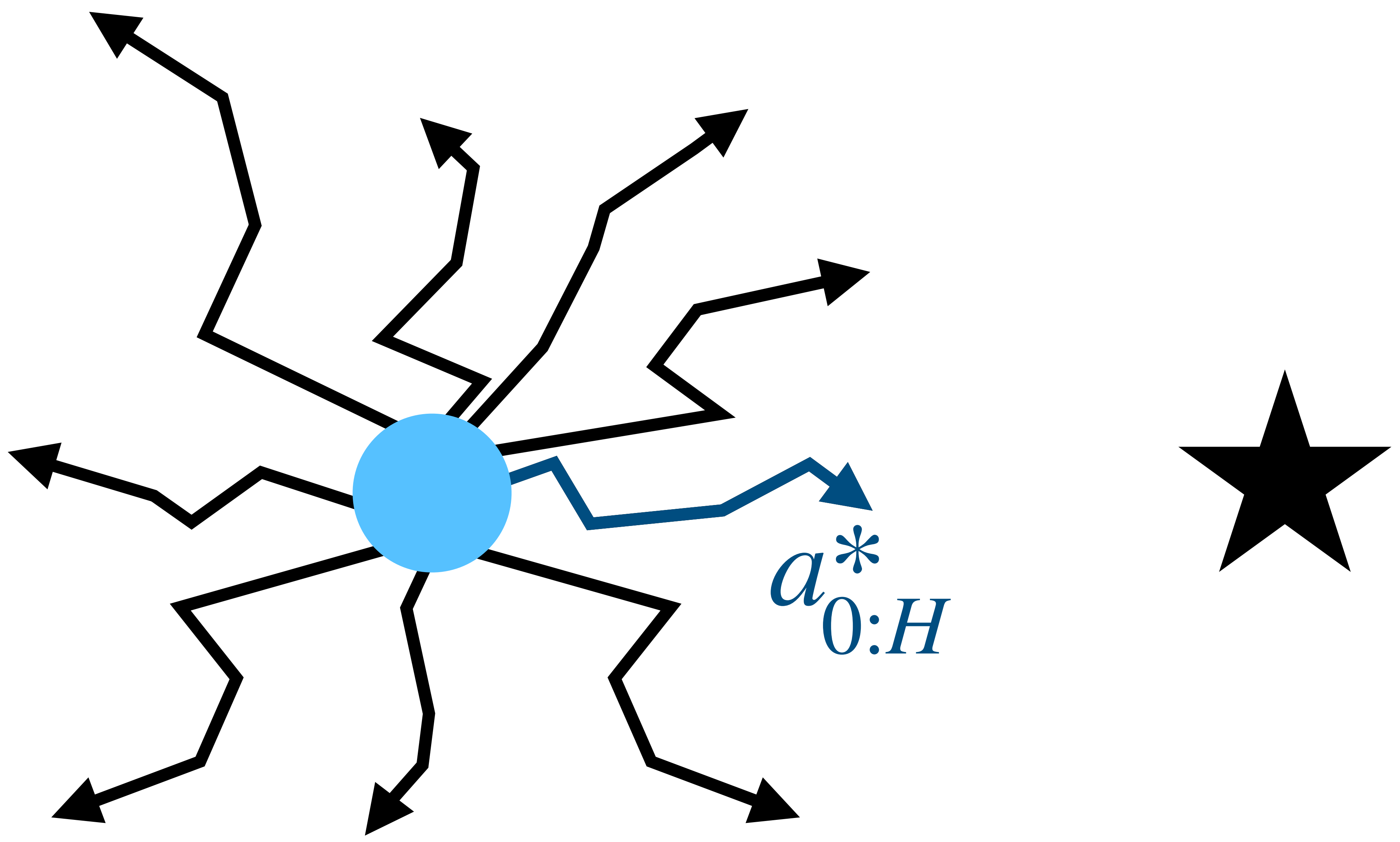


Shooting Methods
Random shooting

Simple

Parallelisable

Sample inefficient



FCAI

fcai.fi

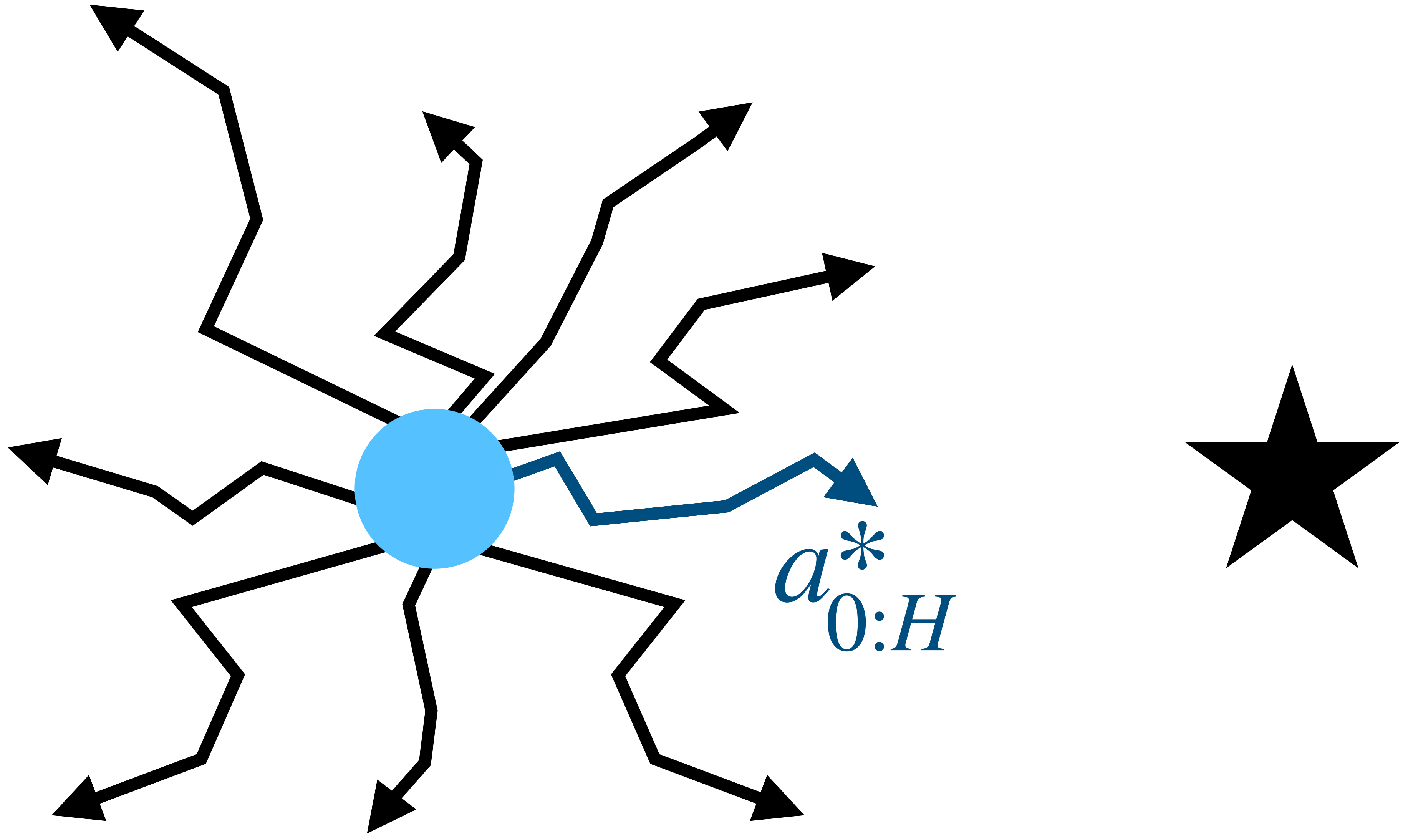
Shooting Methods

Random shooting

Simple

Parallelisable

Sample inefficient



Shooting Methods

Cross-Entropy Method

Iteration 1

Shooting Methods

Cross-Entropy Method

Iteration 1

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

Shooting Methods

Cross-Entropy Method

Iteration 1

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Shooting Methods

Cross-Entropy Method

Iteration 1

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

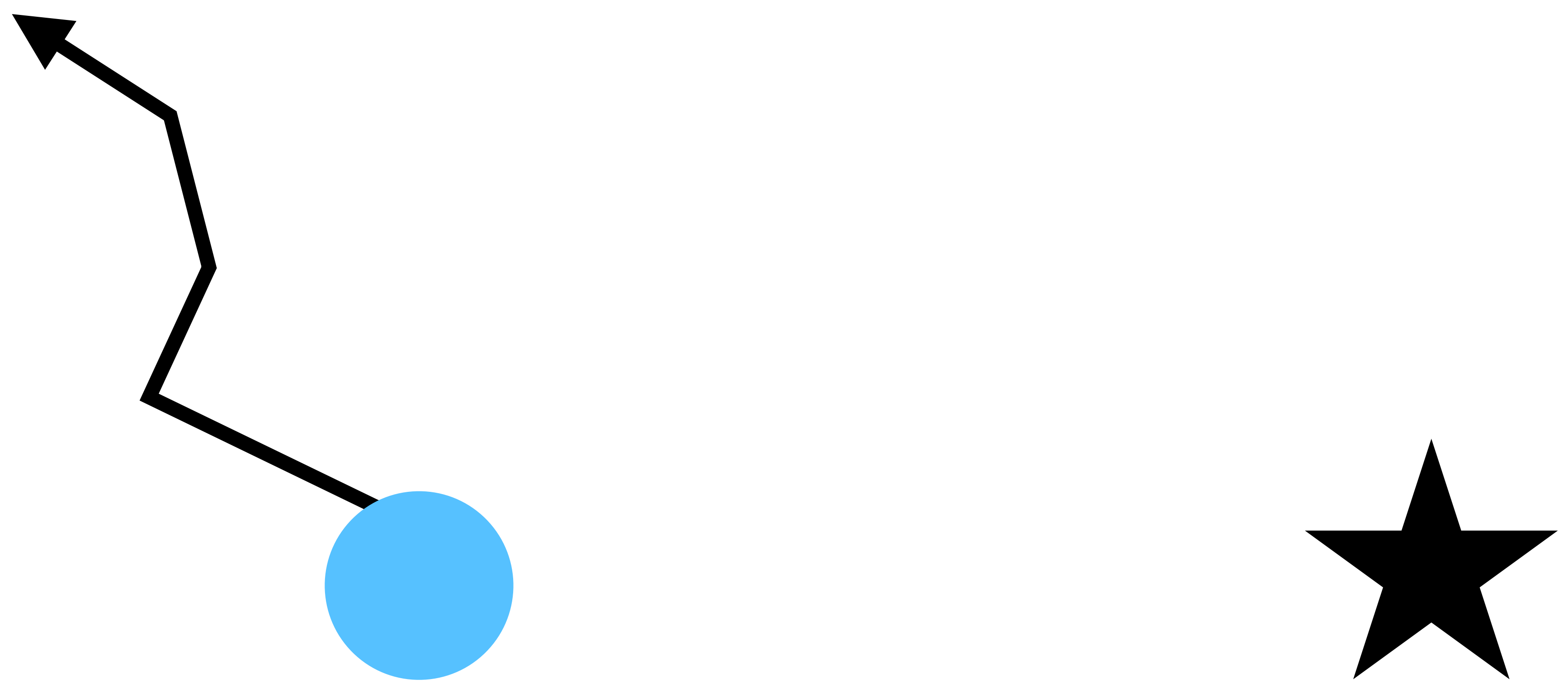
For each iteration

  Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

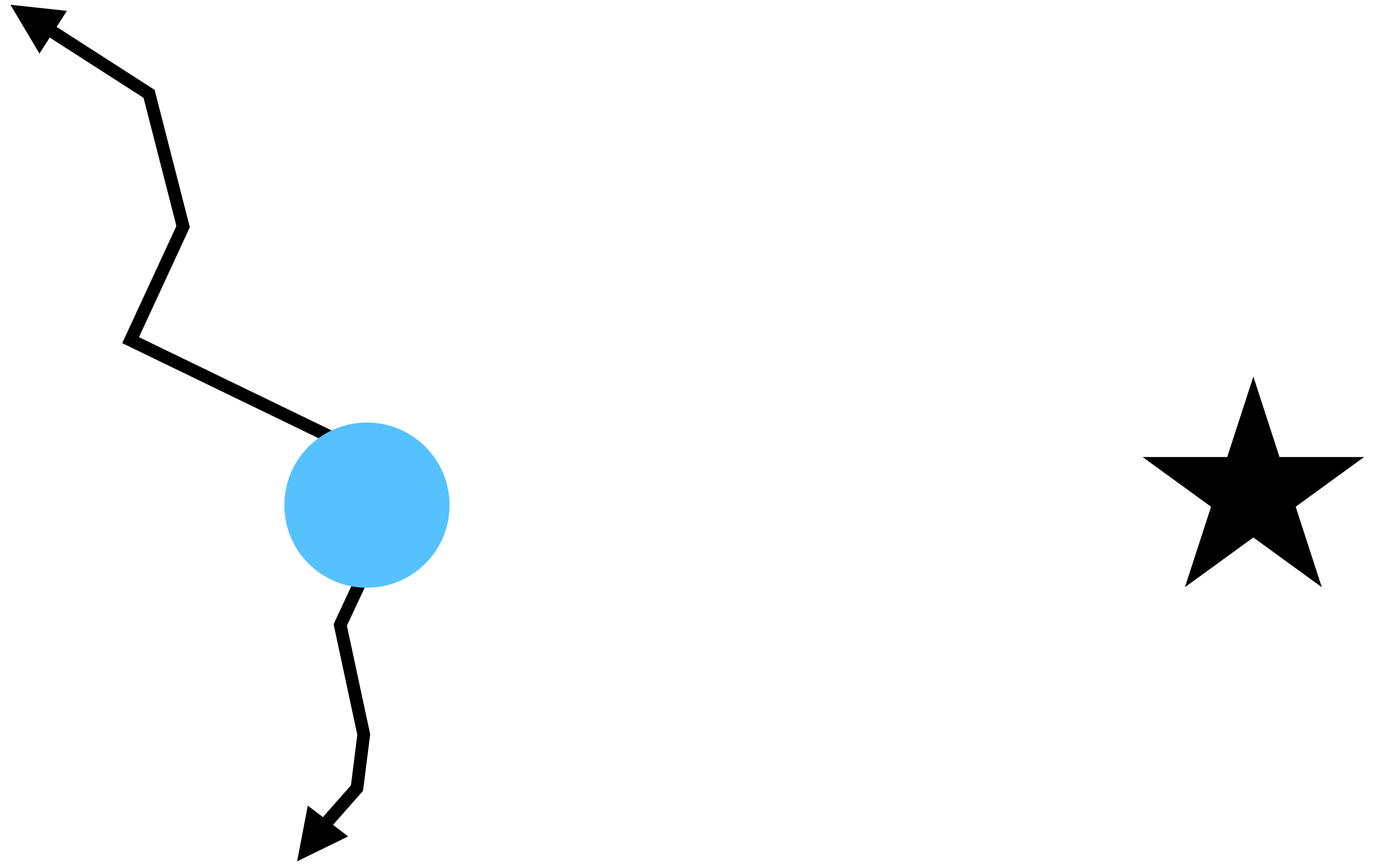
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

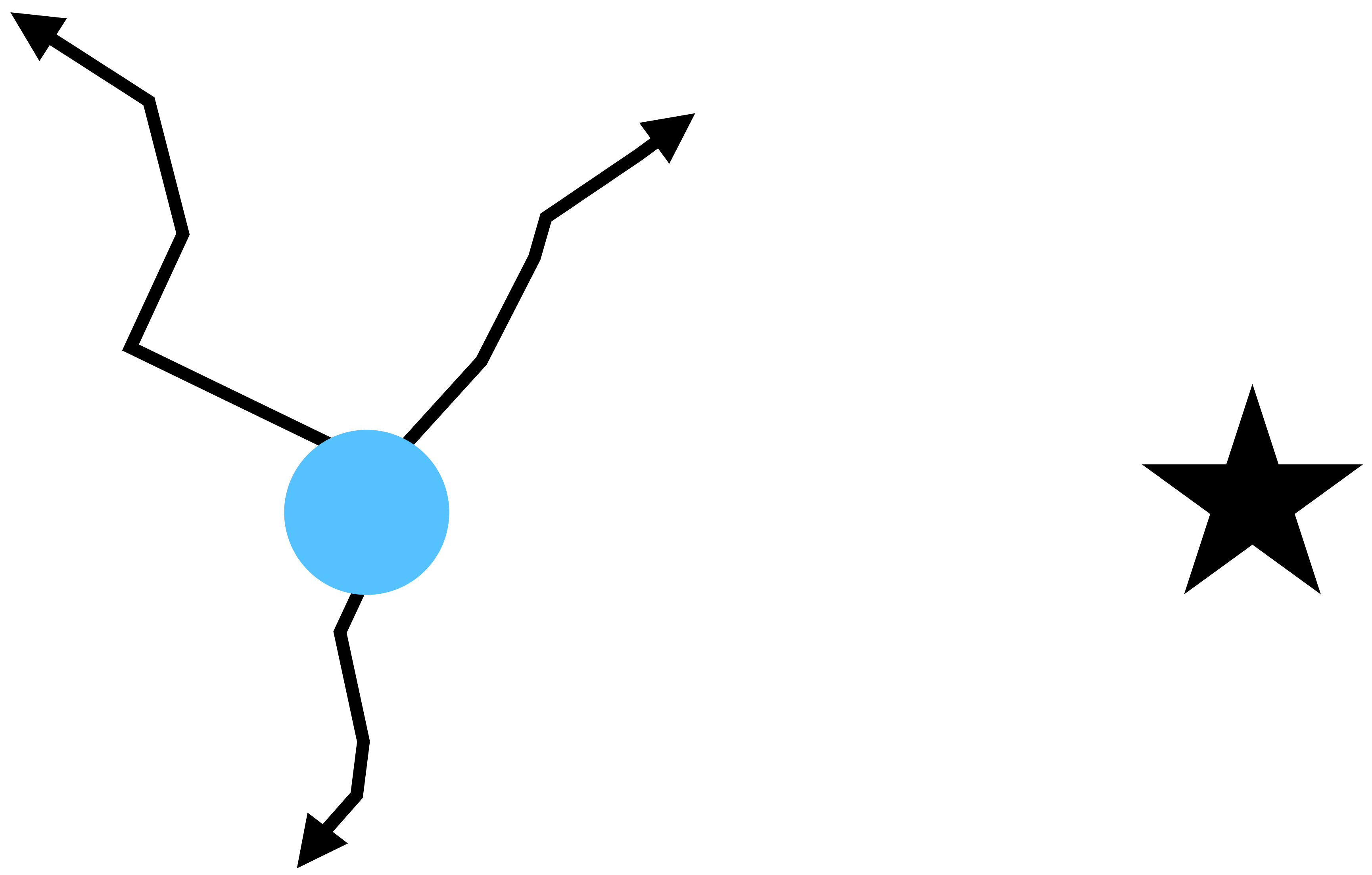
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

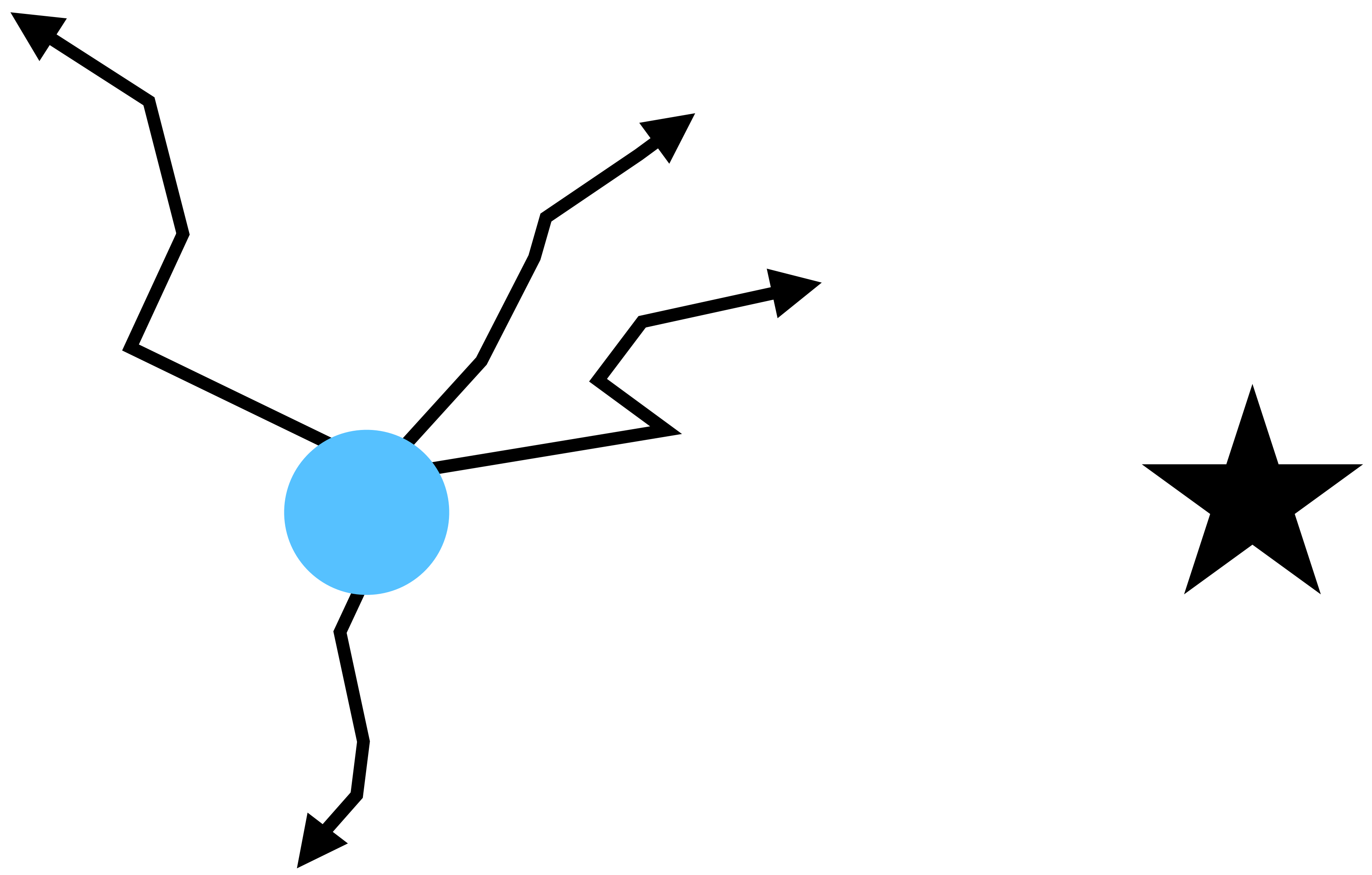
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

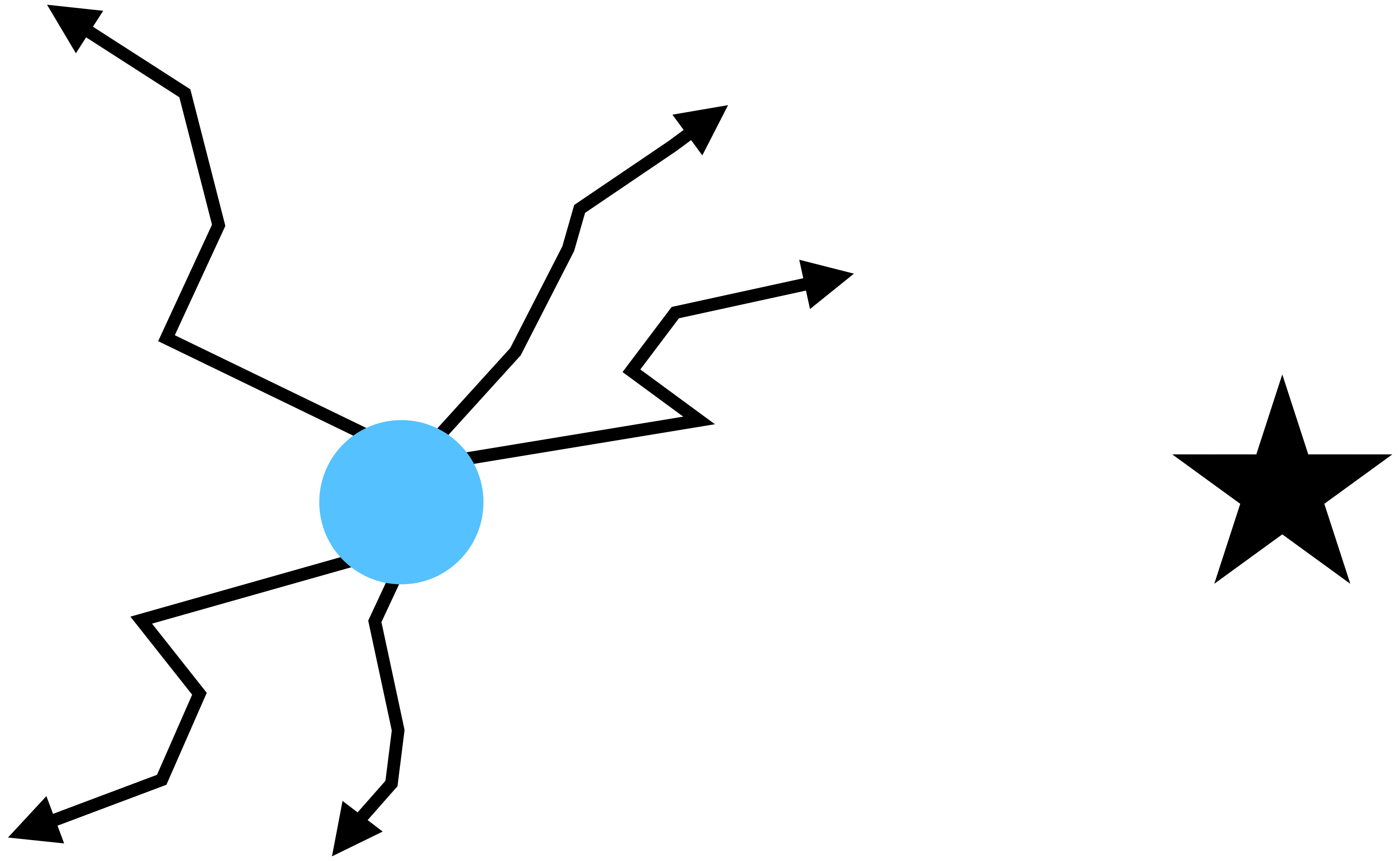
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

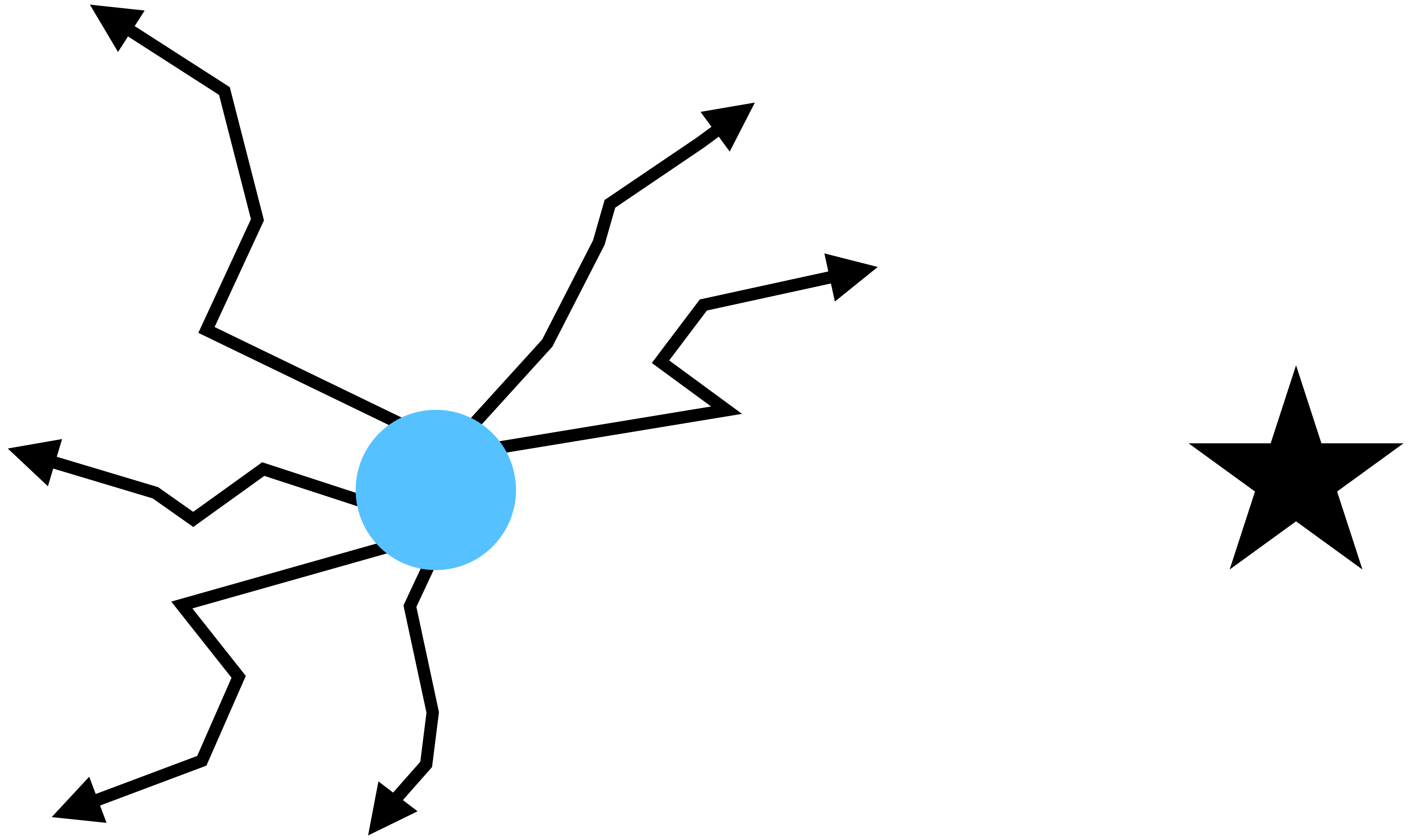
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

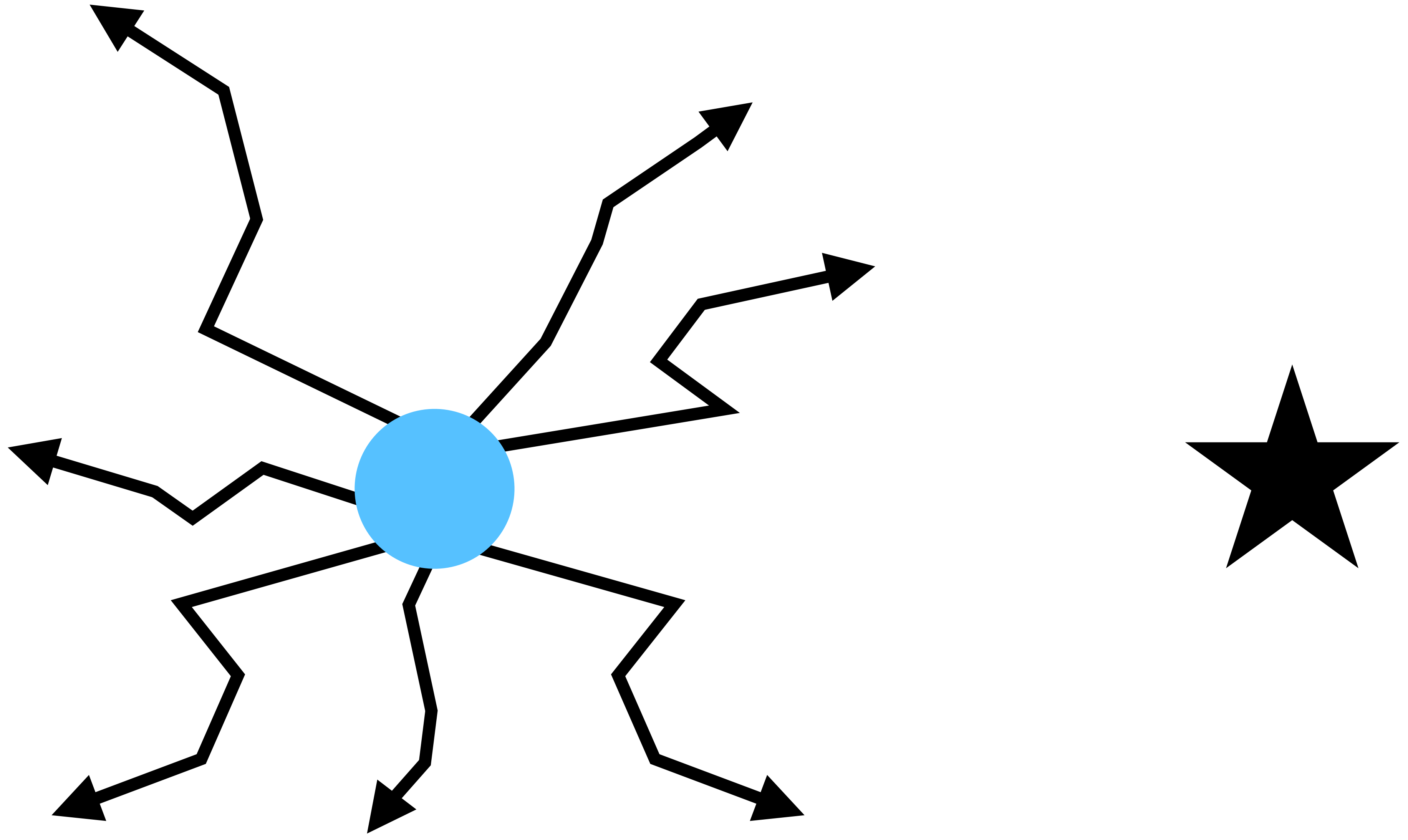
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

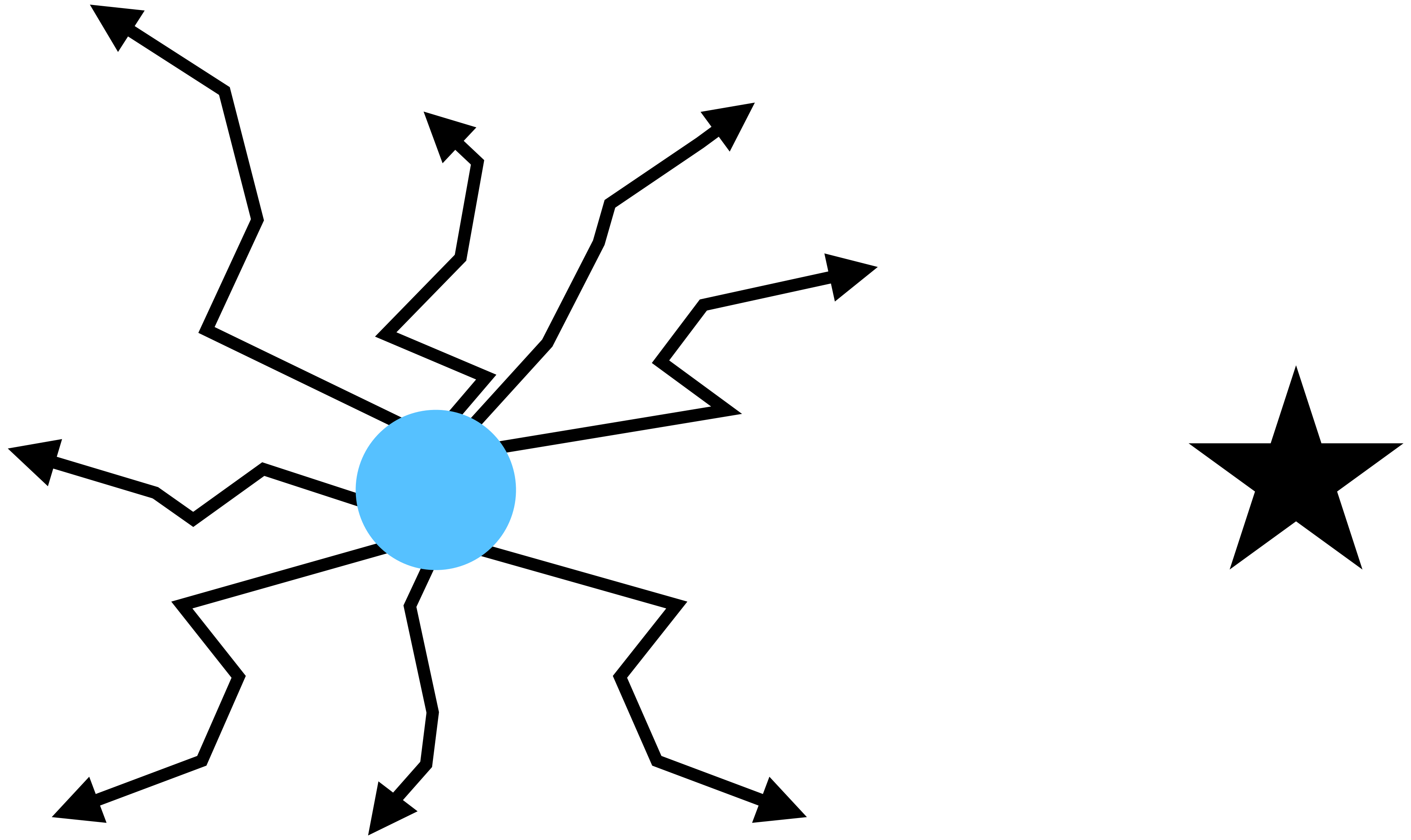
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

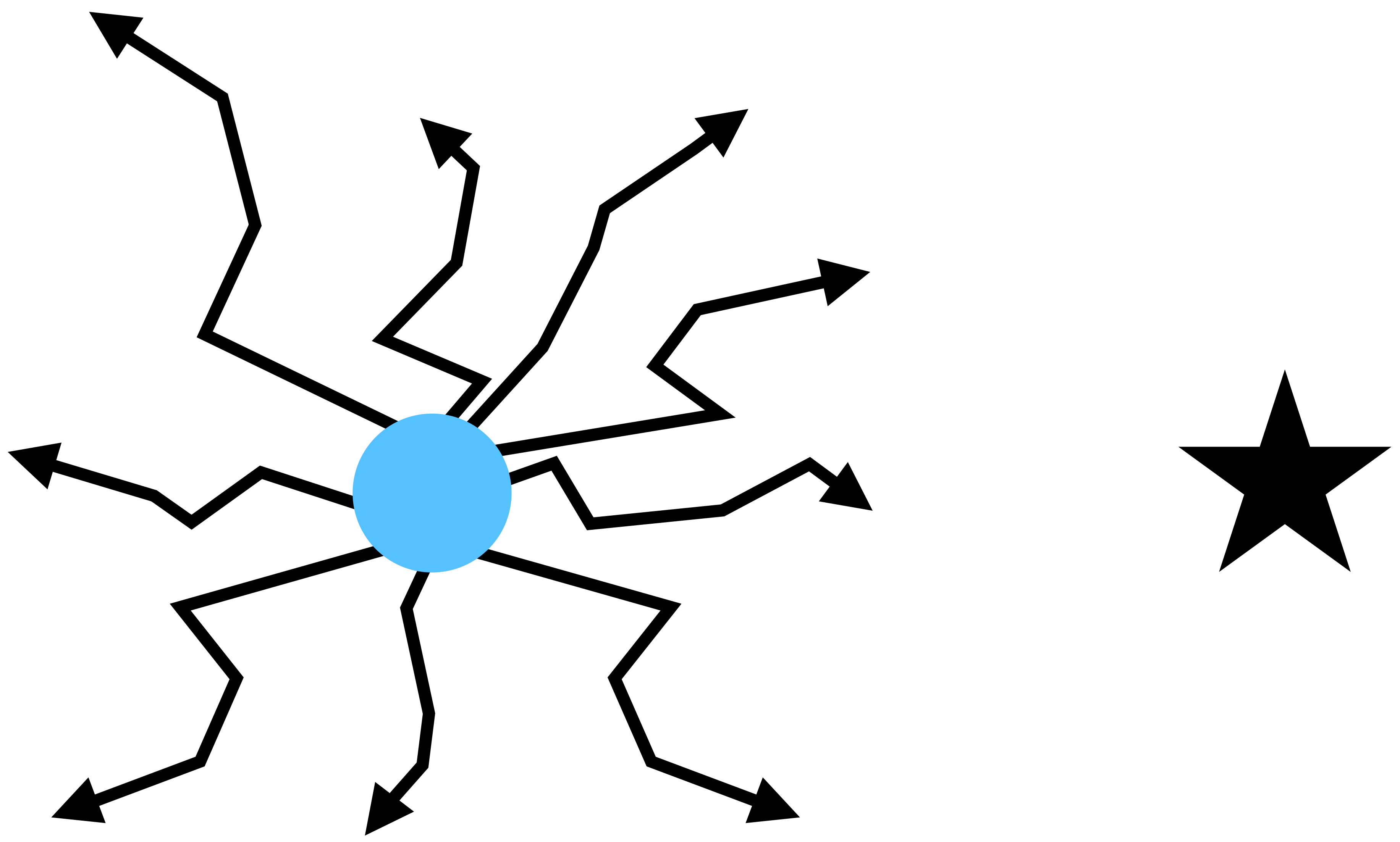
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

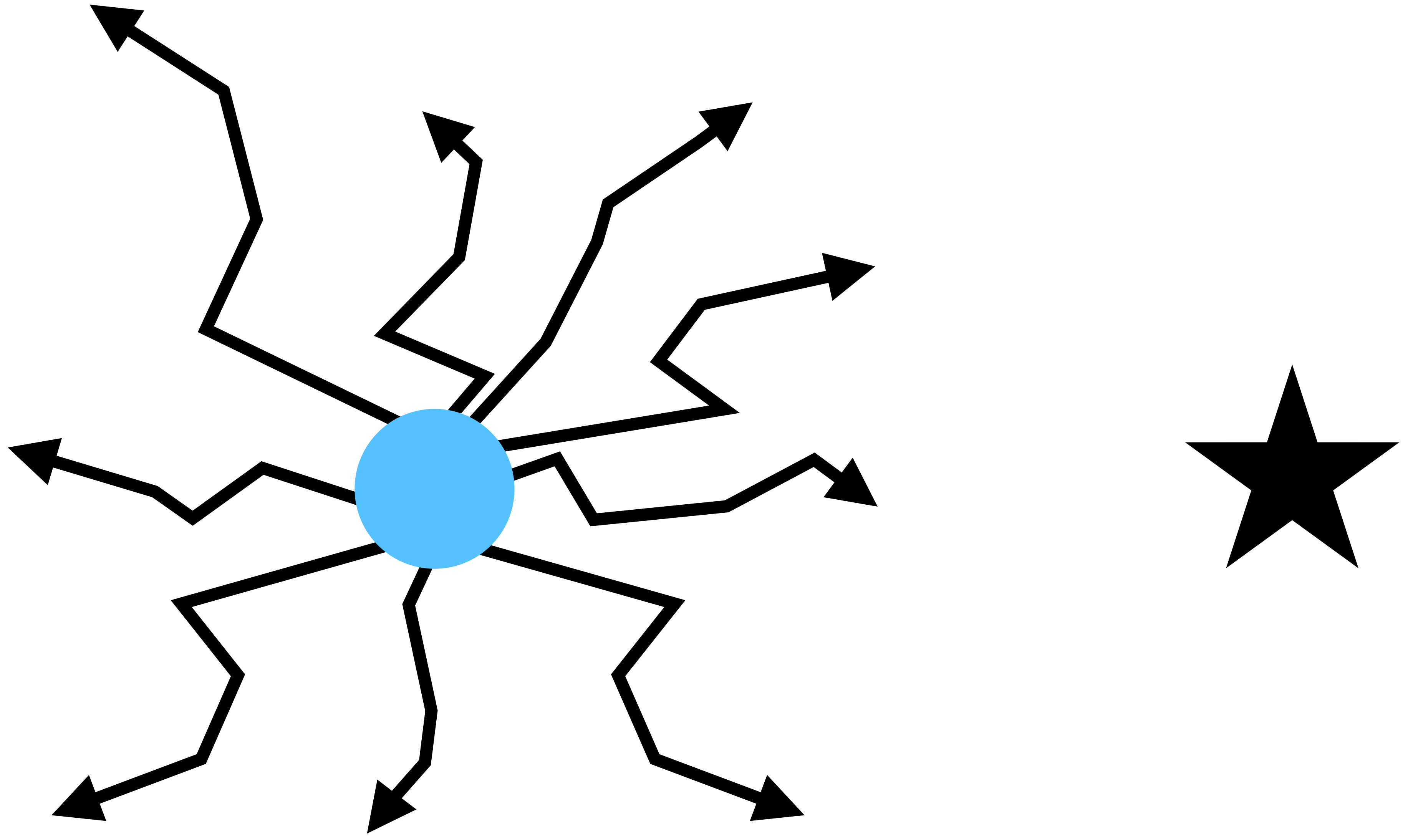
For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

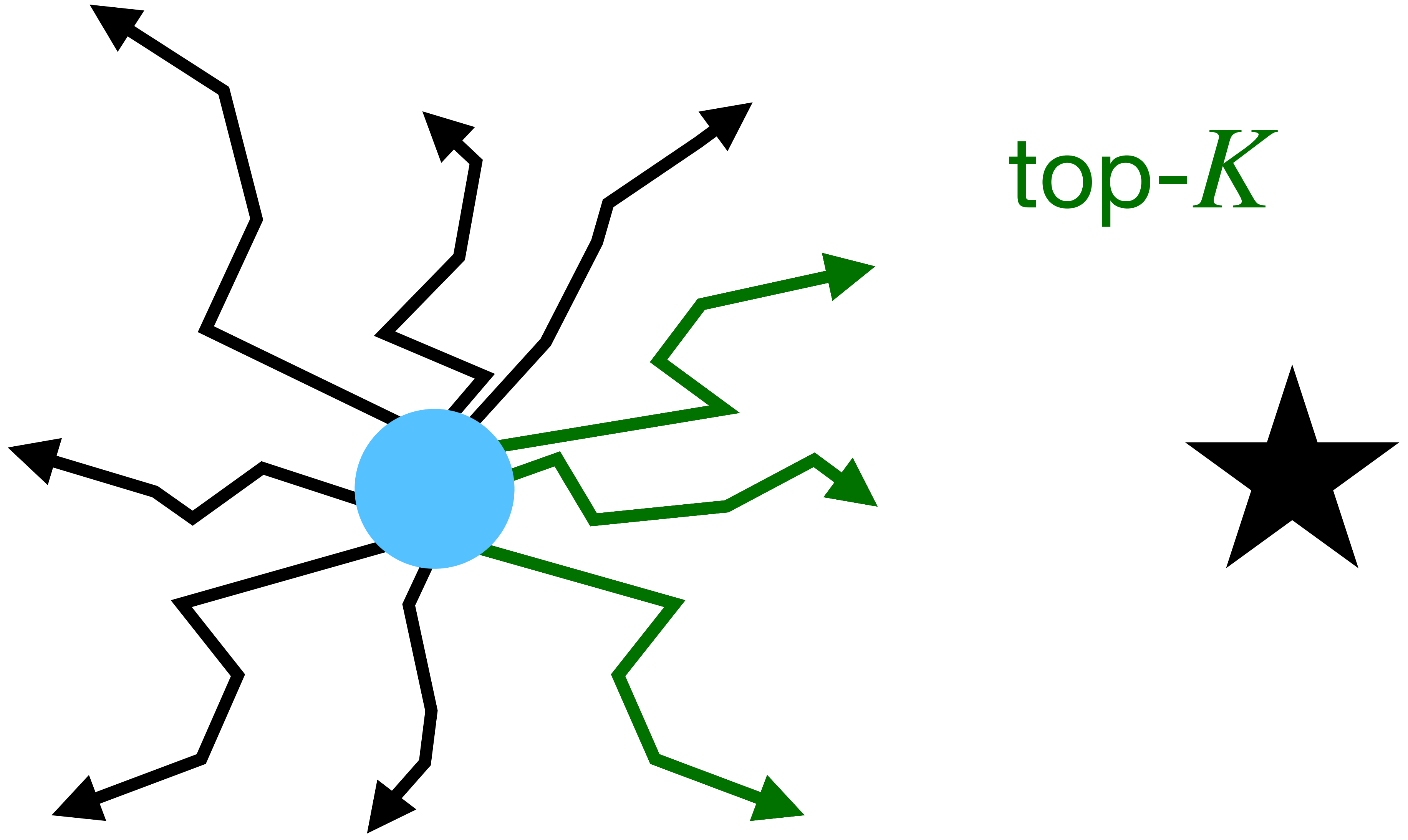
Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

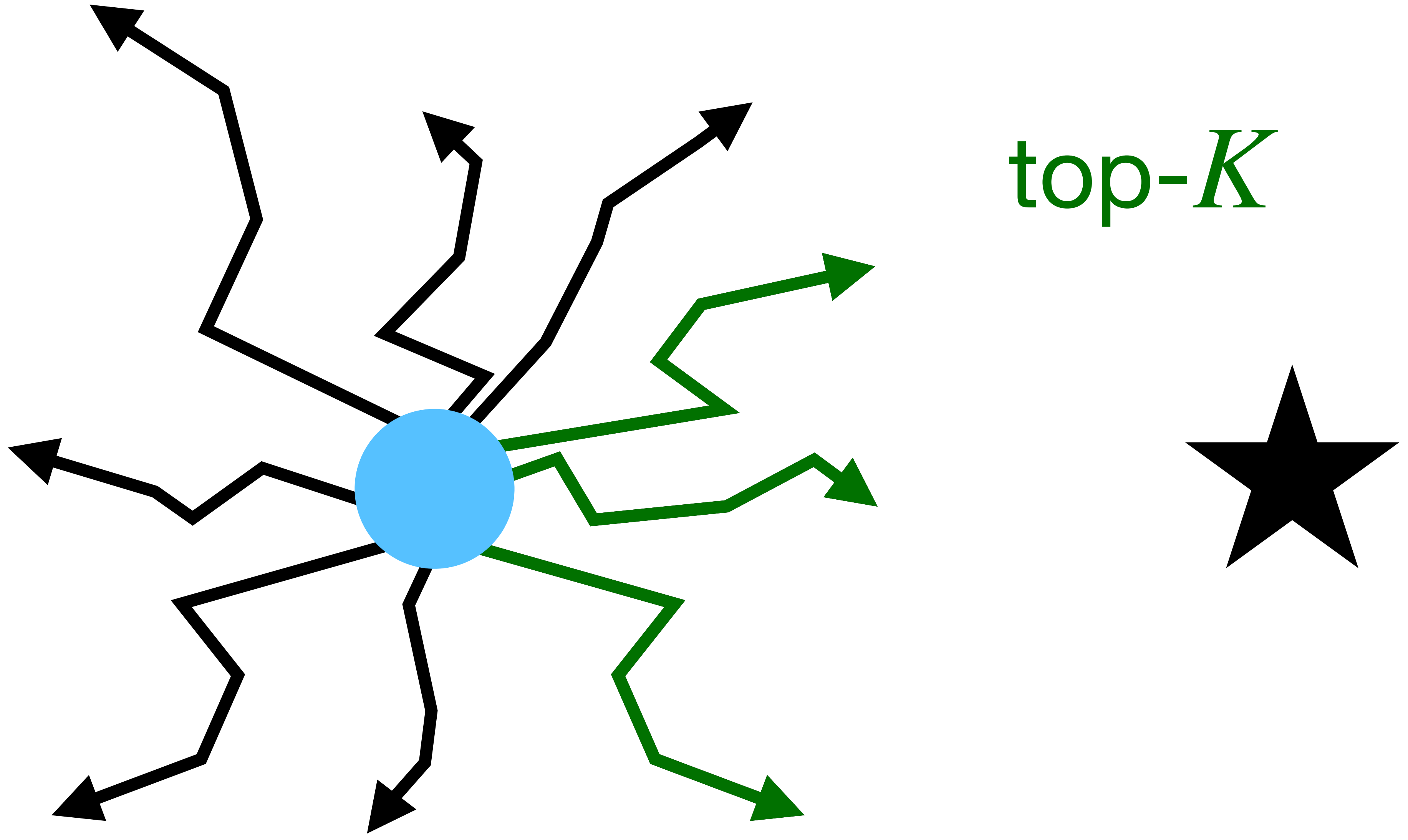
FCAI

fcai.fi

Shooting Methods

Cross-Entropy Method

Iteration 1



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^H$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

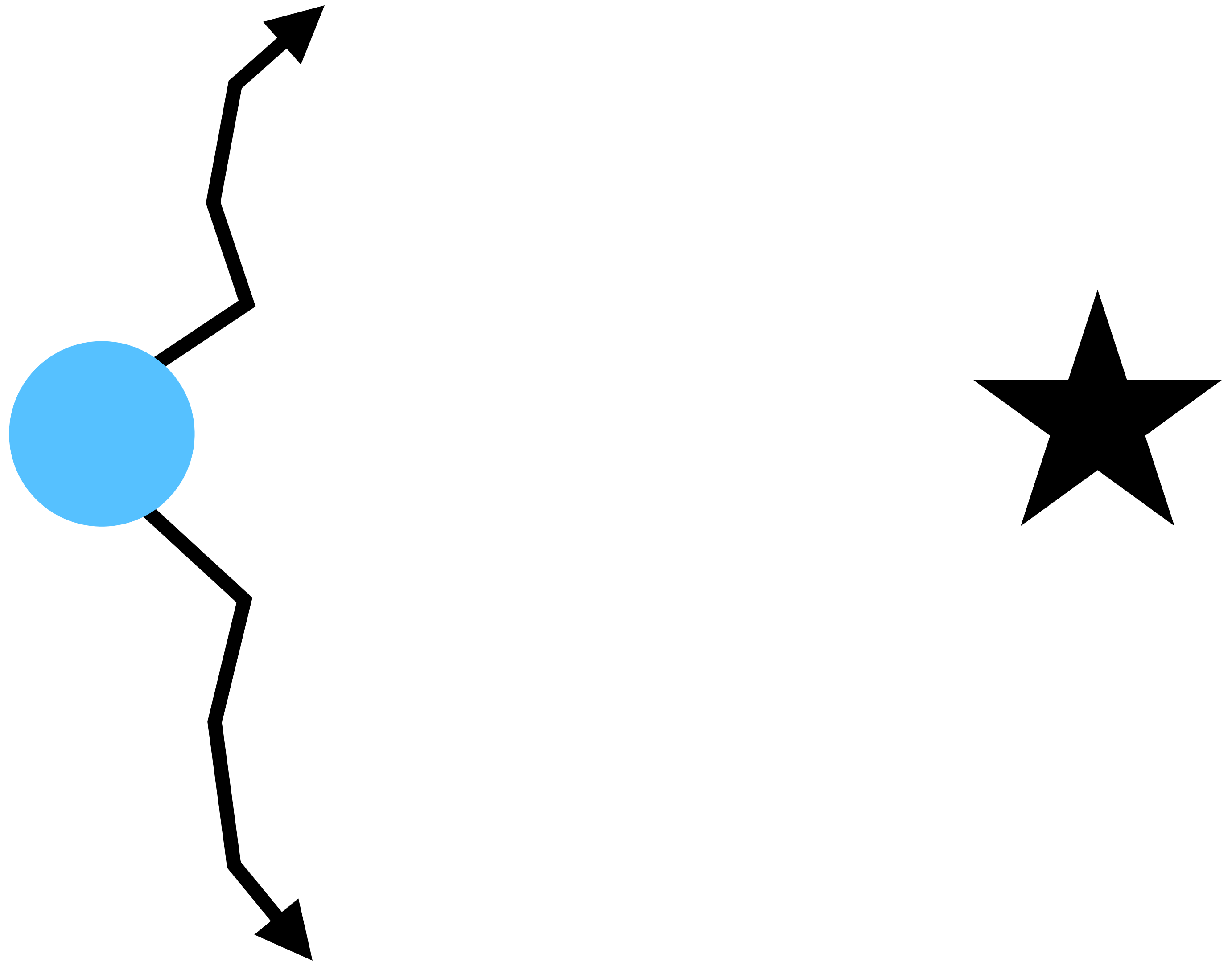
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

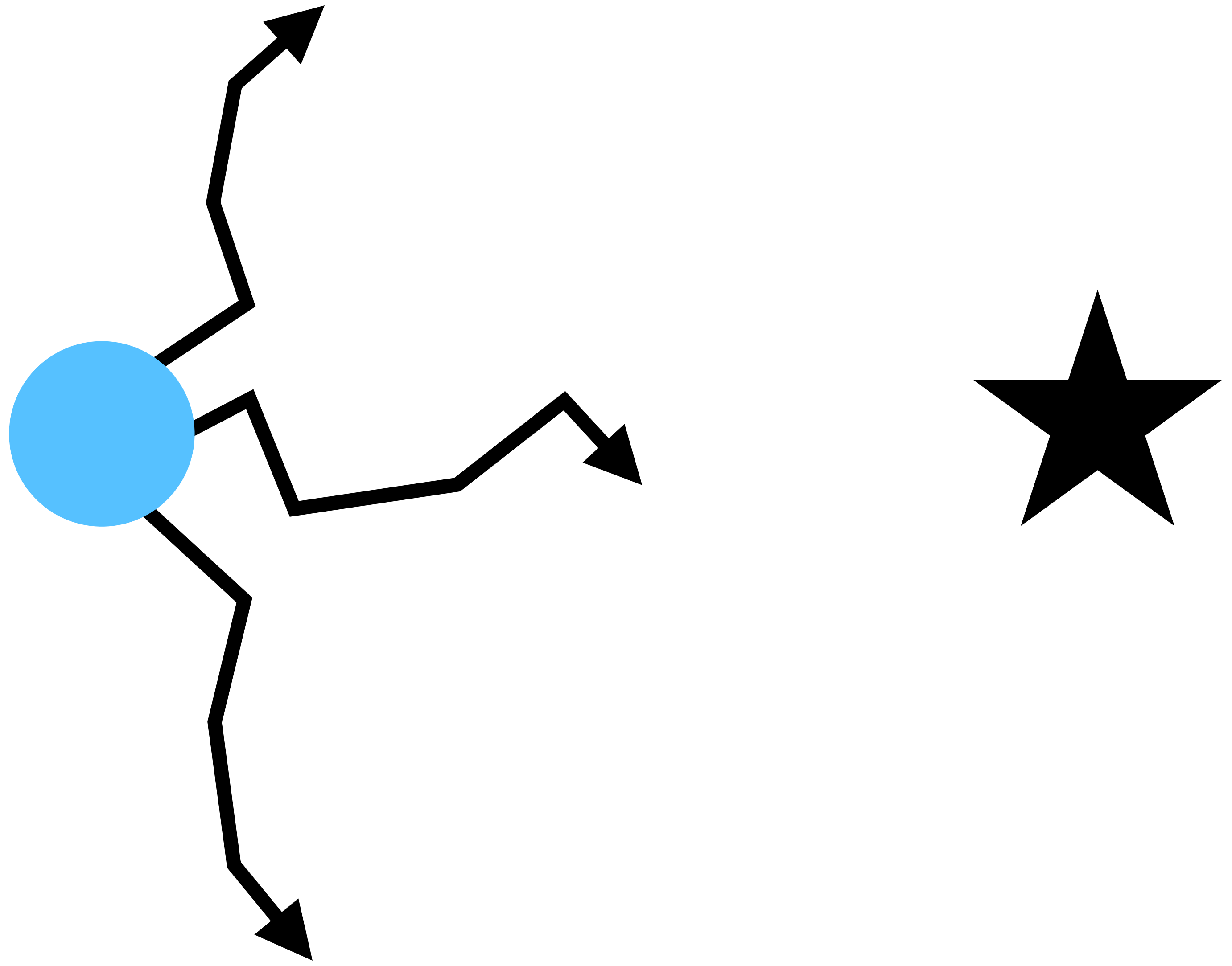
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

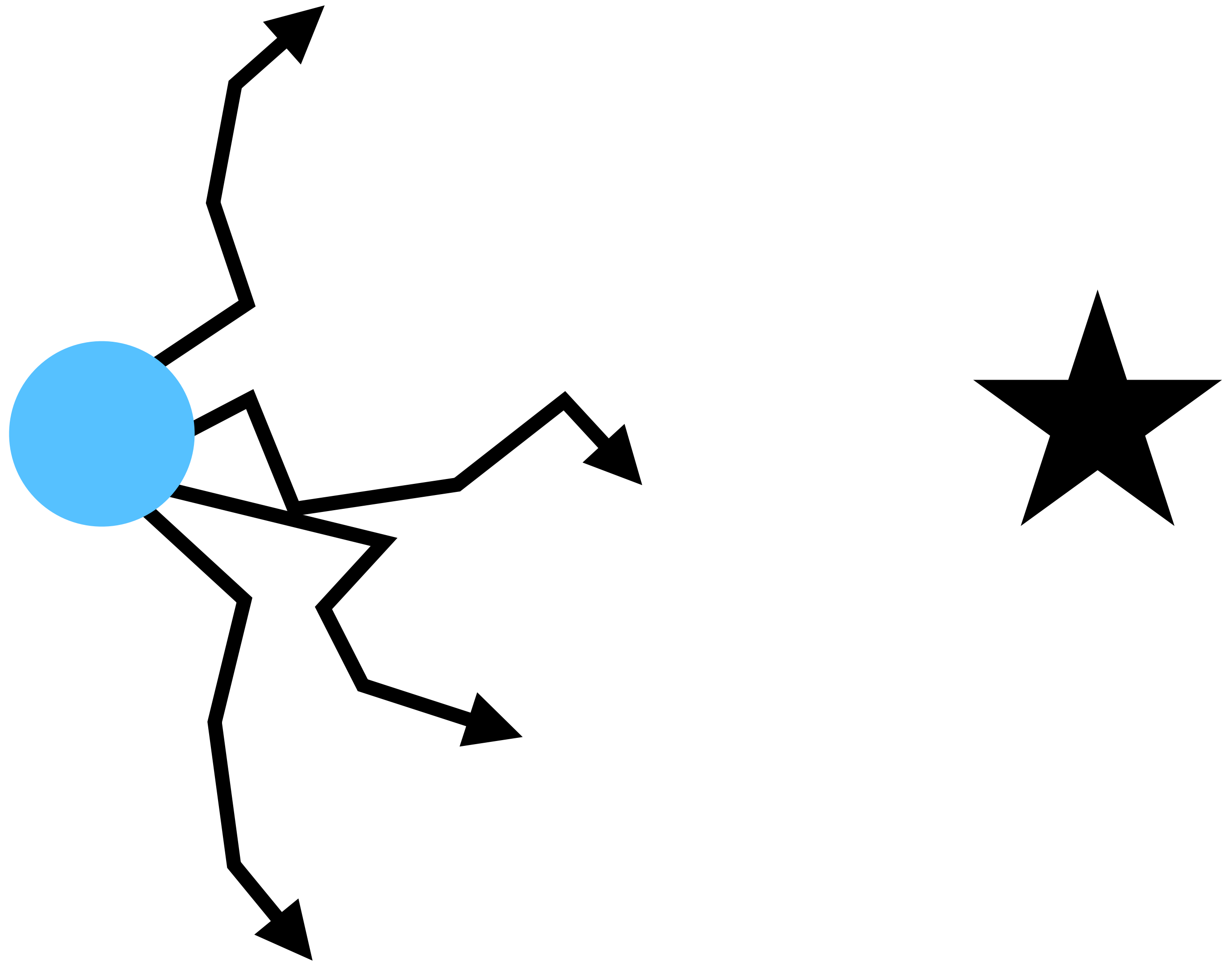
Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

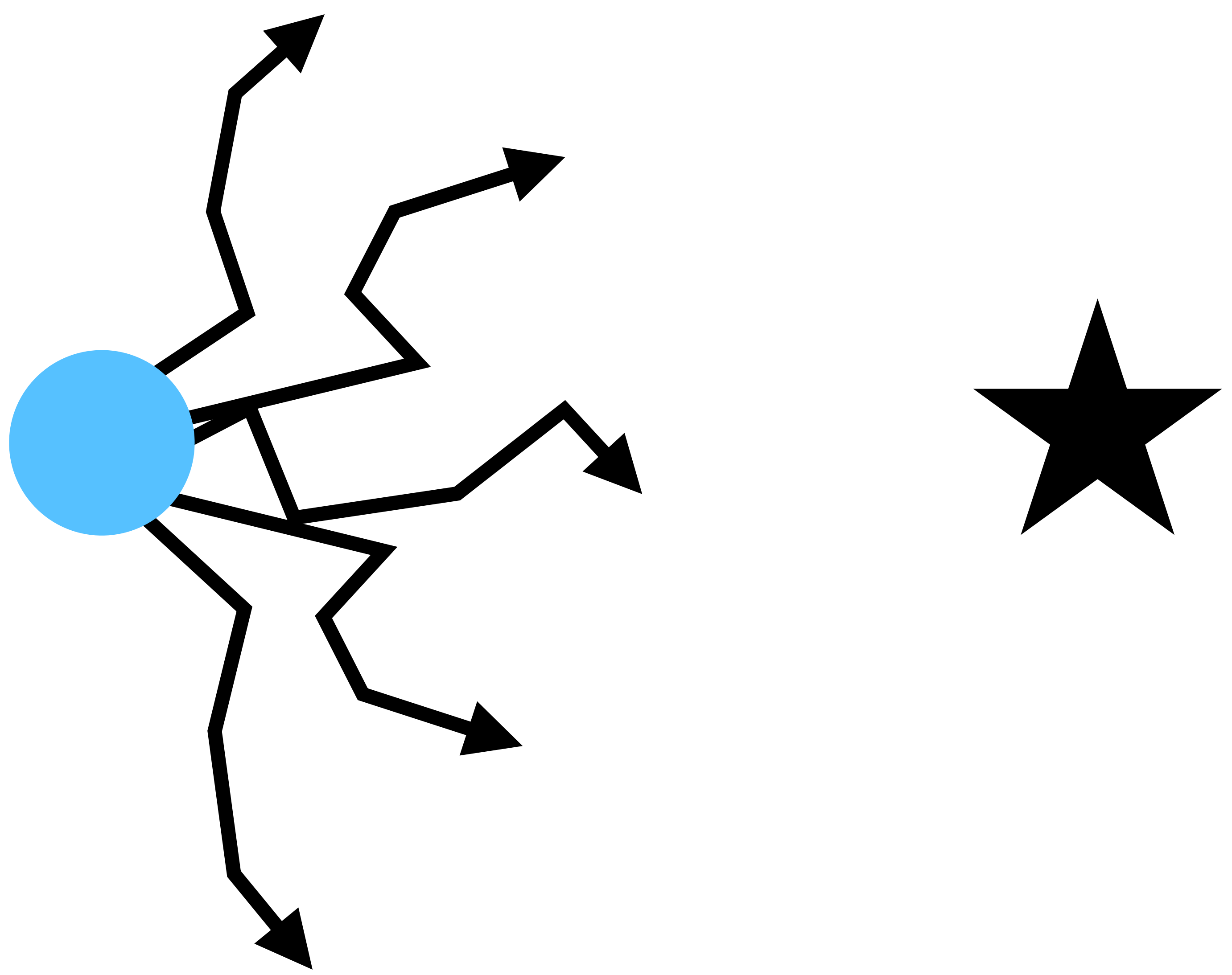
Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

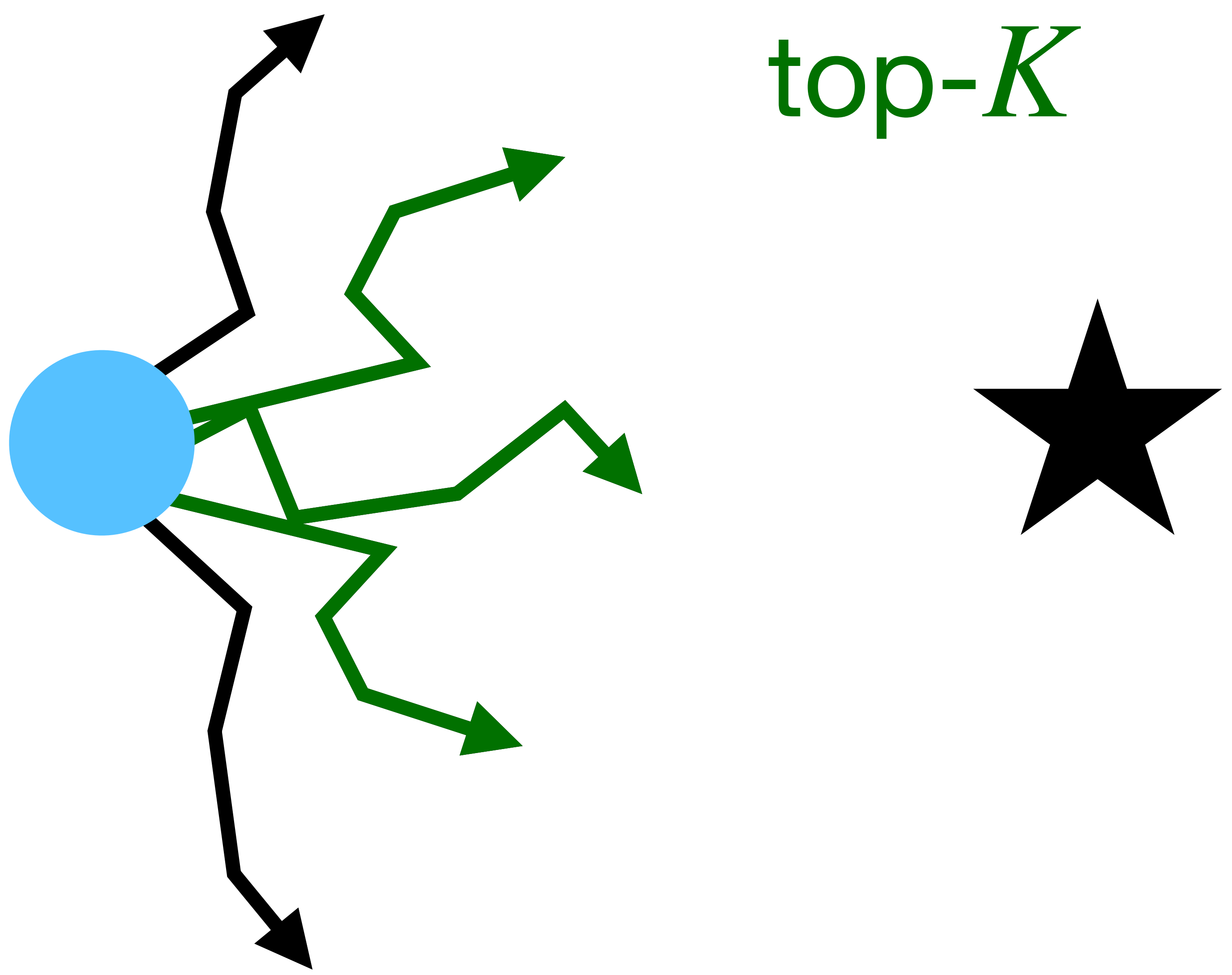
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 2



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3

Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

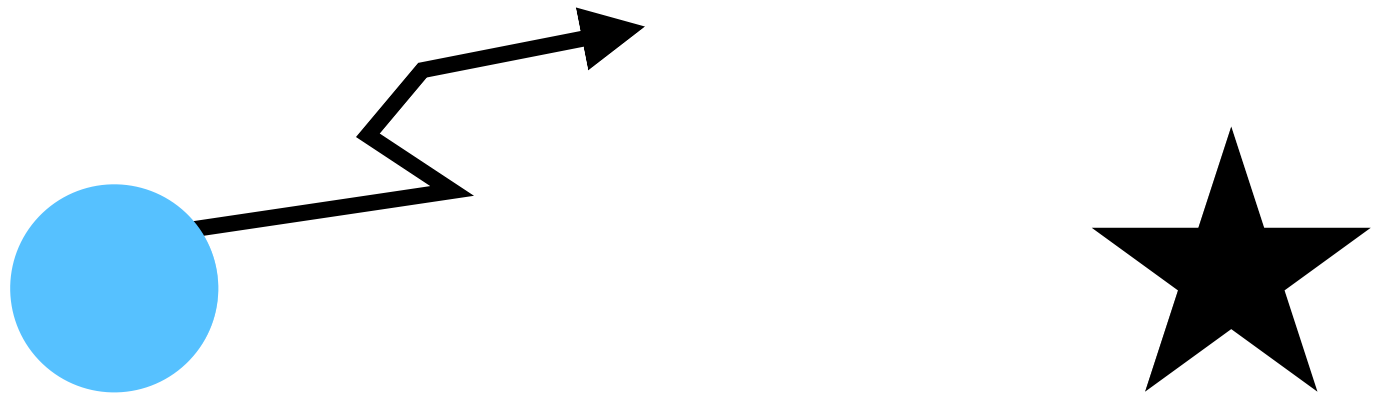
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

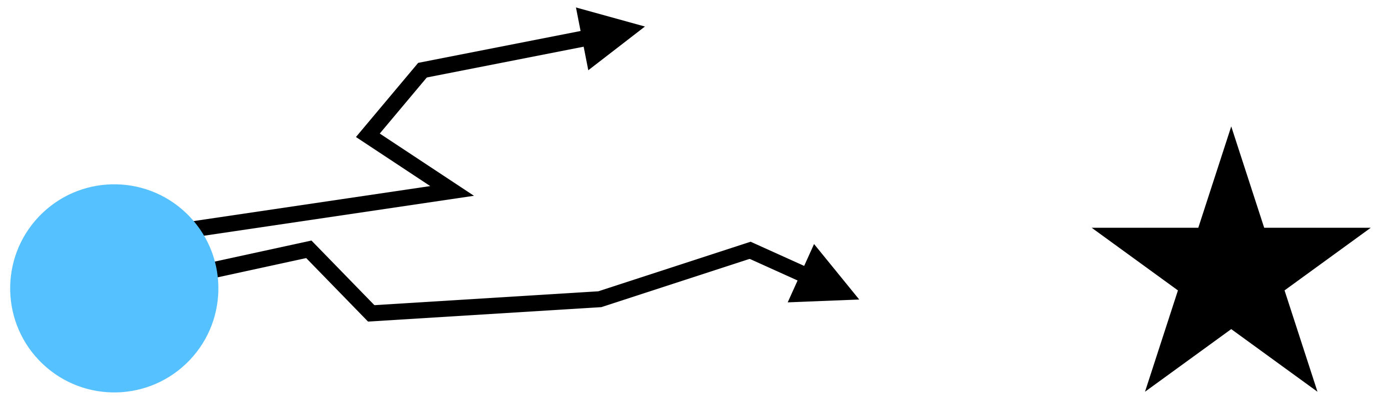
Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

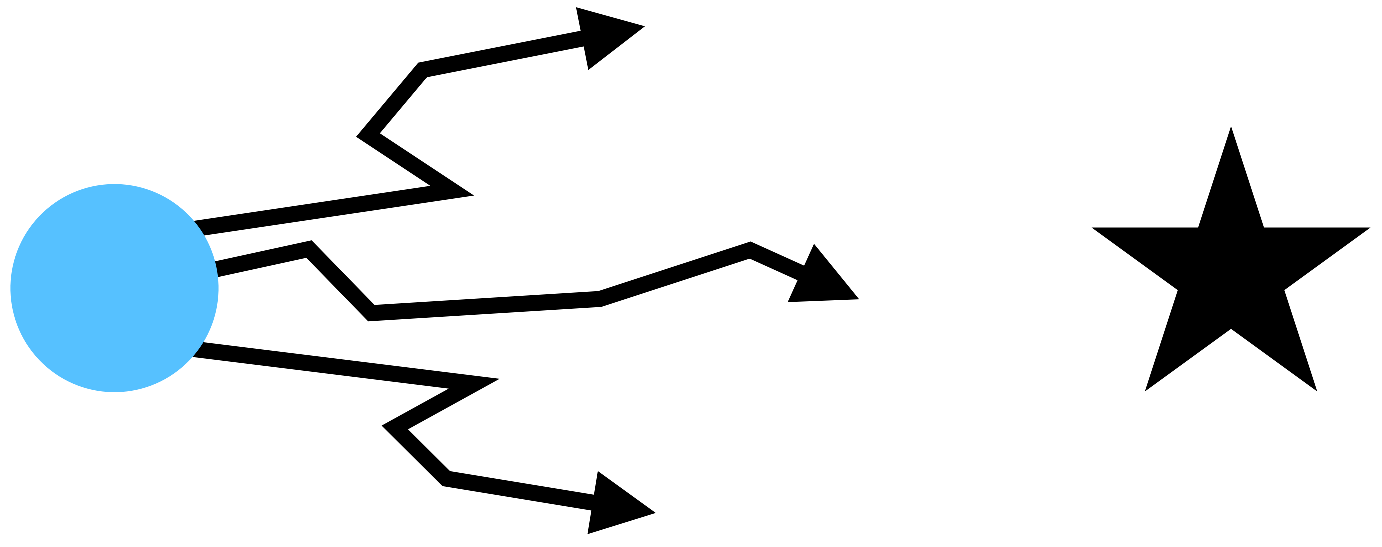
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^H$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

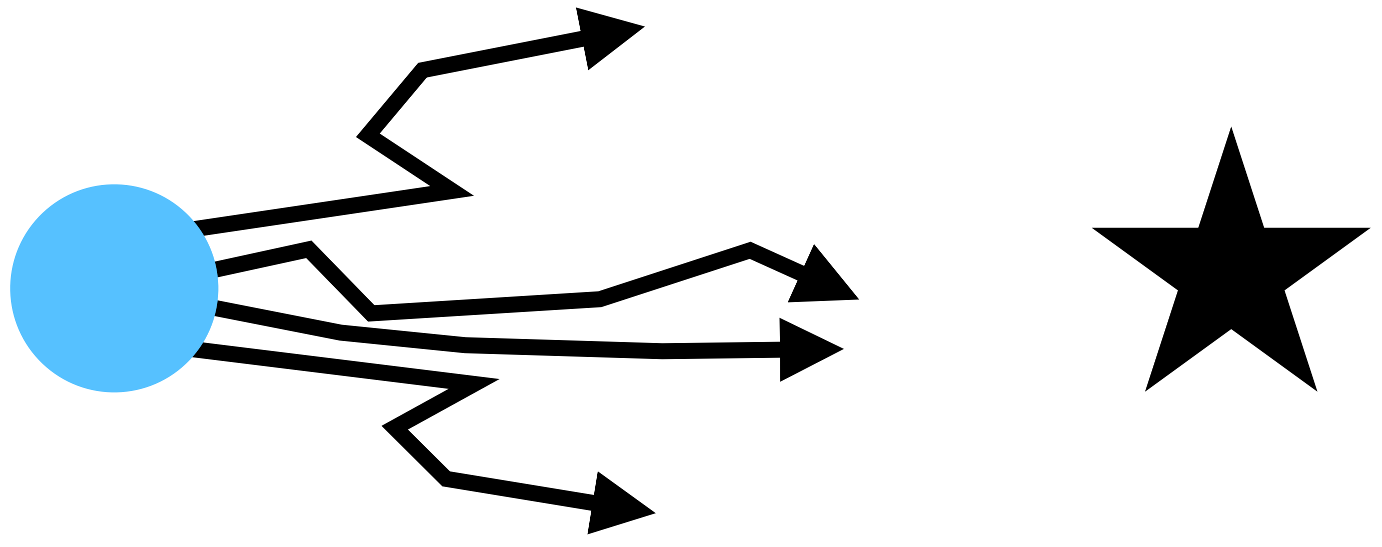
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

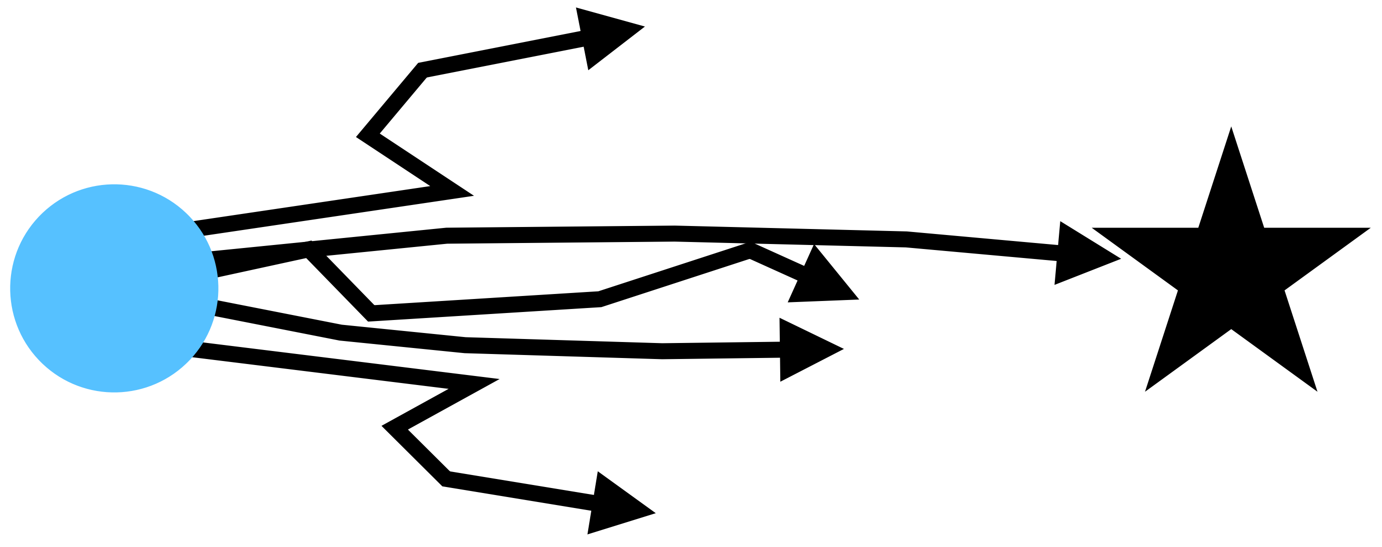
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a_{0:H}^i)$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

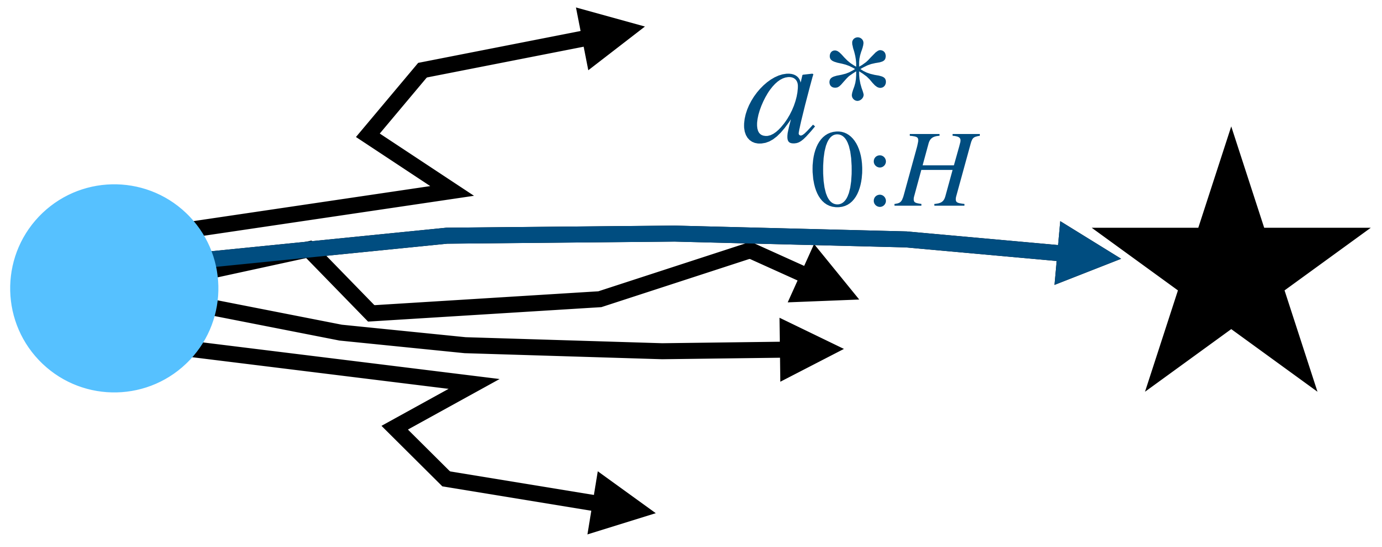
Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a^i_{0:H}\}_{i=1}^{N}$ from sampling distribution

Evaluate objective $J(a^i_{0:H}) = \sum_{t=0}^{H} \gamma^t r(s_t, a^i_t)$ for each sample

Select top K performing samples, i.e. highest value $J(a^i_{0:H})$

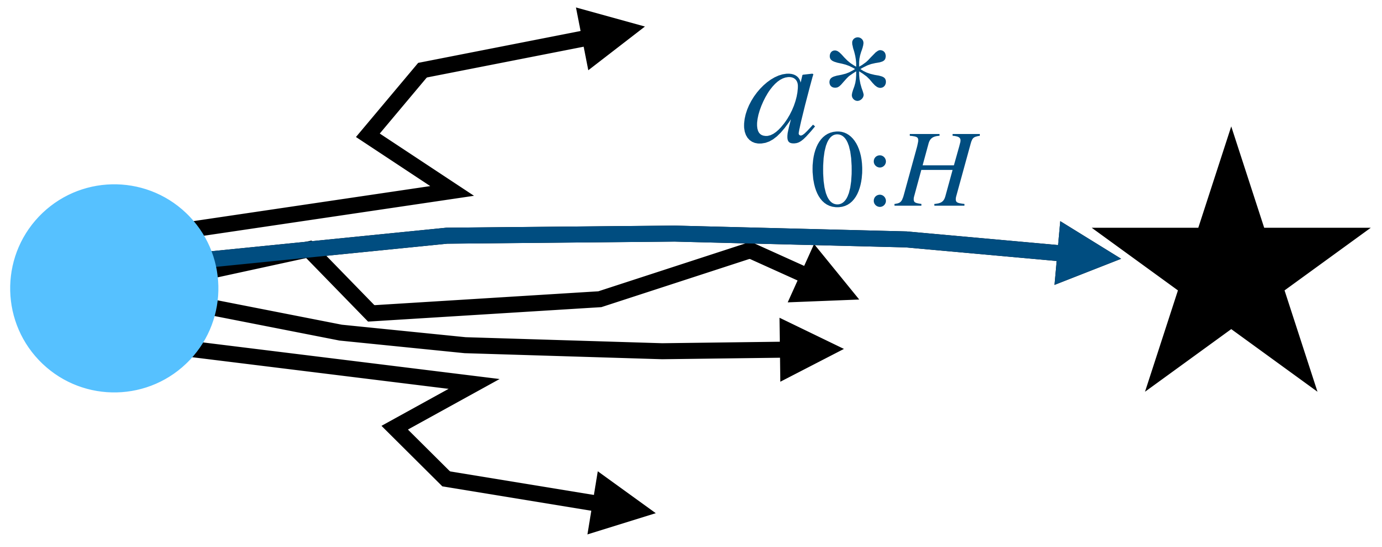
Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

More sample efficient

Shooting Methods

Cross-Entropy Method

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

For each iteration

Sample N action sequences $\{a_{0:H}^i\}_{i=1}^N$ from sampling distribution

Evaluate objective $J(a_{0:H}^i) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t^i)$ for each sample

Select top K performing samples, i.e. highest value $J(a_{0:H}^i)$

Update parameters $\{\mu_t, \sigma_t^2\}_{t=0}^H$ of action dist. using top K samples

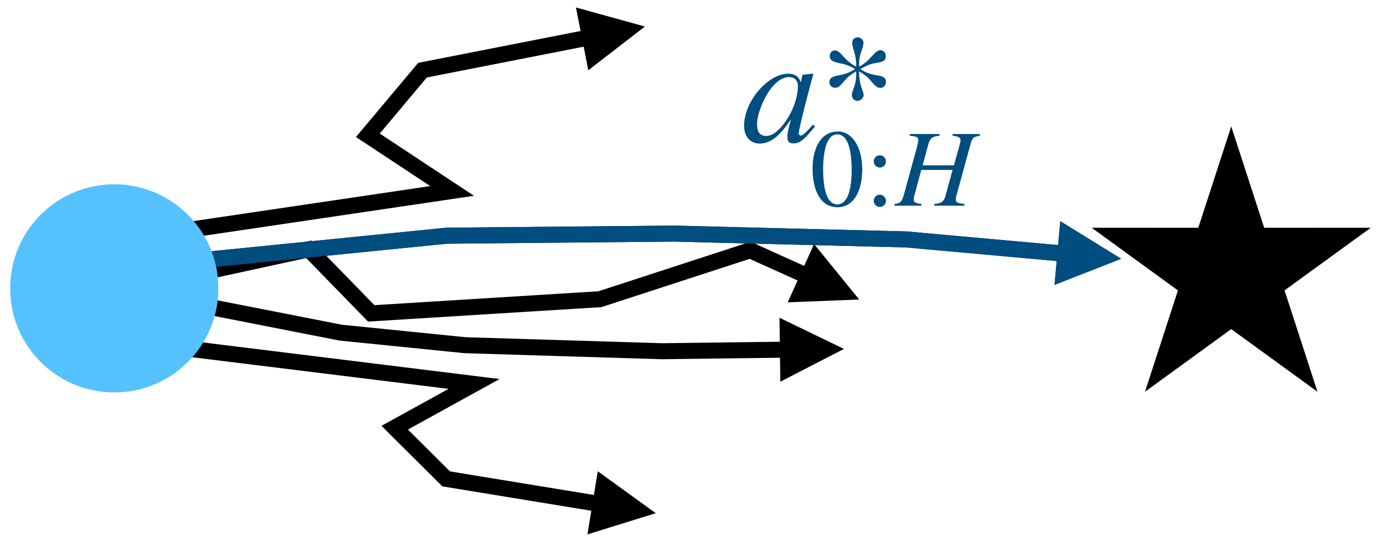
Shooting Methods

Cross-Entropy Method

More sample efficient

Faster convergence

Iteration 3



Initialise action sequence sampling distribution $\{a_t \sim \mathcal{N}(\mu_t, \sigma_t^2)\}_{t=0}^{H}$

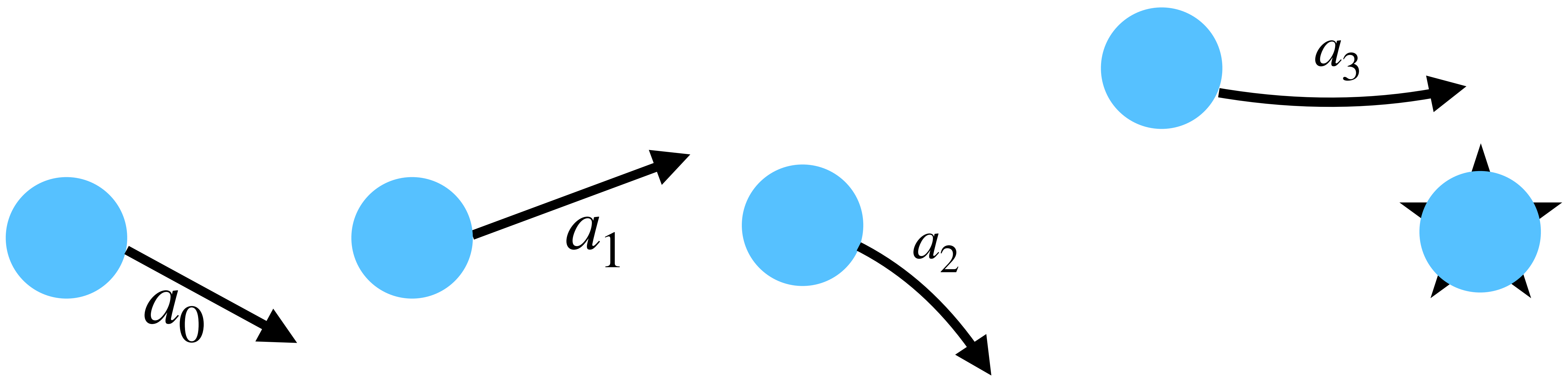
For each iteration

- **Sample** N action sequences $\{a^i_{0:H}\}_{i=1}^{N}$ from sampling distribution
- **Evaluate** objective $J(a^i_{0:H}) = \sum_{t=0}^{H} \gamma^t r(s_t, a^i_t)$ for each sample
- **Select** top K performing samples, i.e. highest value $J(a^i_{0:H})$
- **Update** parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$ of action dist. using top K samples

Collocation methods

Illustration

$$J(a_{0:H}, s_{0:H}) = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t. } \|s_{t+1} - f(s_t, a_t)\| = 0$$

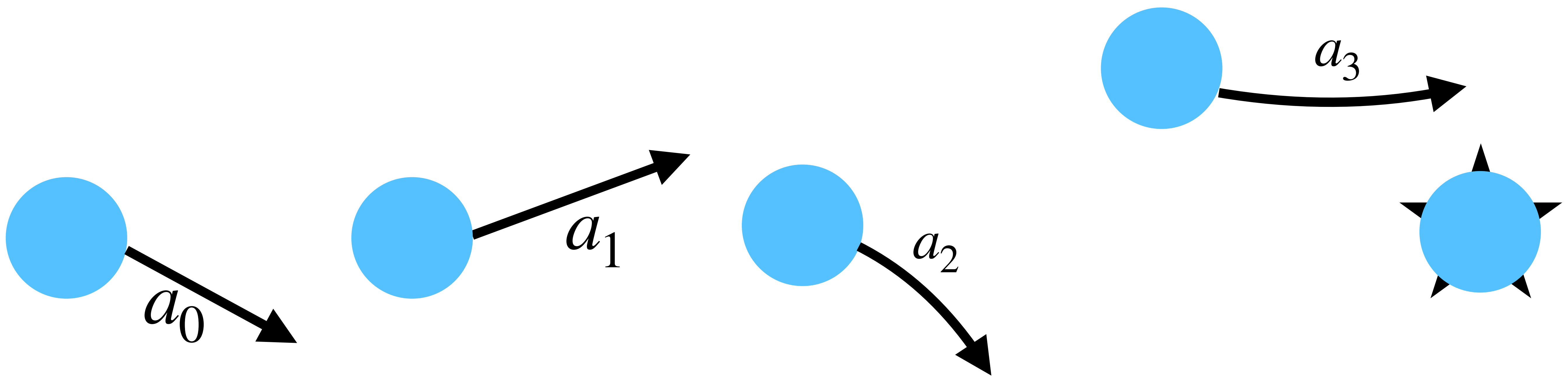


Collocation methods

Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t. } \|s_{t+1} - f(s_t, a_t)\| = 0$$

Optimising states and actions

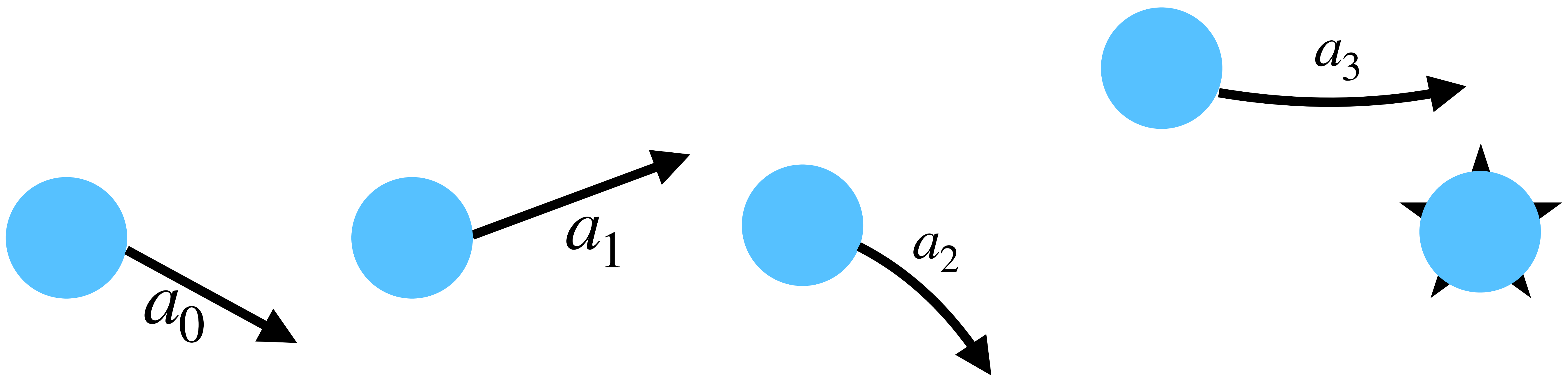


Collocation methods

Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \; \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions



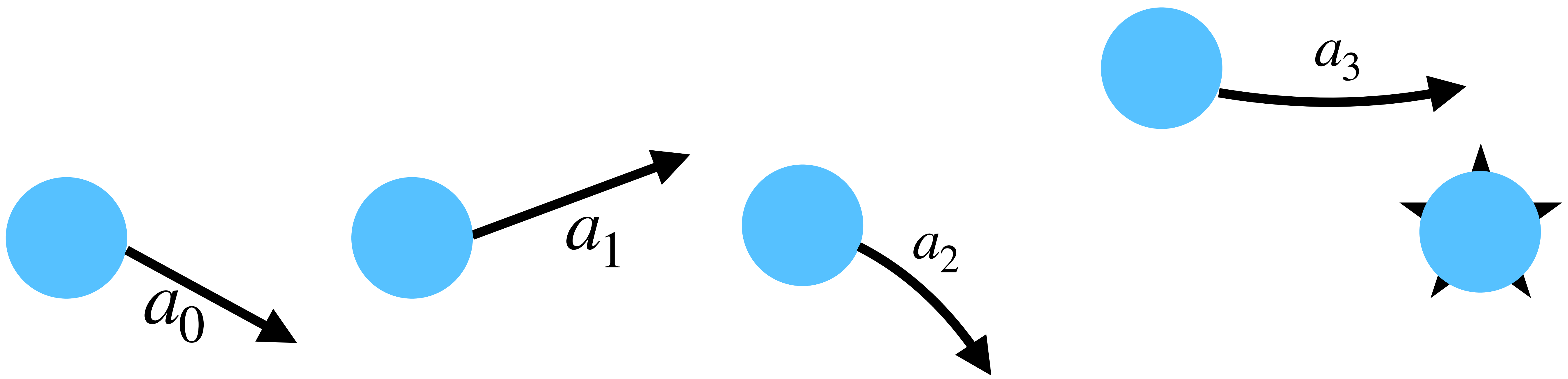
Collocation methods

Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \quad \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint



Collocation methods

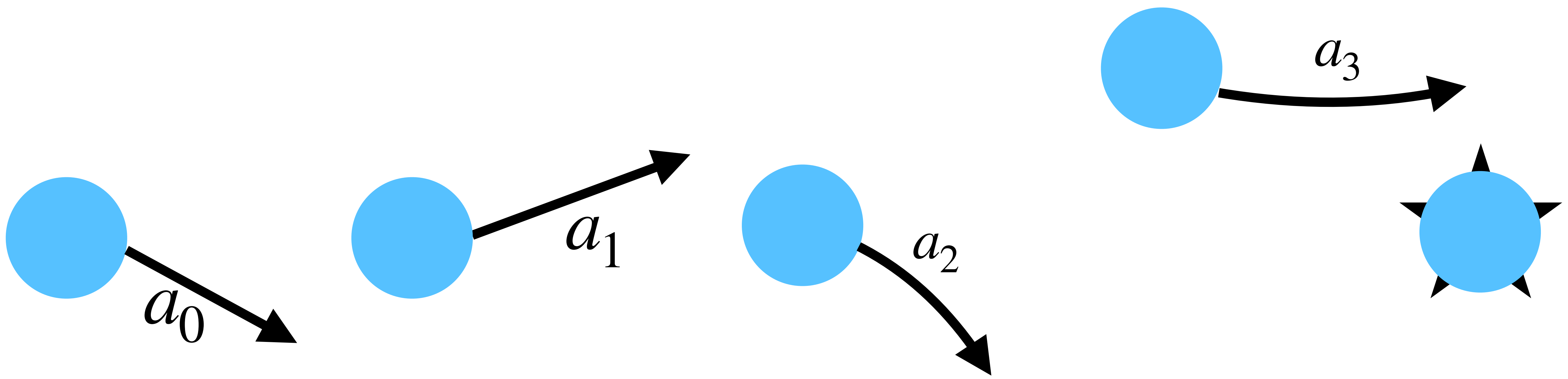
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Collocation methods

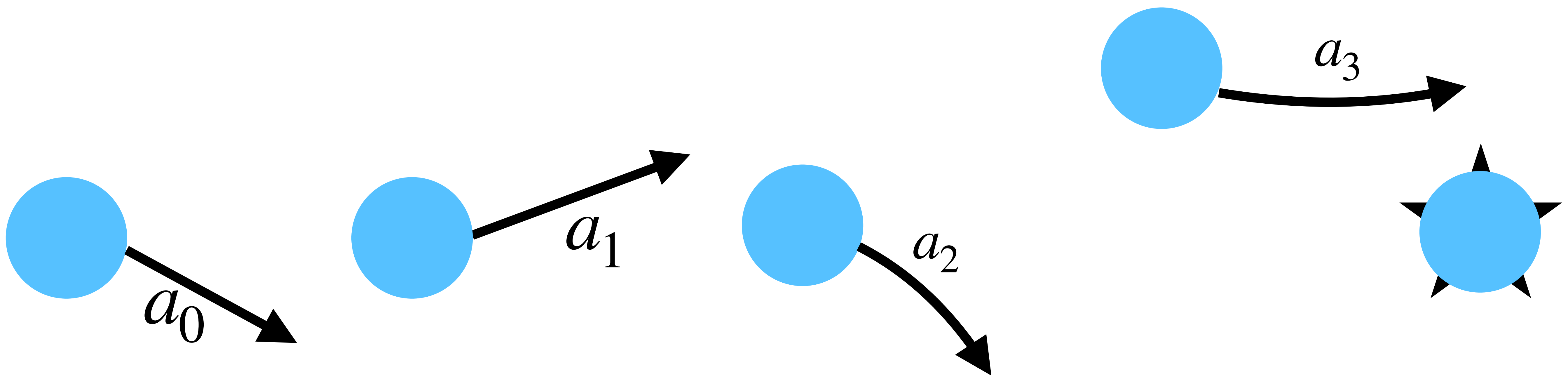
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \quad \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

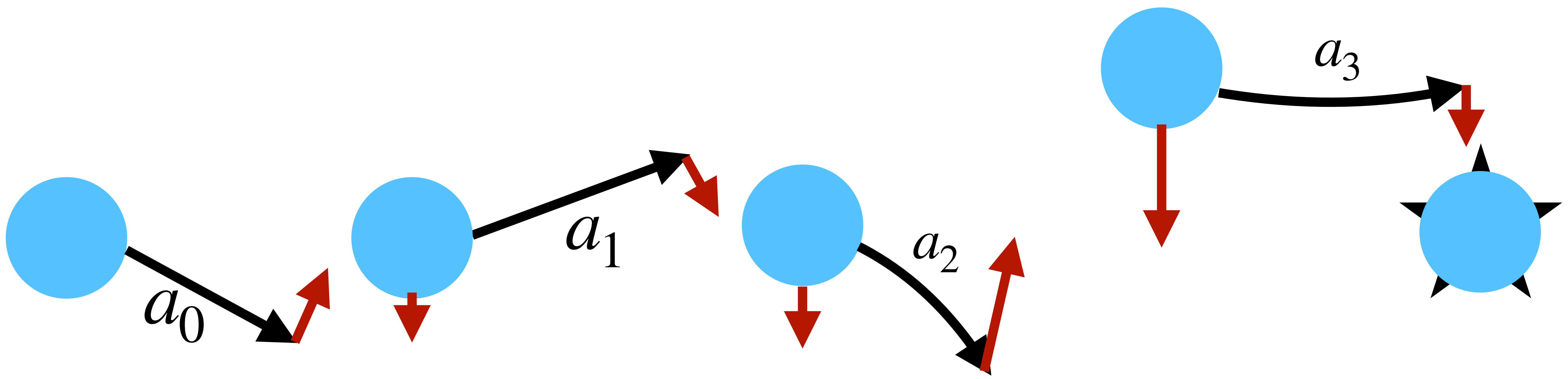
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

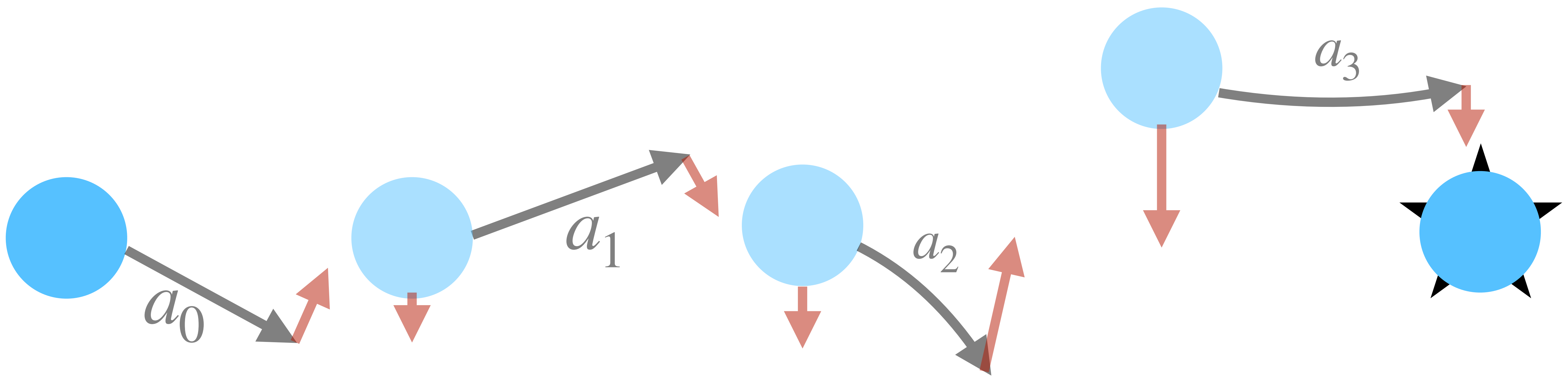
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \ \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

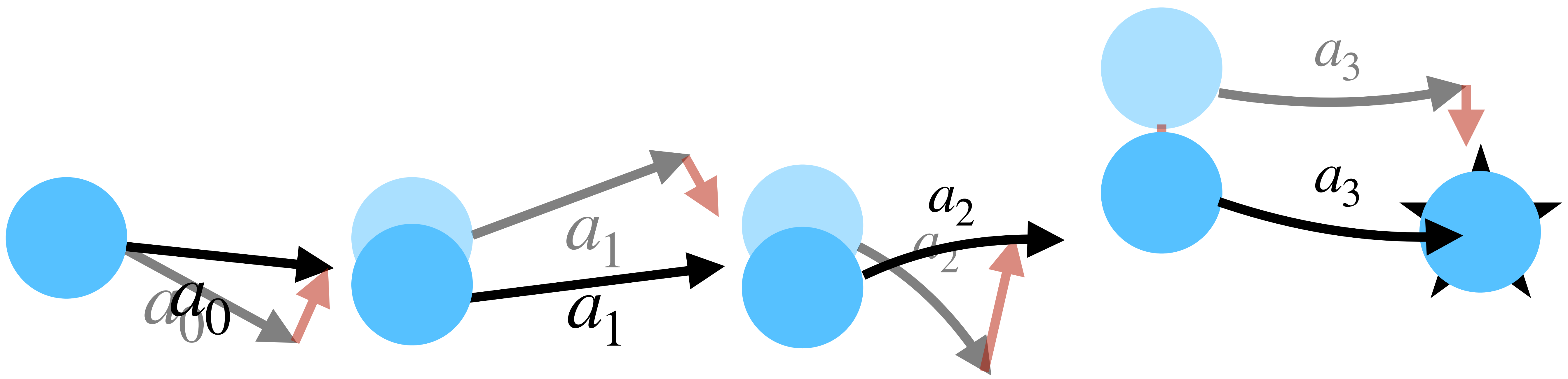
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

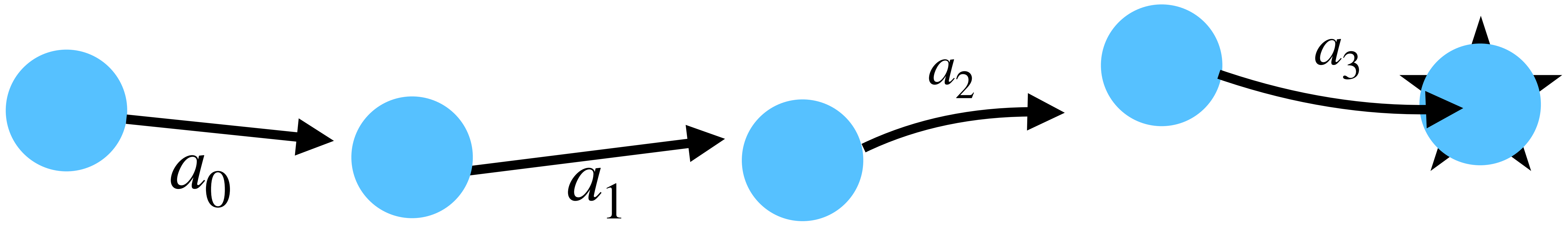
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

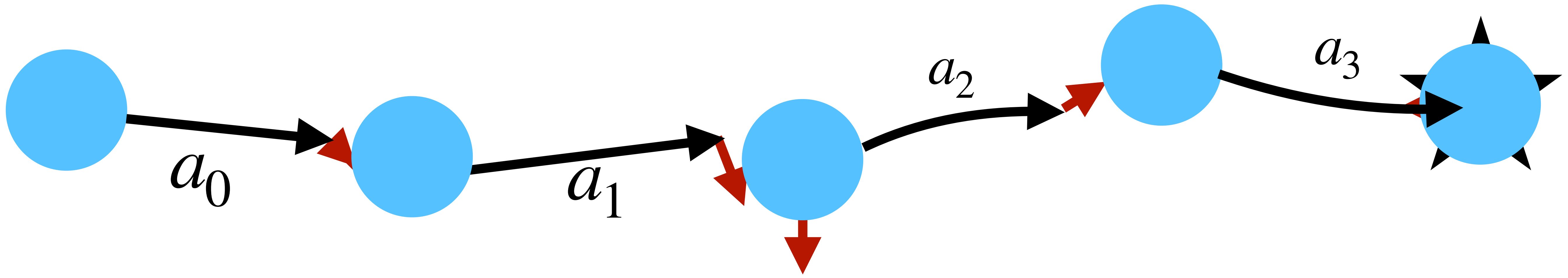
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \ \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

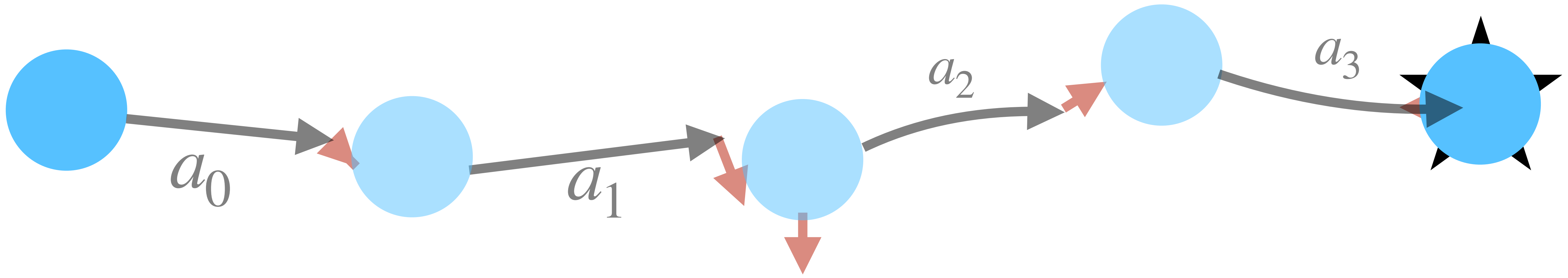
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint not satisfied!

Collocation methods

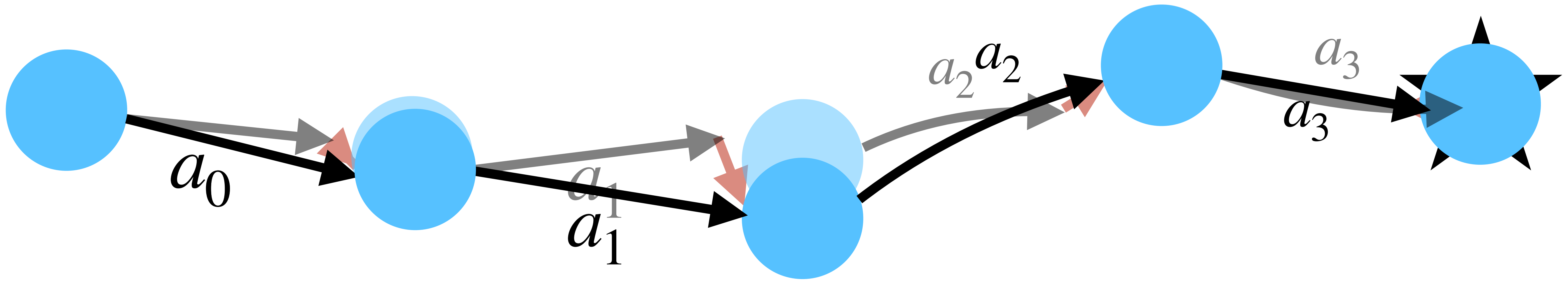
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \quad \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Collocation methods

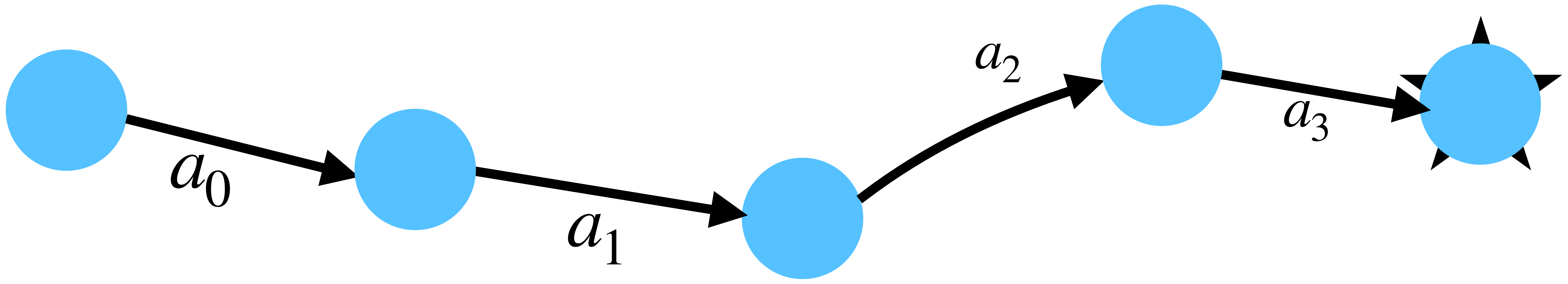
Illustration

$$\boxed{J(a_{0:H}, s_{0:H})} = \sum_{t=0}^{H} \gamma^t r(s_t, a_t) \quad \text{s.t.} \boxed{\|s_{t+1} - f(s_t, a_t)\| = 0}$$

Optimising states and actions

Dynamics constraint

No dynamics rollout



Dynamics constraint satisfied!

Finite Horizon Planning has Limitations

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)$$

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$$

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \boxed{\gamma^H Q_\theta(s_H, a_H)}$$

Approximate infinite horizon return using learned Q-function

FCAI

fcai.fi

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{\infty} \gamma^t Q(s_t, a_t) \approx \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \boxed{\gamma^H Q_\theta(s_H, a_H)}$$

Approximate infinite horizon return using learned Q-function

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{\infty} \gamma^t Q(s_t, a_t) \approx \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \boxed{\gamma^H Q_\theta(s_H, a_H)}$$

Approximate infinite horizon return using learned Q-function

Learned Q-function is common in model-free RL

Finite Horizon Planning has Limitations

$$\sum_{t=0}^{\infty} \gamma^t Q(s_t, a_t) \approx \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \boxed{\gamma^H Q_\theta(s_H, a_H)}$$

Approximate infinite horizon return using learned Q-function

Learned Q-function is common in model-free RL

Best of both worlds!

Trajectory optimisation methods are open loop.

Trajectory optimisation methods are open loop.
We can do better.

Decision-time Planning

Model Predictive Control (MPC)

Decision-time Planning

Model Predictive Control (MPC)

For each environment step

FCAI

fcai.fi

Decision-time Planning

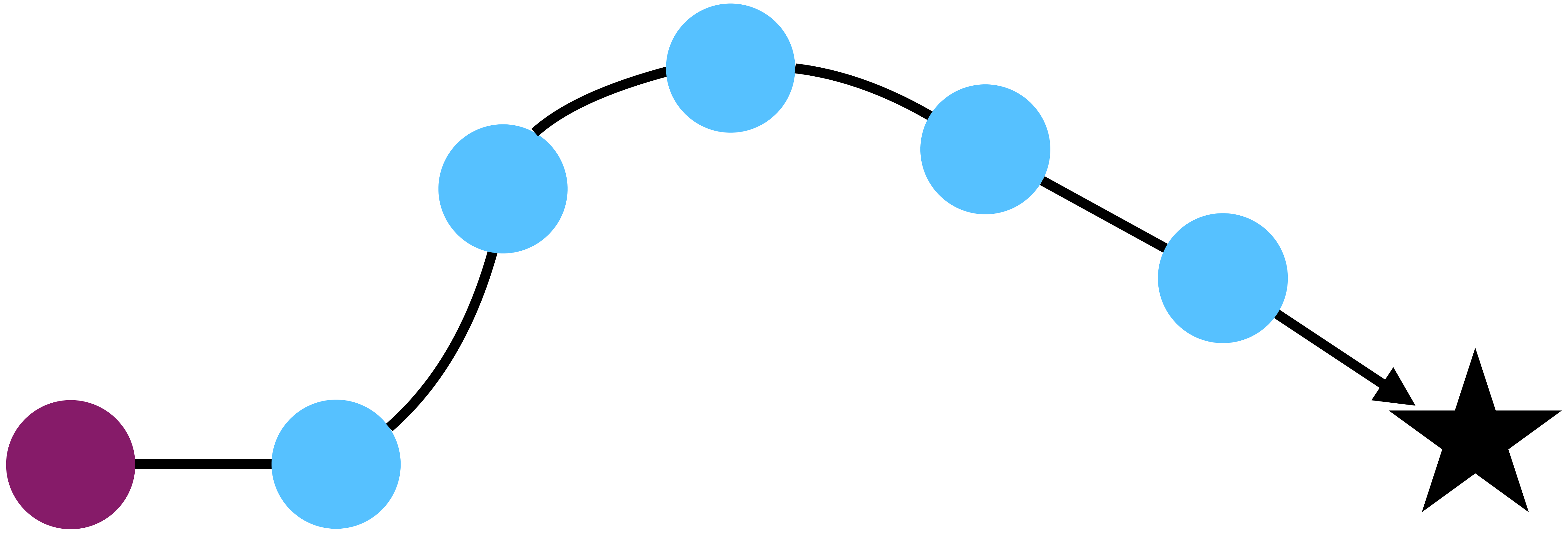
Model Predictive Control (MPC)

For each environment step

Observe state s

Decision-time Planning

Model Predictive Control (MPC)



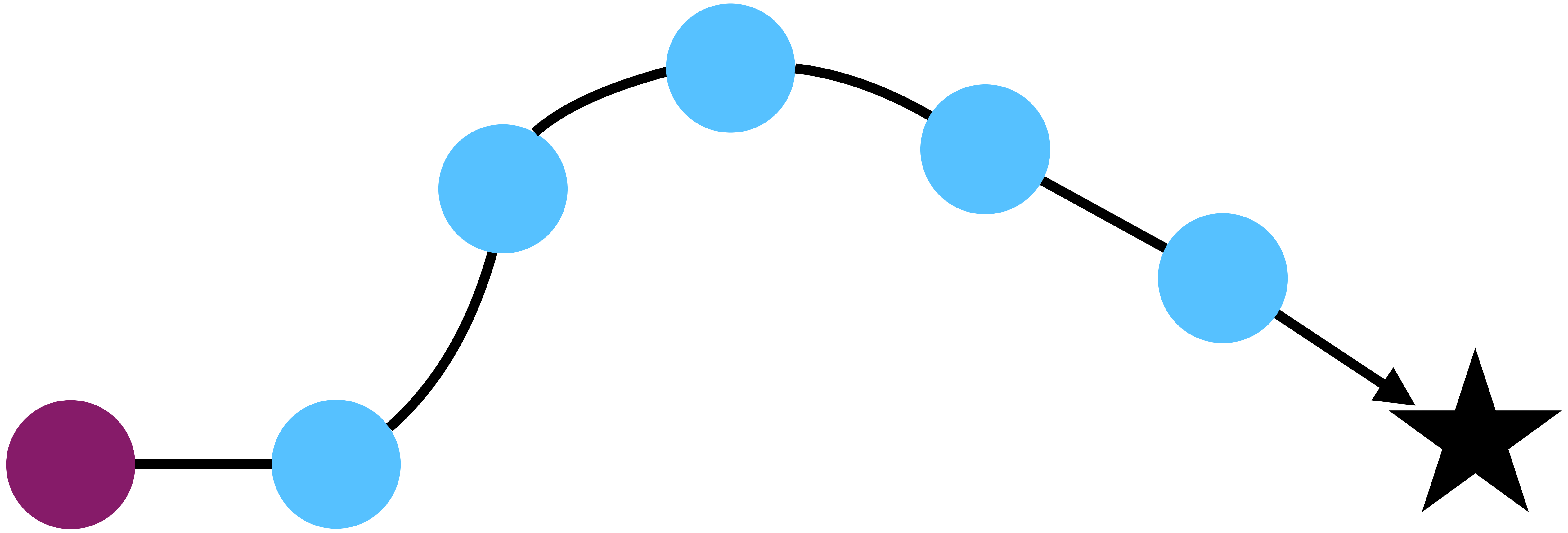
For each environment step

Observe state s

Plan $a_{0:H}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

Any trajectory optimisation method

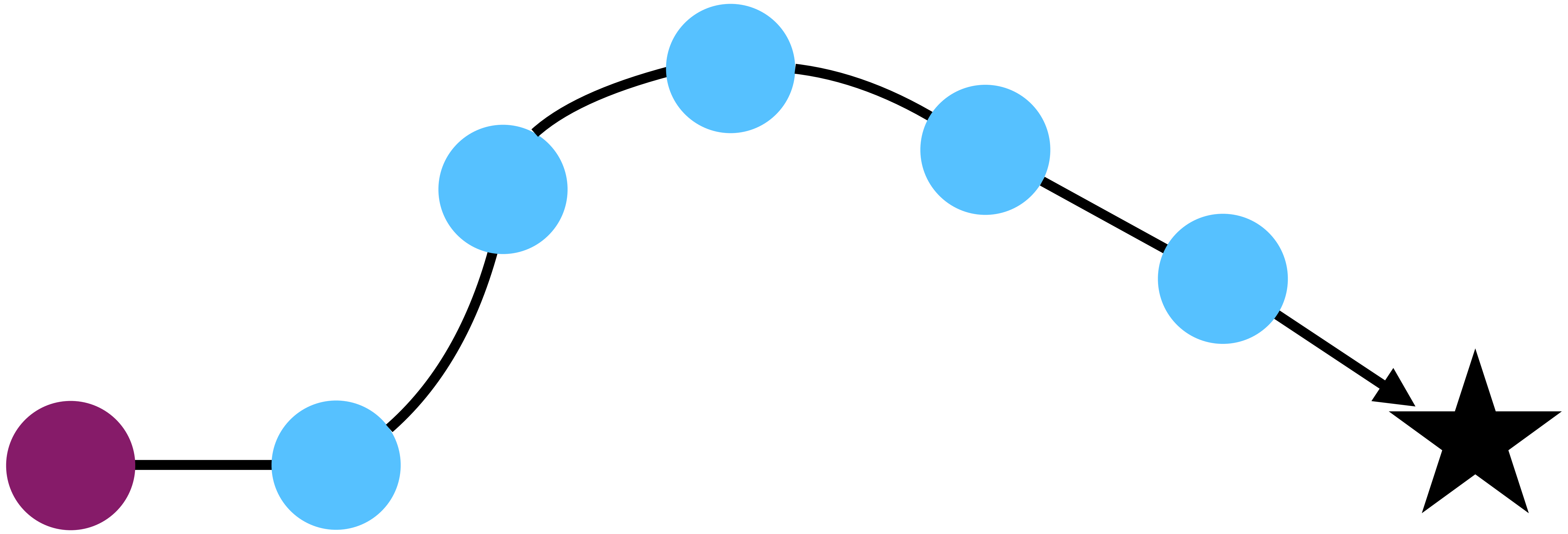
Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

FCAI

fcai.fi

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

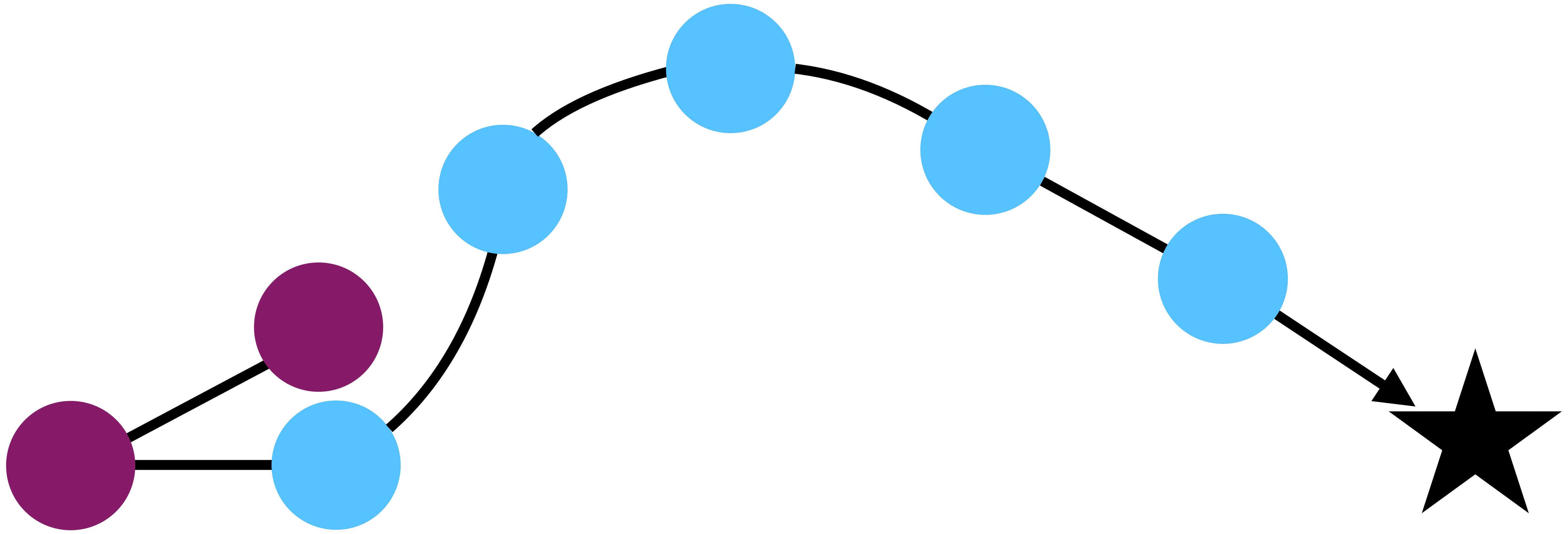
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

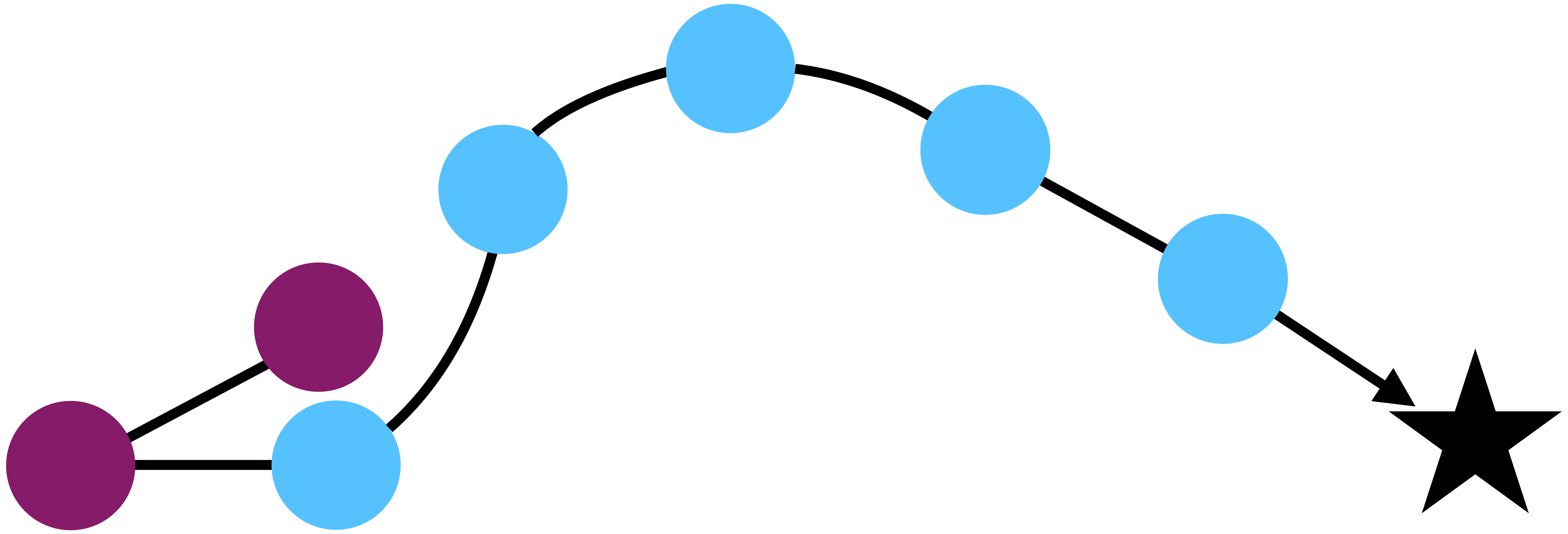
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

For each environment step

Observe state s

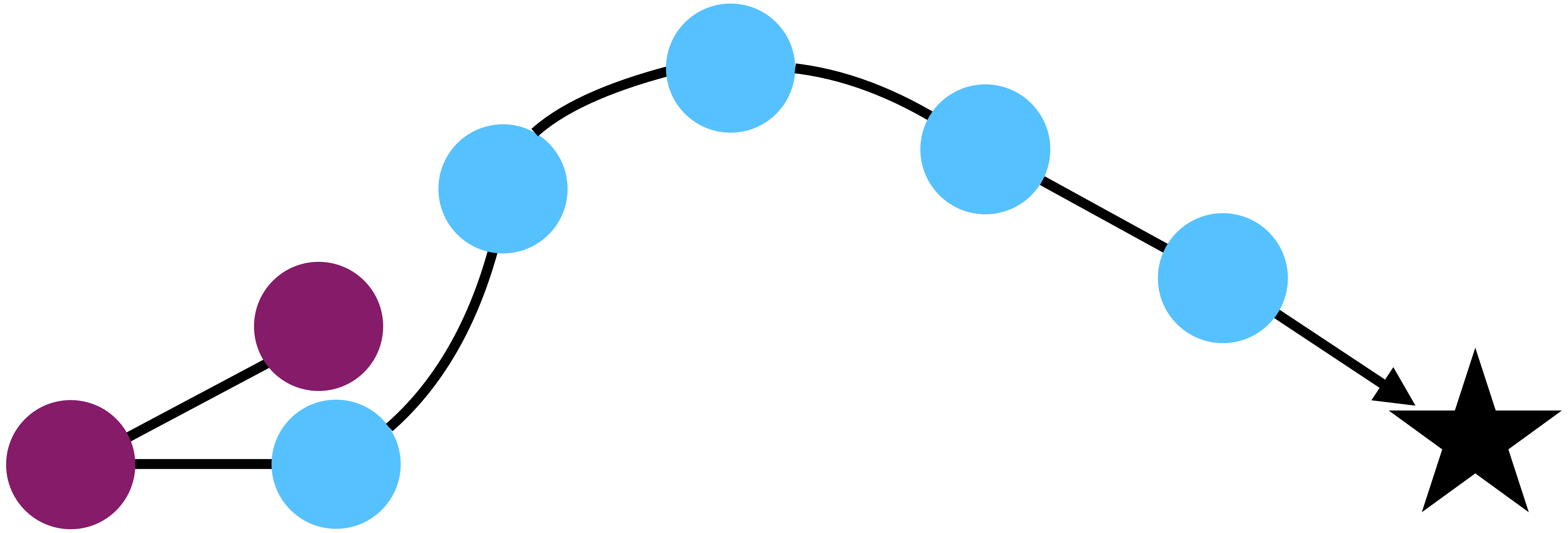
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

Discard $a_1, \ldots, a_H$

For each environment step

Observe state s

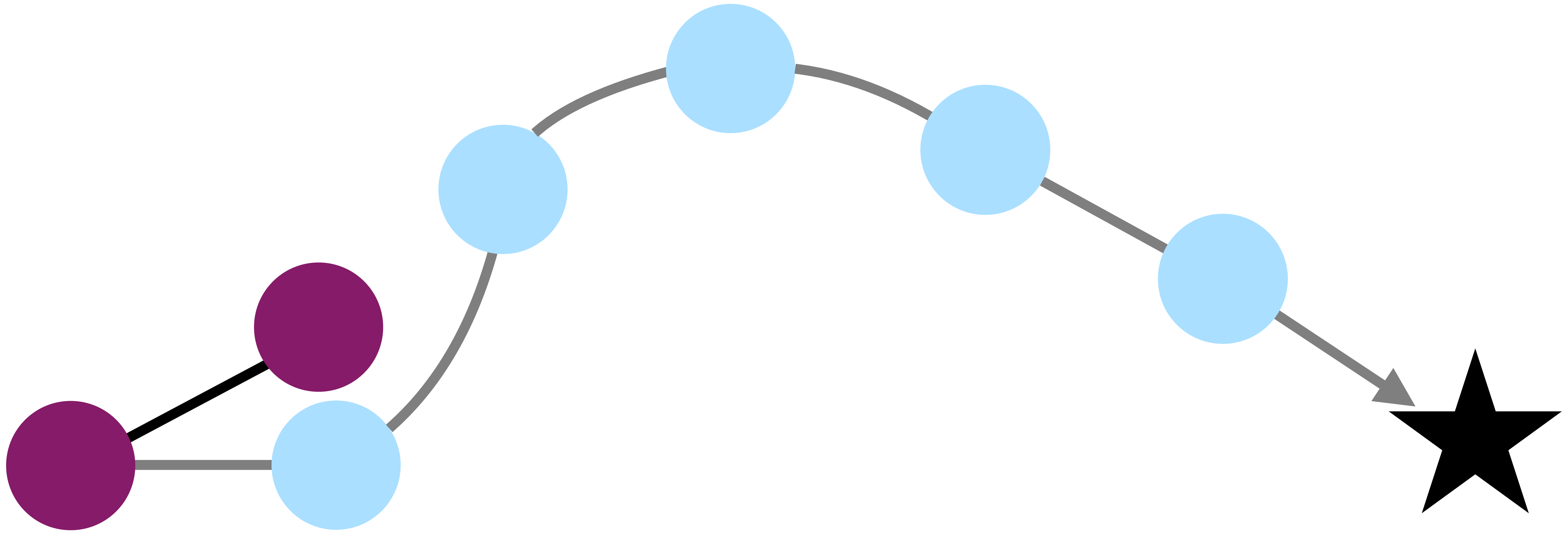
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

Discard $a_1, \ldots, a_H$

For each environment step

Observe state s

Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

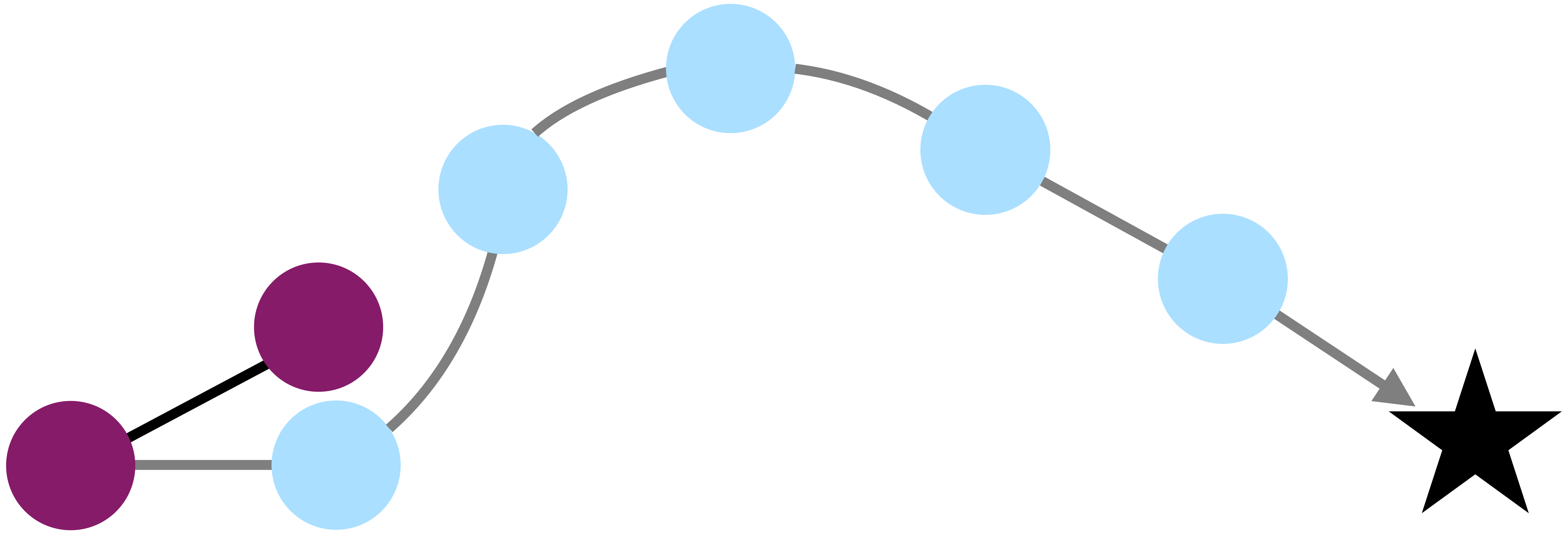
Execute a_0 and discard $a_1, \ldots, a_H$

FCAI

fcai.fi

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

Discard $a_1, \ldots, a_H$

So let's replan.

For each environment step

Observe state s

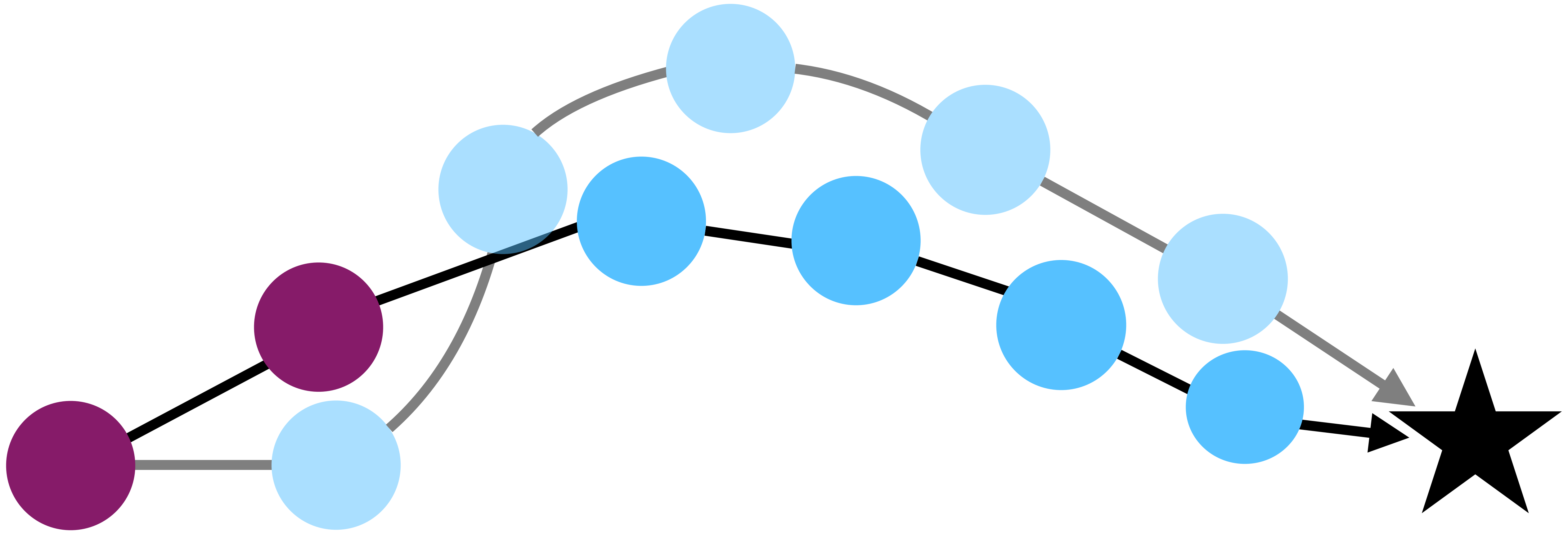
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

Discard $a_1, \ldots, a_H$

So let's replan.

For each environment step

Observe state s

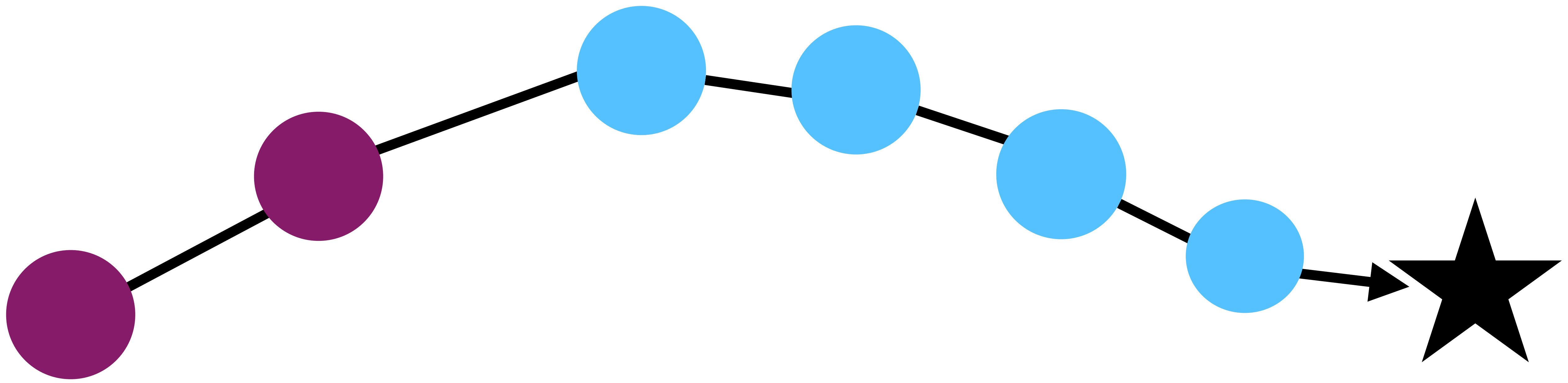
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

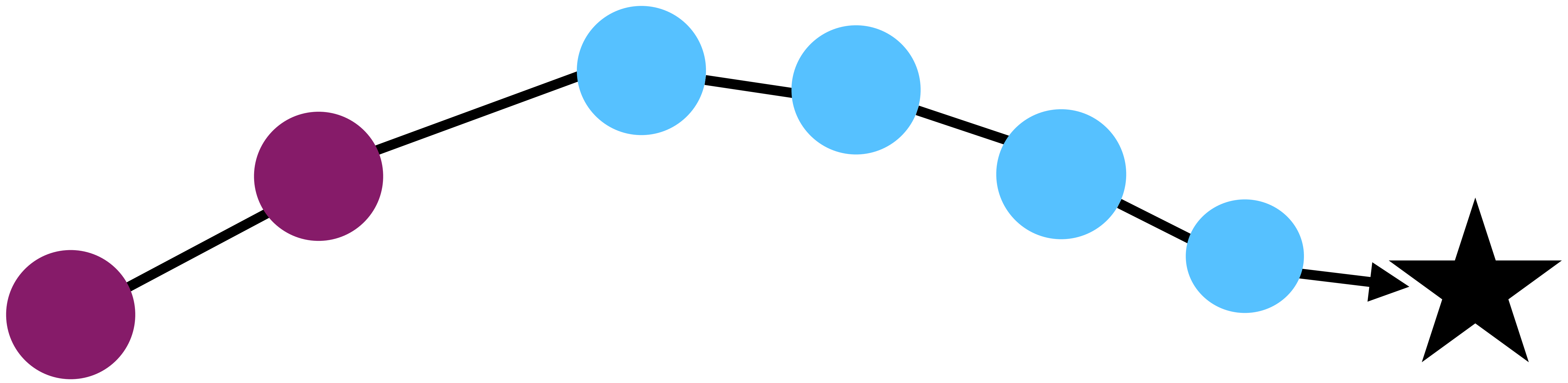
Diverged from planned trajectory…

Discard $a_1, \ldots, a_H$

So let's replan.

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

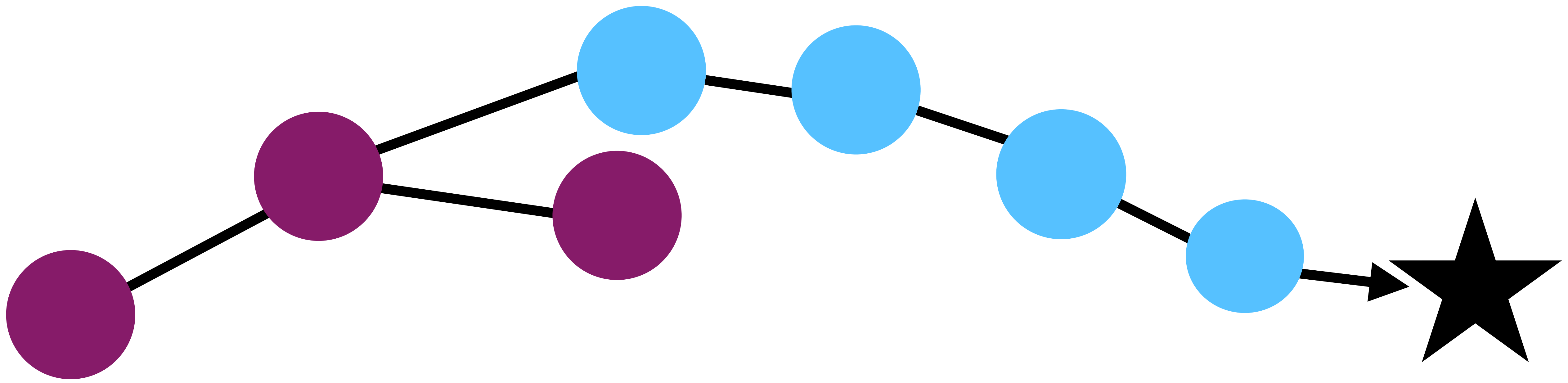
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

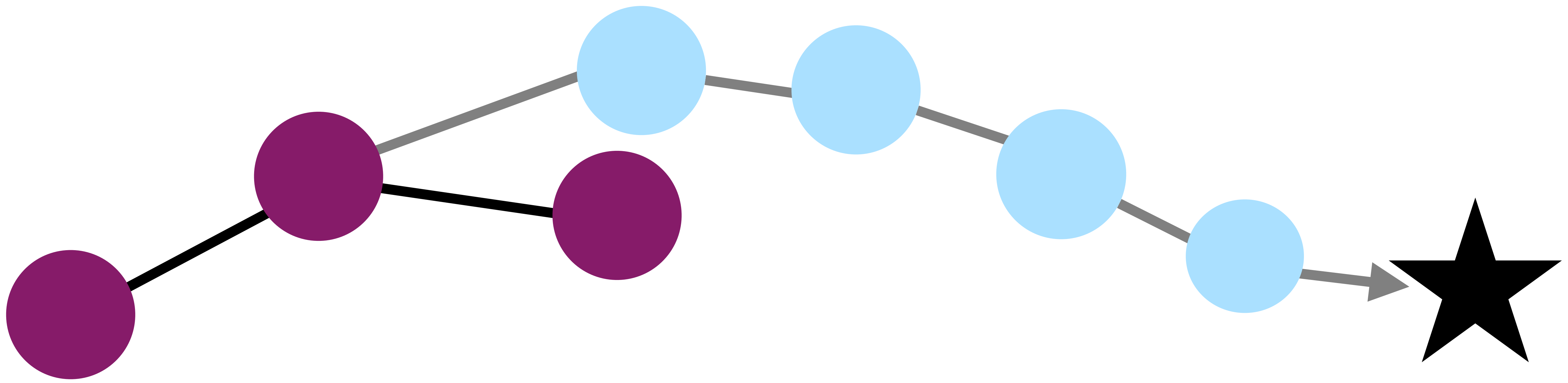
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

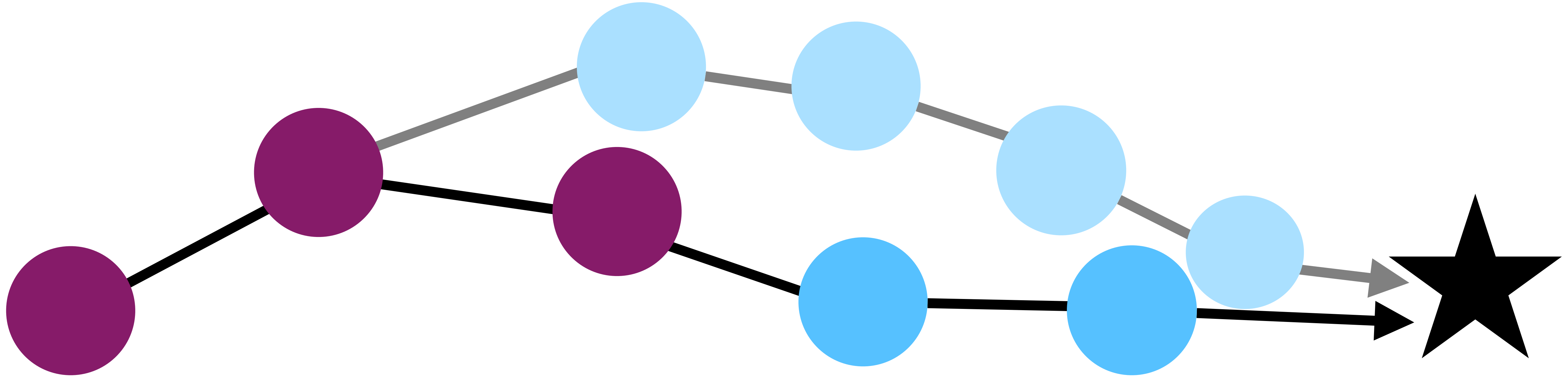
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

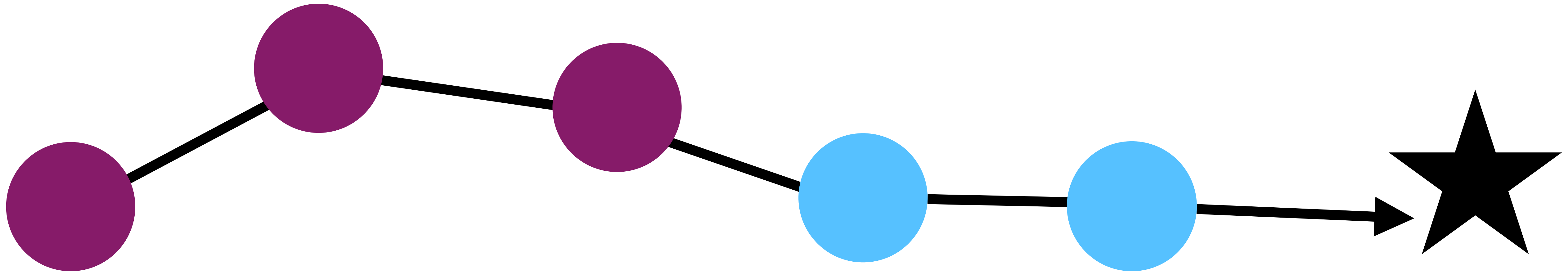
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

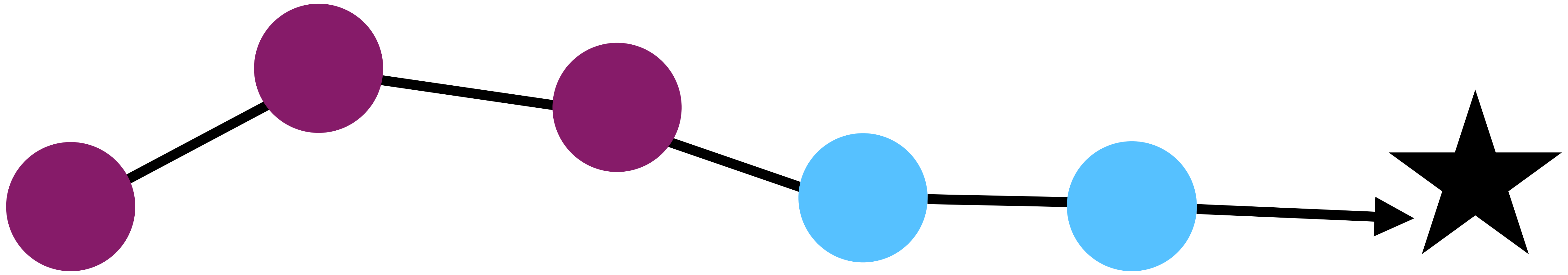
Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



And so on…

For each environment step

Observe state s

Any trajectory optimisation method

Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

Execute a_0 and discard $a_1, \ldots, a_H$

Decision-time Planning

Model Predictive Control (MPC)

$$\pi_{\mathrm{MPC}}(s; f, r, Q_\theta) = \arg\max_{a_0} \max_{a_1,\ldots,a_{H-1}} = \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H) \qquad \text{s.t.} \quad s_{t+1} = f(s_t, a_t)$$

$$s_0 = s$$

FCAI

fcai.fi

Decision-time Planning

Main Takeaways

Decision-time Planning

Main Takeaways

Common to use CEM

Decision-time Planning

Main Takeaways

Common to use CEM

- Avoids local optima

Decision-time Planning

Main Takeaways

Common to use CEM

- Avoids local optima
- Can handle deterministic and stochastic dynamics

Decision-time Planning

Main Takeaways

Common to use CEM

- Avoids local optima
- Can handle deterministic and stochastic dynamics
- Avoids exploding/vanishing gradients

Decision-time Planning

Main Takeaways

Common to use CEM

- Avoids local optima
- Can handle deterministic and stochastic dynamics
- Avoids exploding/vanishing gradients

Use MPC to make CEM closed loop

Decision-time Planning

Main Takeaways

Common to use CEM

- Avoids local optima
- Can handle deterministic and stochastic dynamics
- Avoids exploding/vanishing gradients

Use MPC to make CEM closed loop

Consider infinite horizon via learned $Q_\theta(s, a)$

Learning Objectives

Understand

1. ~~What a “model” is in model-based RL~~
2. ~~How a “model” can aid decision making~~
3. ~~Differences between background and decision-time planning~~
4. ~~Decision-time planning strategies for continuous actions~~
5. Sources of uncertainty in model-based RL
6. Rationale and insights for decision-making under uncertainty

Sources of Uncertainty in Model-Based RL

Sources of Uncertainty

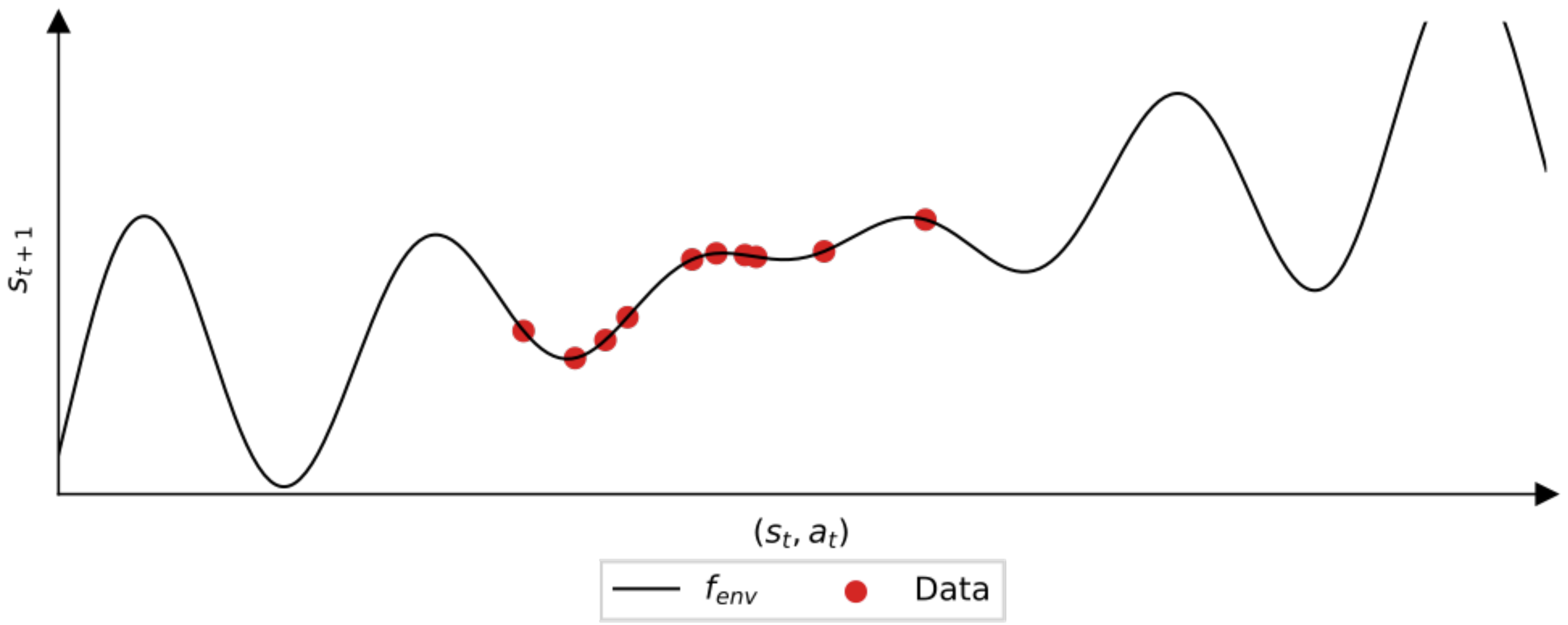
Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$



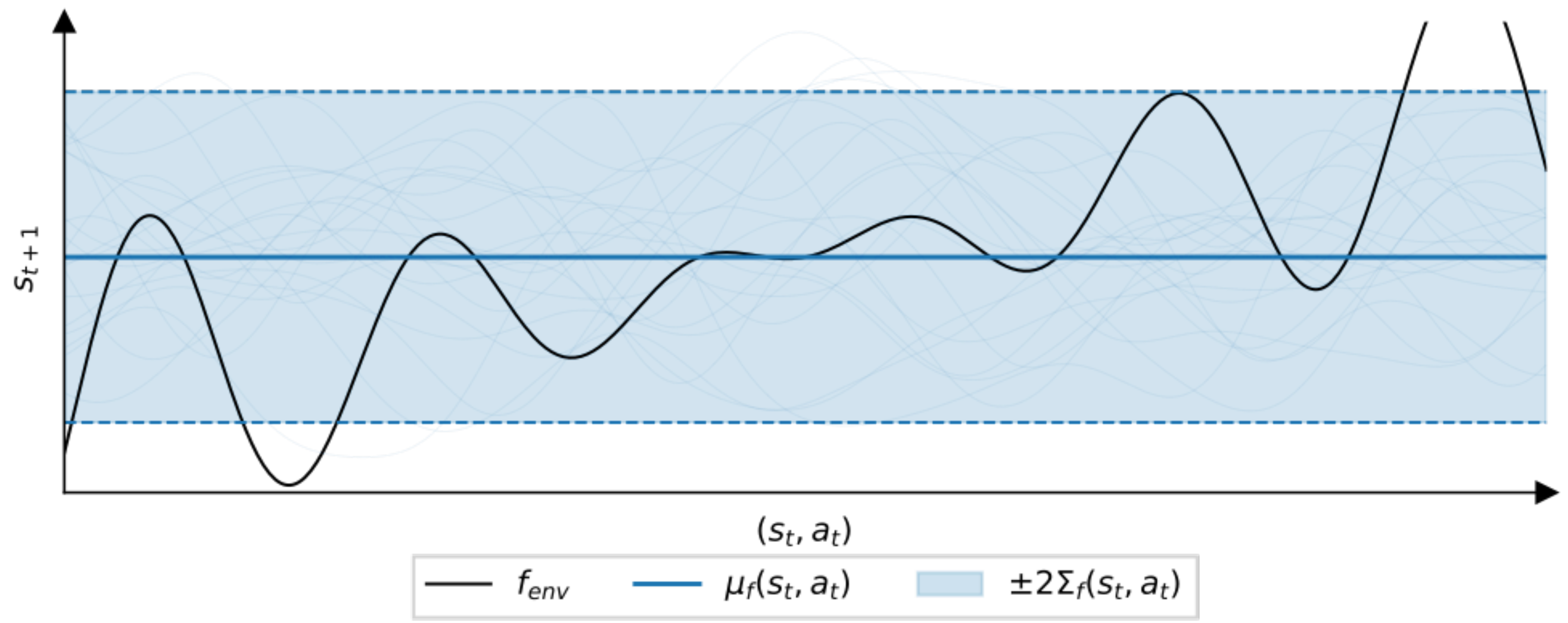
FCAI

fcai.fi

Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

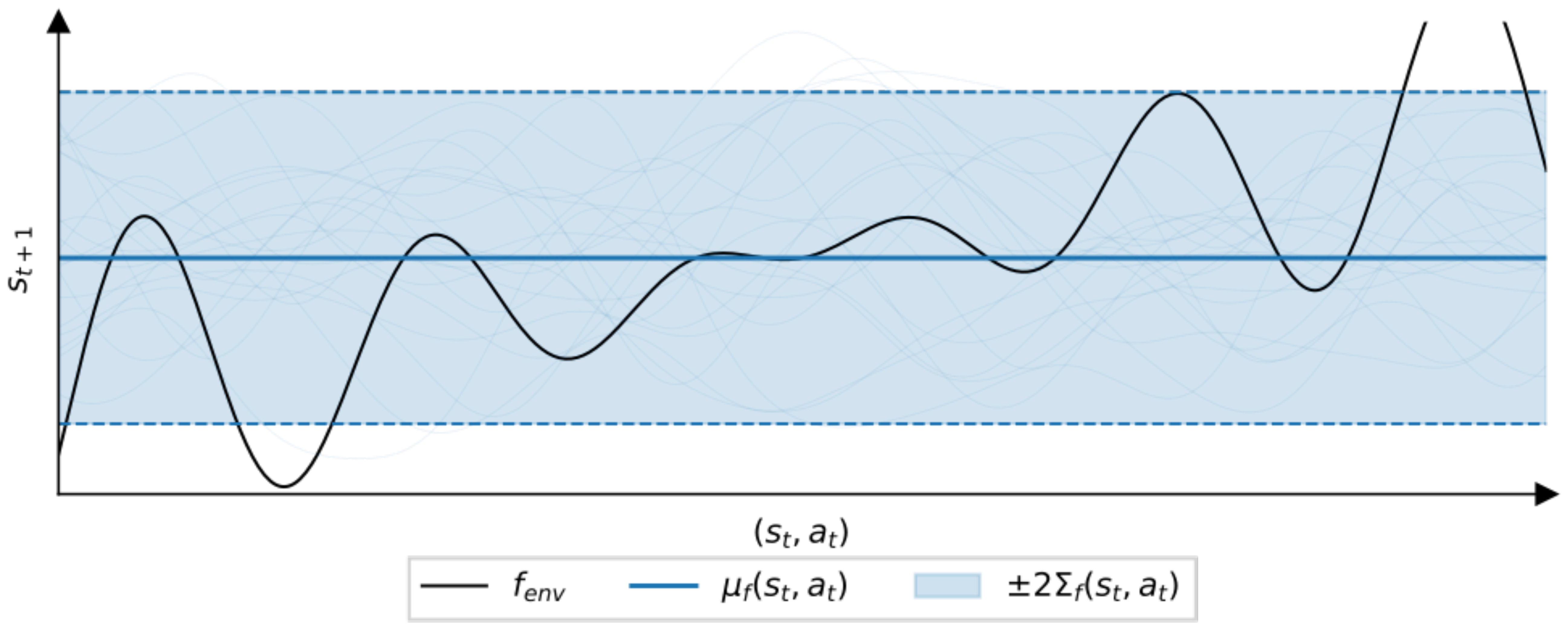


Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

Epistemic uncertainty



FCAI

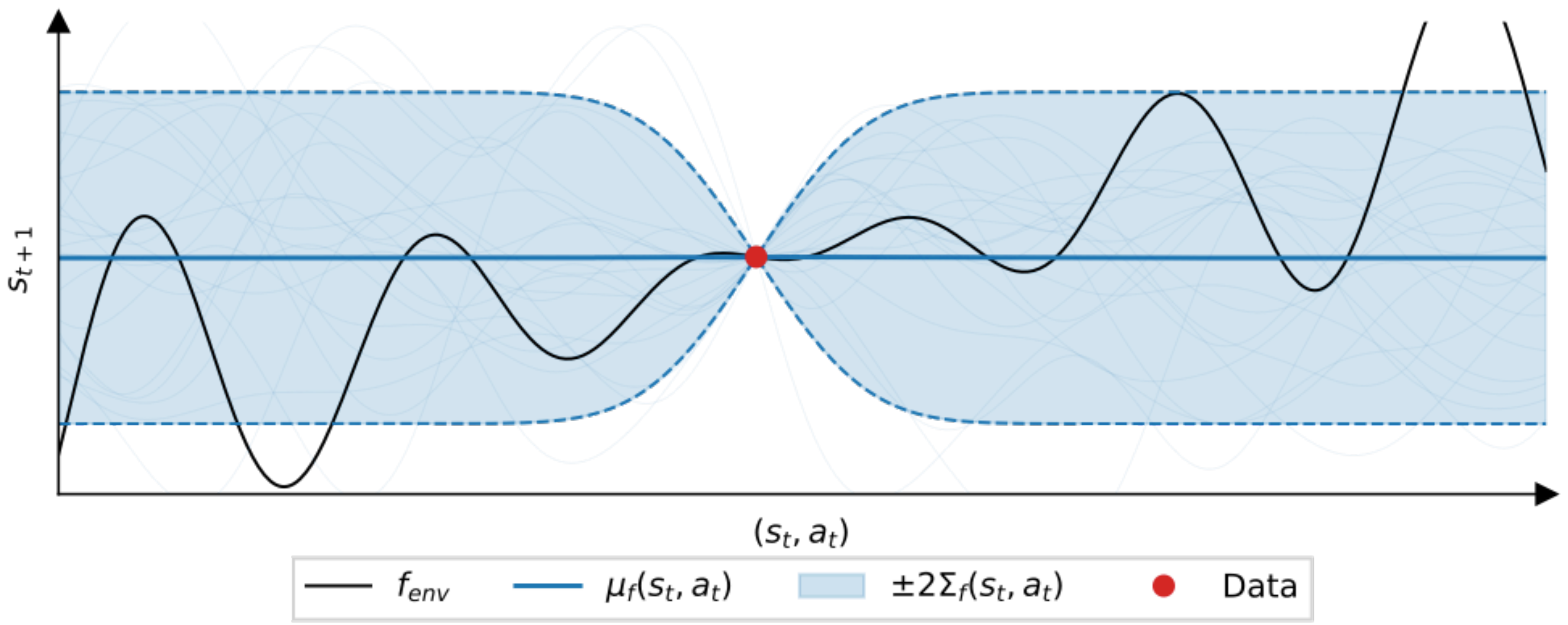
fcai.fi

Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

Epistemic uncertainty



FCAI

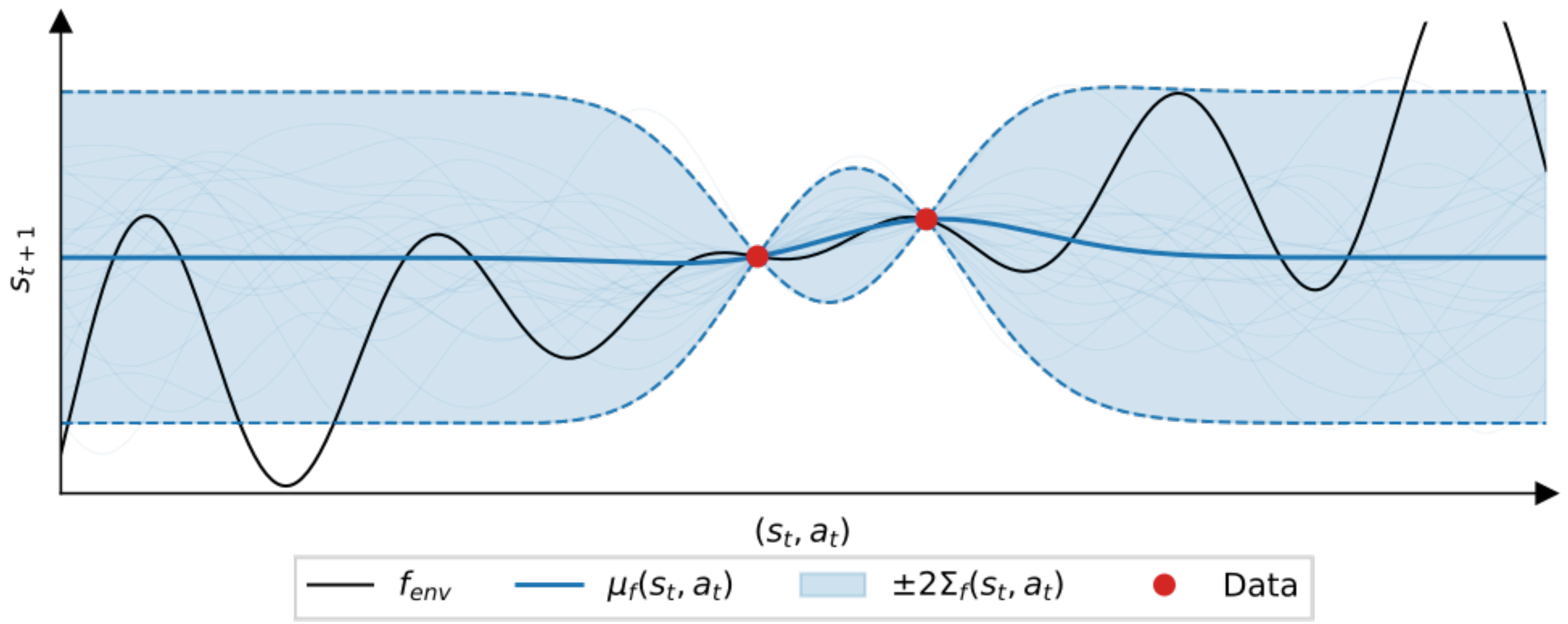
fcai.fi

Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

Epistemic uncertainty



FCAI

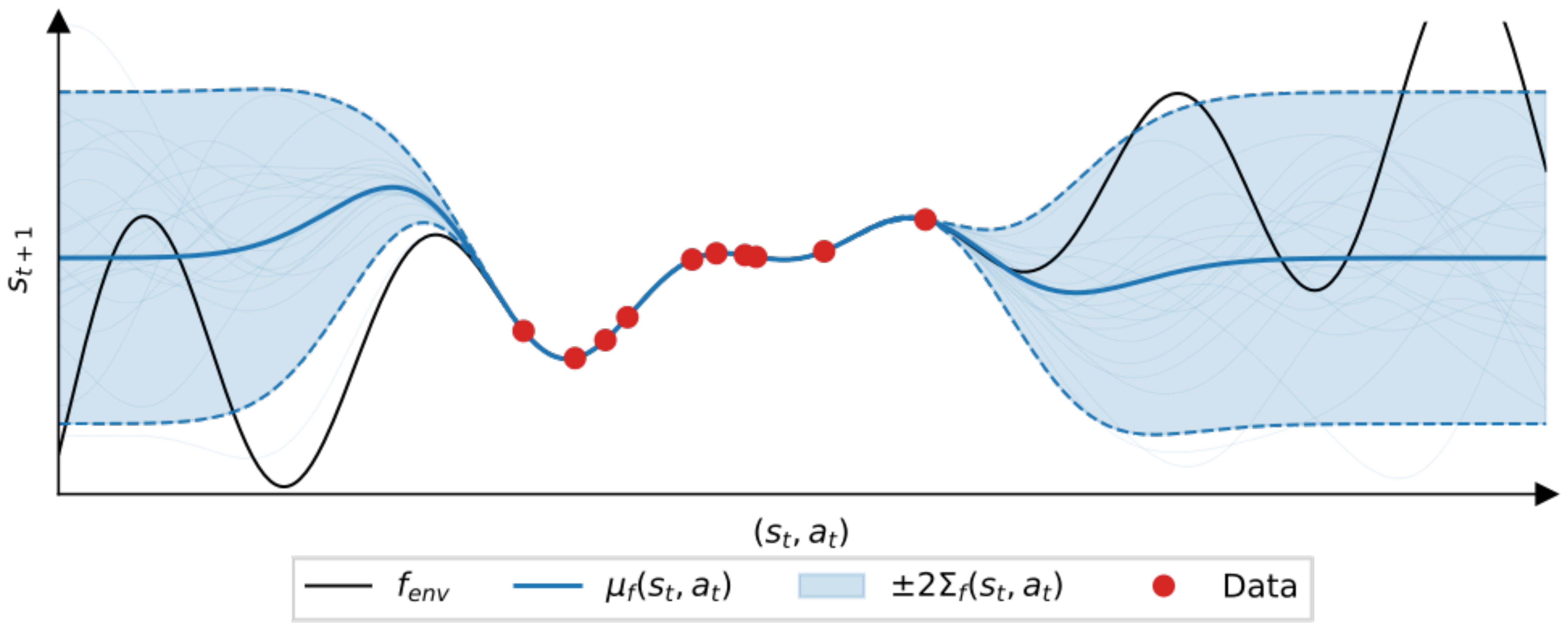
fcai.fi

Sources of Uncertainty

Learning From Limited Data

$$s_{t+1} = f_{env}(s_t, a_t)$$

Epistemic uncertainty



FCAI

fcai.fi

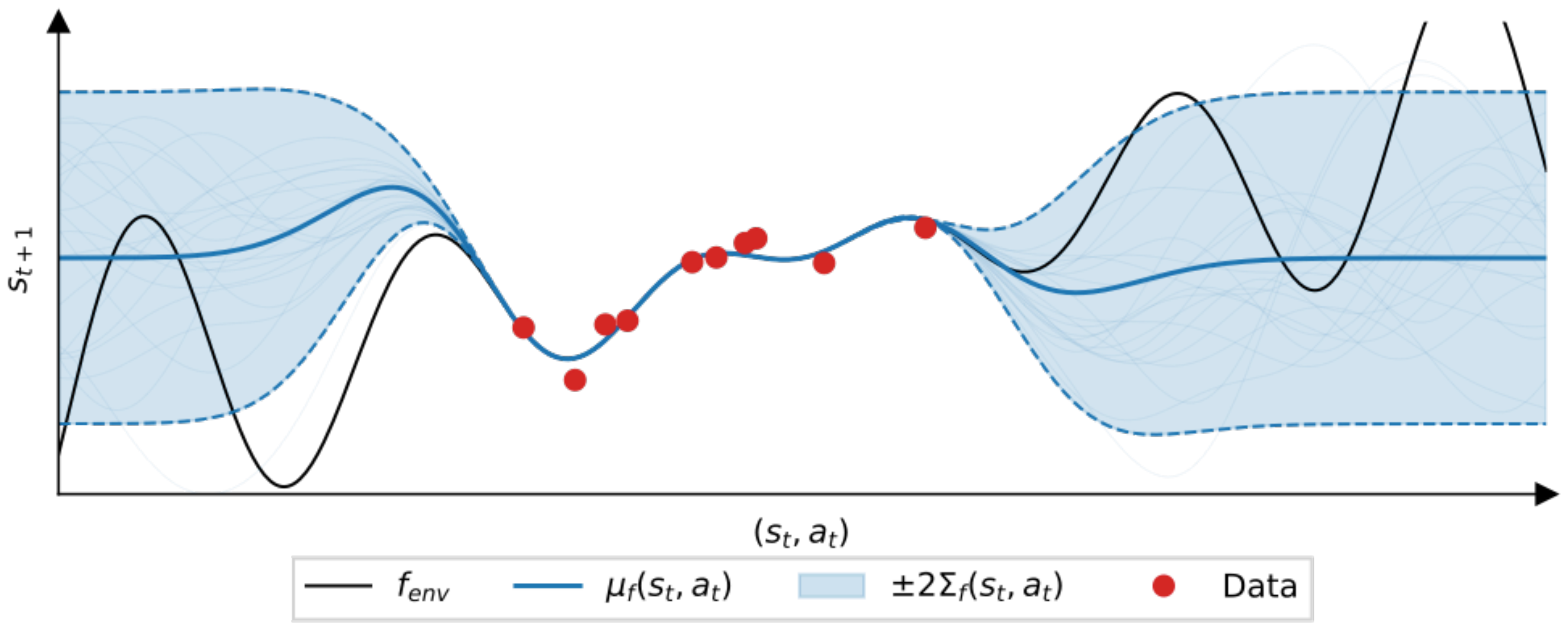
Sources of Uncertainty

Stochastic Environments

$$s_{t+1} = f_{env}(s_t, a_t) + \epsilon_t \quad \text{where} \quad \mathbb{E}[\epsilon_t] = 0$$

Sources of Uncertainty
Stochastic Environments

$$s_{t+1} = f_{env}(s_t, a_t) + \epsilon_t \quad \text{where} \quad \mathbb{E}[\epsilon_t] = 0$$



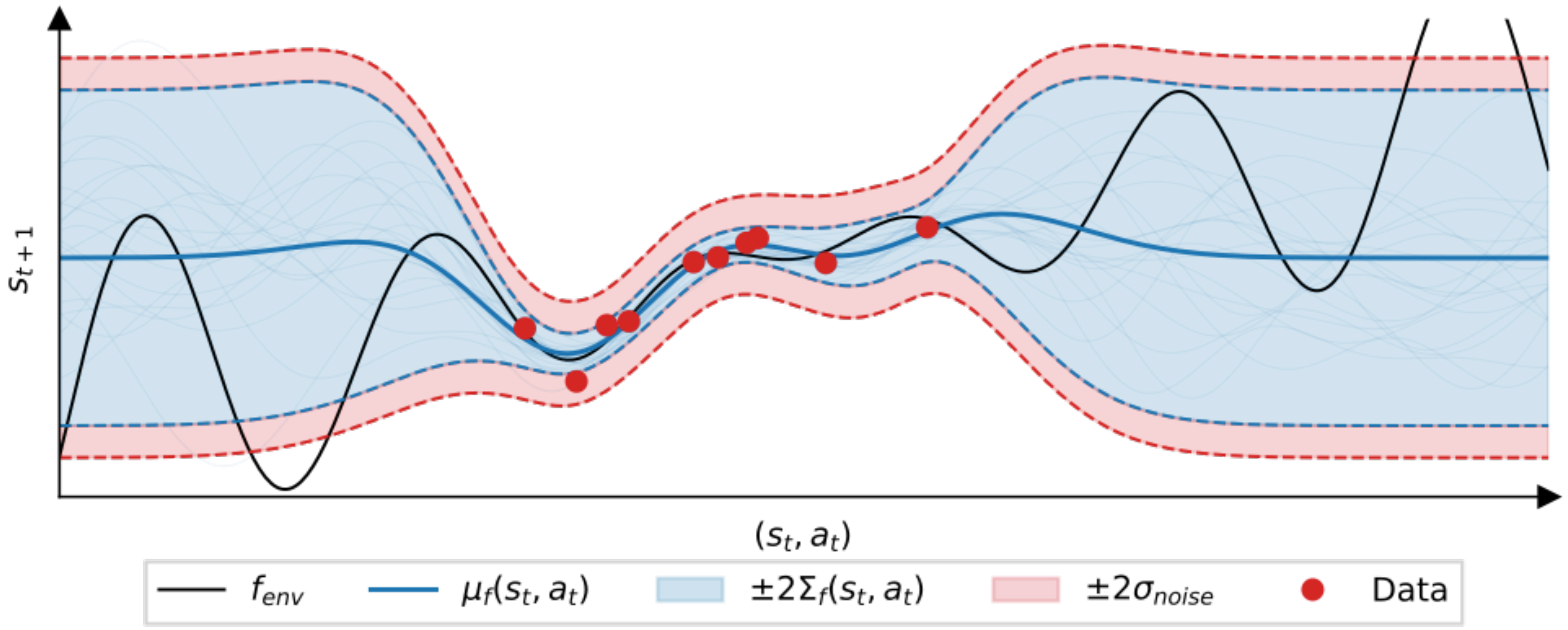
FCAI

fcai.fi

Sources of Uncertainty

Stochastic Environments

$$s_{t+1} = f_{env}(s_t, a_t) + \epsilon_t \quad \text{where} \quad \mathbb{E}[\epsilon_t] = 0$$



FCAI

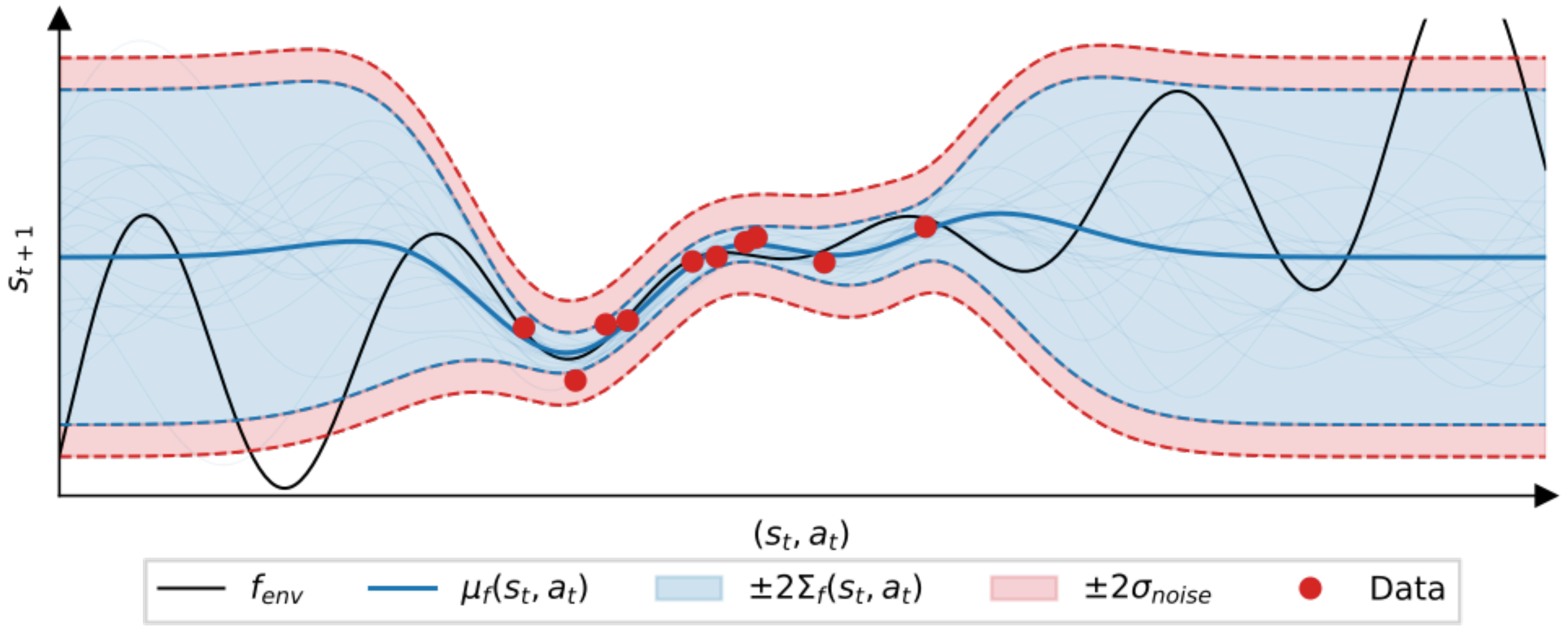
fcai.fi

Sources of Uncertainty

Stochastic Environments

$$s_{t+1} = f_{env}(s_t, a_t) + \epsilon_t \quad \text{where} \quad \mathbb{E}[\epsilon_t] = 0$$

Aleatoric uncertainty



Decision-making Under Uncertainty

Sources of Uncertainty

Decision-making Under Uncertainty

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{???}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

Return = discounted sum of rewards

$$J(\pi; f) = \mathbb{E}_{???}\left[\boxed{\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)} \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

Stochastic dynamics

$$J(\pi; f) = \mathbb{E}_{???}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \boxed{s_{t+1} = f(s_t, a_t) + \epsilon_t,} a_t = \pi(s_t)\right]$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

Deterministic policy

$$J(\pi; f) = \mathbb{E}_{???}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, \boxed{a_t = \pi(s_t)} \right]$$

FCAI

fcai.fi

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \boxed{\mathbb{E}_{???}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t) \right]$$

What is the expectation over?

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{???}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{\epsilon_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{\epsilon_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Expectation is over transition noise, i.e. aleatoric uncertainty

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{\epsilon_{0:\infty}}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\Big]$$

Expectation is over transition noise, i.e. aleatoric uncertainty

Posterior over dynamics models:

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{\epsilon_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Expectation is over transition noise, i.e. aleatoric uncertainty

Posterior over dynamics models:

$$p(f \mid \mathcal{D})$$

Sources of Uncertainty

Decision-making Under Uncertainty

RL objective:

$$J(\pi; f) = \mathbb{E}_{\epsilon_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_{t+1} = f(s_t, a_t) + \epsilon_t, a_t = \pi(s_t)\right]$$

Expectation is over transition noise, i.e. aleatoric uncertainty

Posterior over dynamics models:

$$p(f \mid \mathcal{D})$$

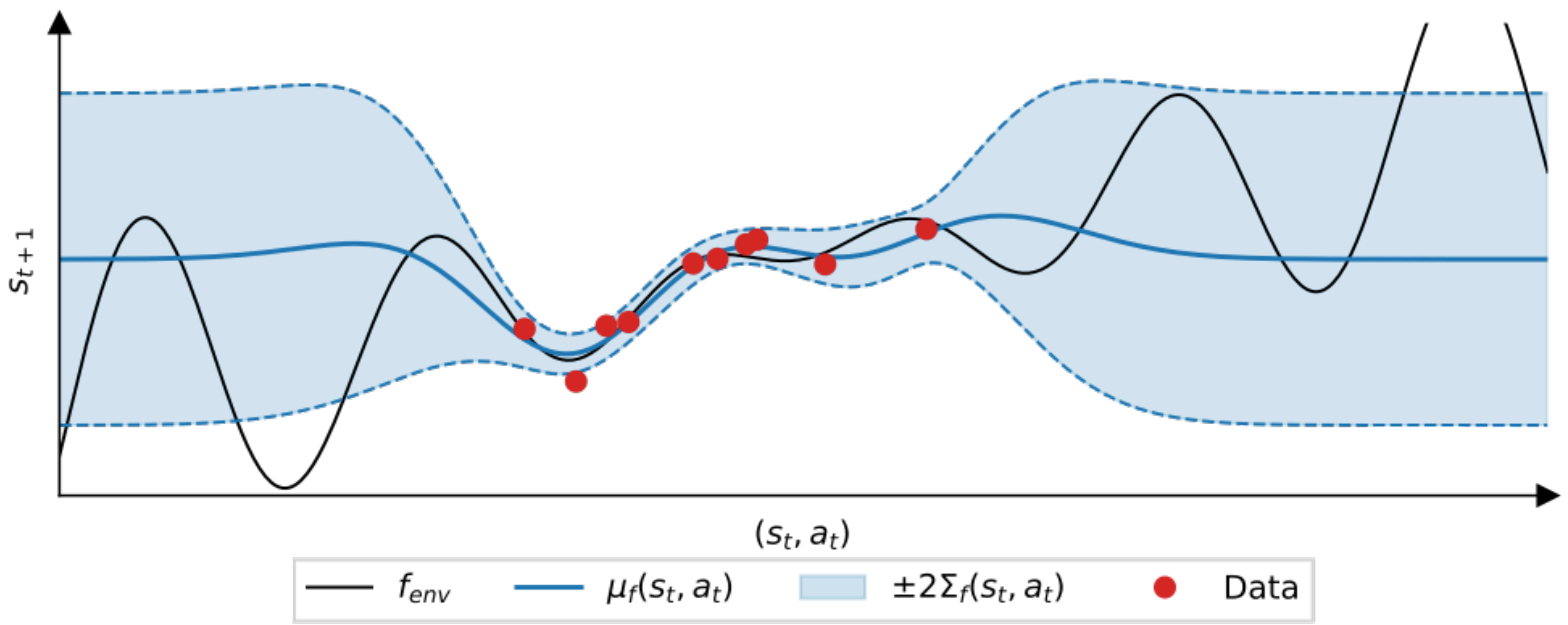
How should we use this?

Model Averaging

$$\pi^{Greedy} = \arg\max_{\pi} \mathbb{E}_{p(f|\mathscr{D})}\Big[J(\pi; f)\Big]$$

Model Averaging

$$\pi^{Greedy} = \arg\max_{\pi} \mathbb{E}_{p(f|\mathscr{D})}\Big[J(\pi; f)\Big]$$

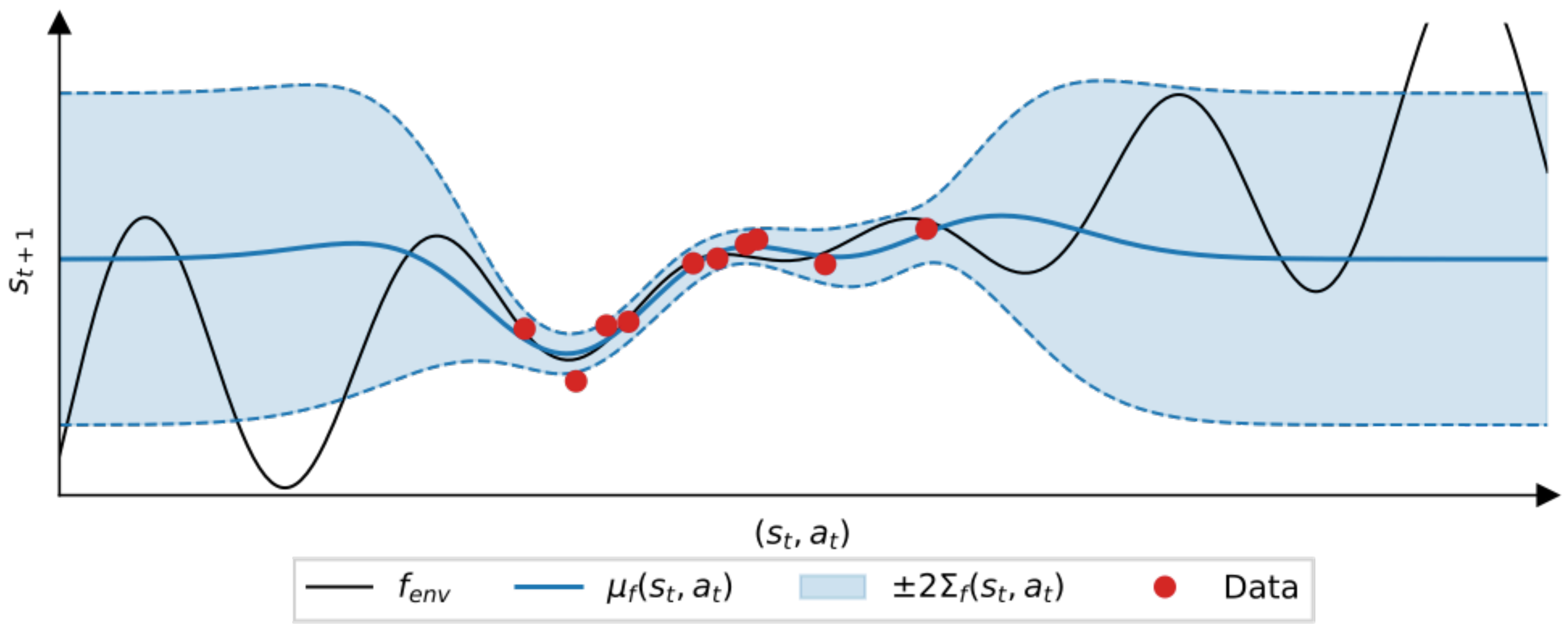


FCAI

fcai.fi

Model Averaging

$$\pi^{Greedy} = \arg\max_{\pi} \mathbb{E}_{p(f|\mathcal{D})}\Big[J(\pi; f)\Big]$$



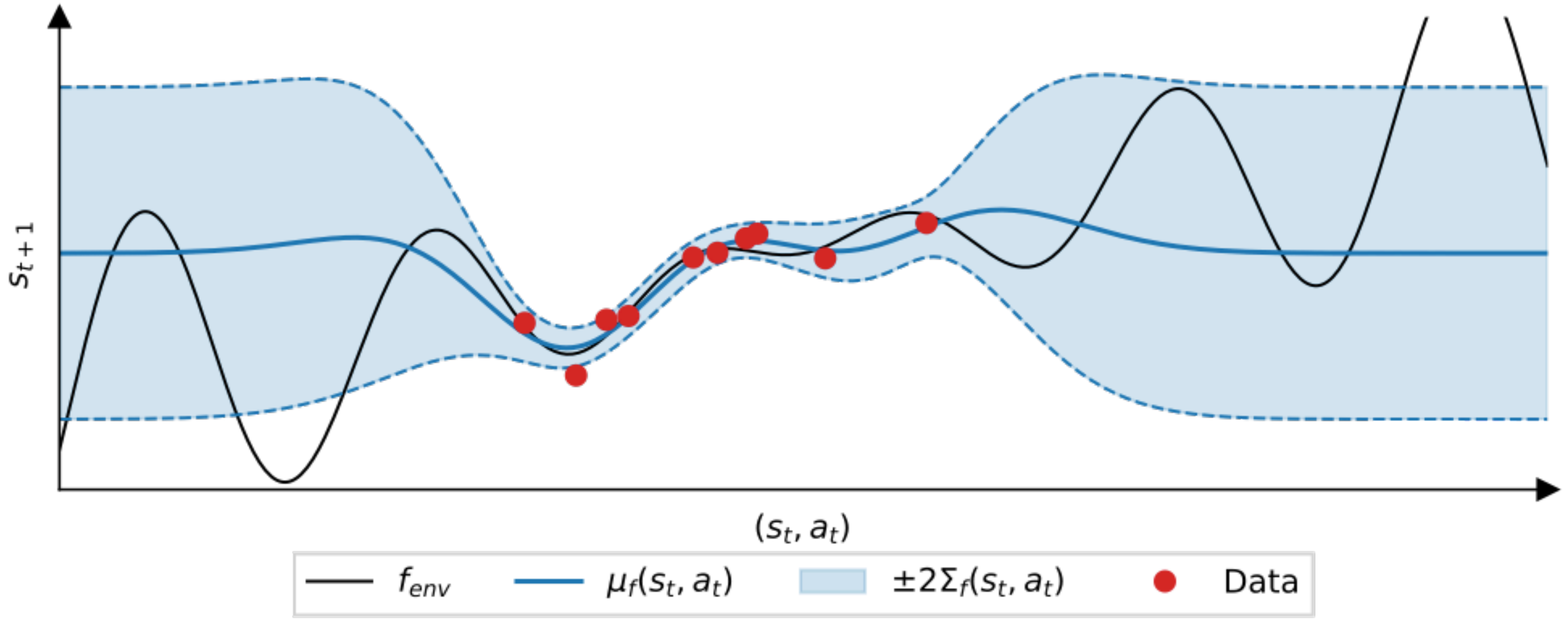
Combats model bias

Model Averaging

$$\pi^{Greedy} = \arg\max_{\pi} \mathbb{E}_{p(f|\mathcal{D})}\left[J(\pi; f)\right]$$

PILCO, PETS, etc

Combats model bias

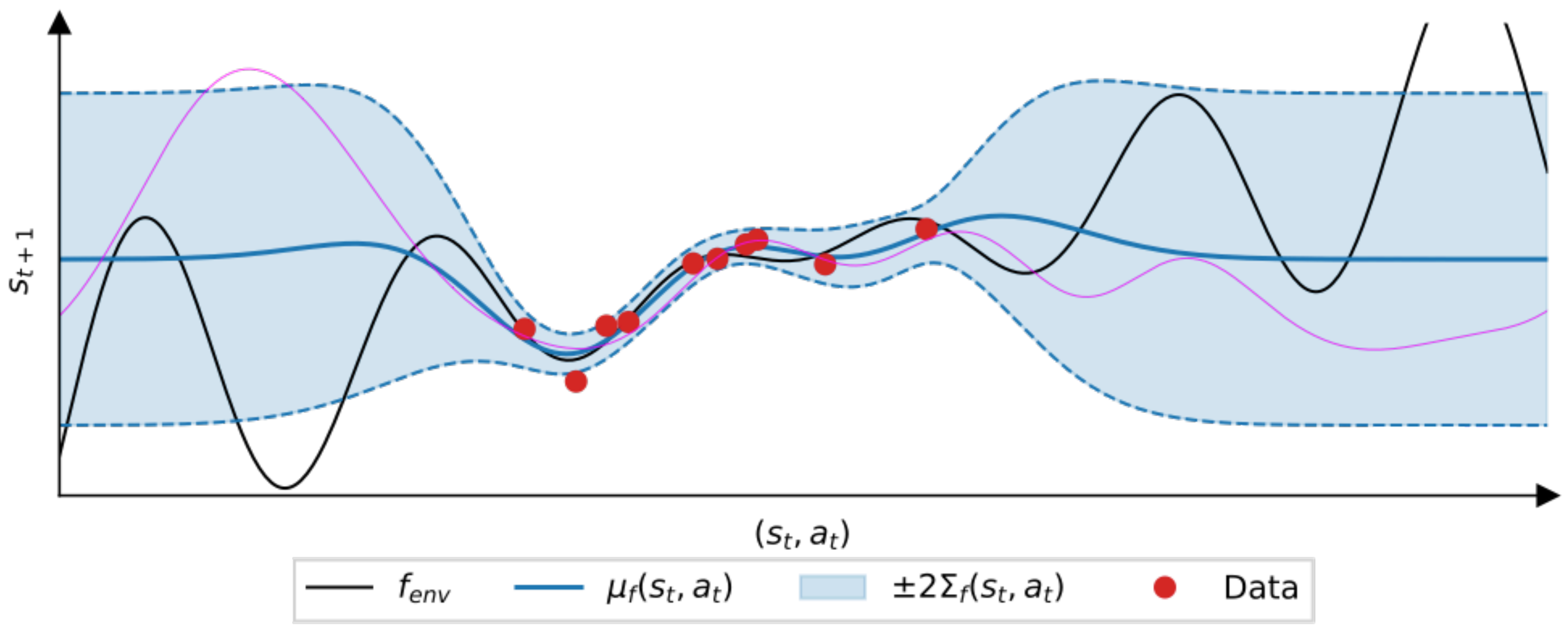


Deisenroth et al. (2011). PILCO: A Model-Based and Data-Efficient Approach to Policy Search. ICML.

Kurtland et al. (2018). Deep Reinforcement Learning in a Handful of Trials using Probabilisitic Dynamics Models. NeurIPS.

Exploration via Posterior Sampling

$$\pi^{PS} = \arg\max_{\pi} J(\pi; \tilde{f}), \quad \tilde{f} \sim p(f \mid \mathcal{D})$$



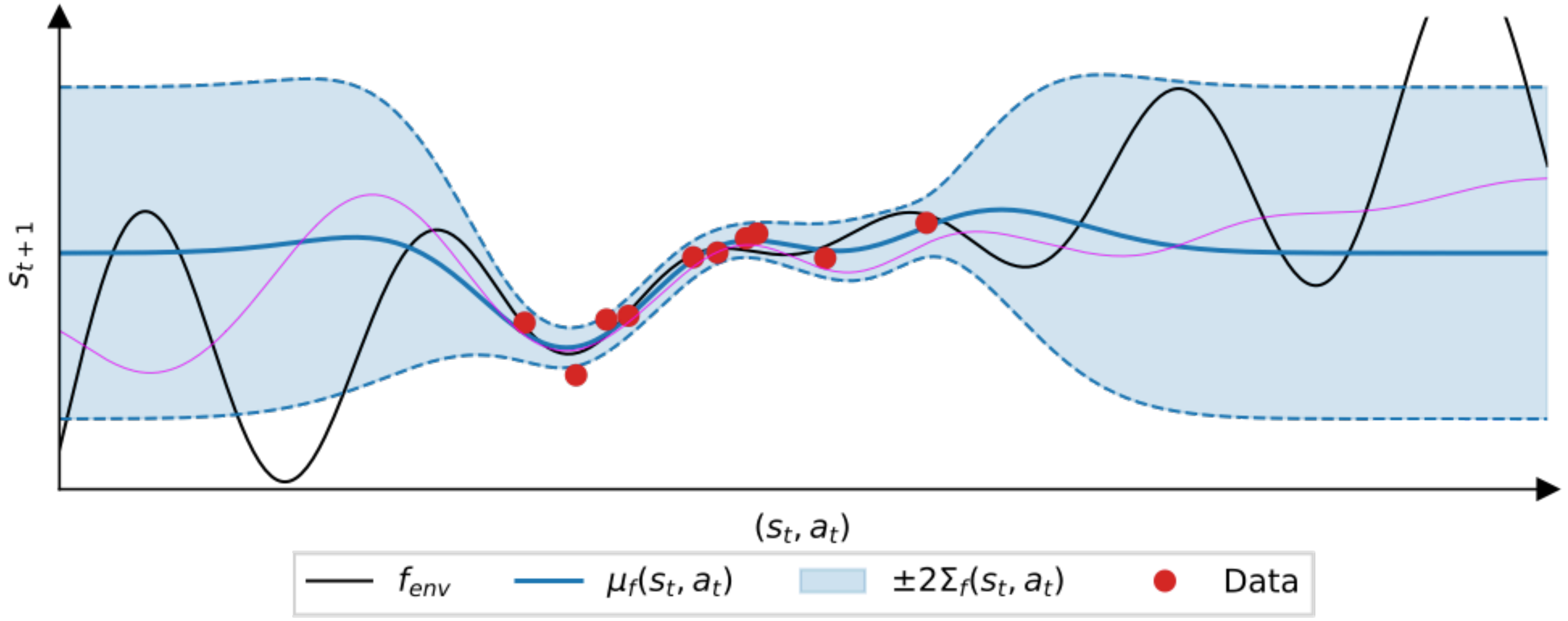
FCAI

Osband et al. (2013). (More) Efficient Reinforcement Learning via Posterior Sampling. NeurIPS.

fcai.fi

Exploration via Posterior Sampling

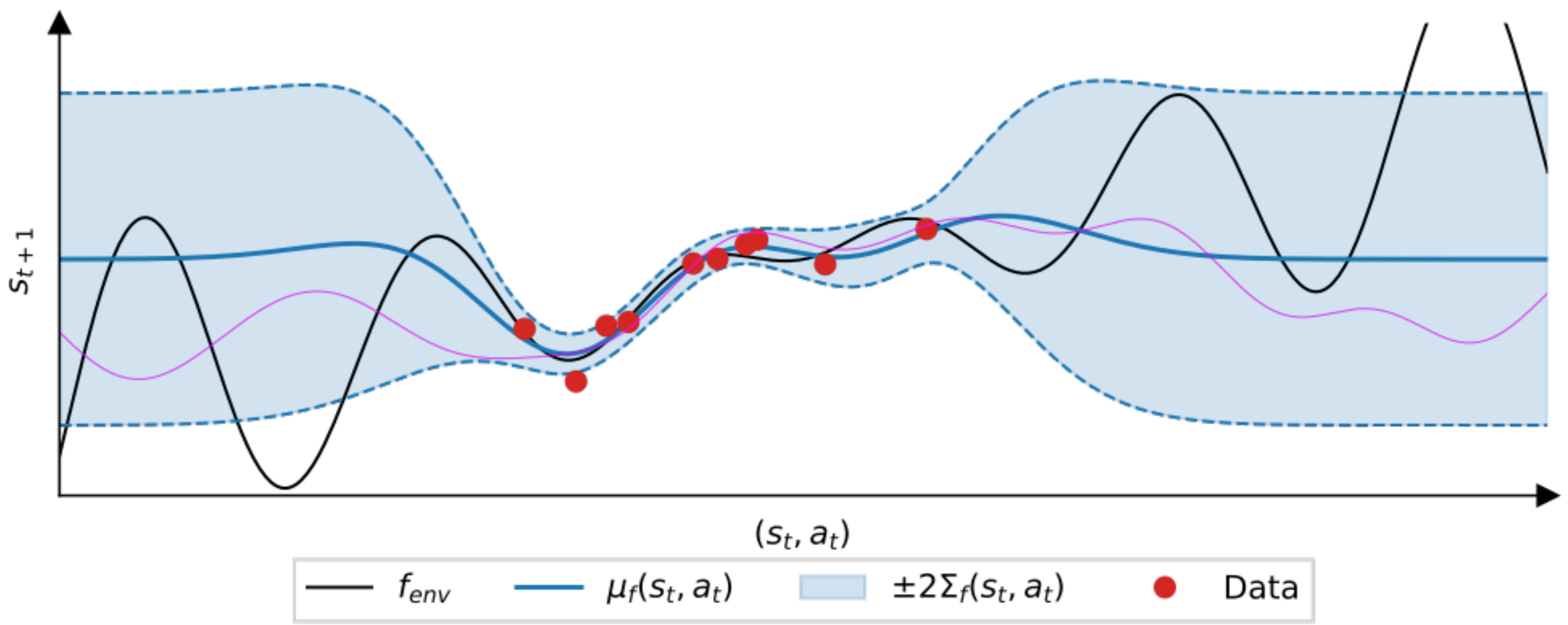
$$\pi^{PS} = \arg\max_{\pi} J(\pi; \tilde{f}), \quad \tilde{f} \sim p(f \mid \mathscr{D})$$



Osband et al. (2013). (More) Efficient Reinforcement Learning via Posterior Sampling. NeurIPS.

Exploration via Posterior Sampling

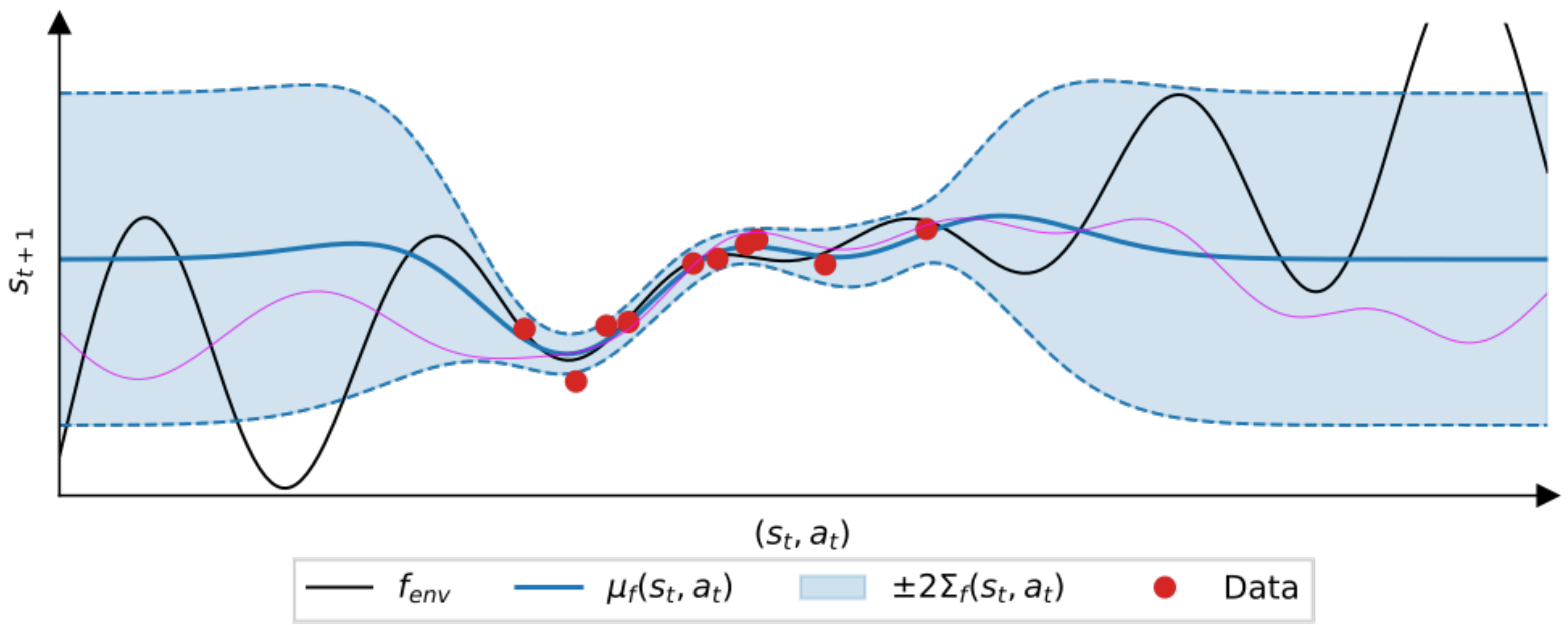
$$\pi^{PS} = \arg\max_{\pi} J(\pi; \tilde{f}), \quad \tilde{f} \sim p(f \mid \mathscr{D})$$



Osband et al. (2013). (More) Efficient Reinforcement Learning via Posterior Sampling. NeurIPS.

Exploration via Posterior Sampling

$$\pi^{PS} = \arg\max_{\pi} J(\pi; \tilde{f}), \quad \tilde{f} \sim p(f \mid \mathscr{D})$$



No extra hyperparamters

Osband et al. (2013). (More) Efficient Reinforcement Learning via Posterior Sampling. NeurIPS.

Exploration via Upper Confidence Bound

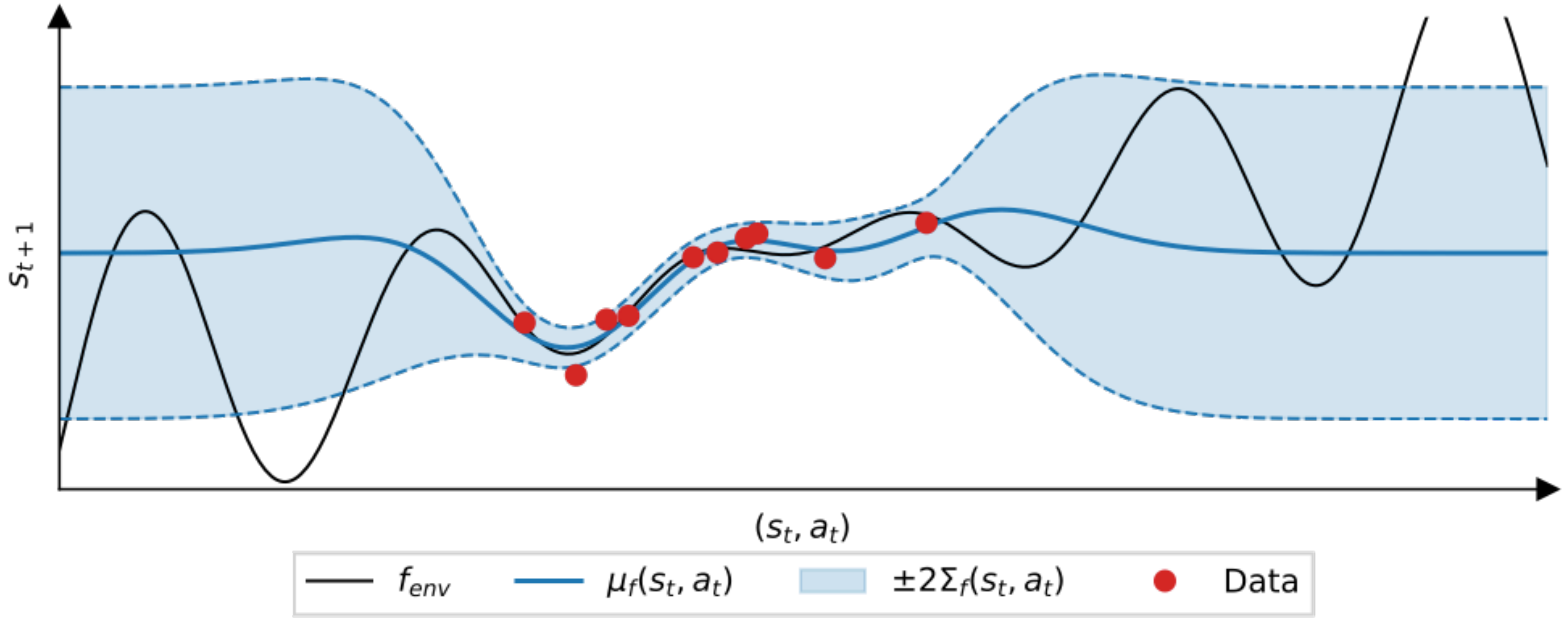
$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f) \qquad \mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$

Exploration via Upper Confidence Bound

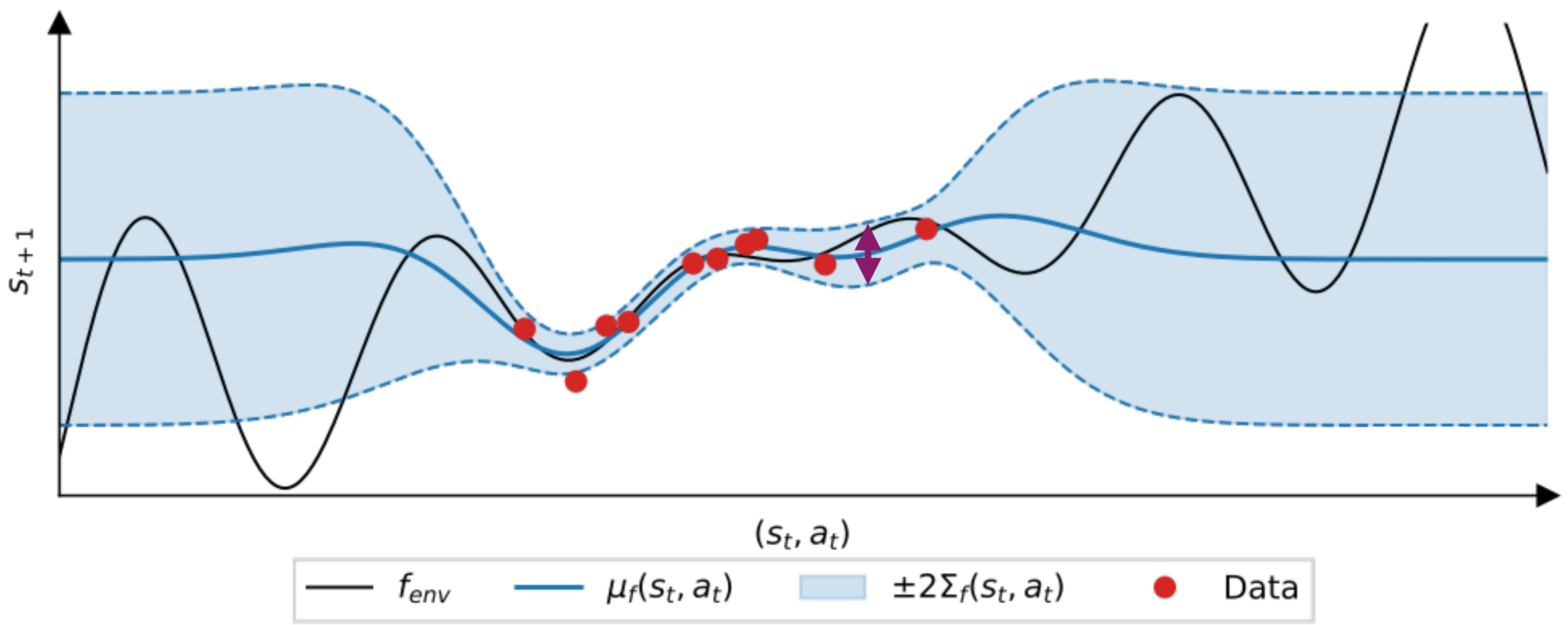
$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f) \qquad \mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$



Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

$$\mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$



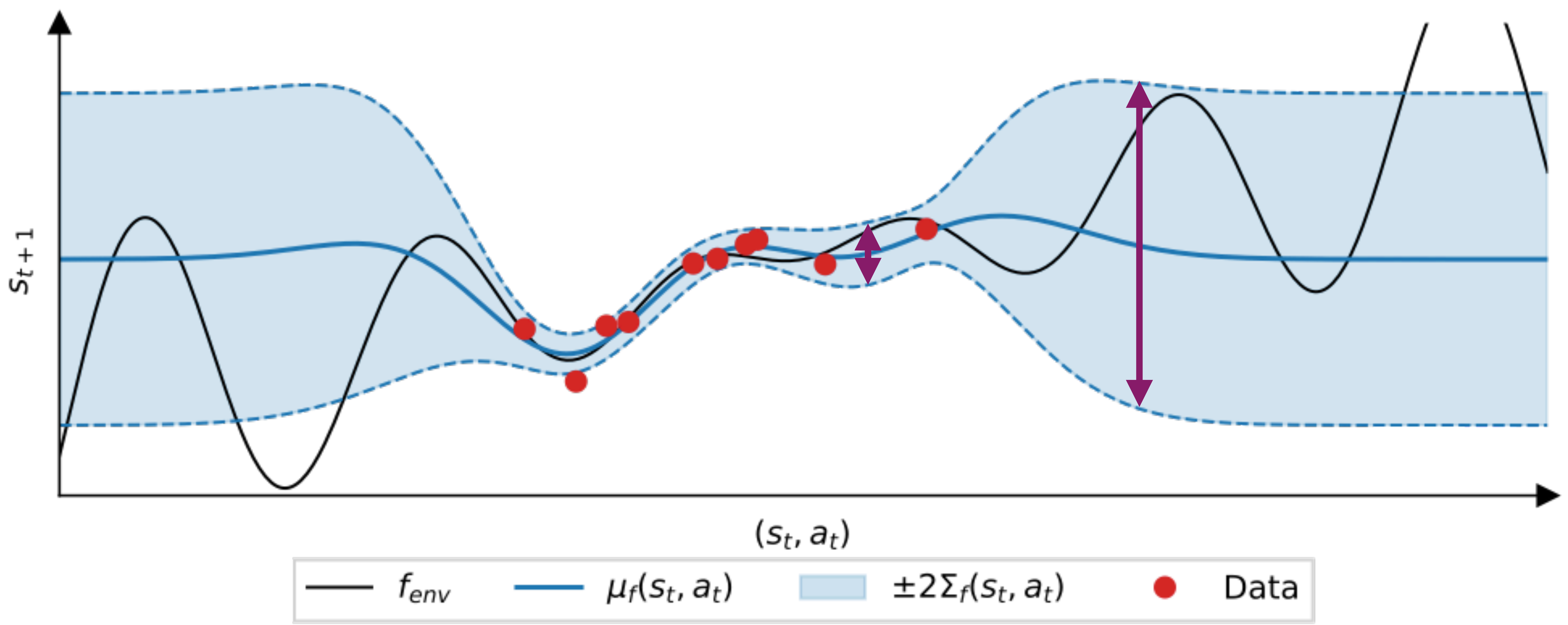
FCAI

fcai.fi

Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

$$\mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \le \beta \Sigma_f(s,a) \right\}$$



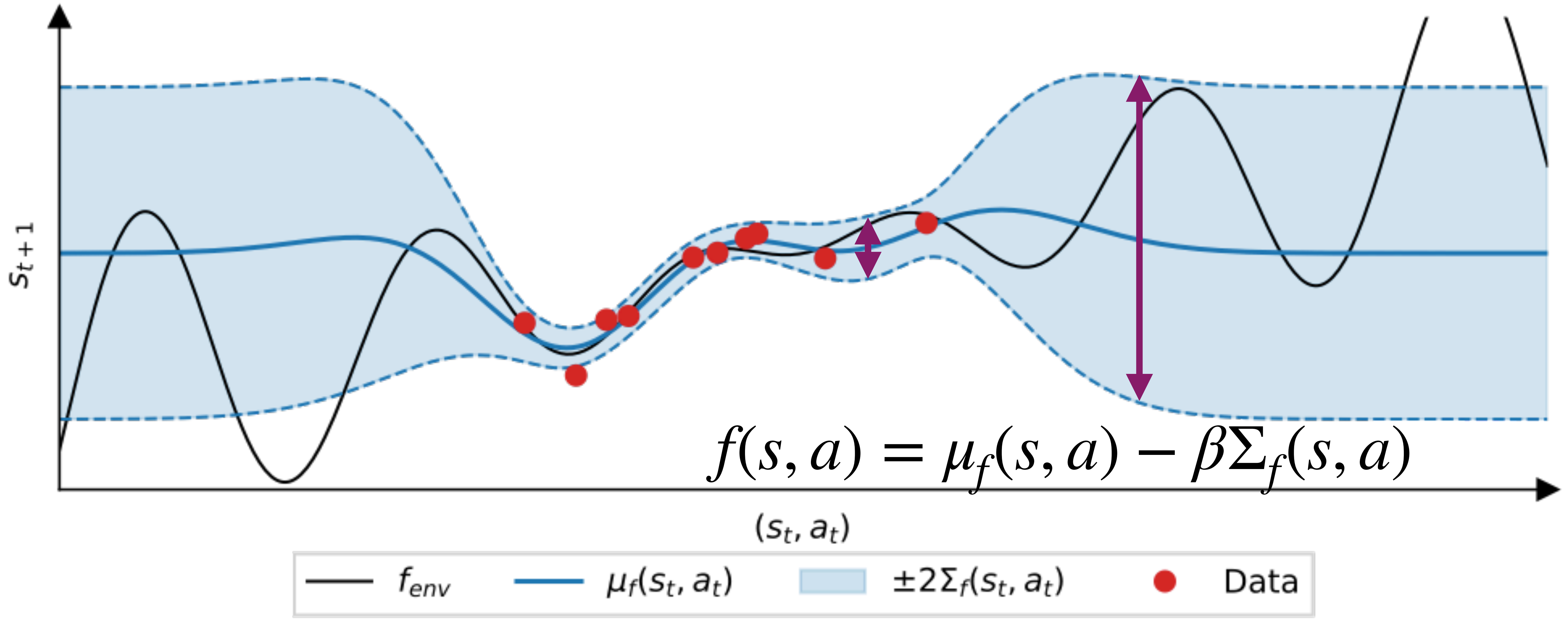
FCAI

fcai.fi

Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

$$\mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$



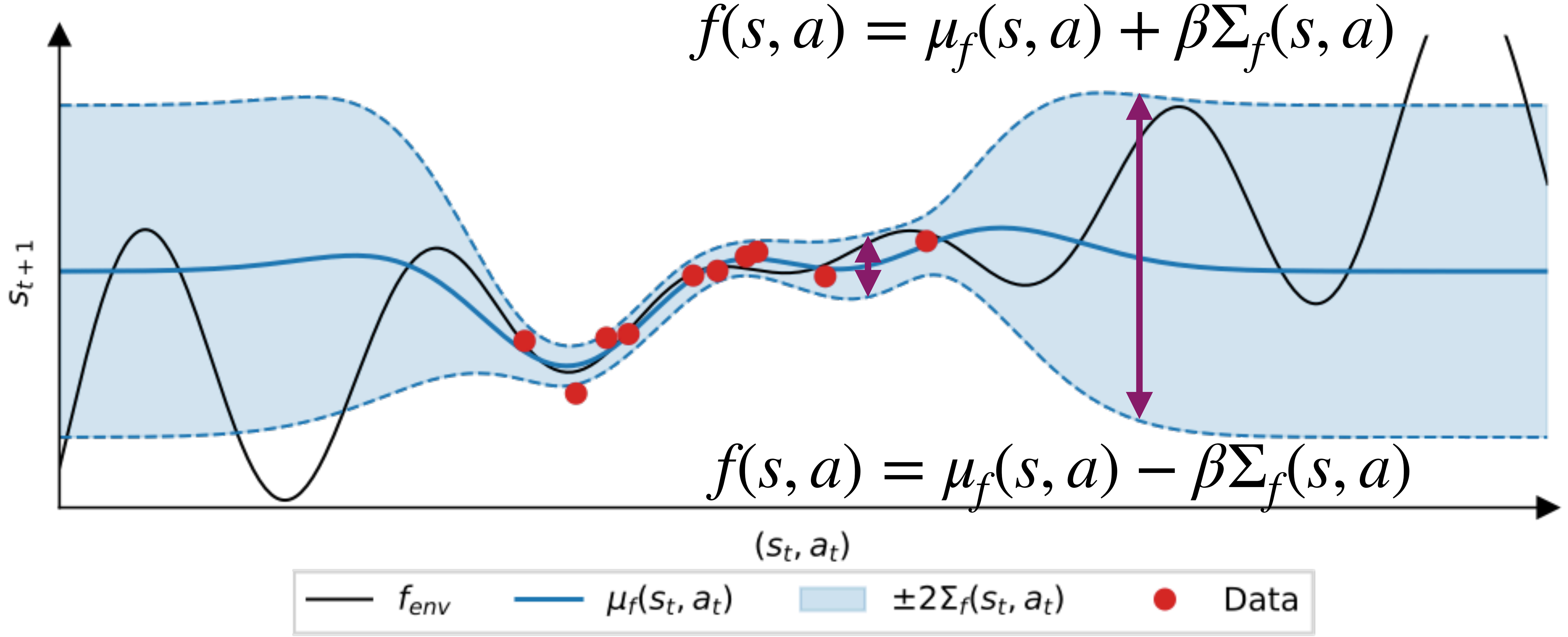
FCAI

fcai.fi

Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

$$\mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$



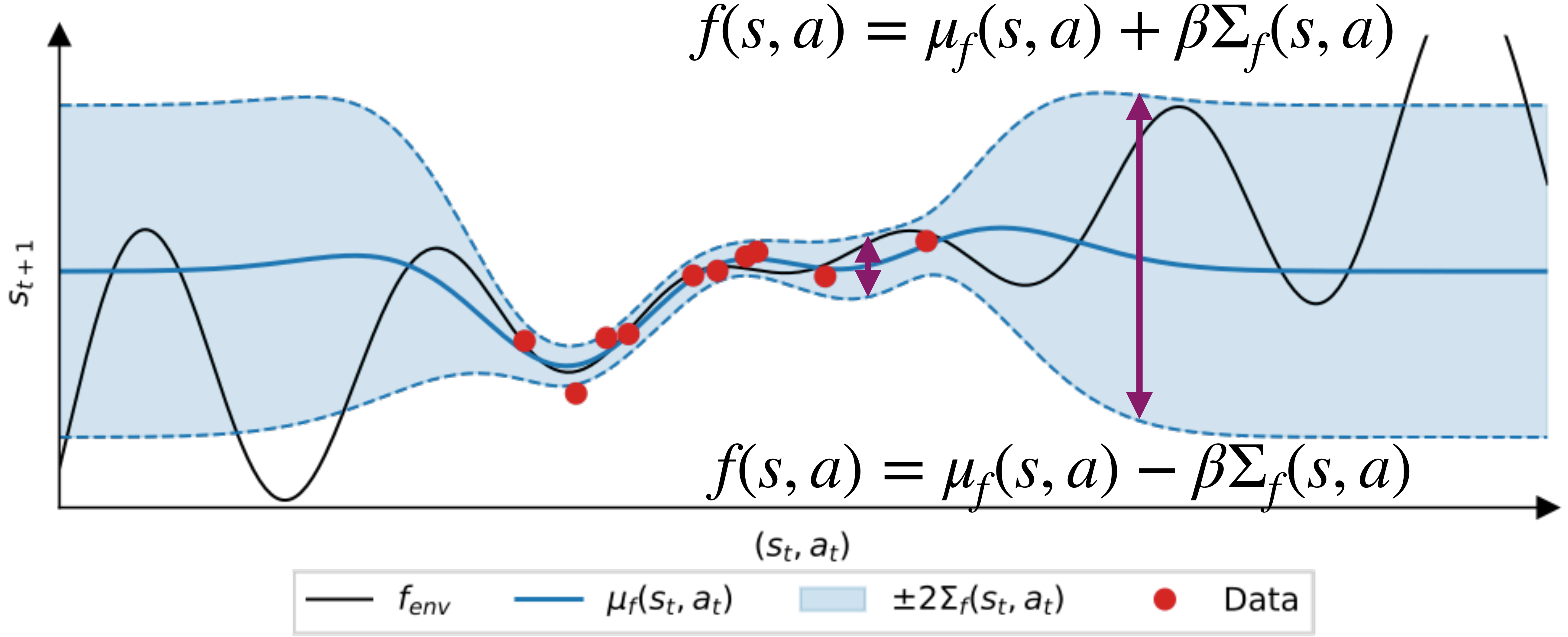
FCAI

fcai.fi

Exploration via Upper Confidence Bound

$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f)$$

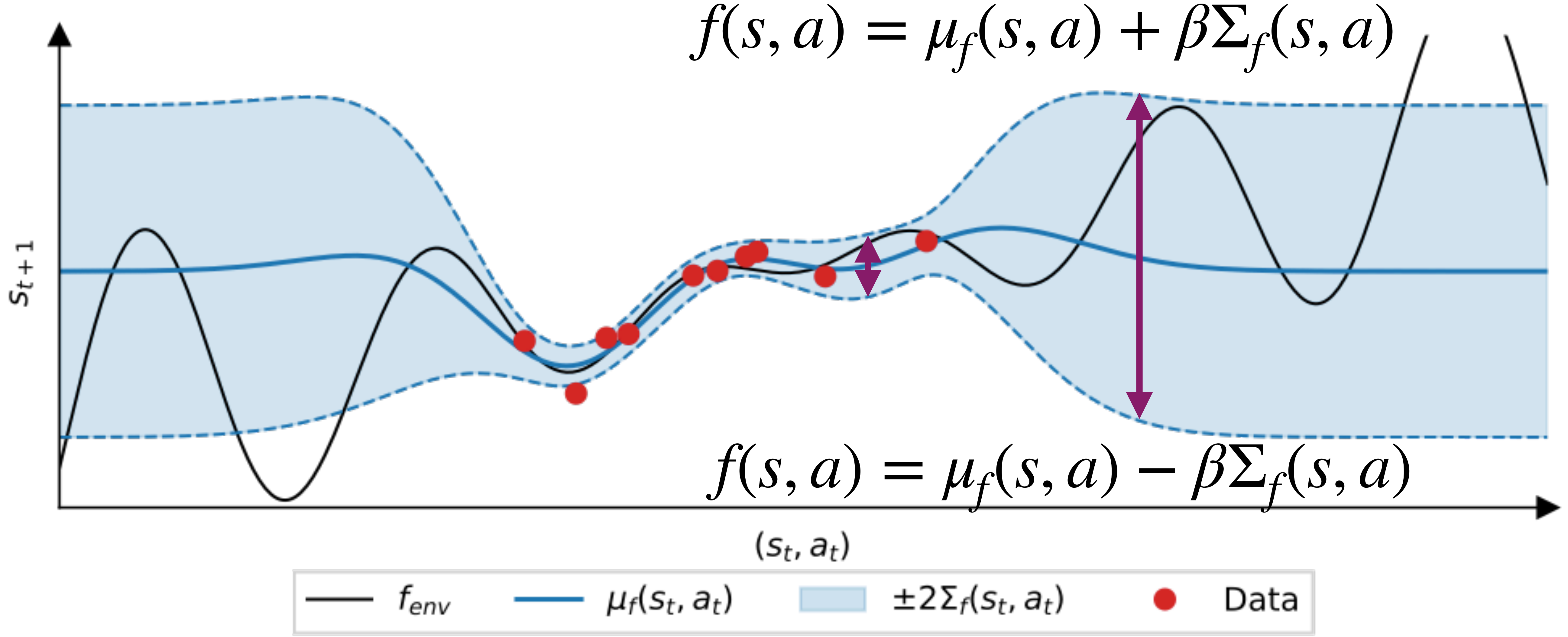
$$\mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$



FCAI Curi et al. (2020). Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning. NeurIPS. fcai.fi

Exploration via Upper Confidence Bound

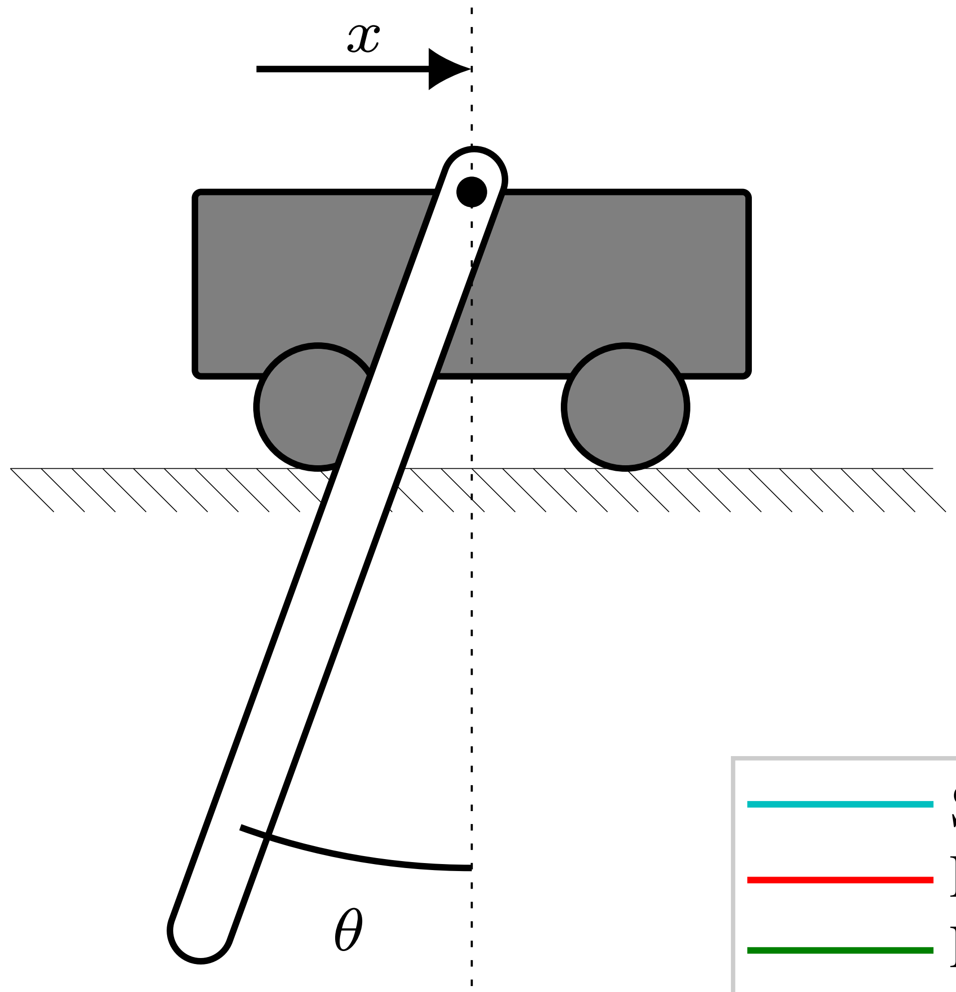
$$\pi^{UCB} = \arg\max_{\pi} \max_{f \in \mathcal{M}} J(\pi; f) \qquad \mathcal{M} = \left\{ f \mid \|f(s,a) - \mu_f(s,a)\| \leq \beta \Sigma_f(s,a) \right\}$$

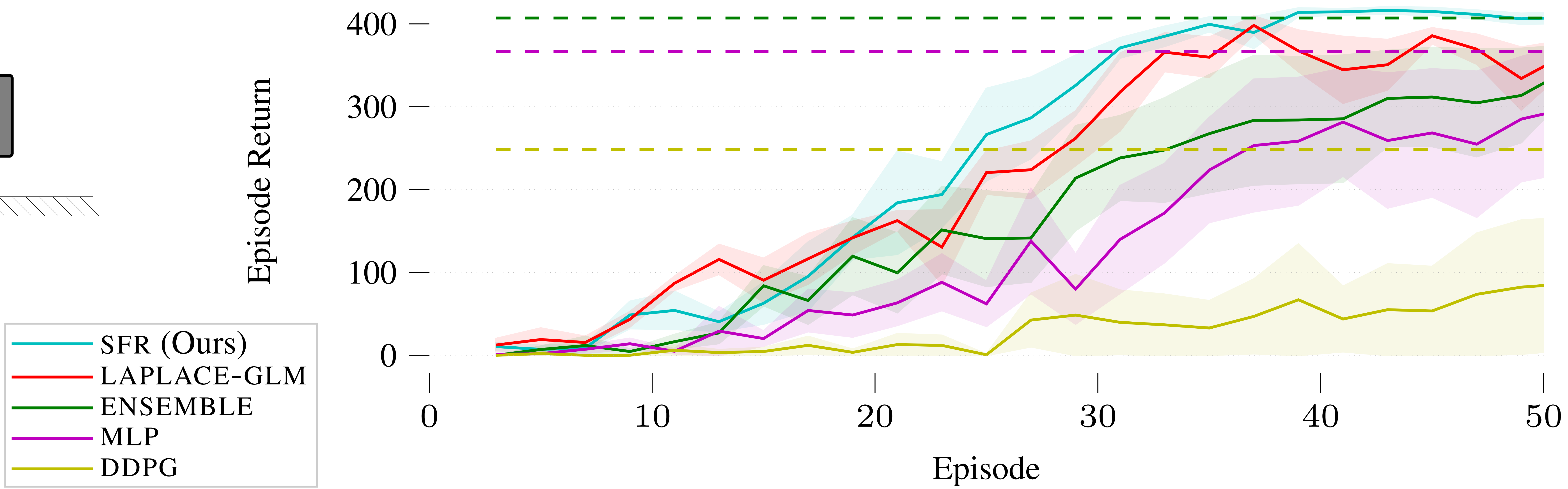


Extra hyperparamter β

 Curi et al. (2020). Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning. NeurIPS.

How to Quantify Uncertainty in Dynamics?





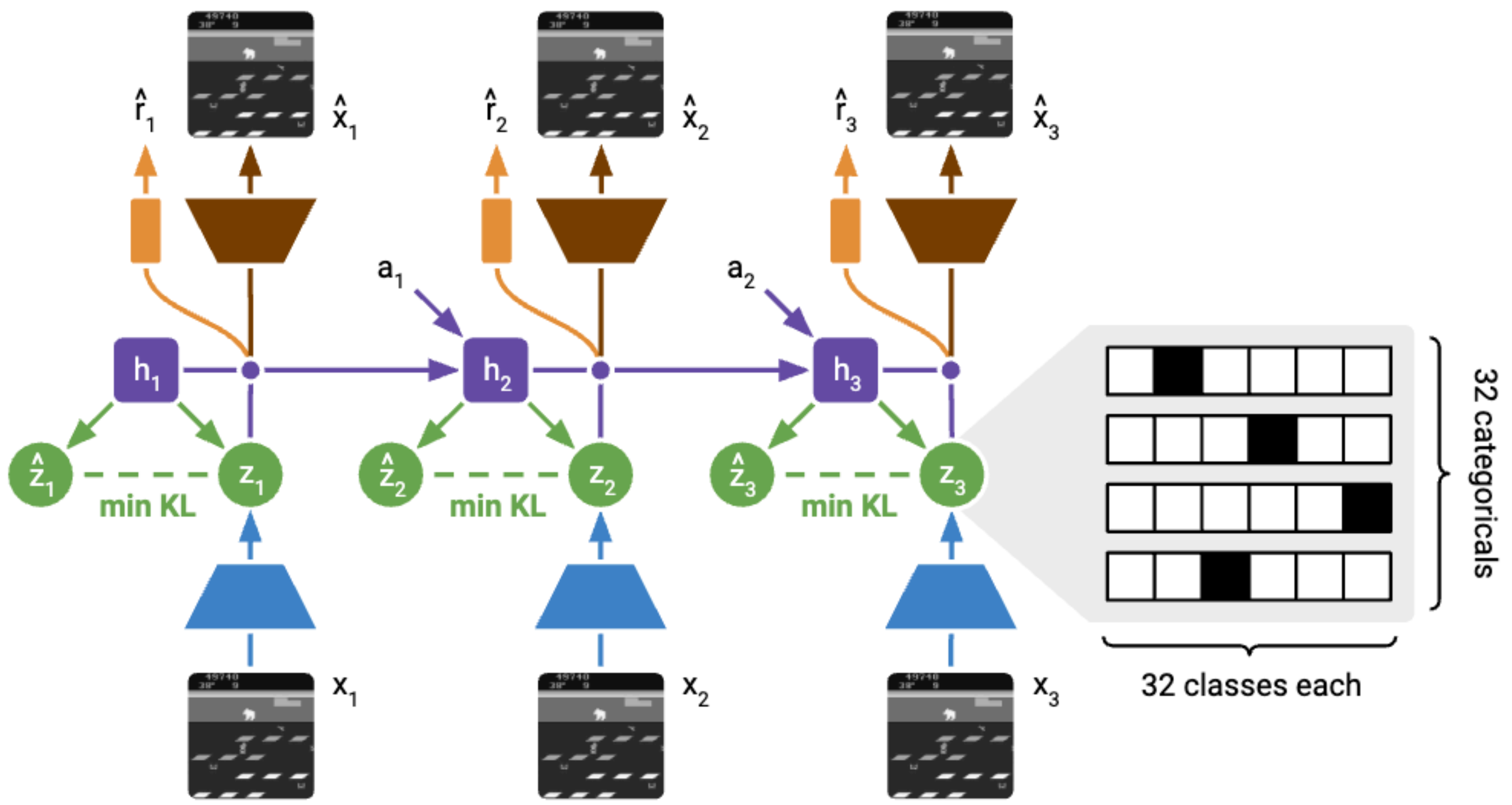
Scannell et al. (2024). Function-space Parameterisation of Neural Networks for Sequential Learning. ICLR.

Learning Objectives

Understand

1. ~~What a “model” is in model-based RL~~
2. ~~How a “model” can aid decision making~~
3. ~~Differences between background and decision-time planning~~
4. ~~Decision-time planning strategies for continuous actions~~
5. ~~Sources of uncertainty in model-based RL~~
6. ~~Rationale and insights for decision-making under uncertainty~~

Case Study: Dreamer (v2/v3)

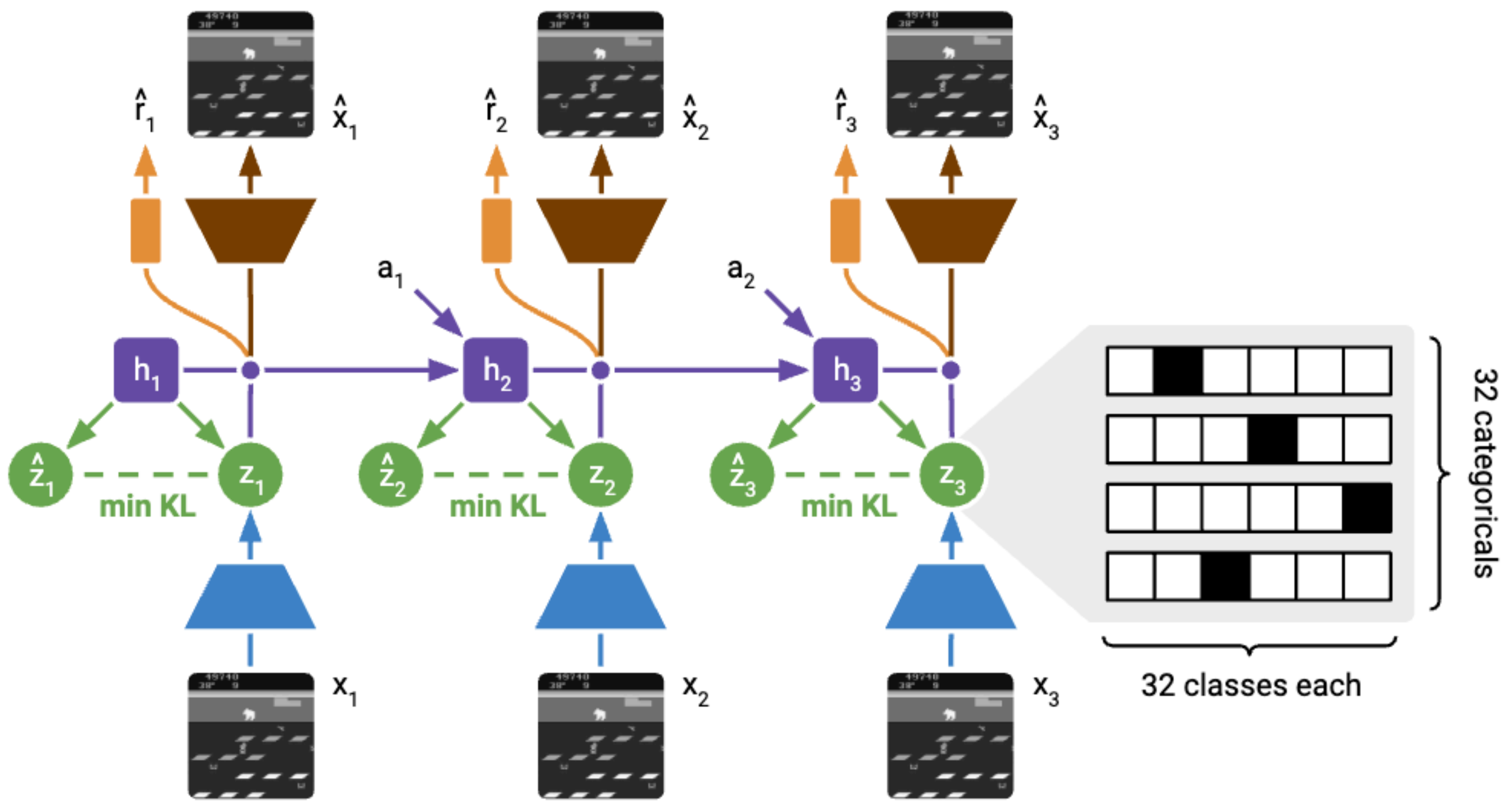


World Model Learning

FCAI

Hafner et al. (2021). Mastering Atari with Discrete World Models. ICLR.

Case Study: Dreamer (v2/v3)



World Model Learning

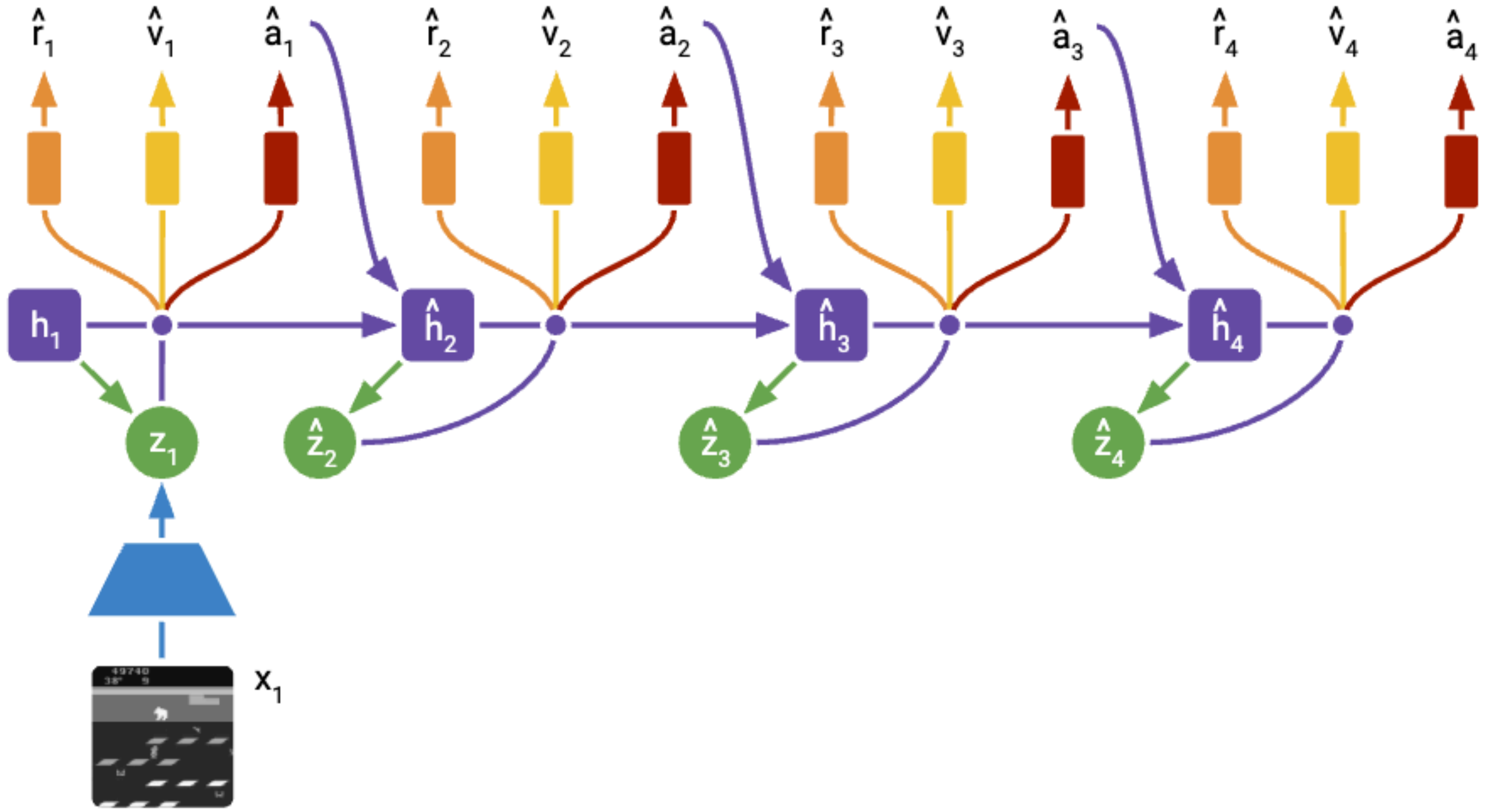
Latent dynamics with encoder/decoder

FCAI

Hafner et al. (2021). Mastering Atari with Discrete World Models. ICLR.

fcai.fi

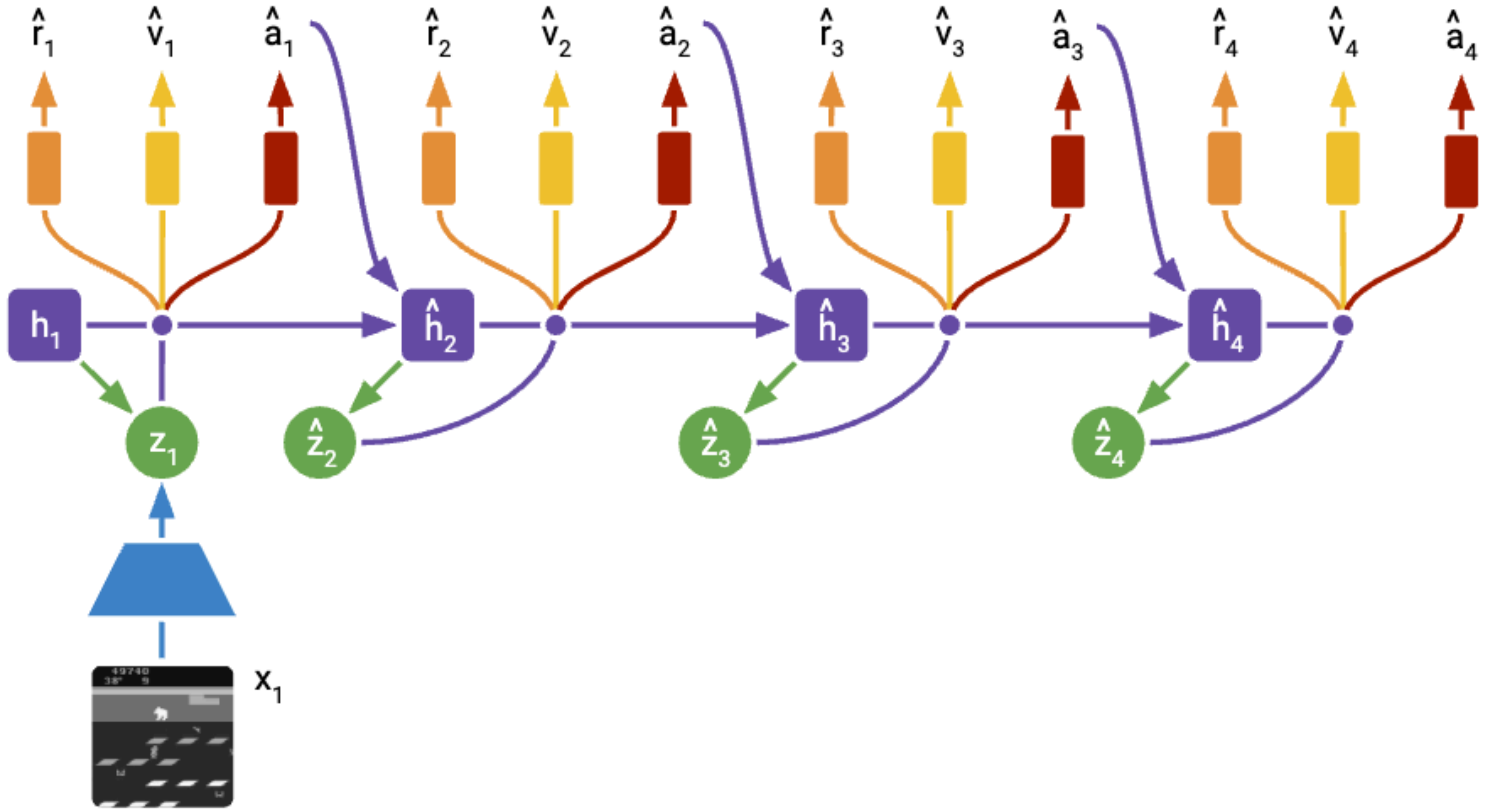
Case Study: Dreamer (v2/v3)



Actor Critic Learning

Latent dynamics with encoder/decoder

Case Study: Dreamer (v2/v3)



Actor Critic Learning

Latent dynamics with encoder/decoder

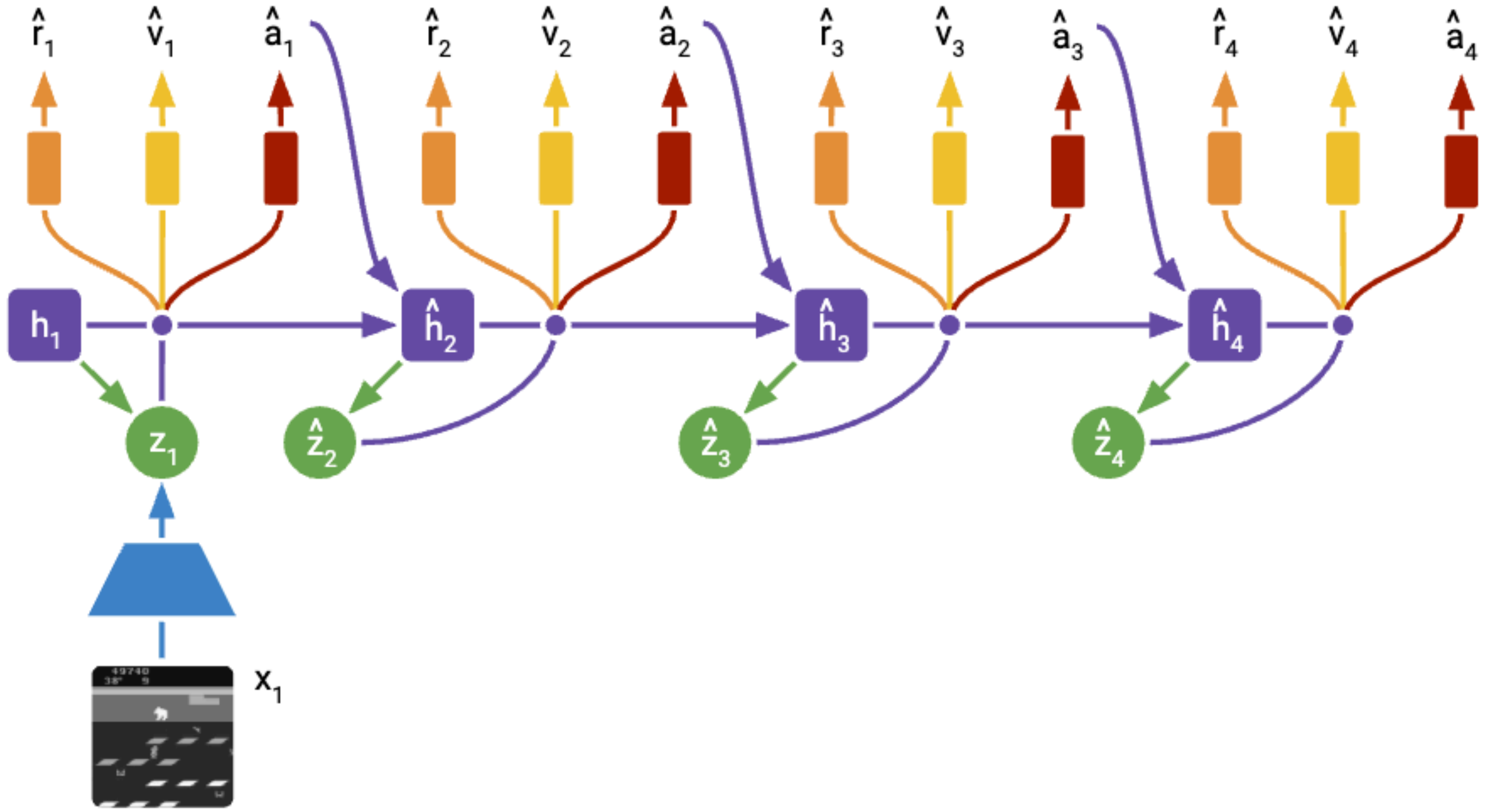
Actor $\pi_\theta(z)$ & critic $Q_\theta(z, a)$ in latent space

FCAI

Hafner et al. (2021). Mastering Atari with Discrete World Models. ICLR.

fcai.fi

Case Study: Dreamer (v2/v3)



Actor Critic Learning

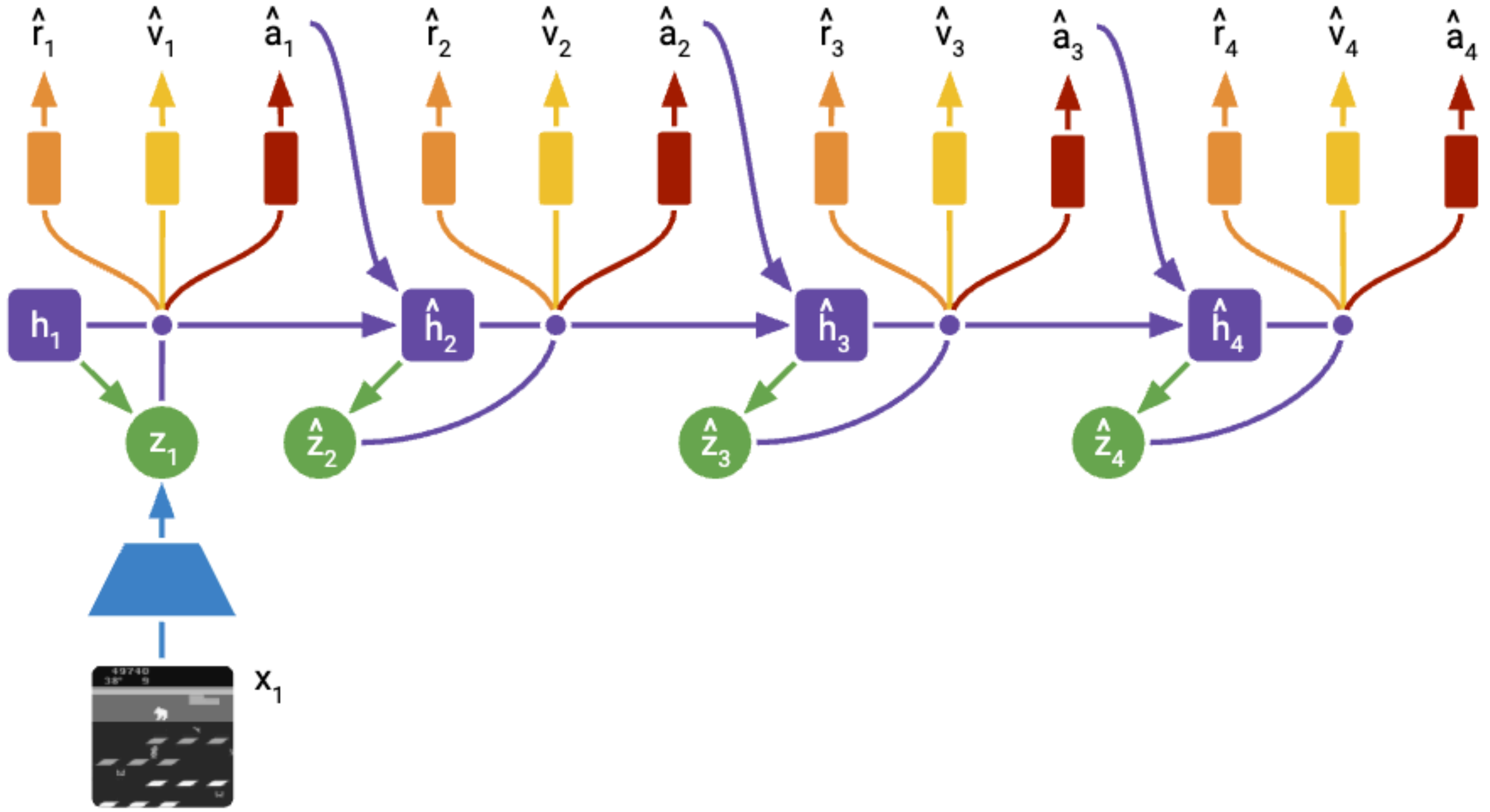
Latent dynamics with encoder/decoder

Actor $\pi_\theta(z)$ & critic $Q_\theta(z, a)$ in latent space

Actor/critic leverage "imagined" outcomes

Hafner et al. (2021). Mastering Atari with Discrete World Models. ICLR.

Case Study: Dreamer (v2/v3)



Actor Critic Learning

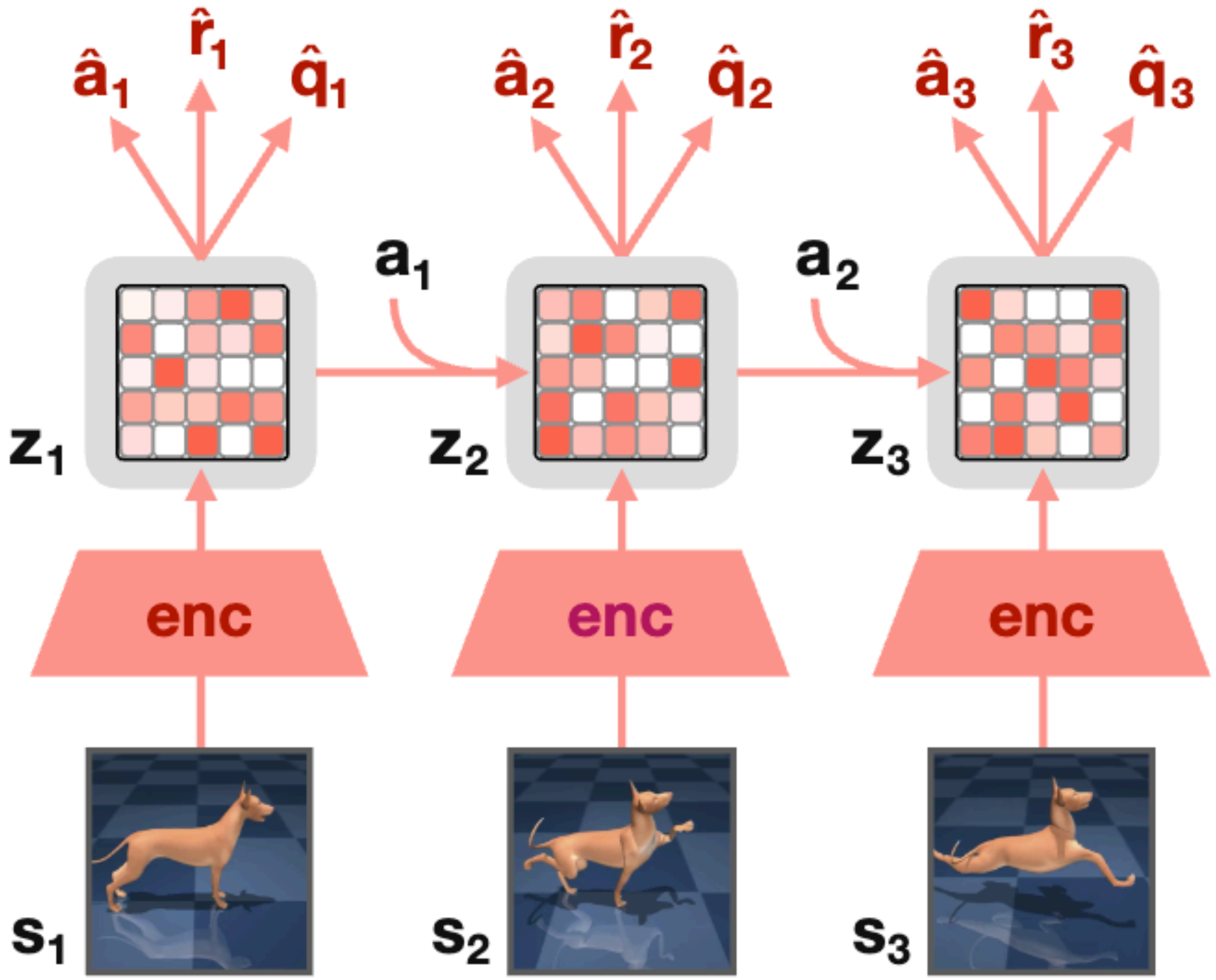
Latent dynamics with encoder/decoder

Actor $\pi_\theta(z)$ & critic $Q_\theta(z, a)$ in latent space

Actor/critic leverage “imagined” outcomes

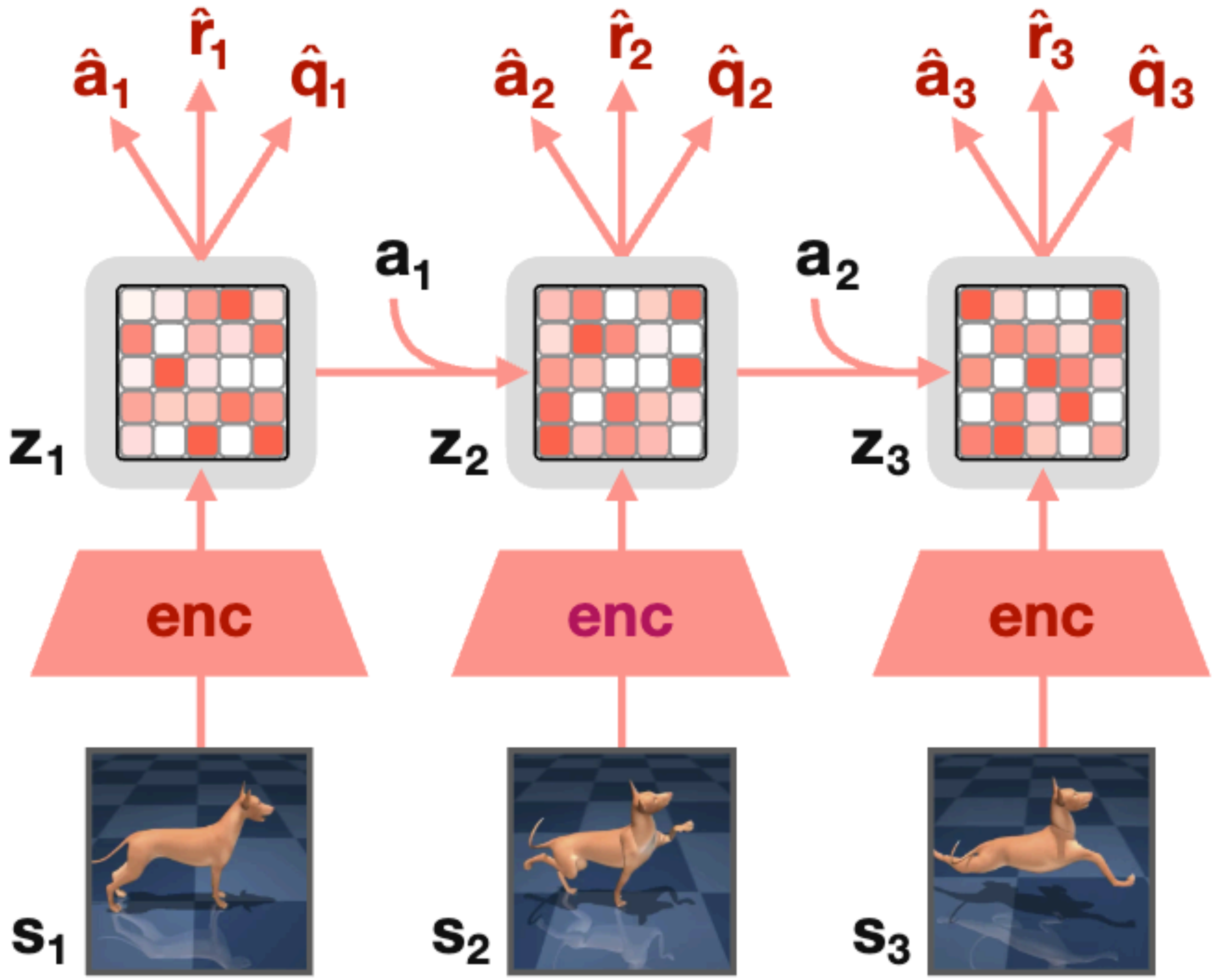
Background planning

Case Study: TD-MPC2



Hansen et al. (2024). TD-MPC2: Robust, Scalable World Models for Continuous Control. ICLR.

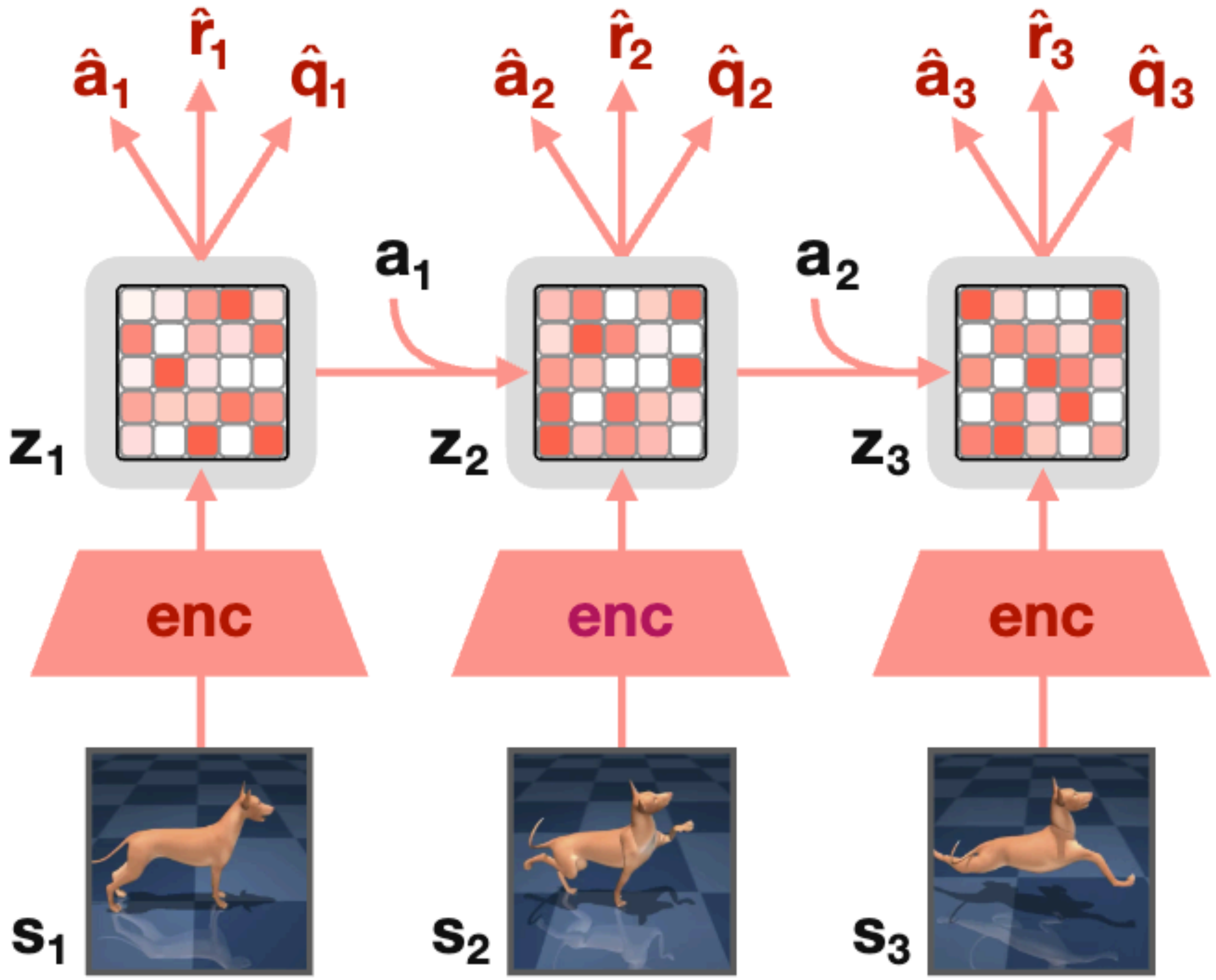
Case Study: TD-MPC2



Latent dynamics

Hansen et al. (2024). TD-MPC2: Robust, Scalable World Models for Continuous Control. ICLR.

Case Study: TD-MPC2

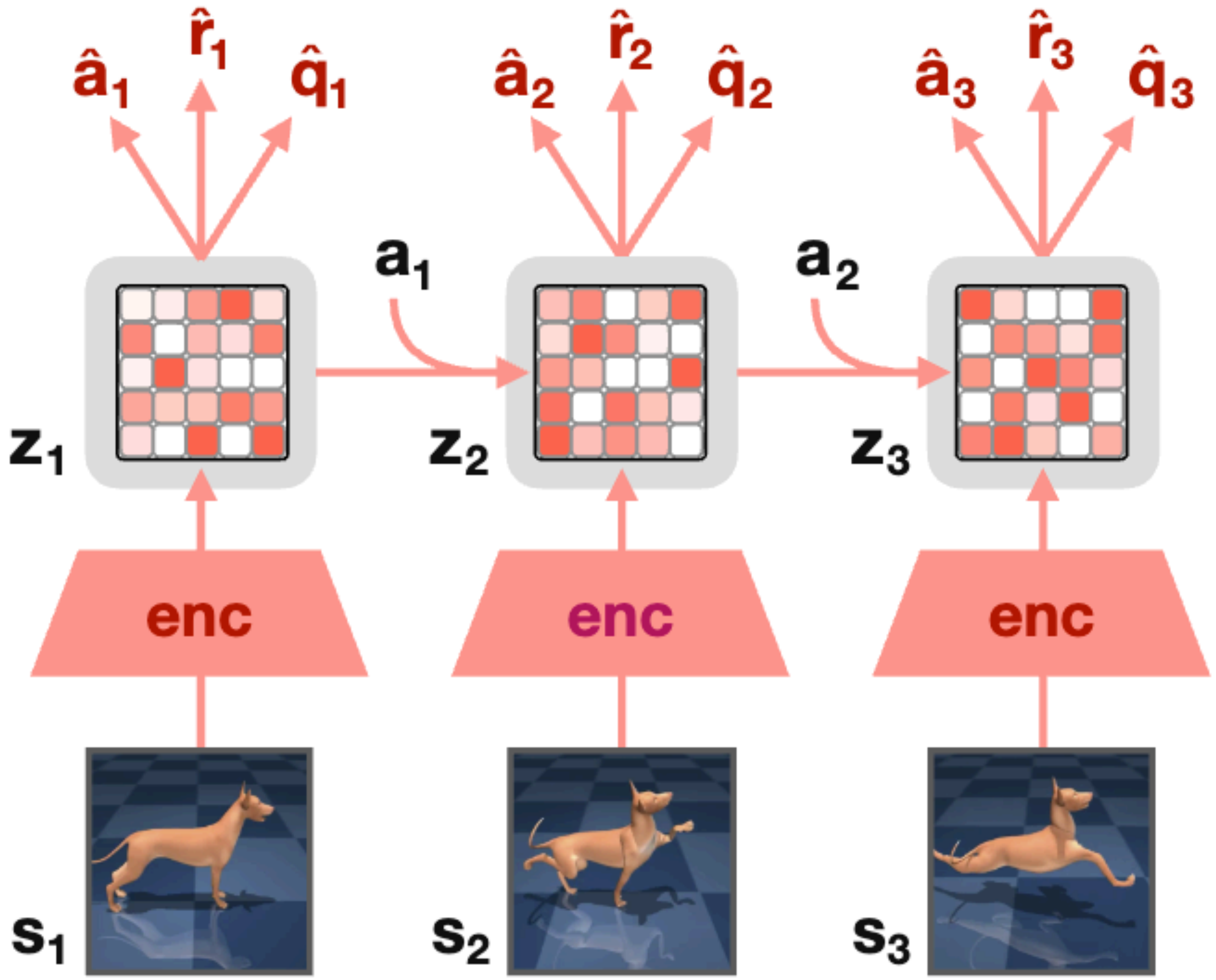


Latent dynamics

No decoder

Hansen et al. (2024). TD-MPC2: Robust, Scalable World Models for Continuous Control. ICLR.

Case Study: TD-MPC2



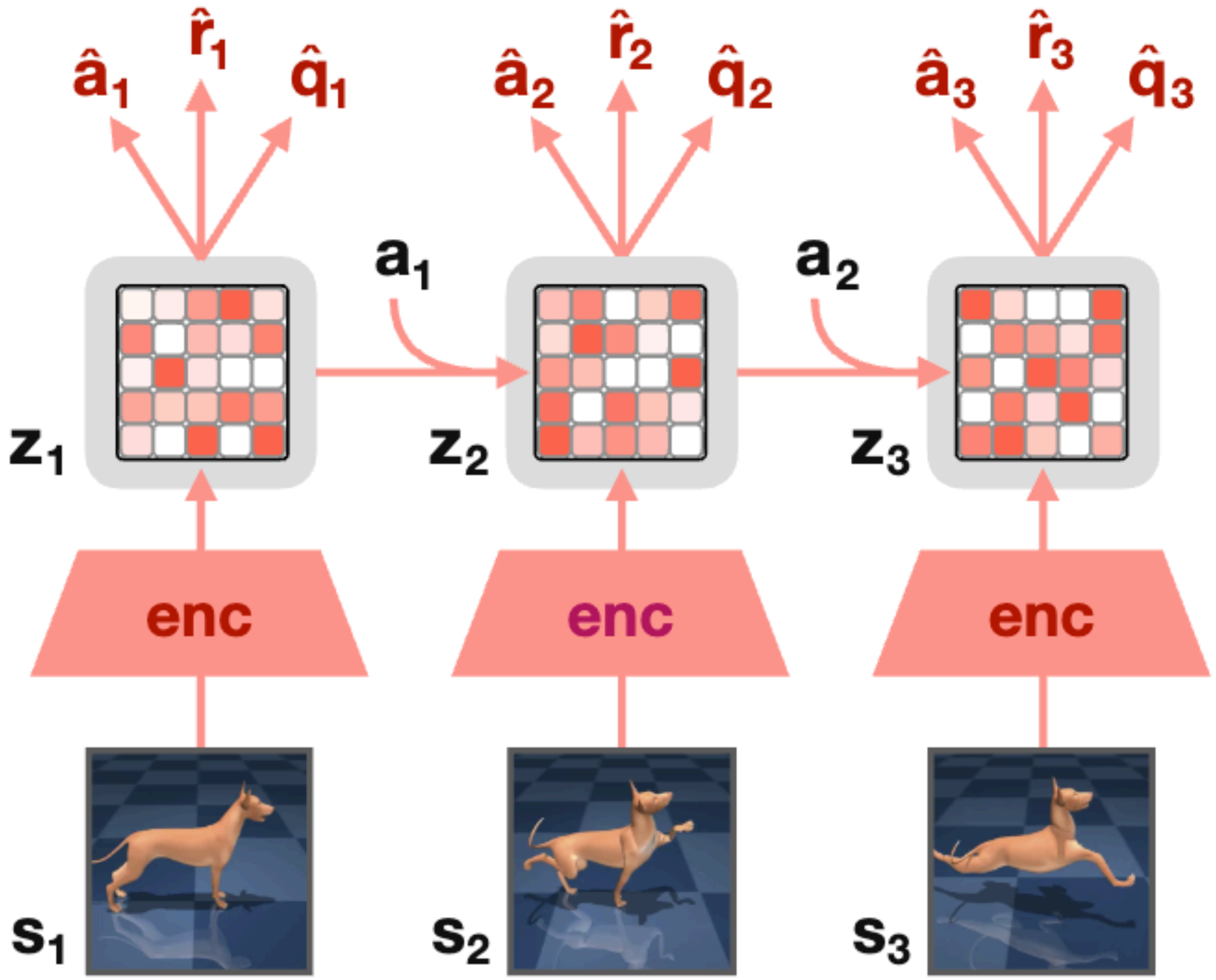
Latent dynamics

No decoder

Actor $\pi_\theta(z)$ & critic $Q_\theta(z, a)$ in latent space

Hansen et al. (2024). TD-MPC2: Robust, Scalable World Models for Continuous Control. ICLR.

Case Study: TD-MPC2



Latent dynamics

No decoder

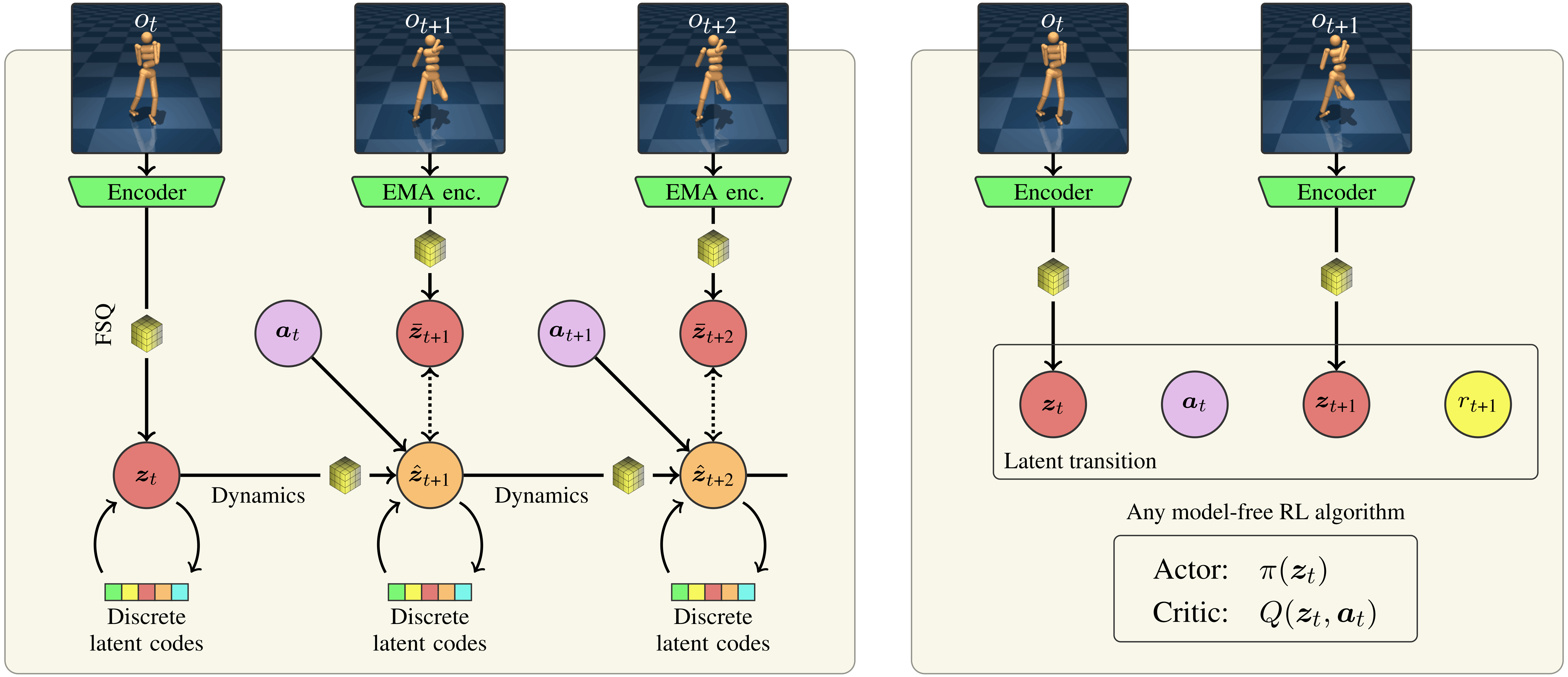
Actor $\pi_\theta(z)$ & critic $Q_\theta(z, a)$ in latent space

Decision-time planning

Hansen et al. (2024). TD-MPC2: Robust, Scalable World Models for Continuous Control. ICLR.

Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔



FCAI

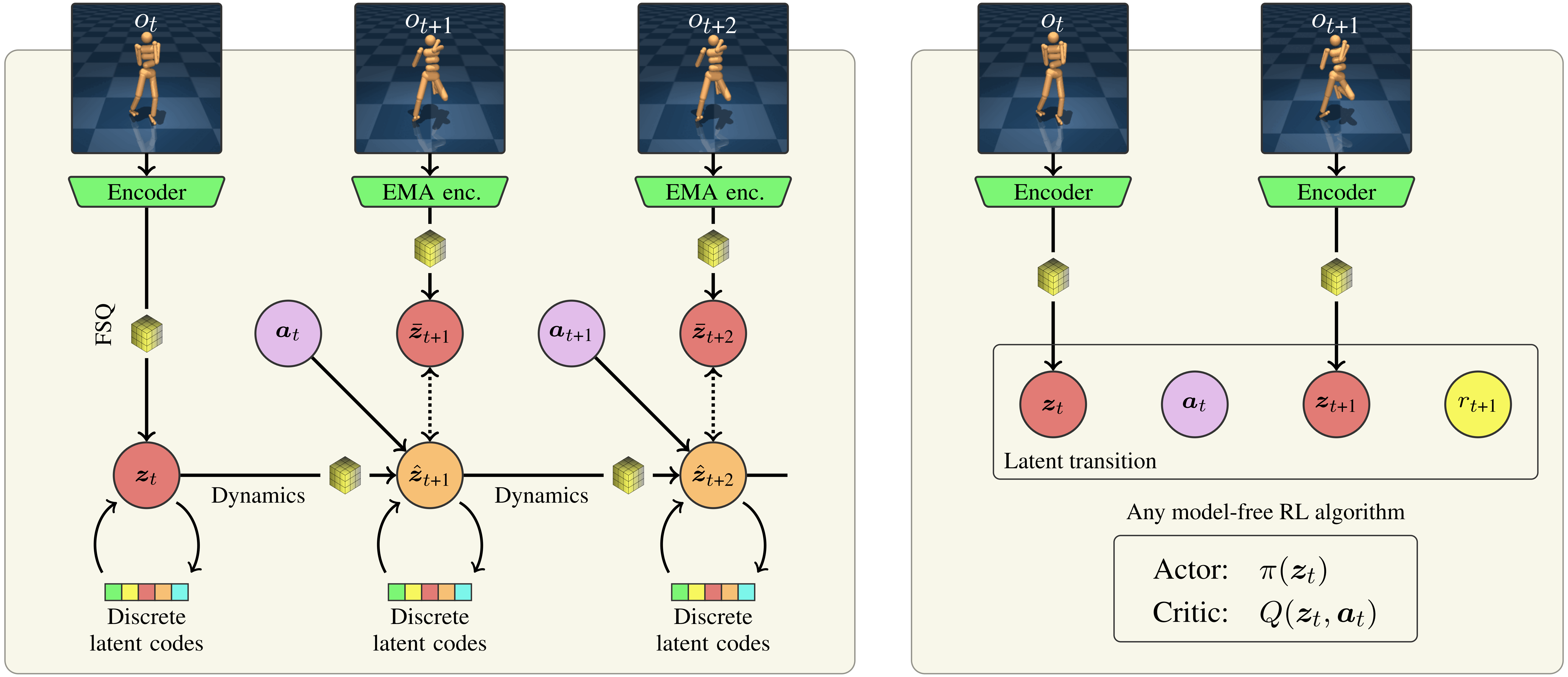
Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

fcai.fi

Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning



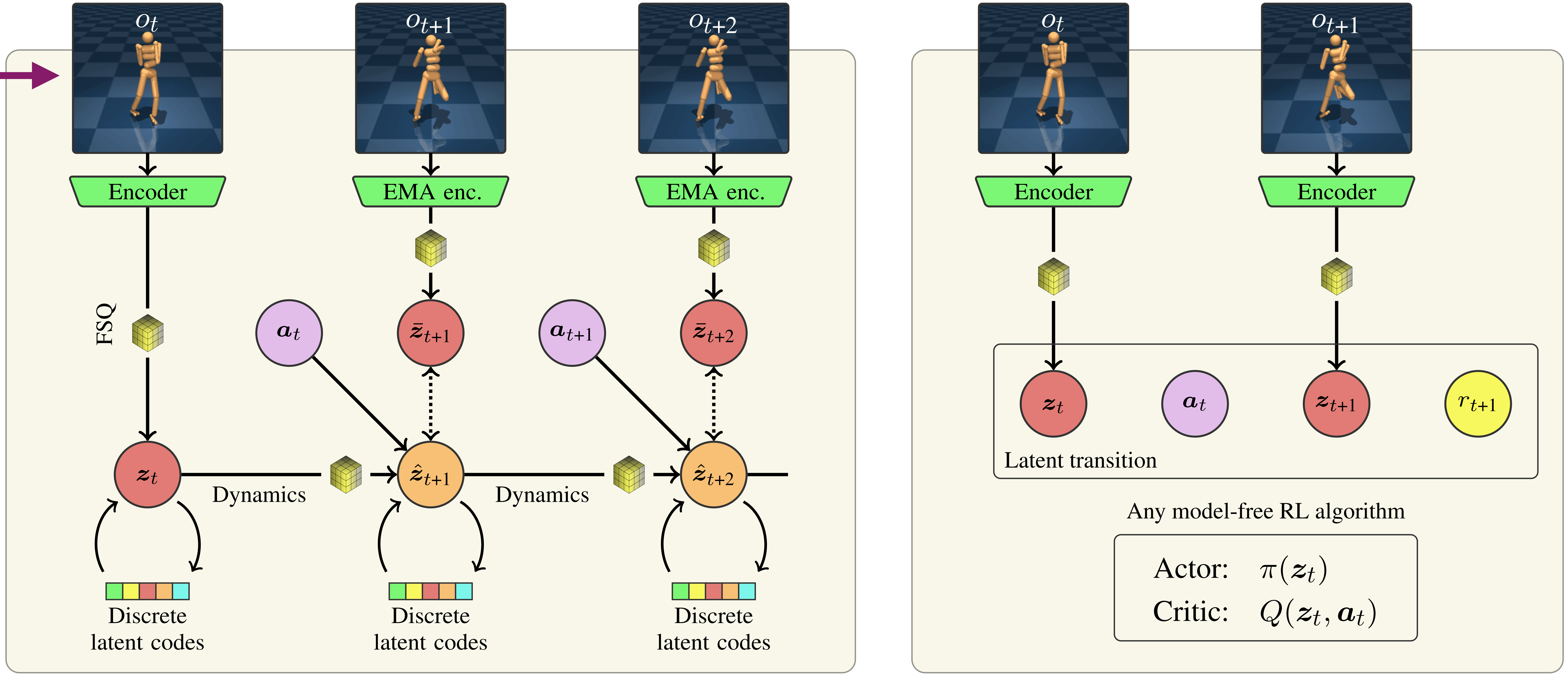
Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning

Observation



FCAI

Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

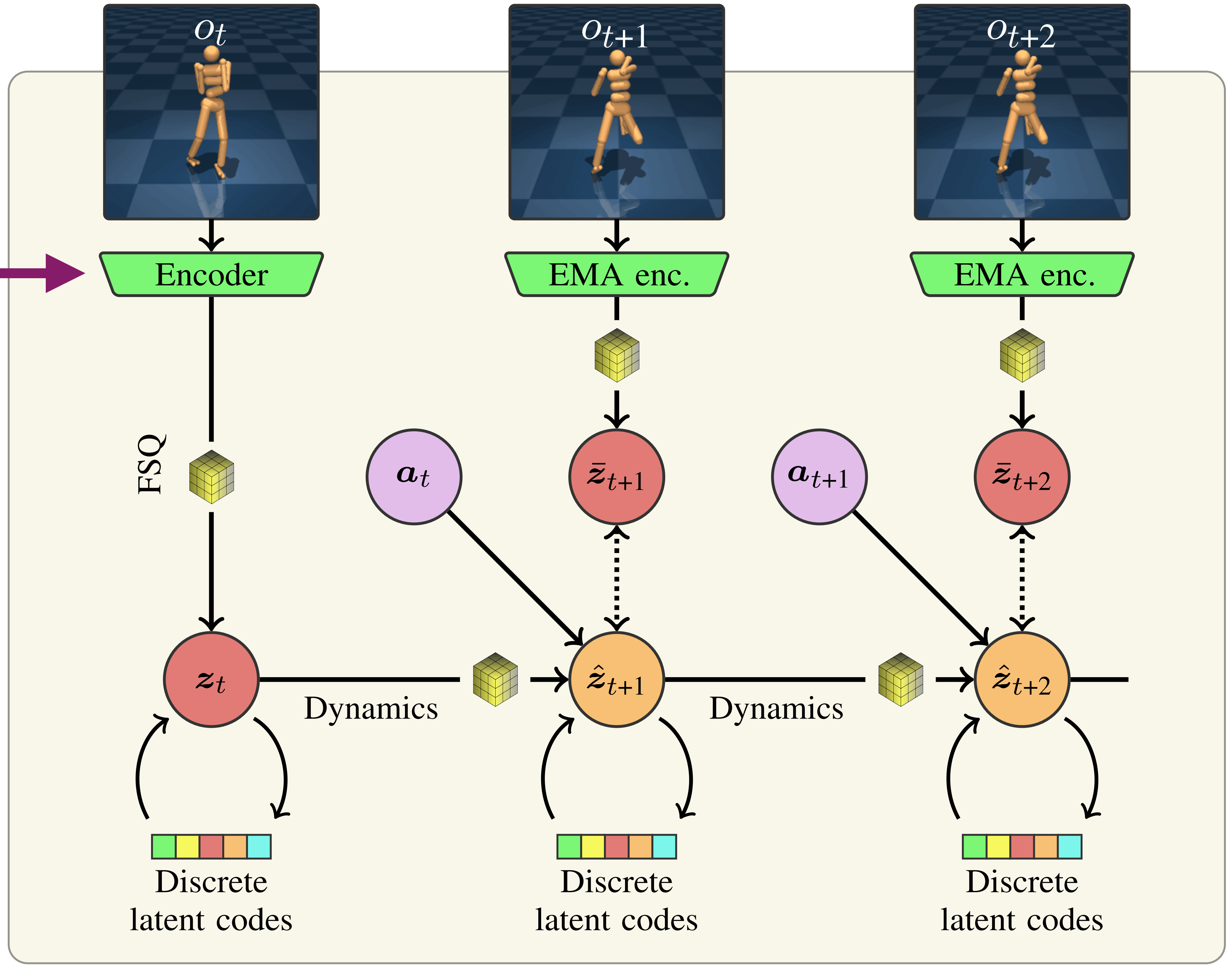
fcai.fi

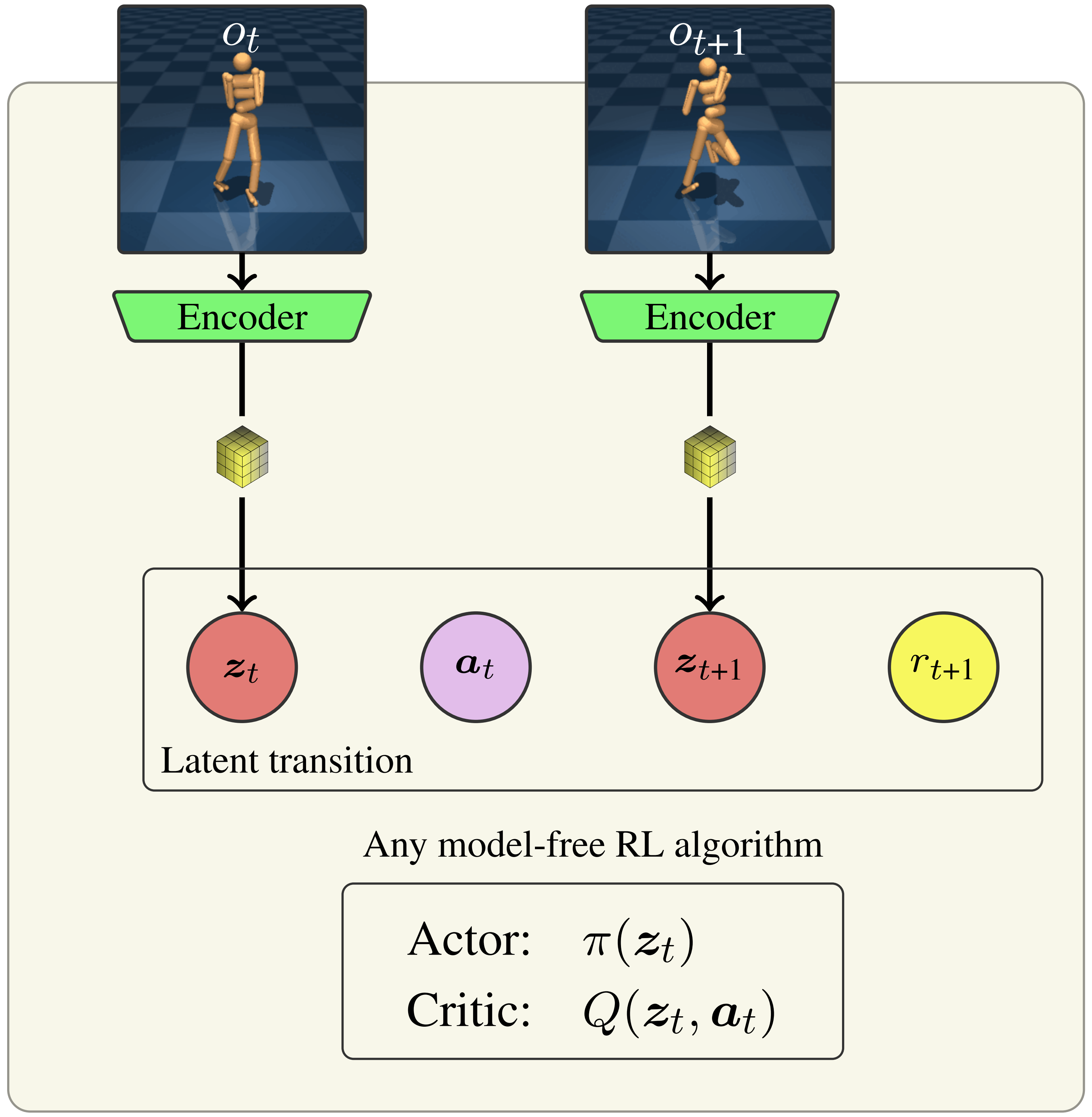
Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning

Encoder





FCAI

Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

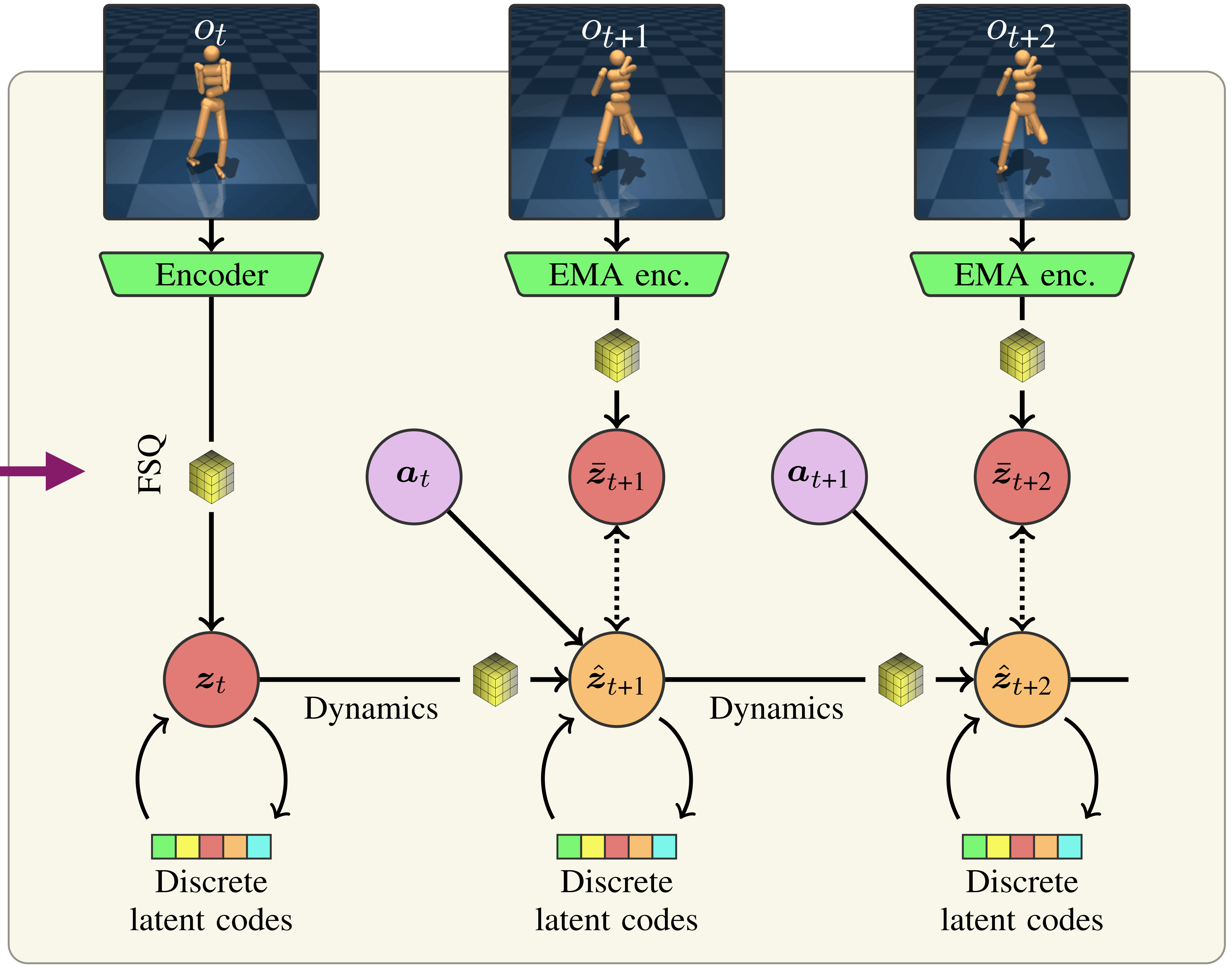
fcai.fi

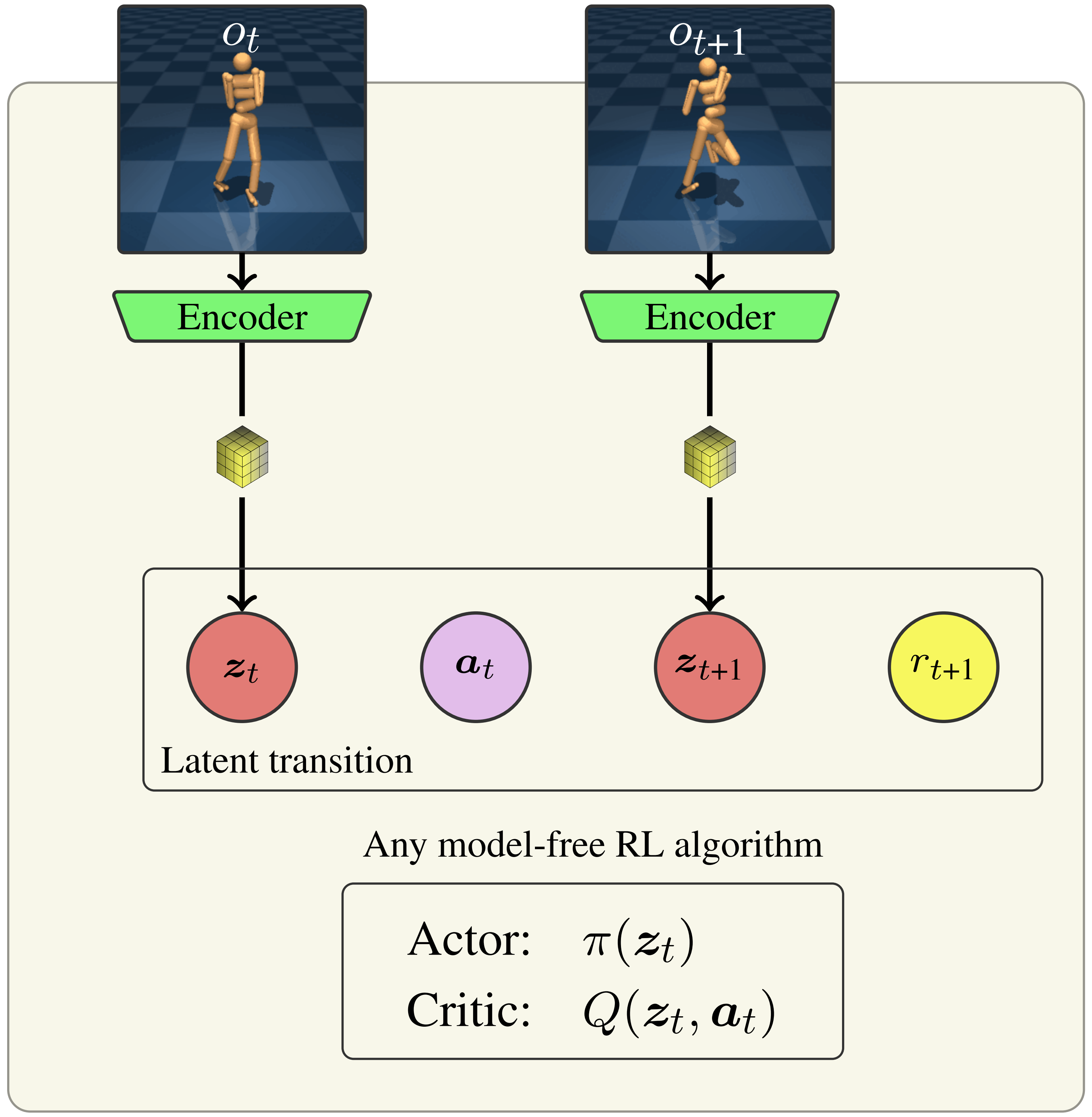
Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning

Quantization





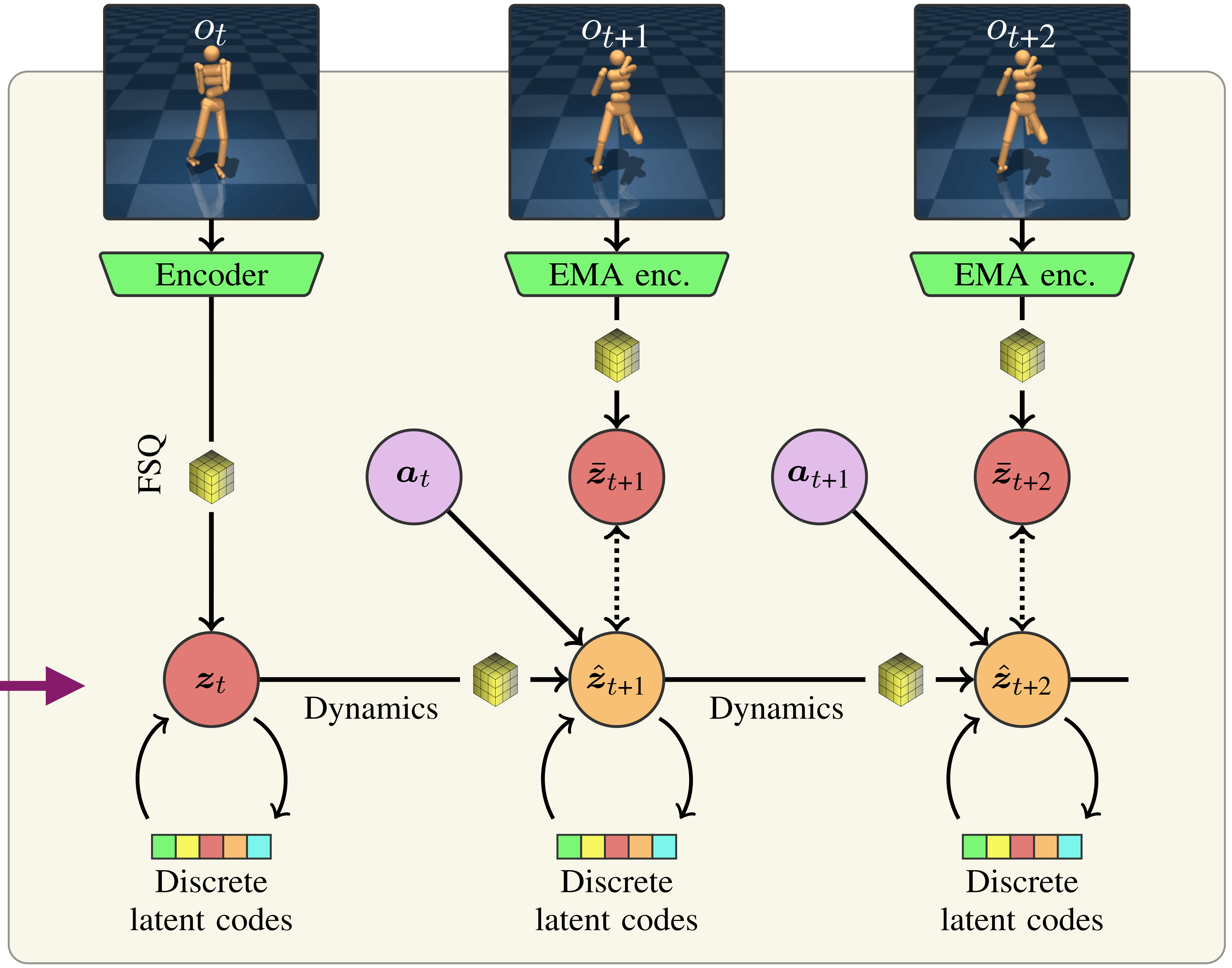
Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

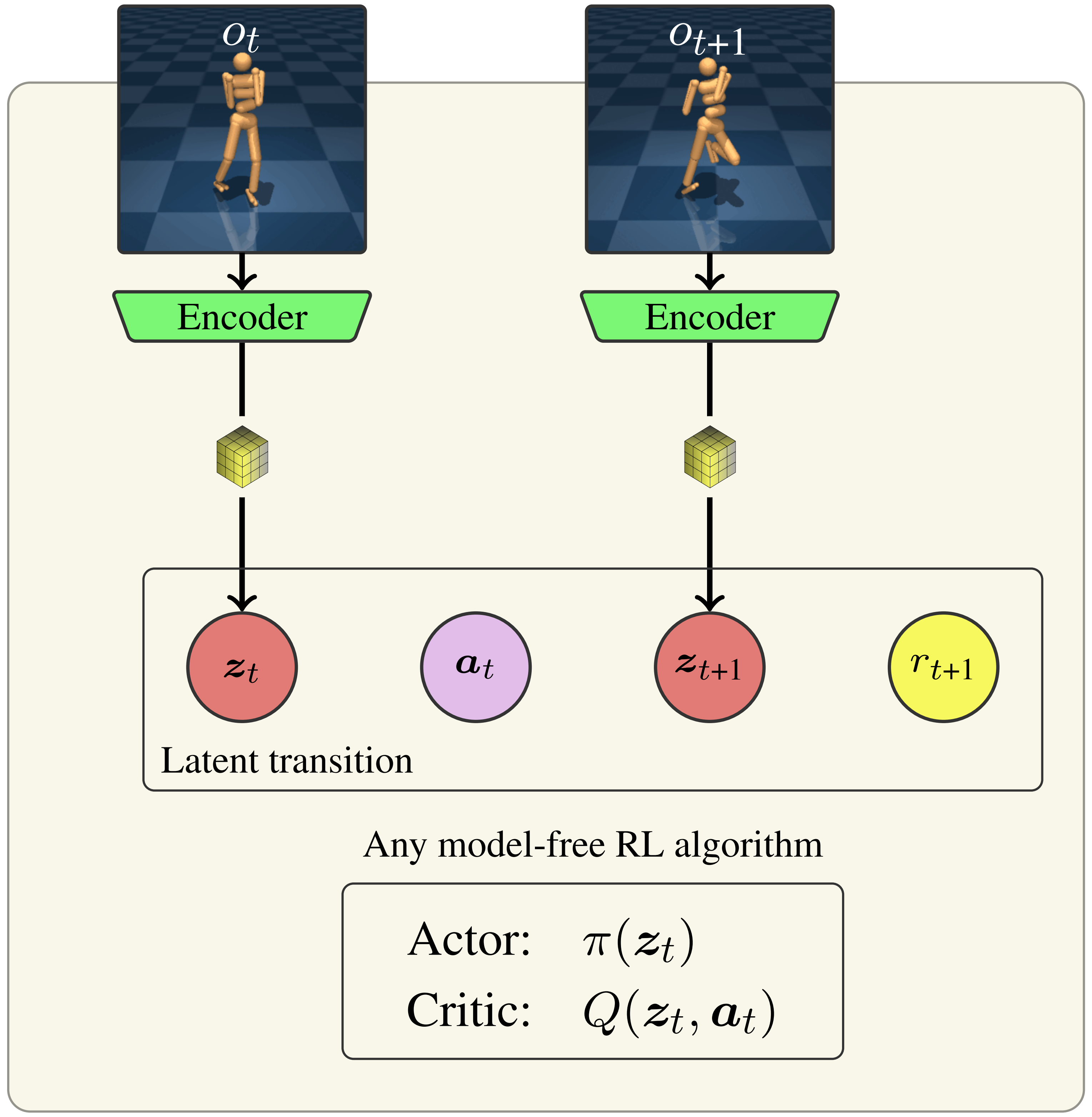
Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning

Latent dynamics



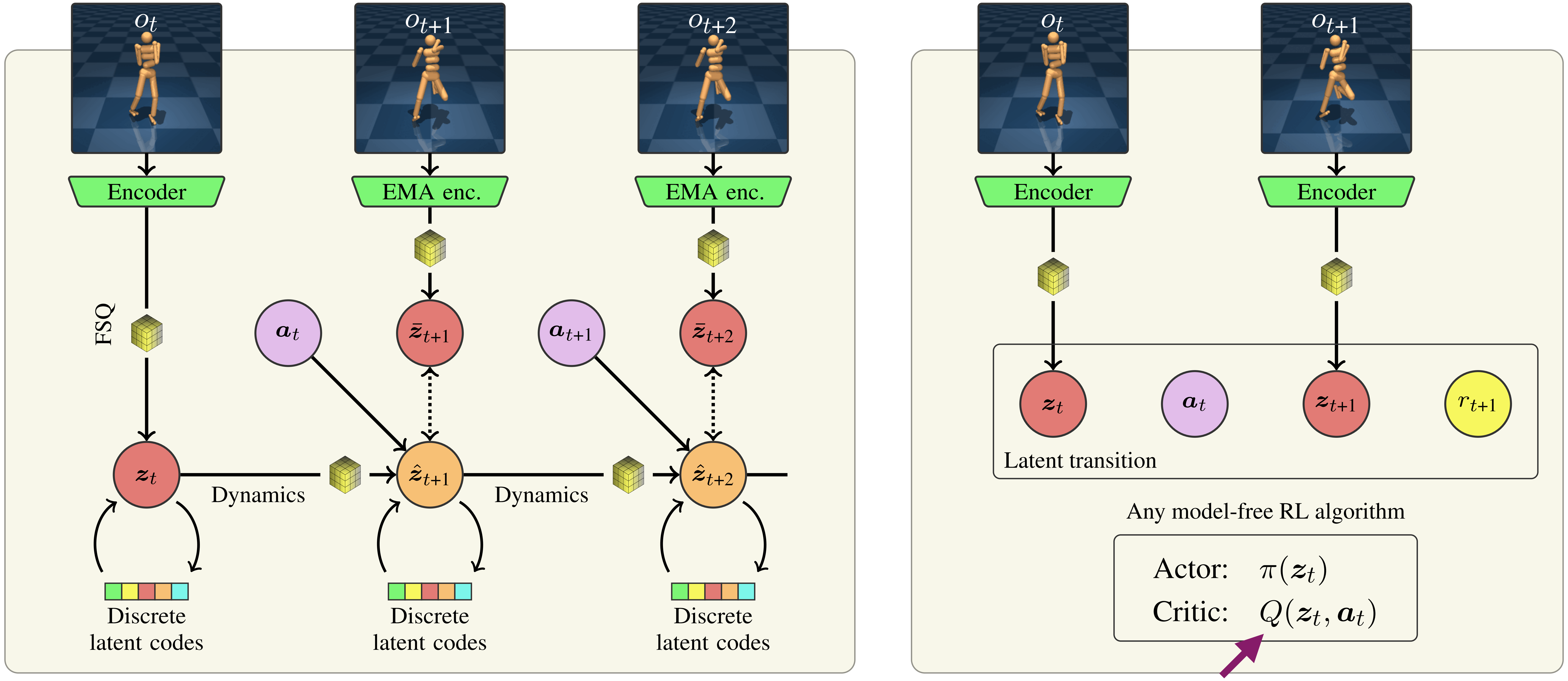


Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

Case Study: iQRL

Dynamics Model but Not Model-based RL? 🤔

Use dynamics for representation learning



Model-free RL in latent space

Scannell et al. (2024). iQRL - Implicitly Quantized Representations for Sample-efficient RL. arXiv:2406.02696.

Outlook

Outlook

Email: aidan.scannell@aalto.fi

Website: www.aidanscannell.com

FCAI

fcai.fi

Outlook

1. Model-based RL is a powerful tool

Email: aidan.scannell@aalto.fi

Website: www.aidanscannell.com

FCAI

fcai.fi

Outlook

1. Model-based RL is a powerful tool
2. Leveraging predictive models improves sample efficiency

Email: aidan.scannell@aalto.fi

Website: www.aidanscannell.com

FCAI

fcai.fi

Outlook

1. Model-based RL is a powerful tool
2. Leveraging predictive models improves sample efficiency
3. Lots more exciting work to be done

Email: aidan.scannell@aalto.fi

Website: www.aidanscannell.com

FCAI

fcai.fi