Sample-efficient Reinforcement Learning with Implicitly Quantized Representations

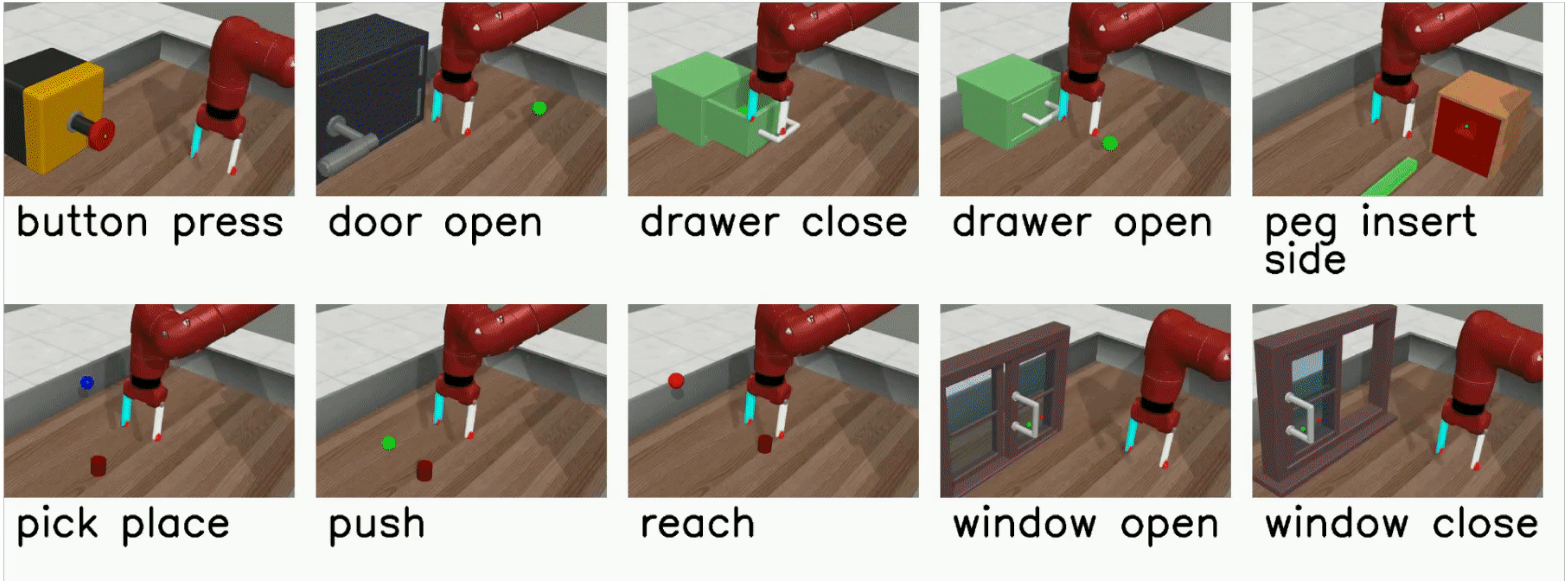
Aidan Scannell, Mohammadreza Nakhaei, Kalle Kujanpää, Yi Zhao,
Kevin Luck, Arno Solin, Joni Pajarinen

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

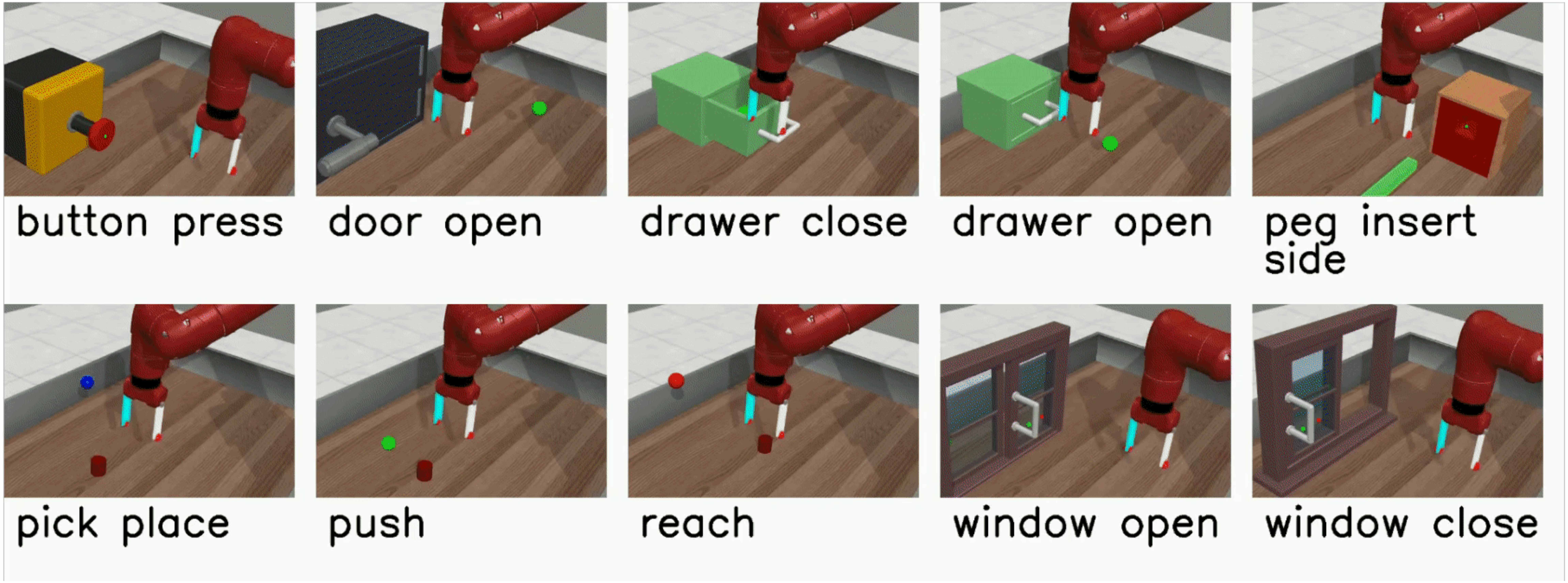
FCAI

fcai.fi

Motivation: Robotic Manipulation



Motivation: Robotic Manipulation

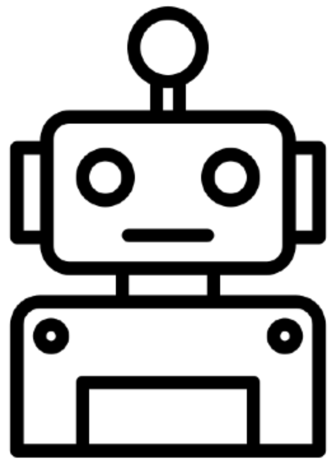


Reinforcement Learning (RL)

Markov Decision Process (MDP)

Reinforcement Learning (RL)

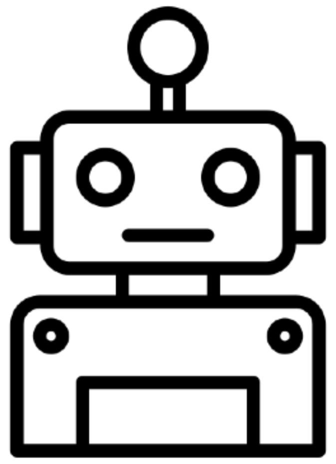
Markov Decision Process (MDP)



Reinforcement Learning (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

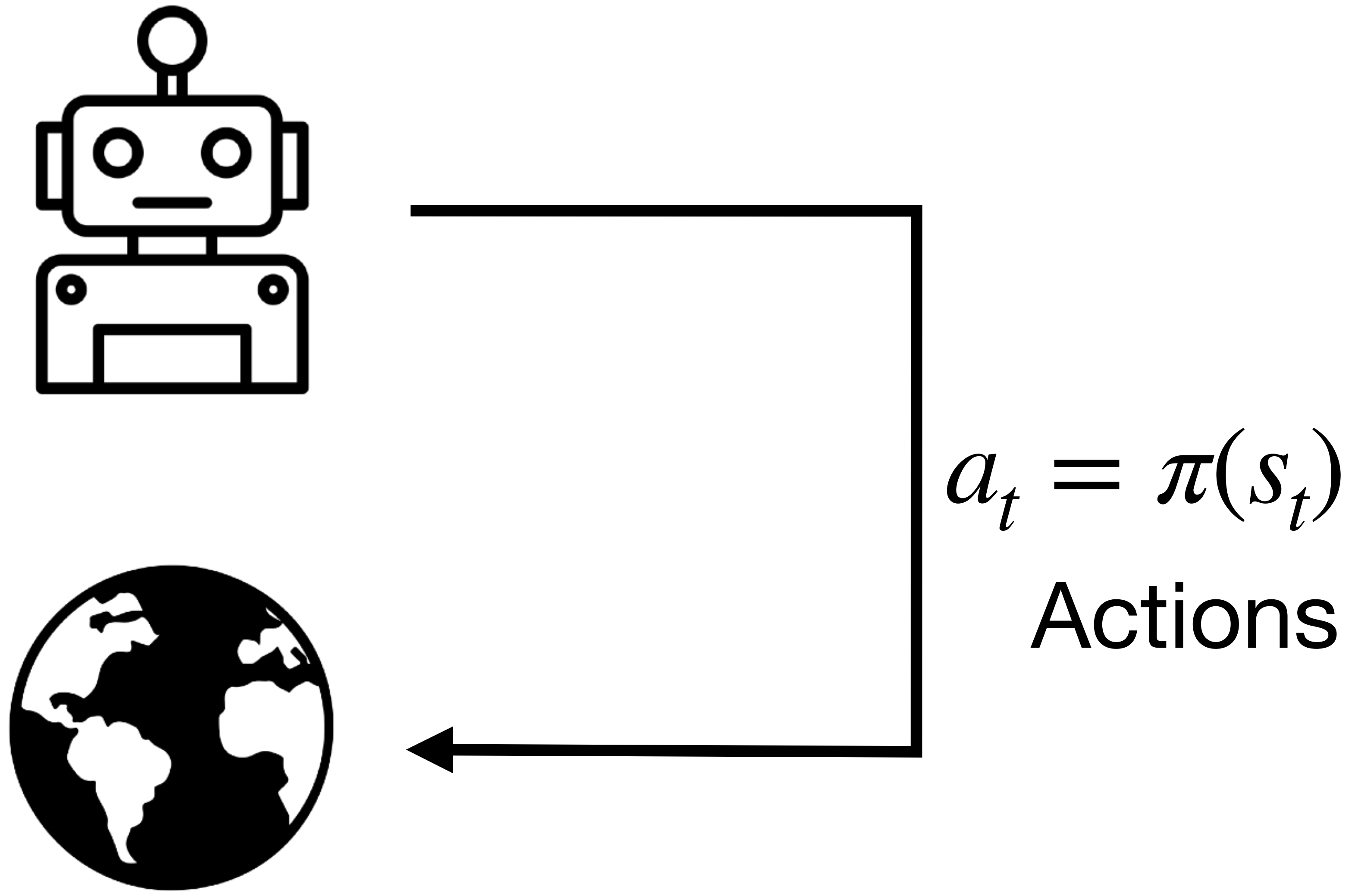


Reinforcement Learning (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$



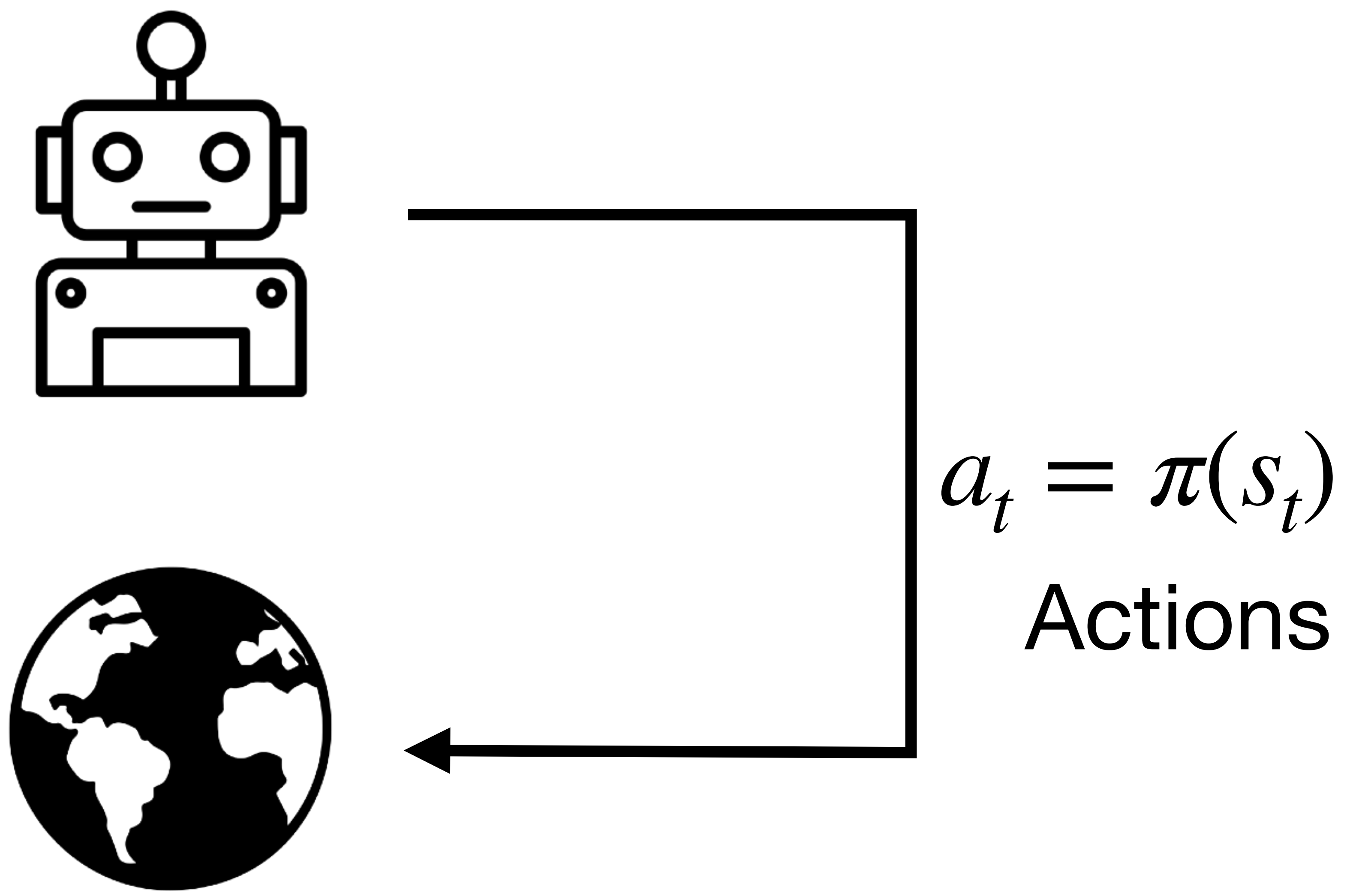
Reinforcement Learning (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

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



Reinforcement Learning (RL)

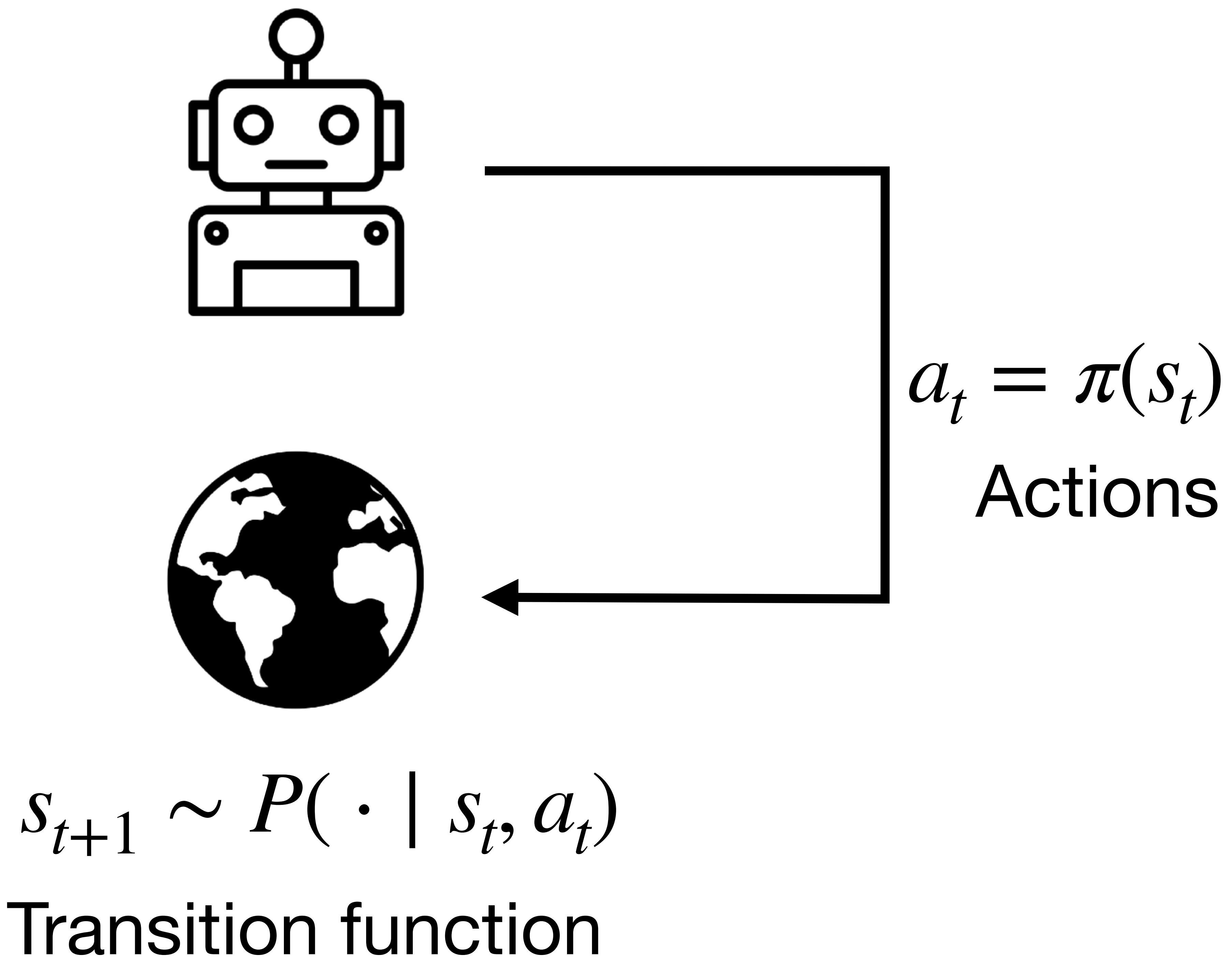
Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

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

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



Reinforcement Learning (RL)

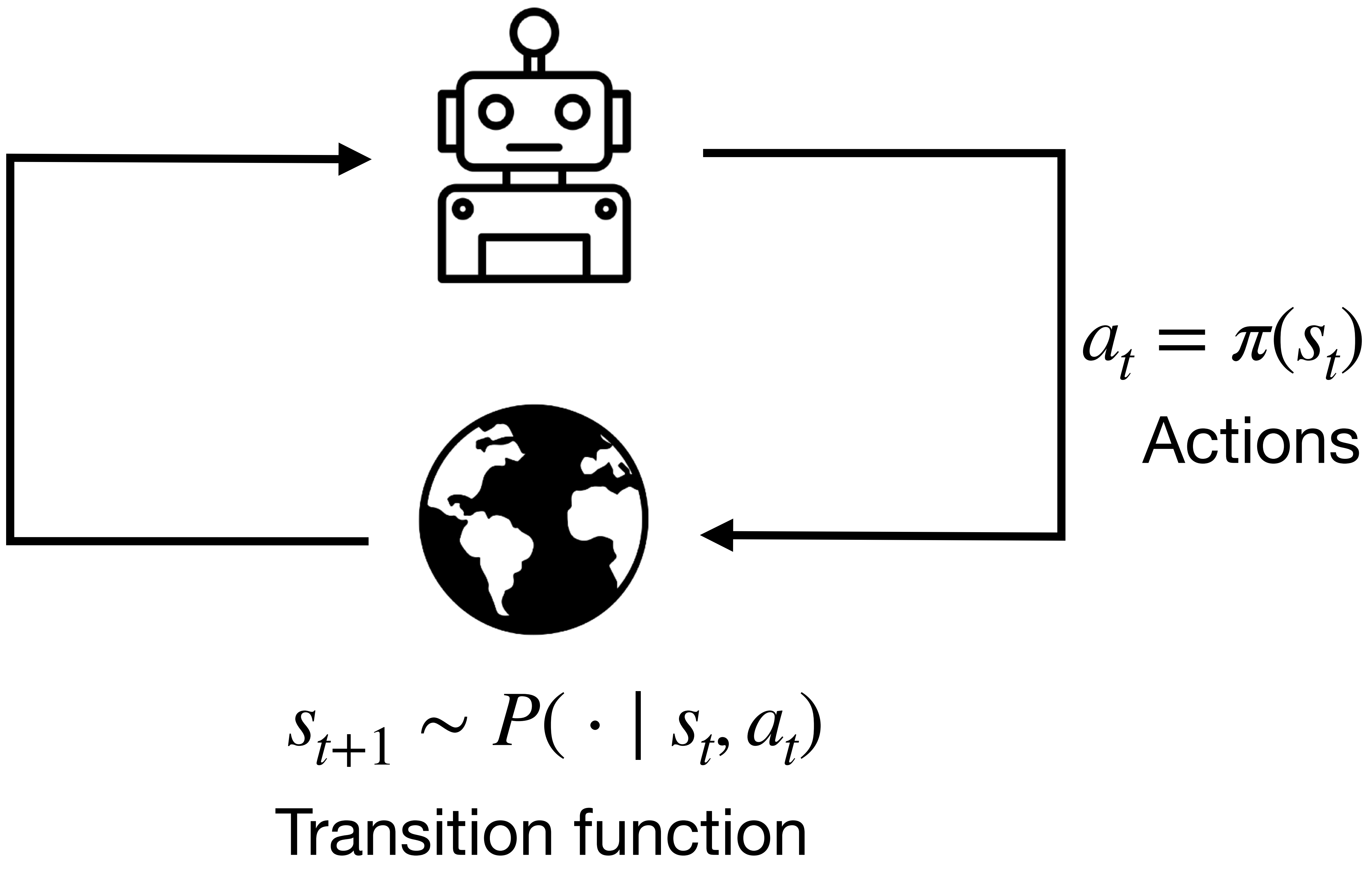
Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

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

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



Reinforcement Learning (RL)

Markov Decision Process (MDP)

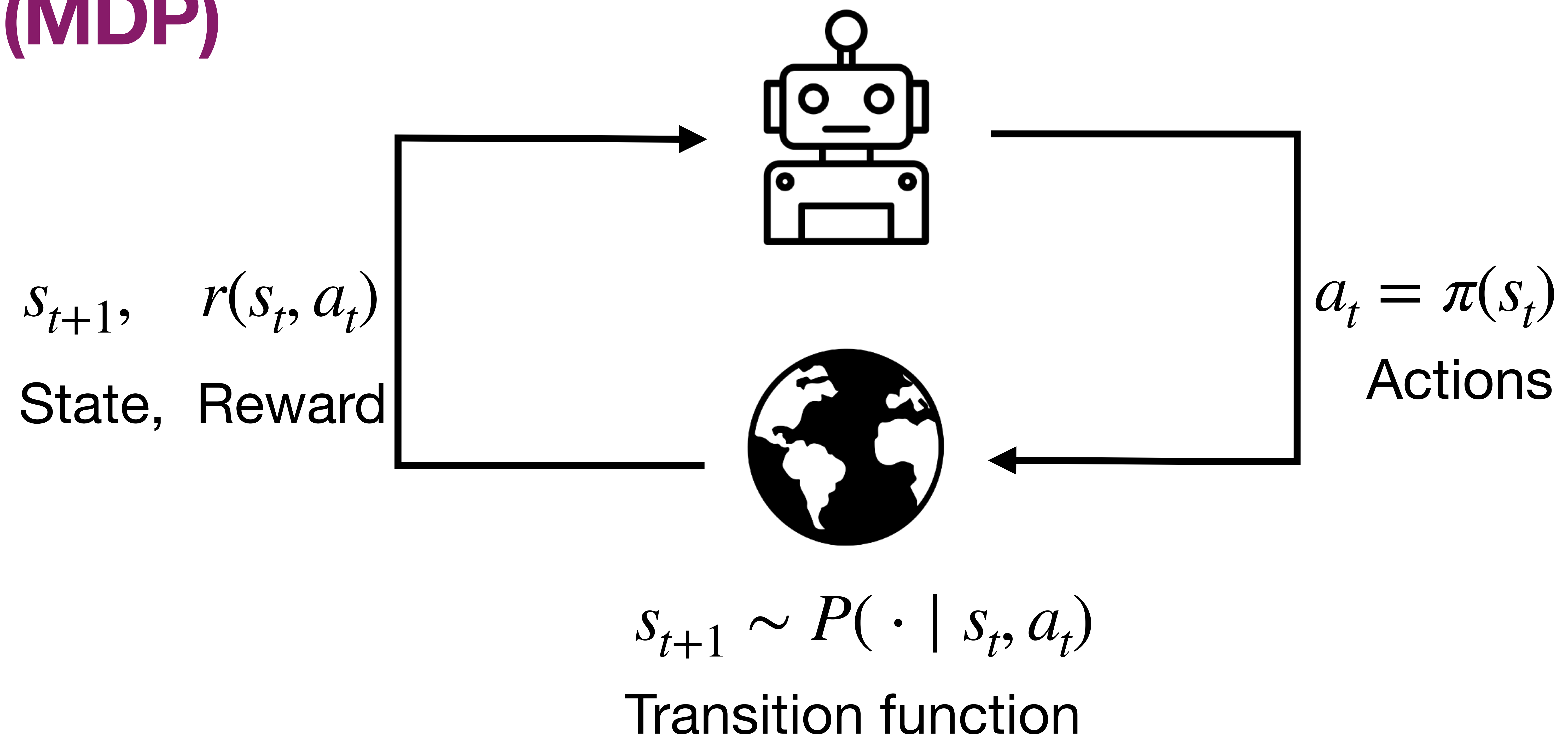
States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

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

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

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



Reinforcement Learning (RL)

Markov Decision Process (MDP)

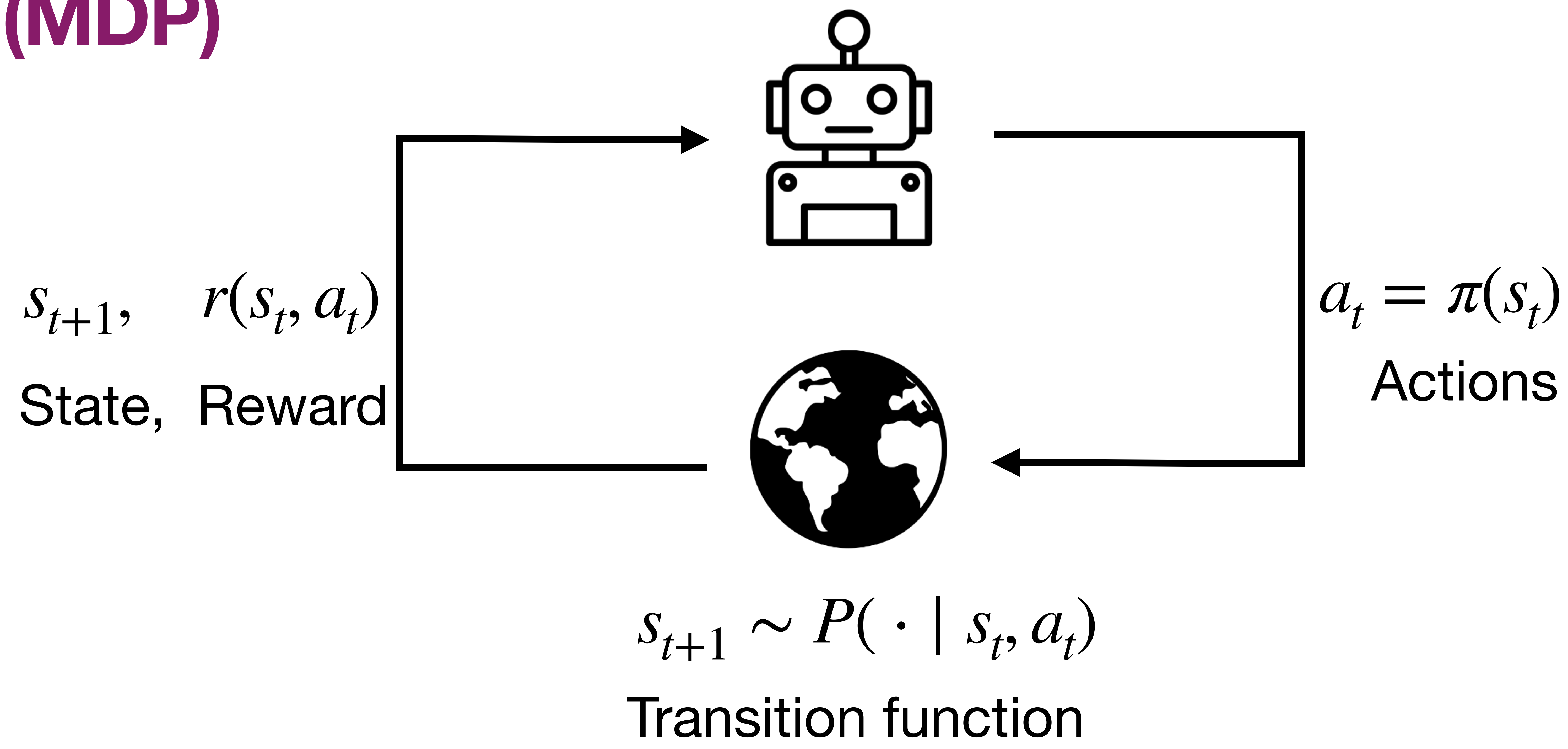
States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

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

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

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



Goal:

Reinforcement Learning (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

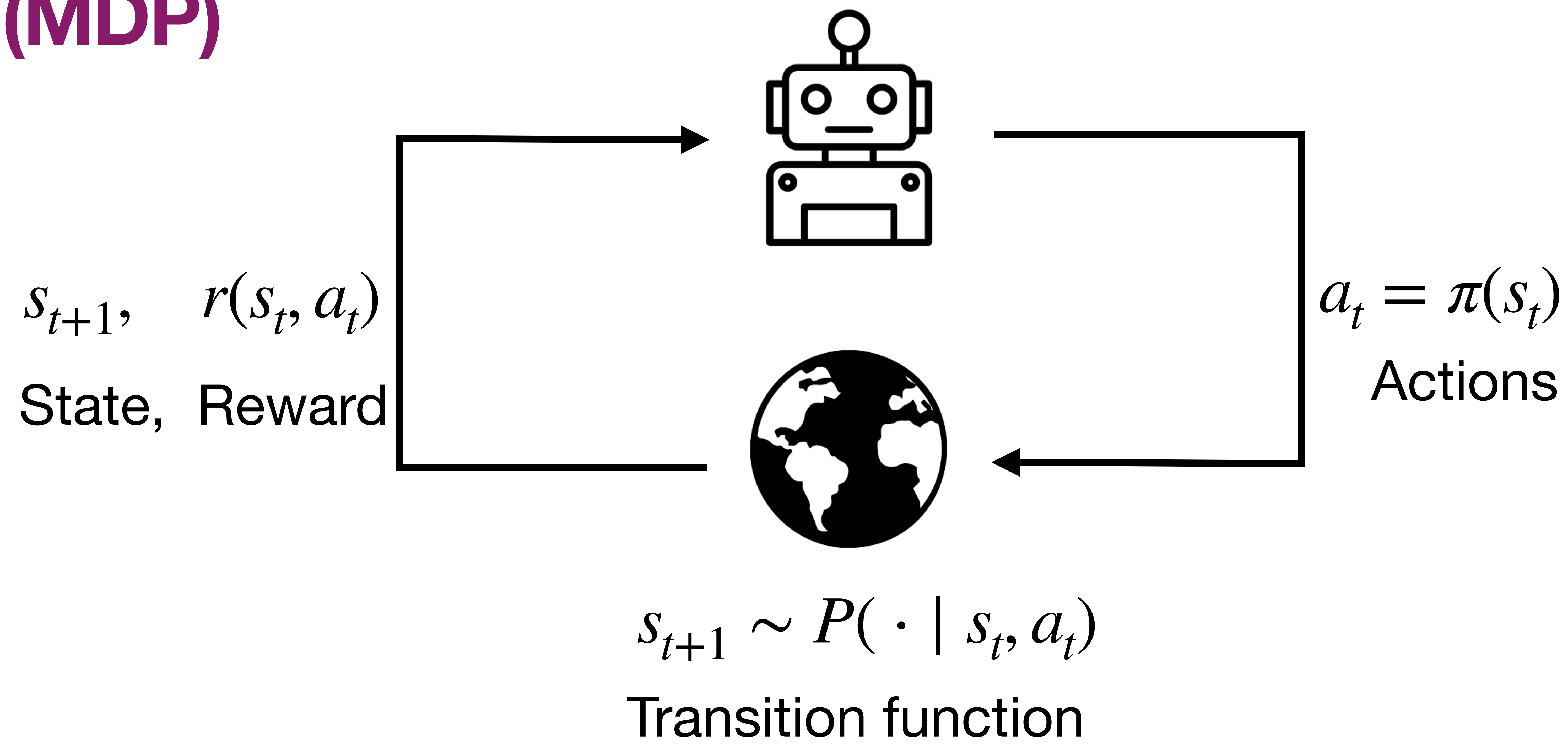
Actions $a \in \mathcal{A}$

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

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

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

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



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 (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

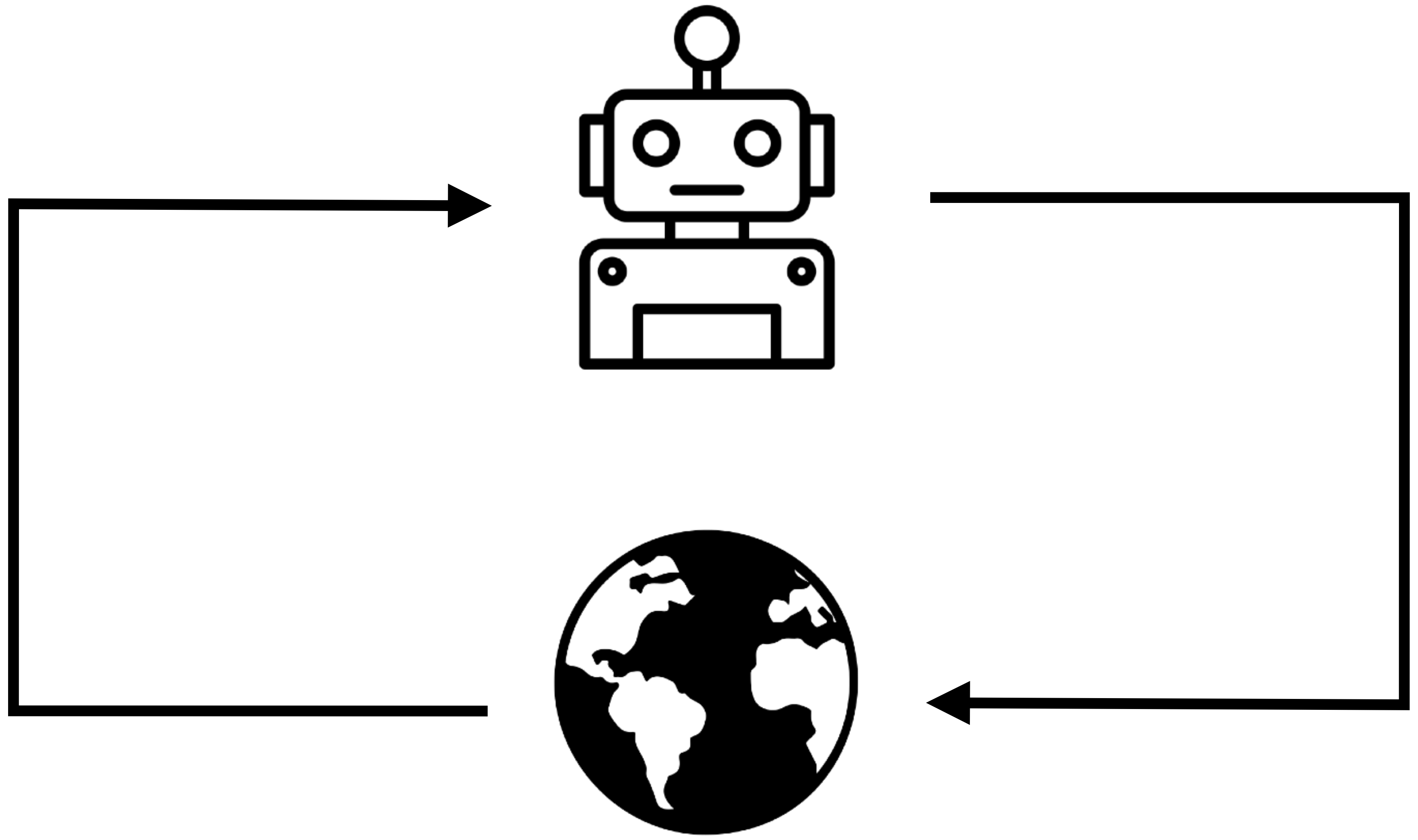
Policy $\pi : \mathcal{S} \to \mathcal{A}$

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

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

In model-based RL these are the "model"

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



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 (RL)

Markov Decision Process (MDP)

States $s \in \mathcal{S}$

Actions $a \in \mathcal{A}$

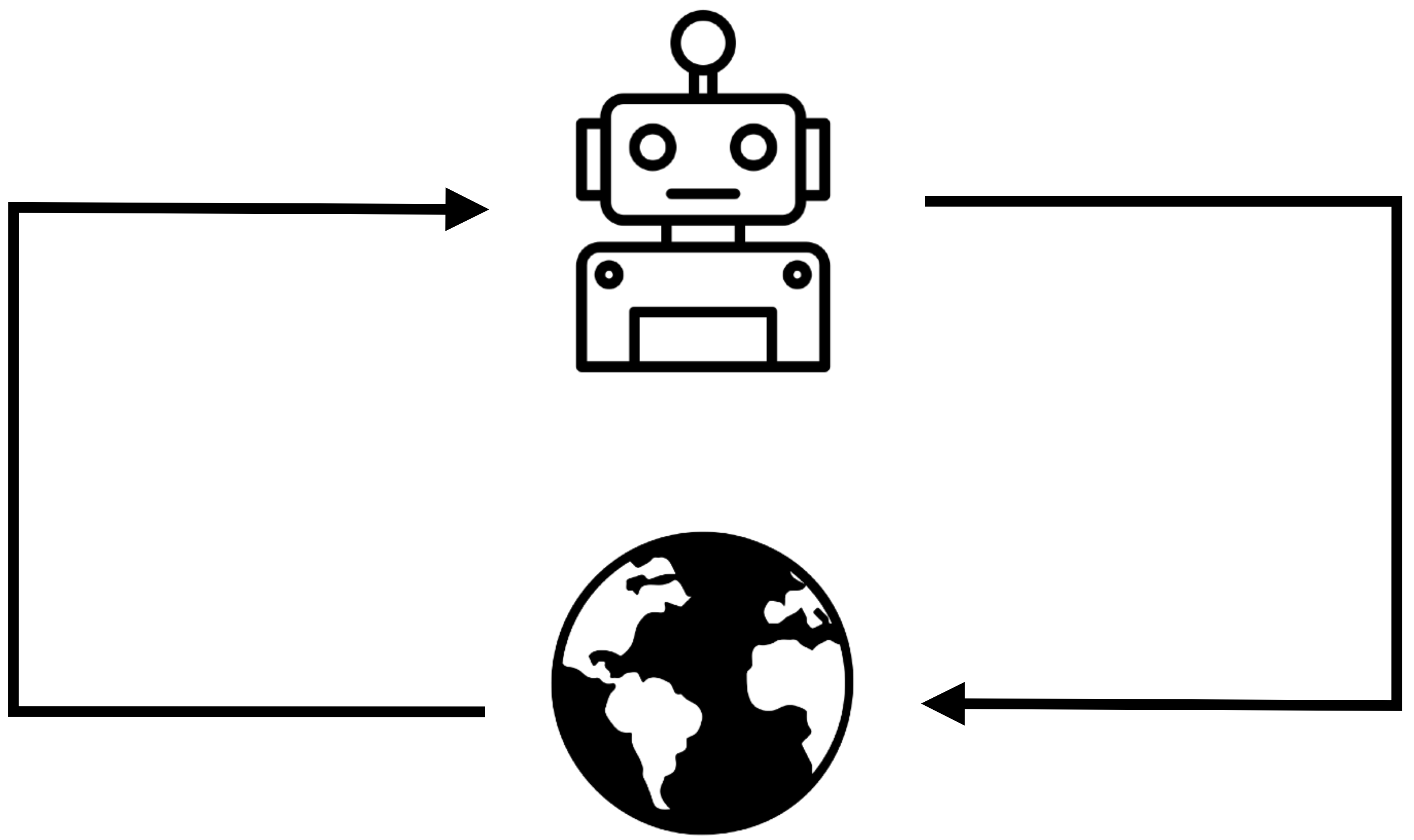
Policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$

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

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

In model-based RL these are the "model"

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



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

State, Reward

$a_t = \pi(s_t)$

Actions

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

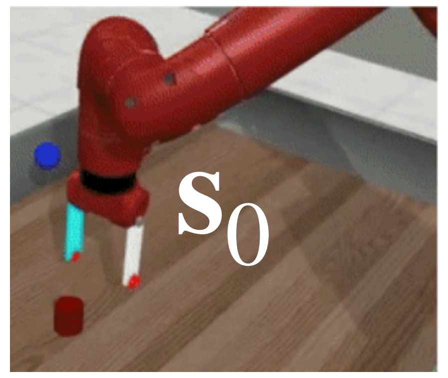
Transition function

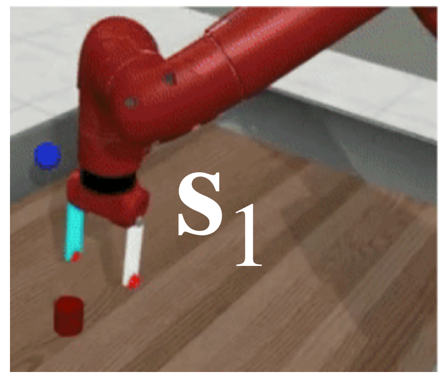
Goal:

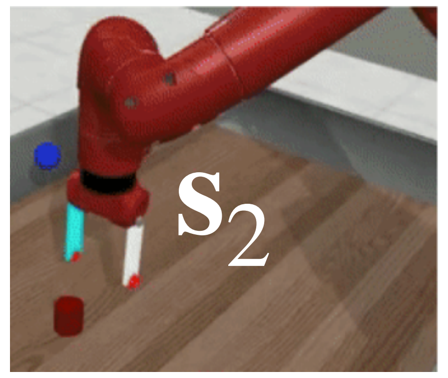
$$\max_{\pi} \mathbb{E}_{\pi, P_\phi}\left[\sum_{t=0}^{\infty} \gamma^t r_\xi(s_t, a_t) \mid s_0 = s, \pi\right]$$

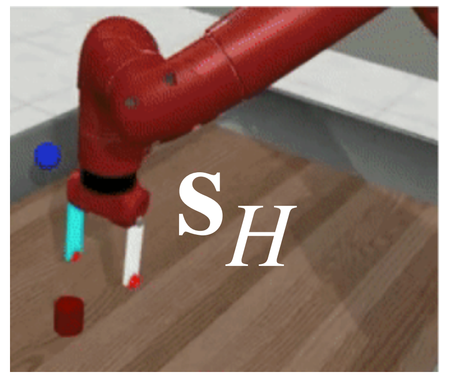
World Models

World Models

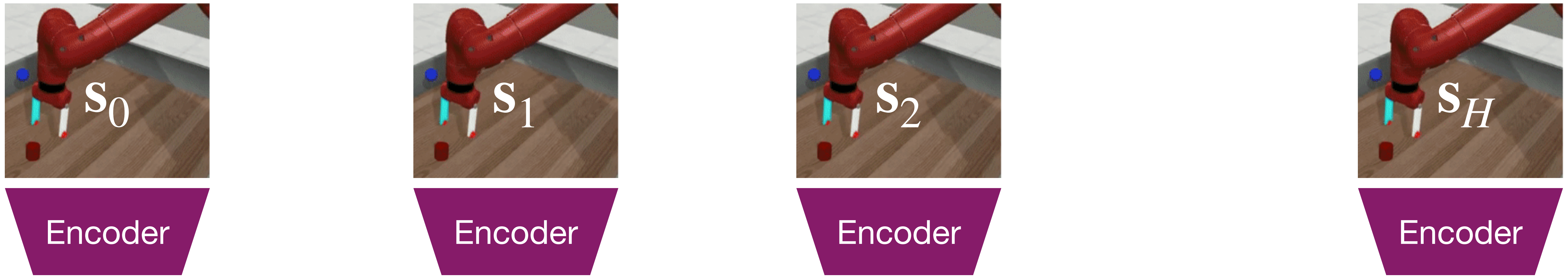




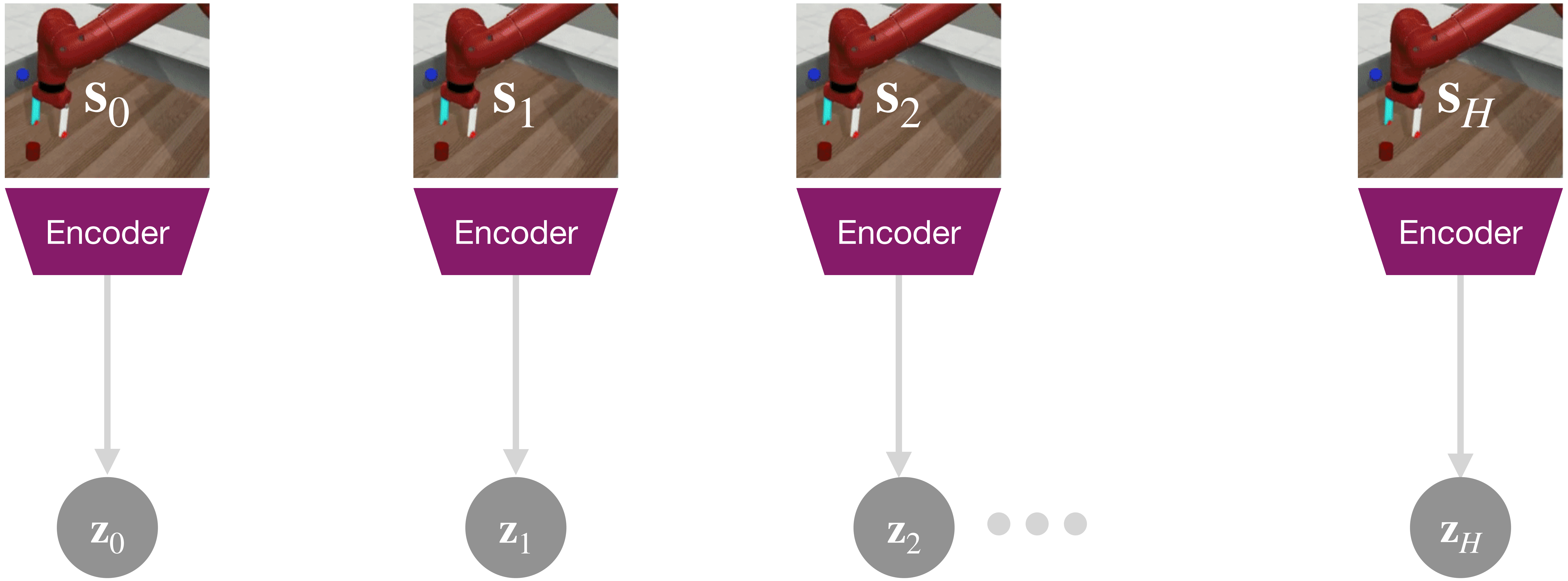




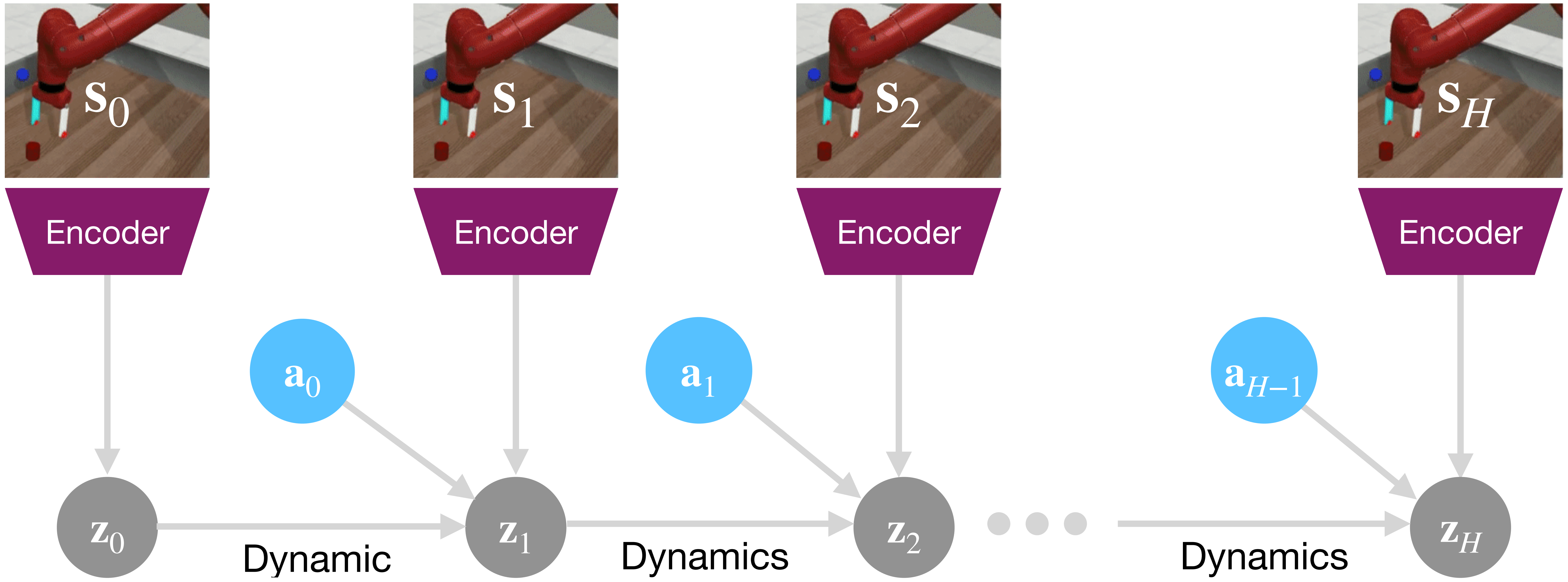
World Models



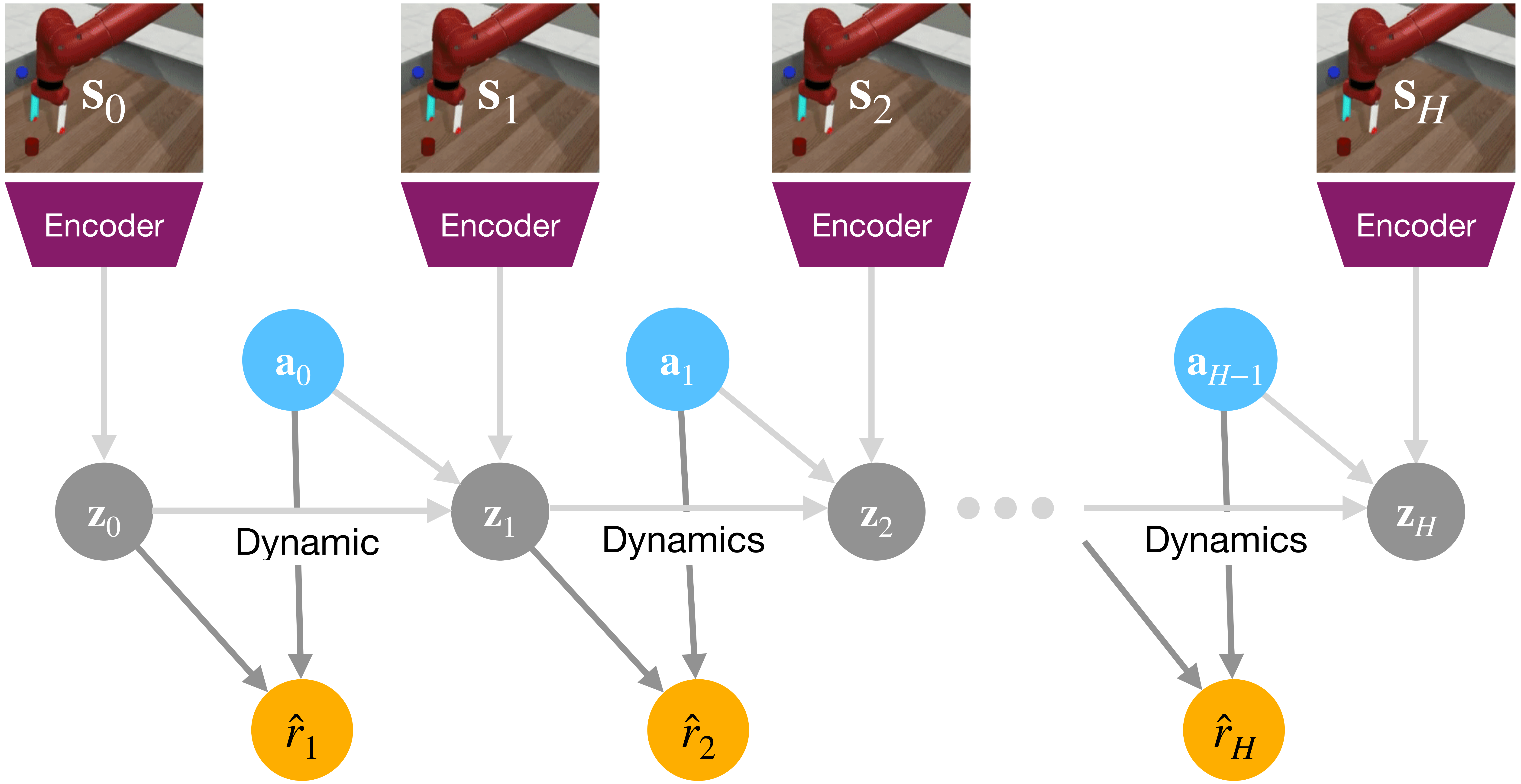
World Models



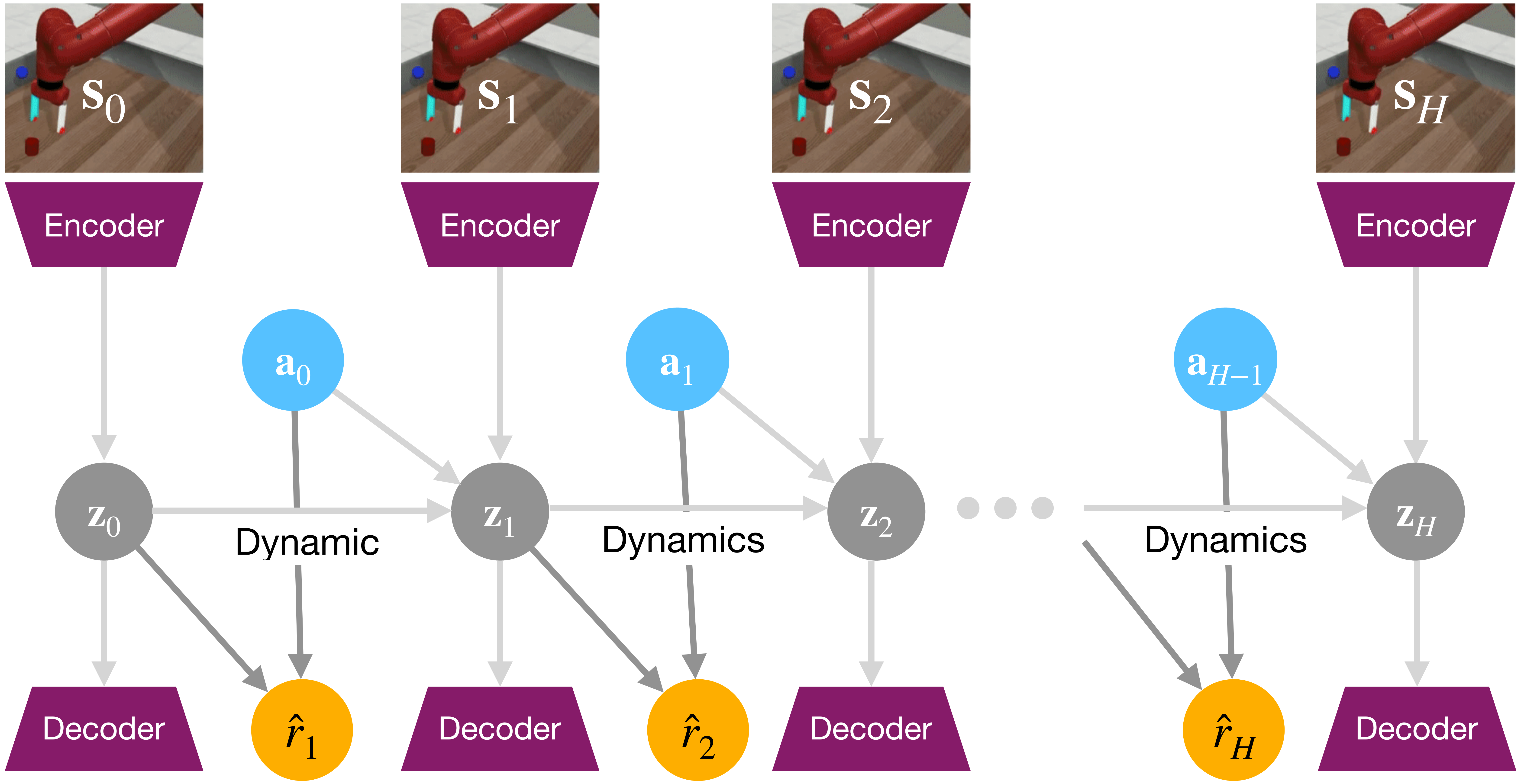
World Models



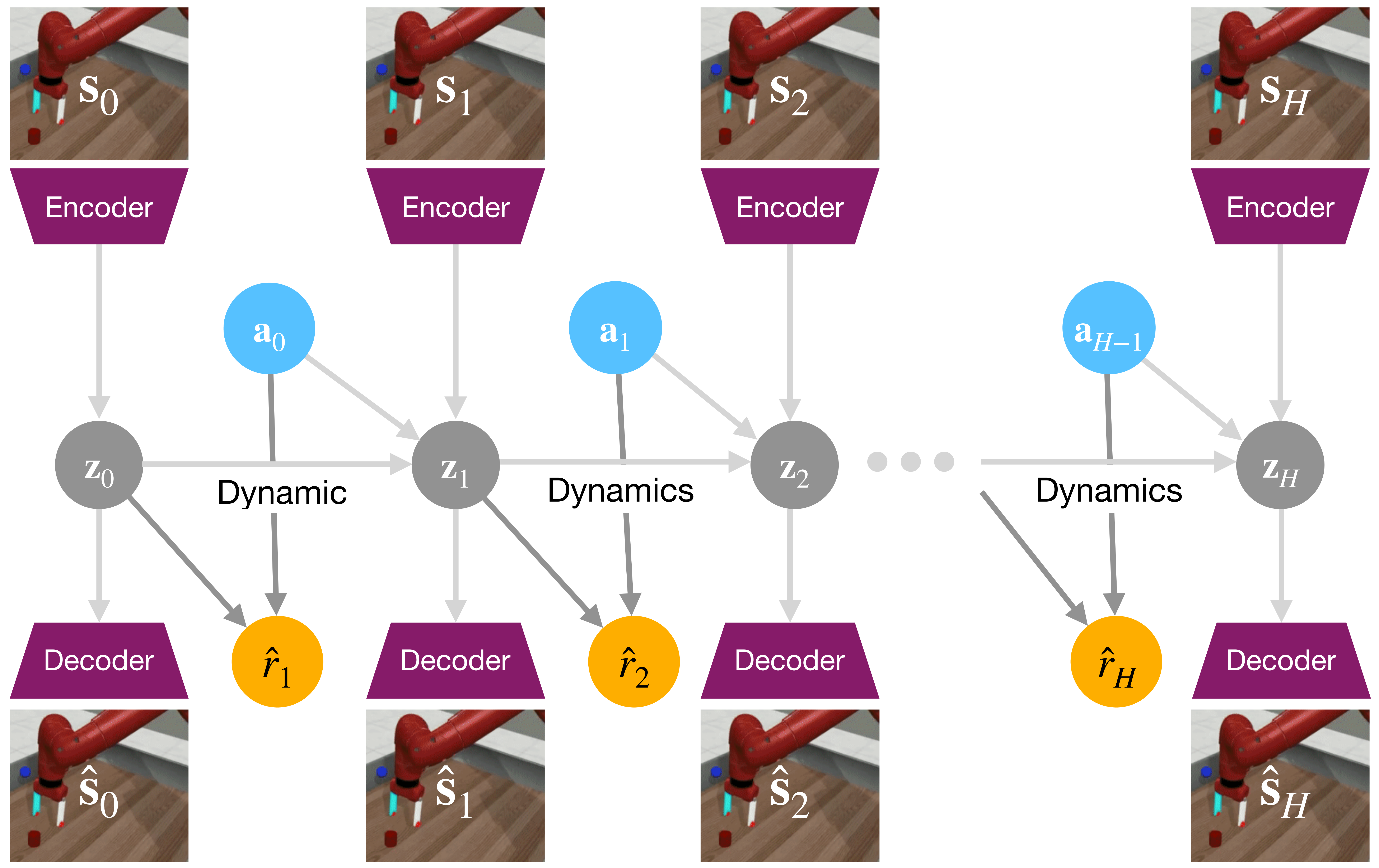
World Models



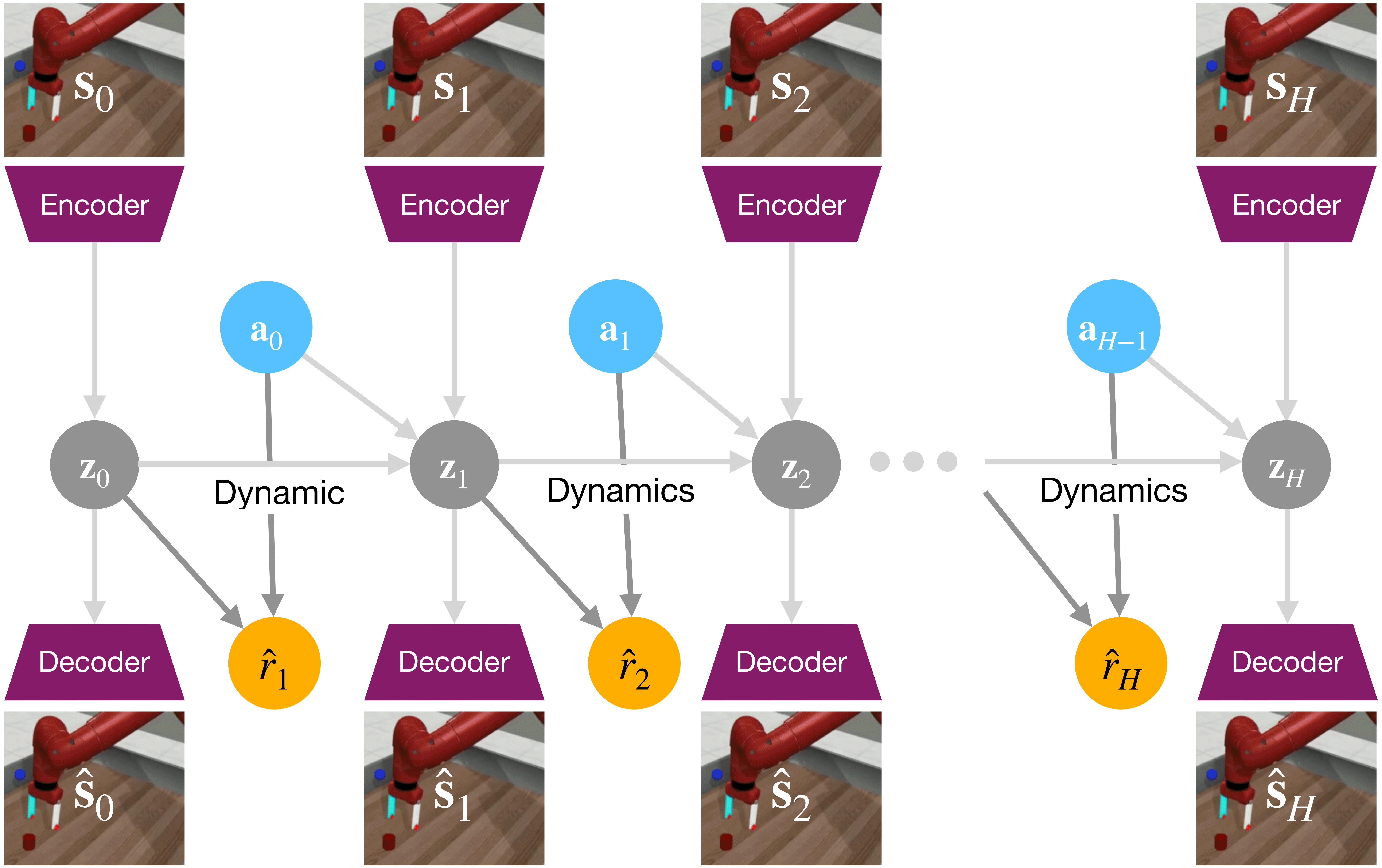
World Models



World Models

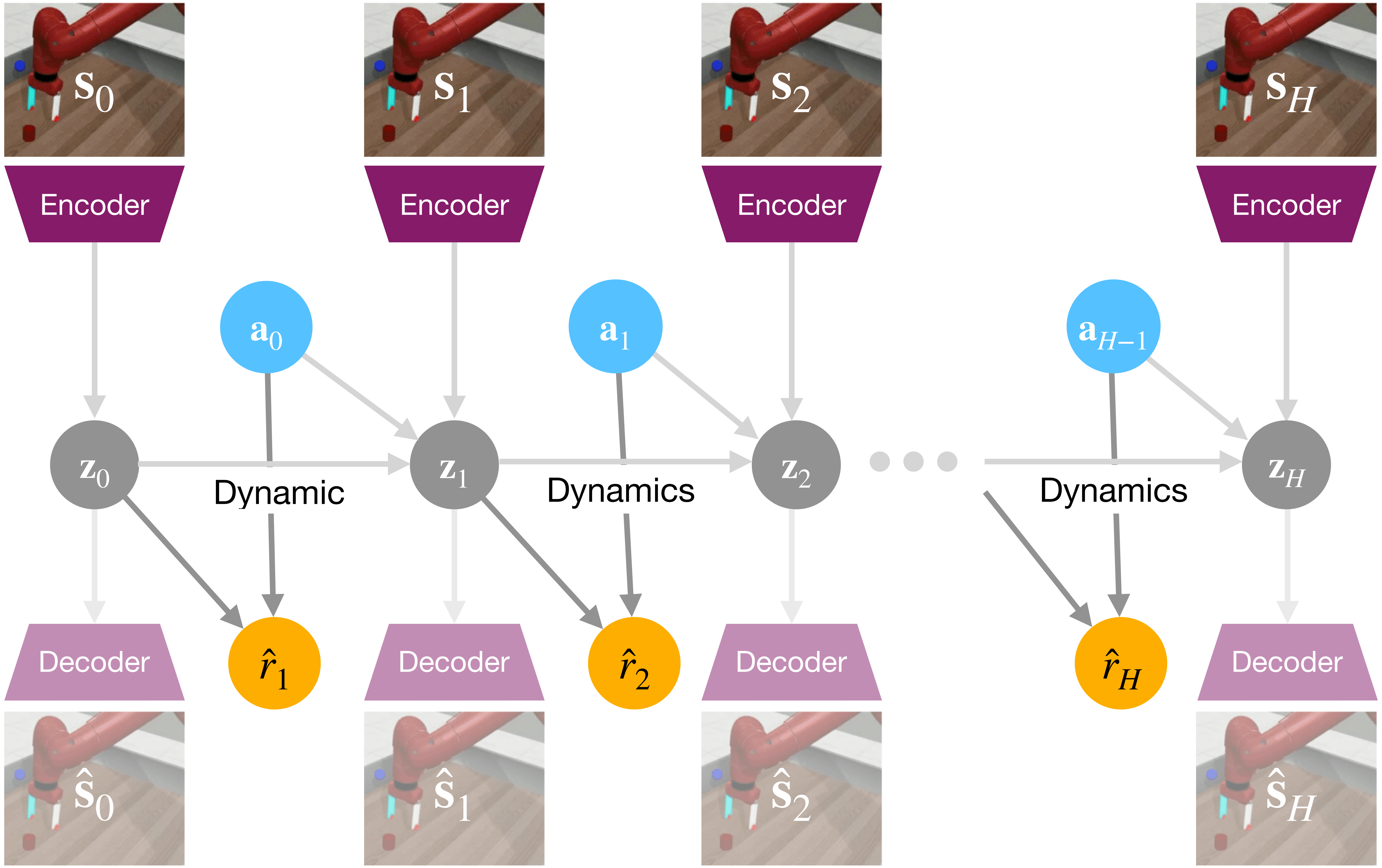


World Models



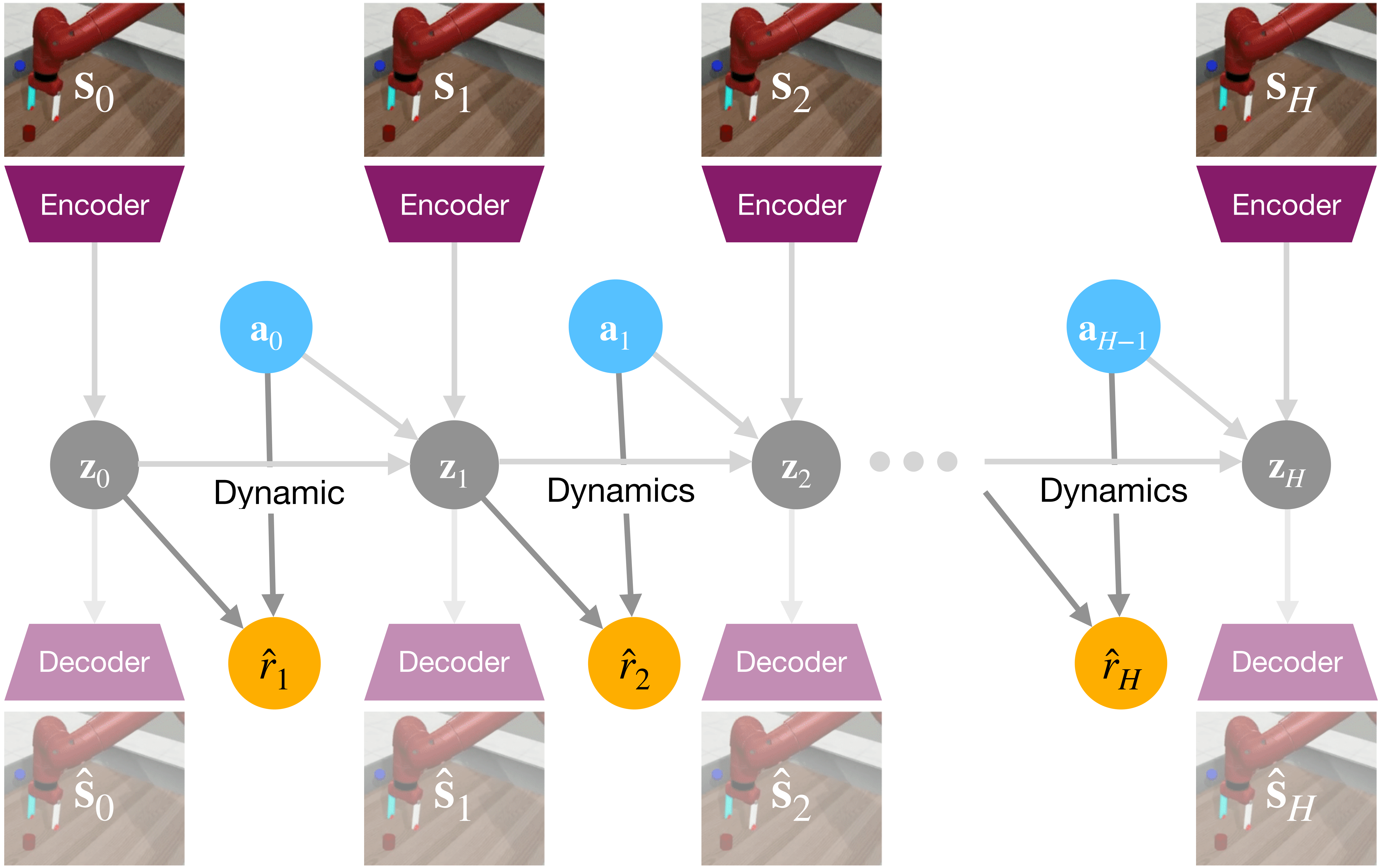
- Decoder vs self-supervised?

World Models



- Decoder vs self-supervised?

World Models



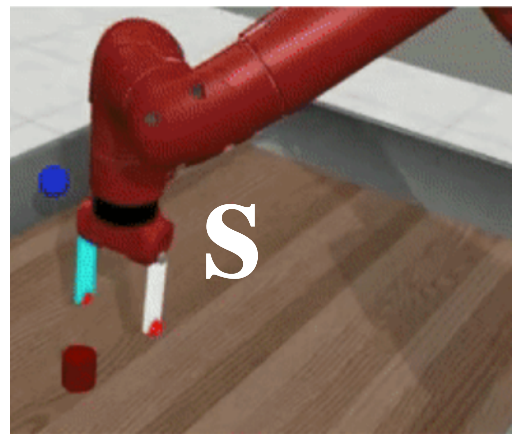
- Decoder vs self-supervised?
- Continuous vs discrete?

DCWM: Discrete Codebook World Model

Discrete Codebook

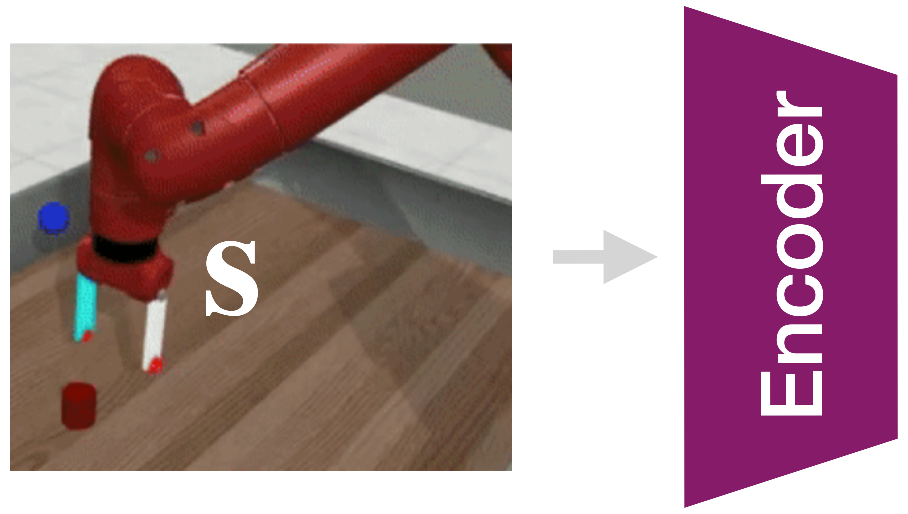
DCWM: Discrete Codebook World Model

Discrete Codebook



DCWM: Discrete Codebook World Model

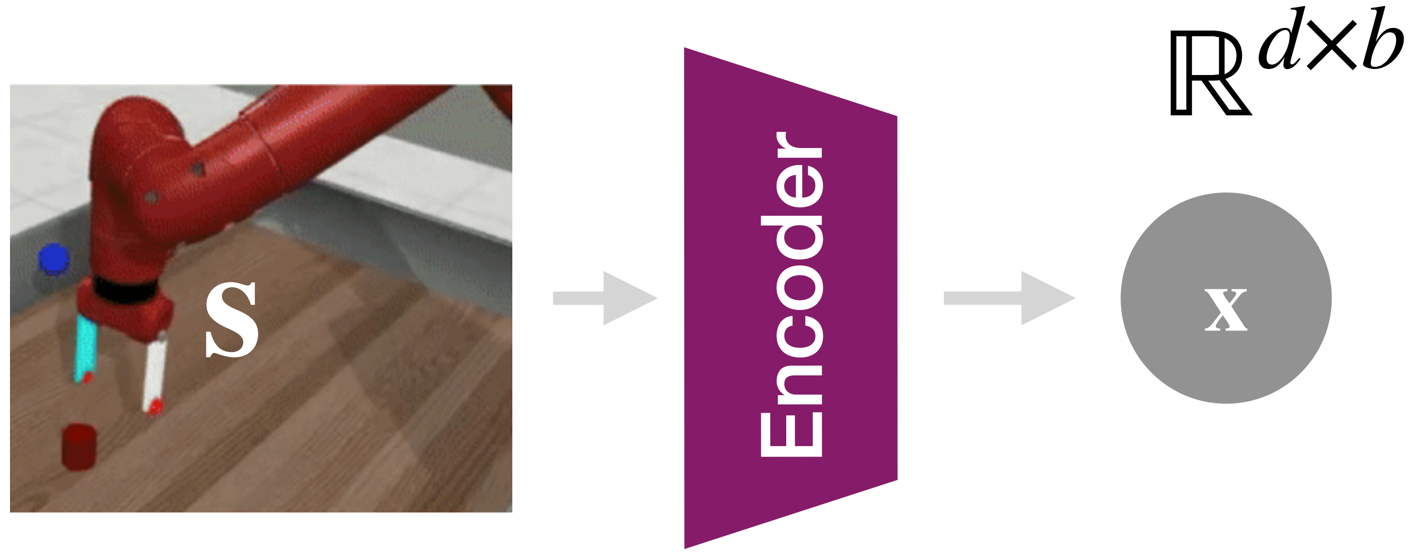
Discrete Codebook



DCWM: Discrete Codebook World Model
Discrete Codebook

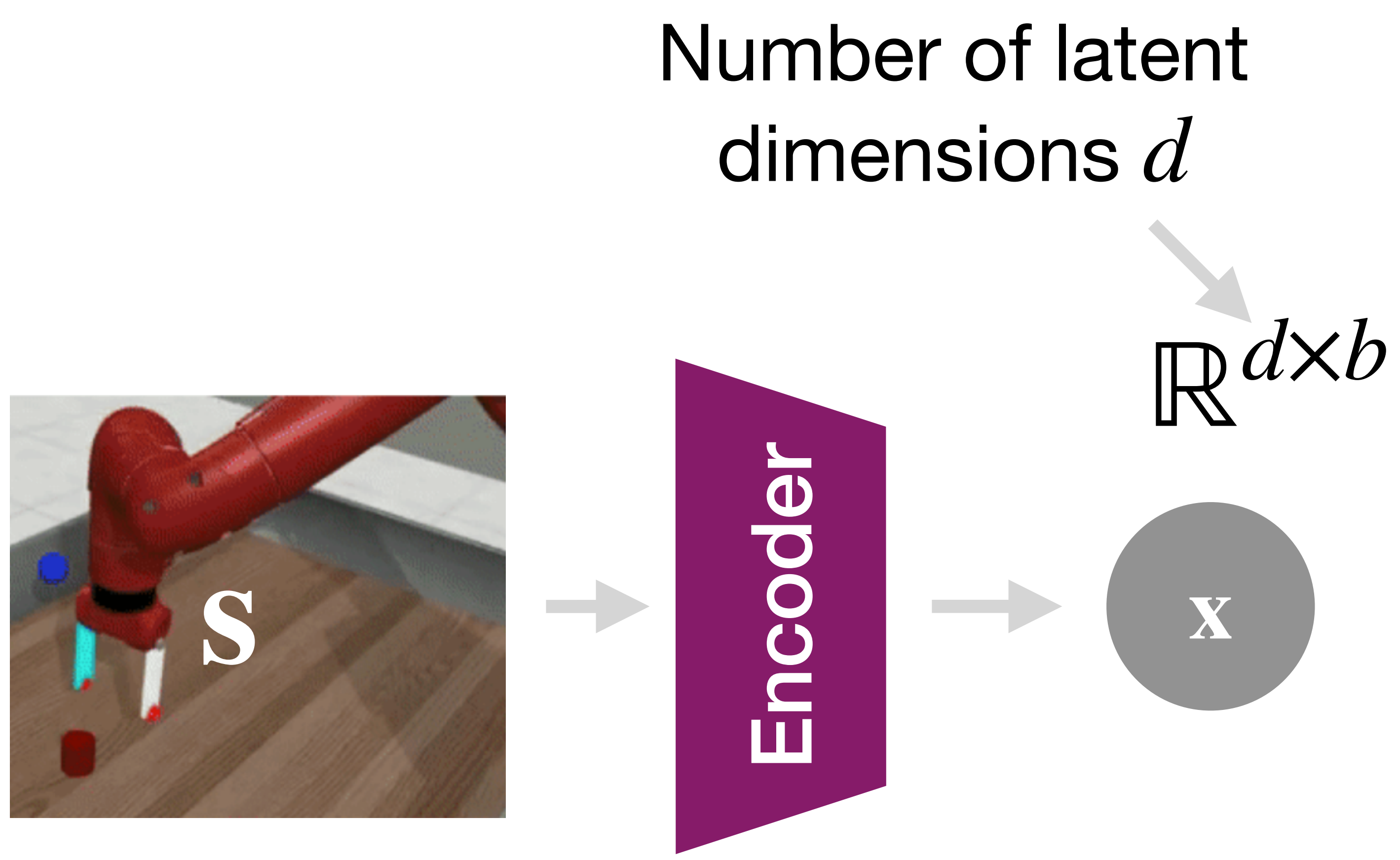
DCWM: Discrete Codebook World Model

Discrete Codebook



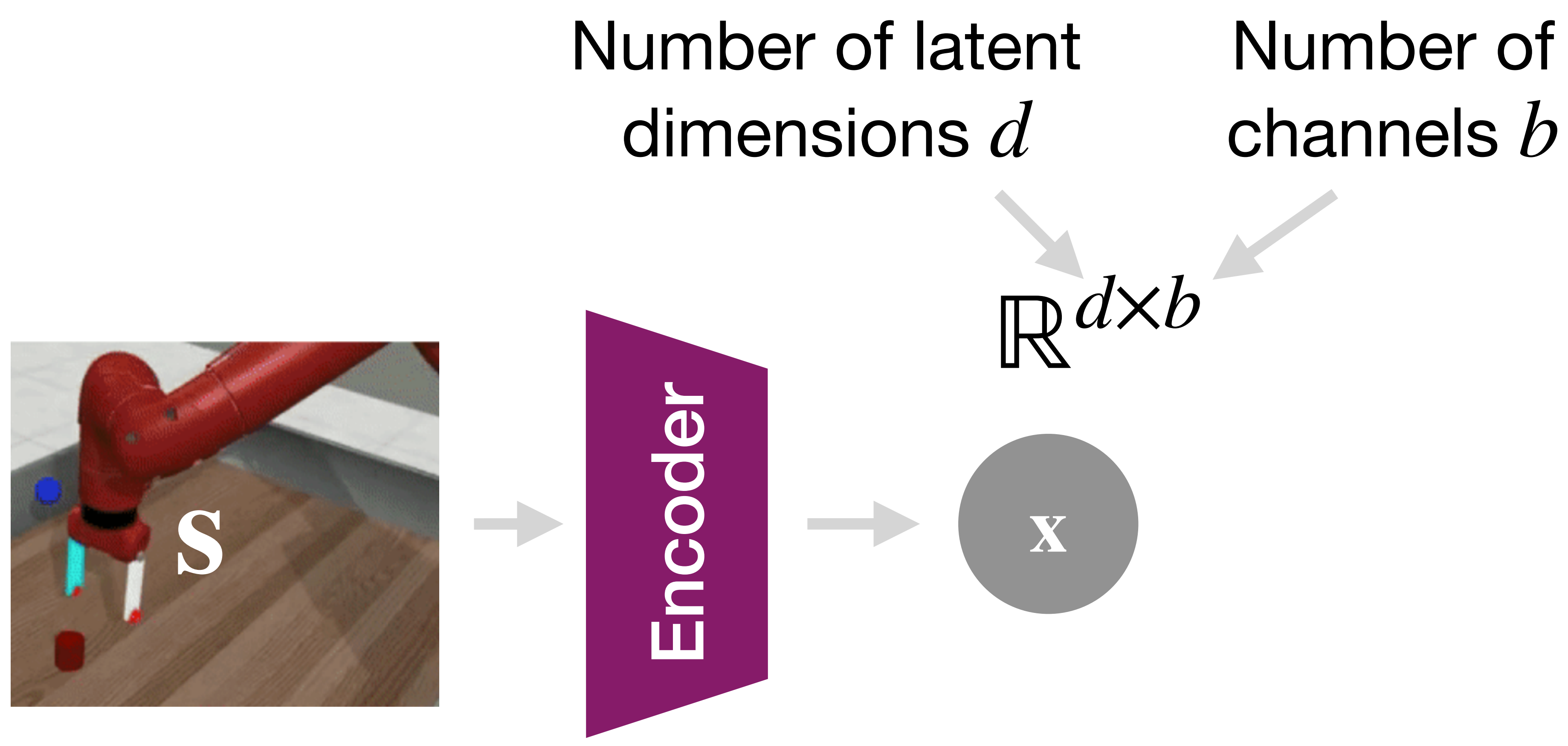
DCWM: Discrete Codebook World Model

Discrete Codebook



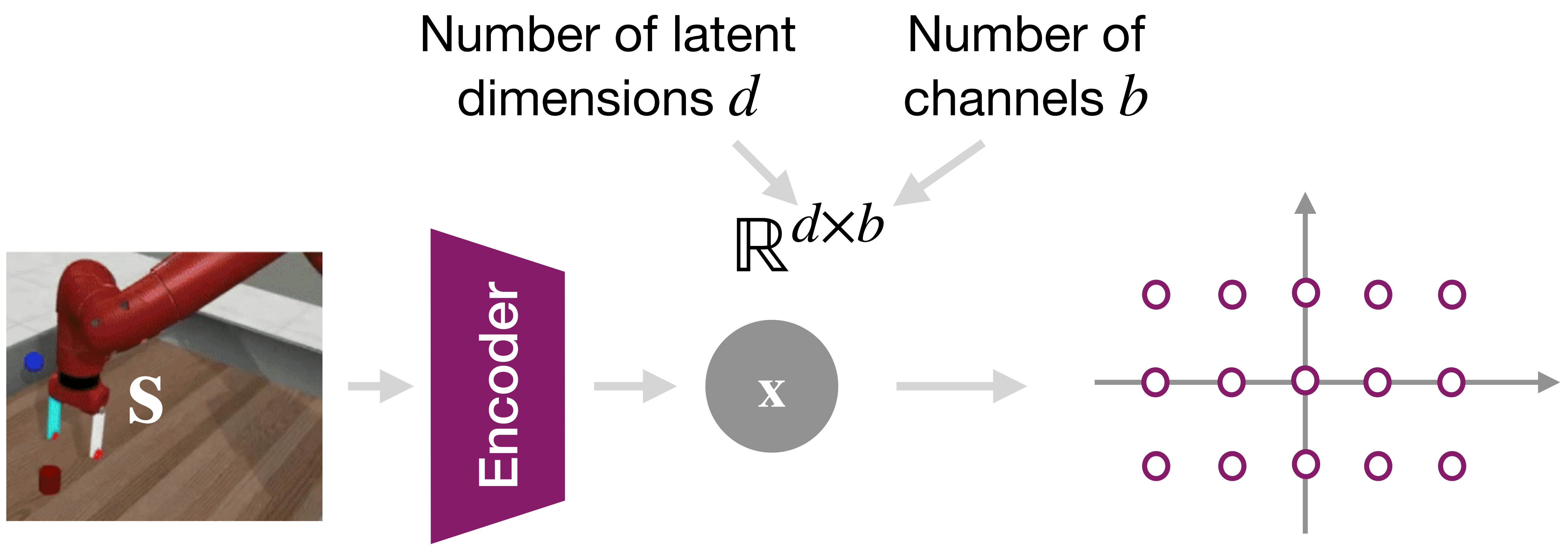
DCWM: Discrete Codebook World Model

Discrete Codebook



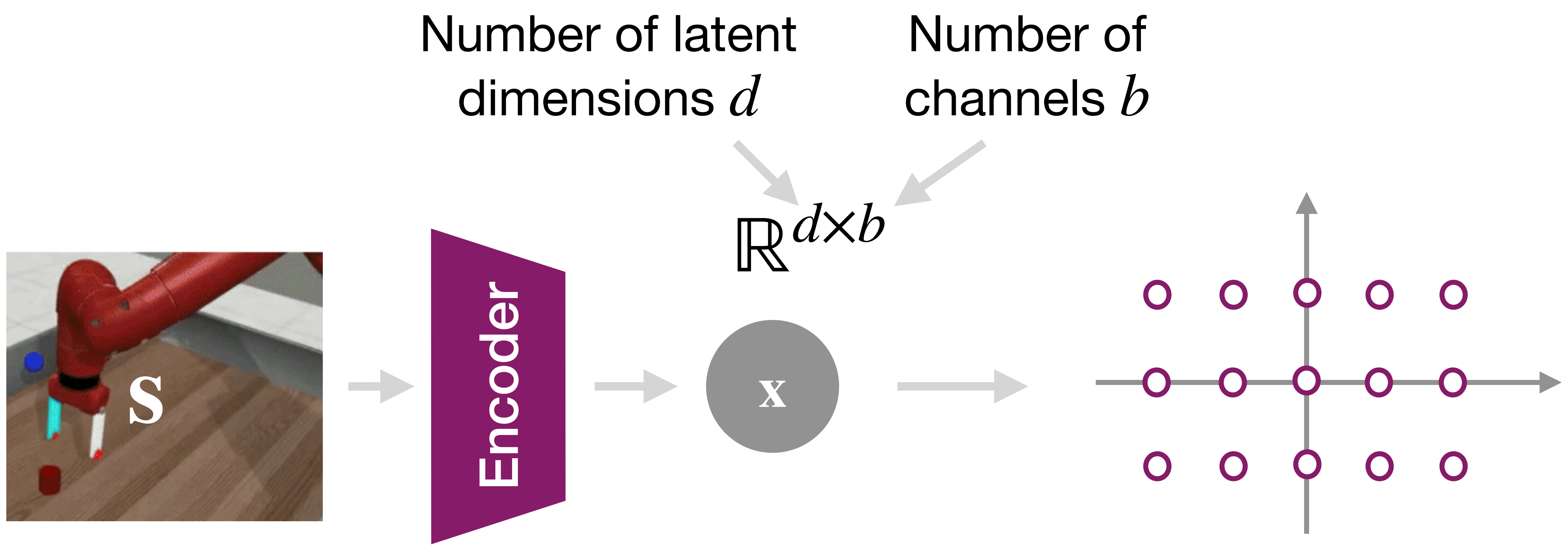
DCWM: Discrete Codebook World Model

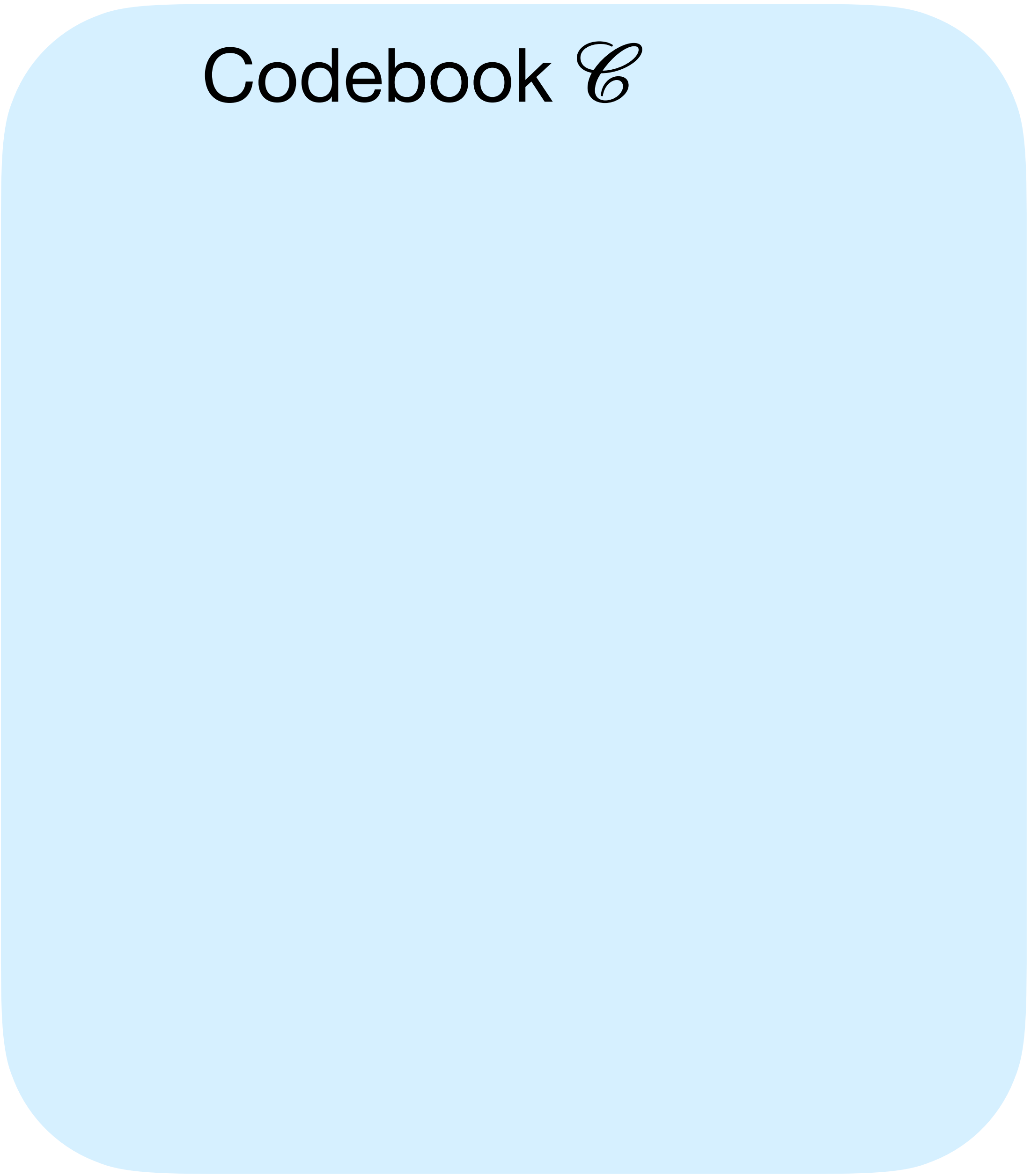
Discrete Codebook



DCWM: Discrete Codebook World Model

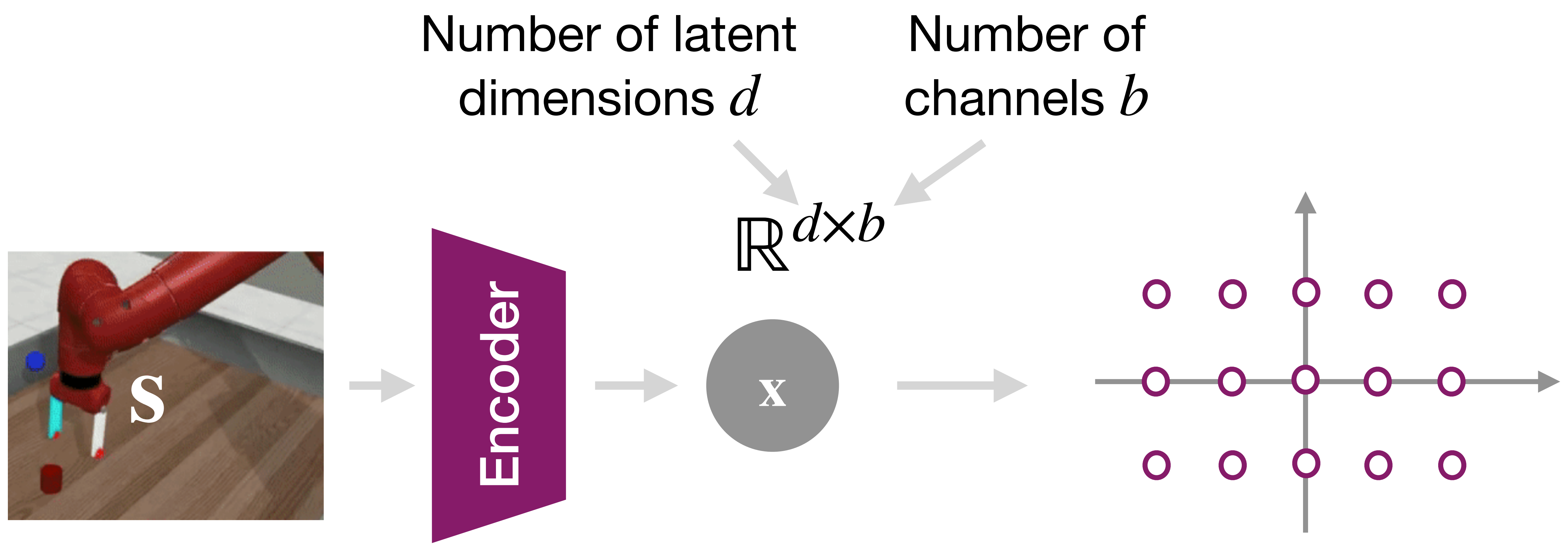
Discrete Codebook

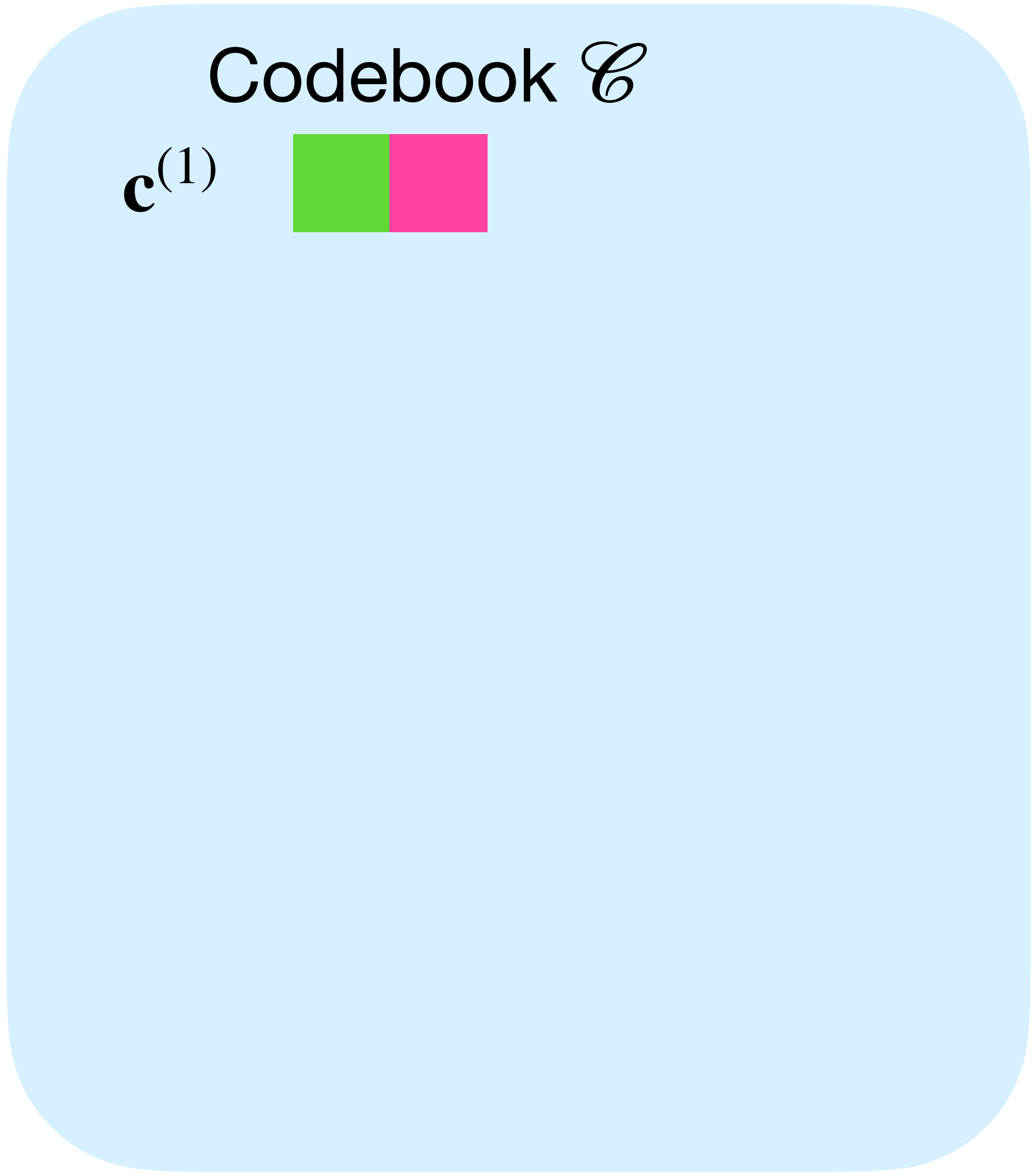




DCWM: Discrete Codebook World Model

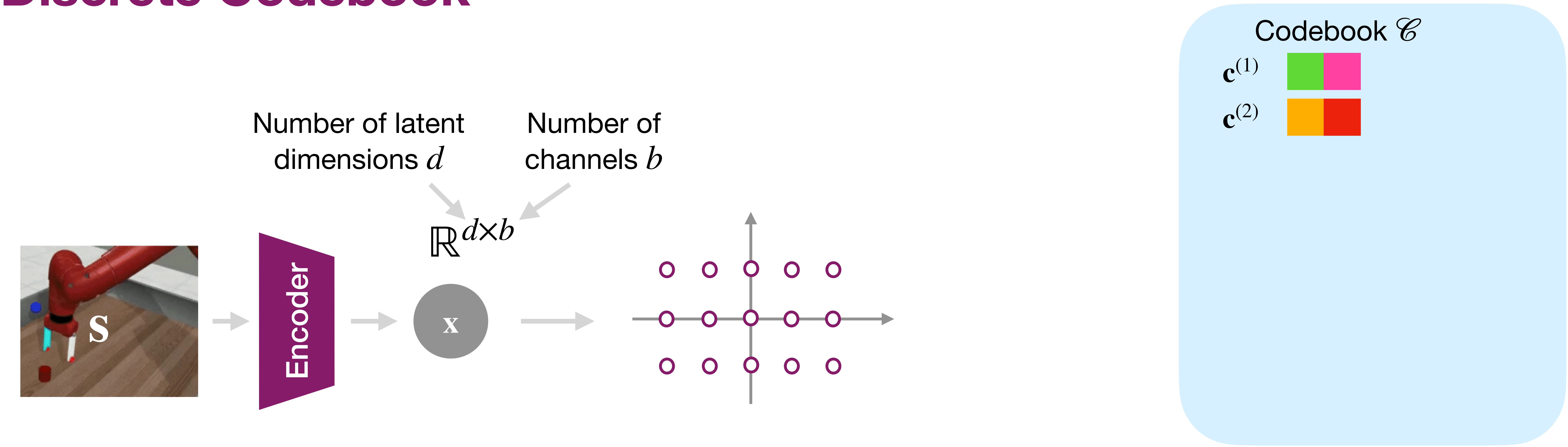
Discrete Codebook





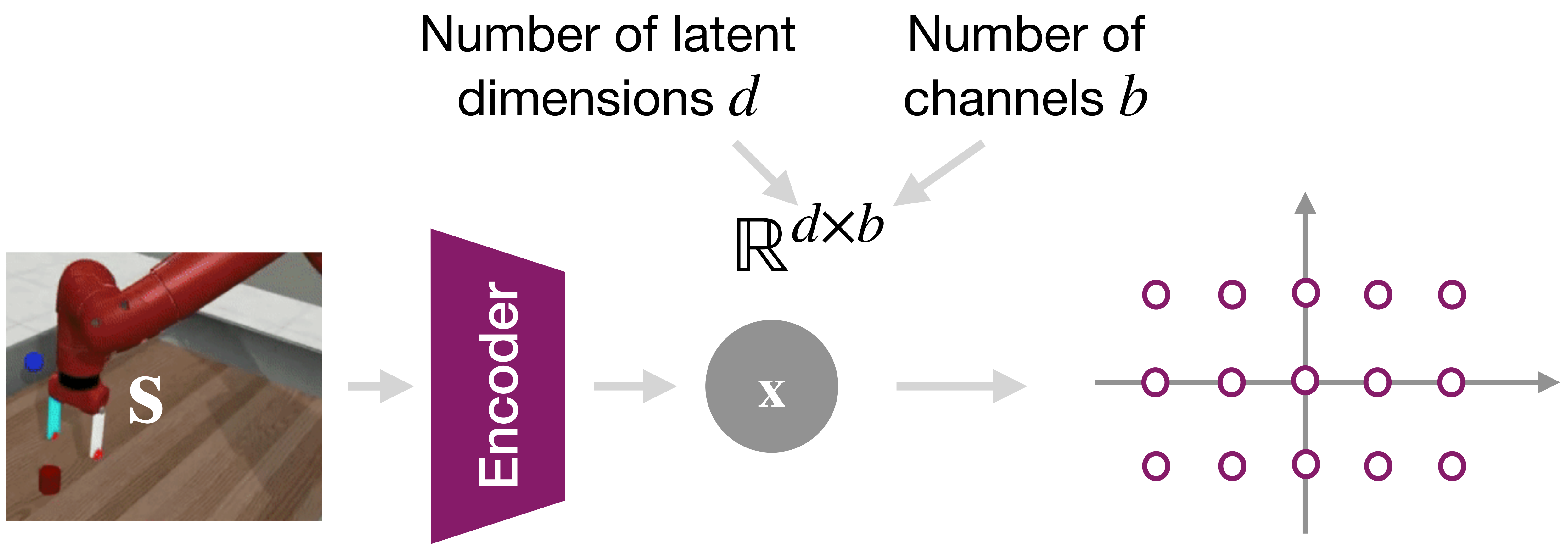
DCWM: Discrete Codebook World Model

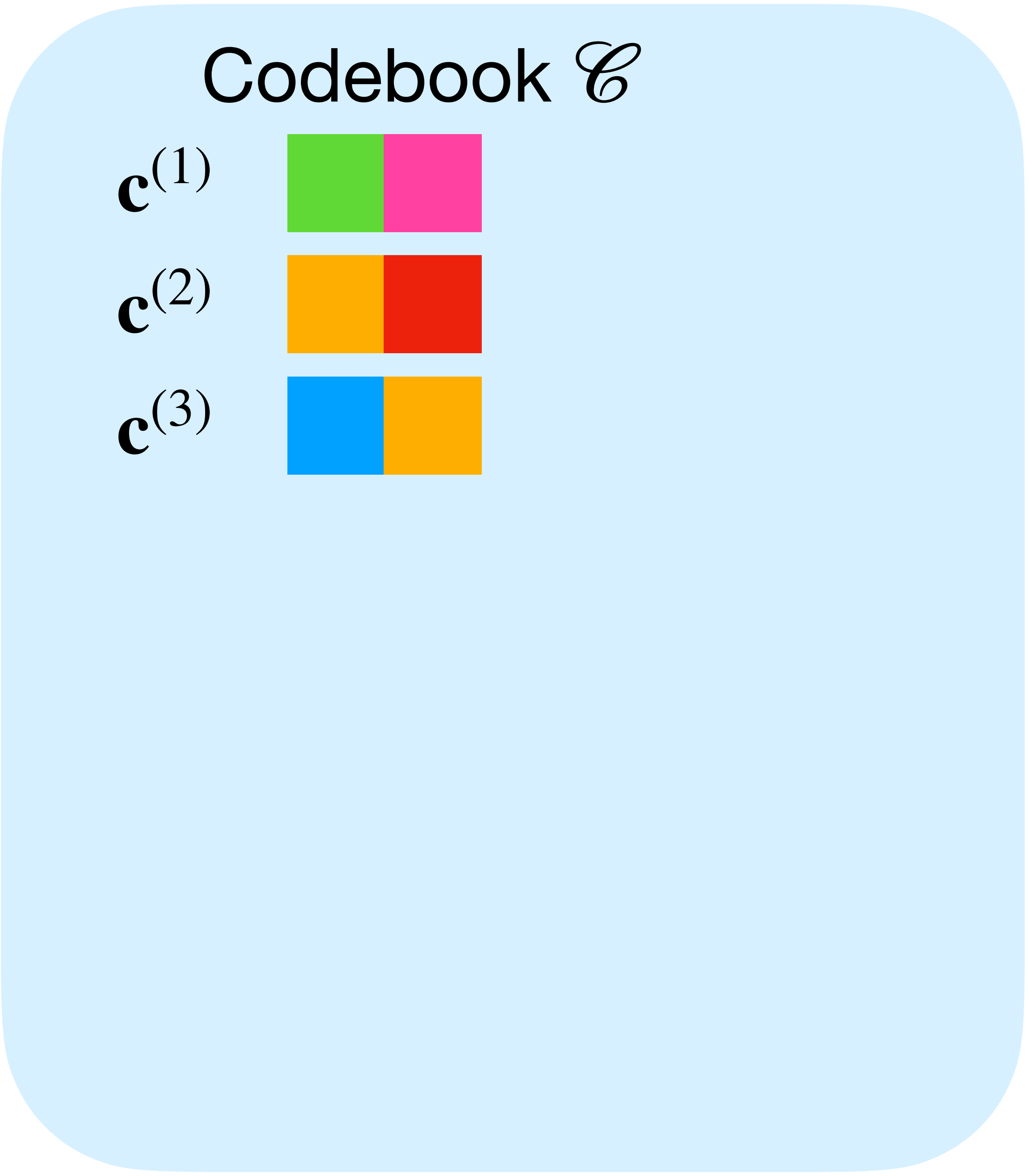
Discrete Codebook



DCWM: Discrete Codebook World Model

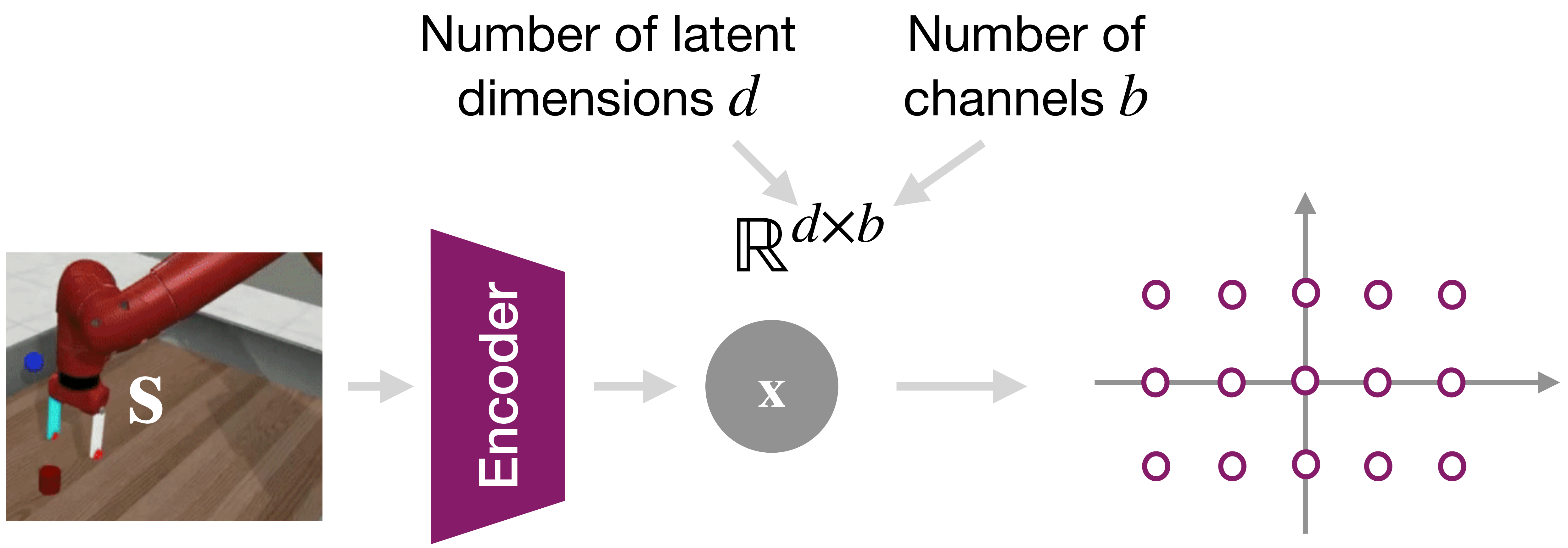
Discrete Codebook

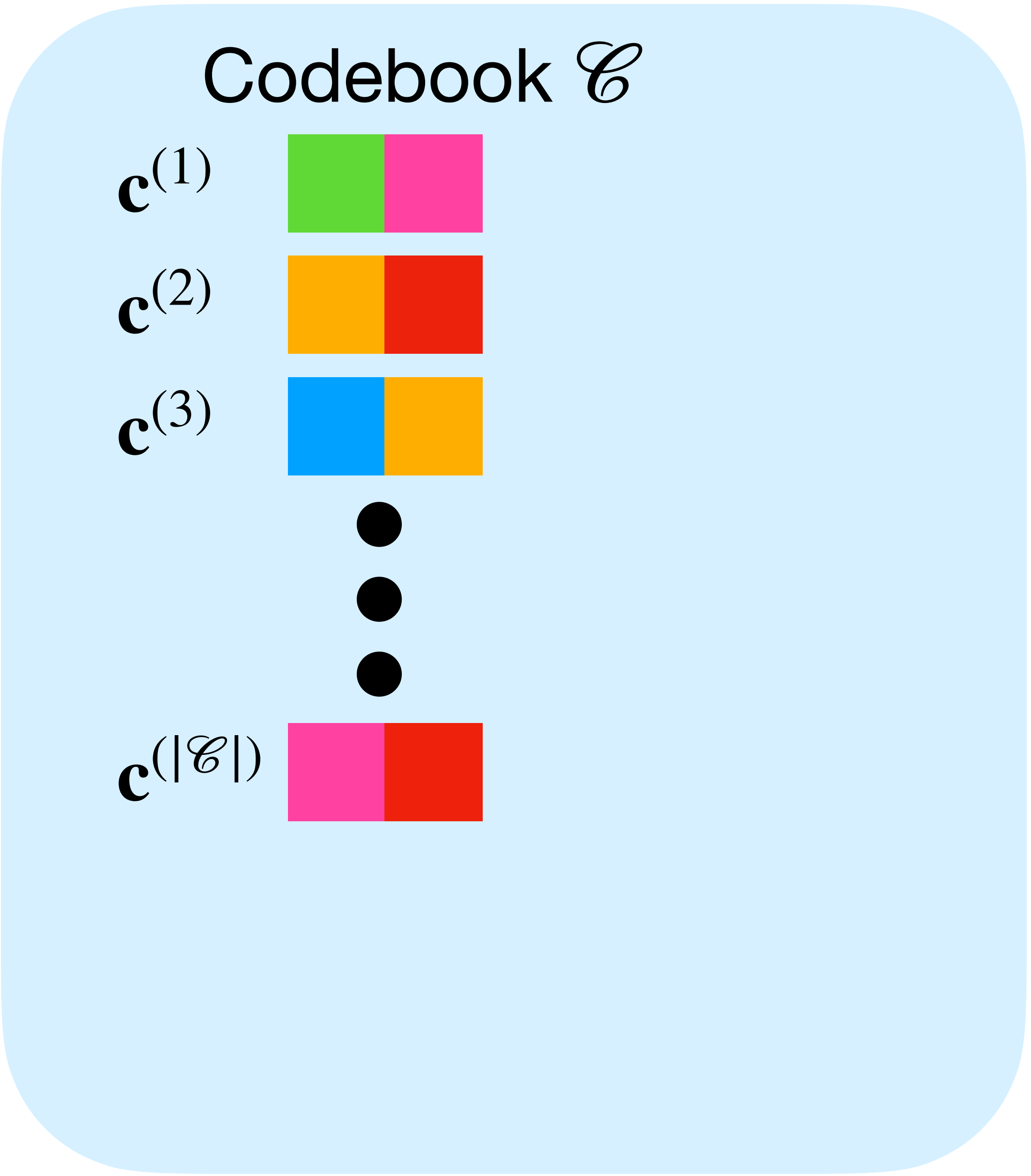




DCWM: Discrete Codebook World Model

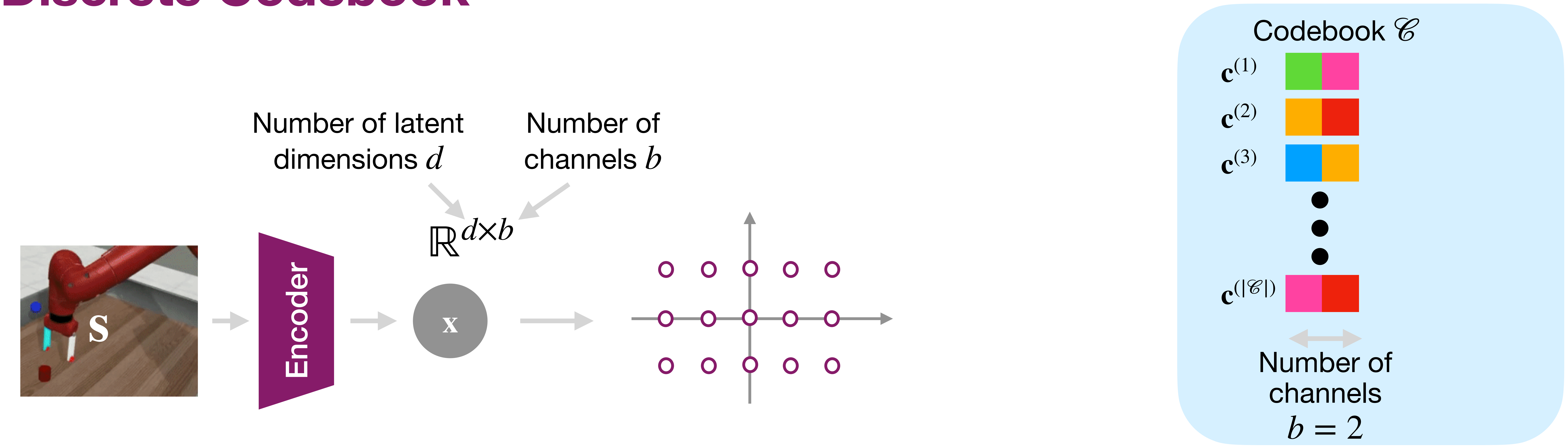
Discrete Codebook





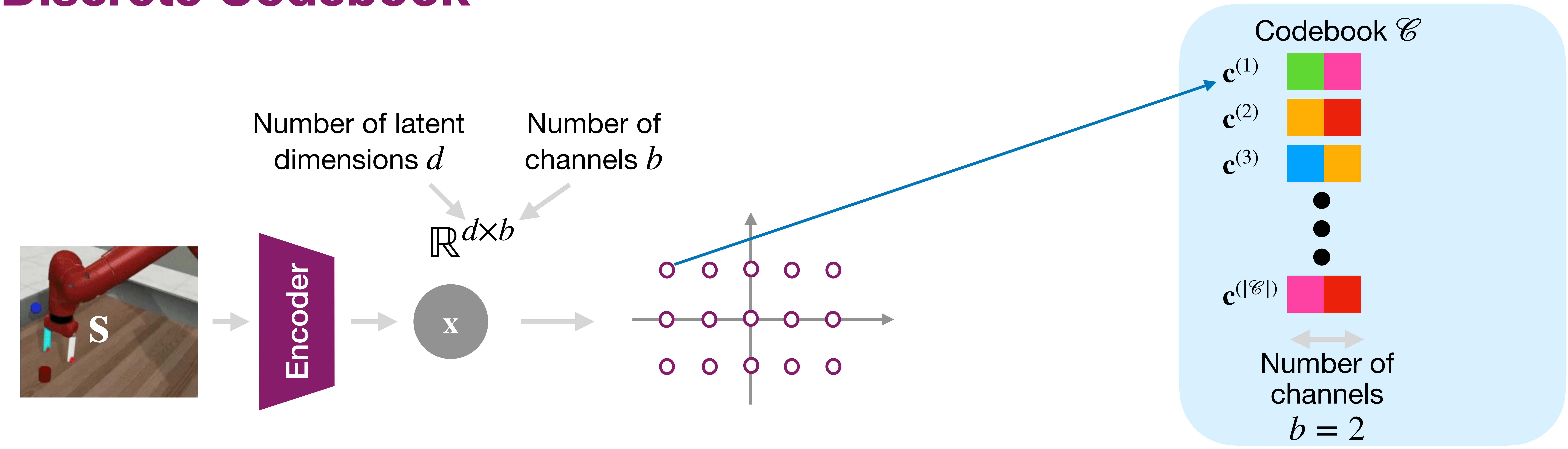
DCWM: Discrete Codebook World Model

Discrete Codebook



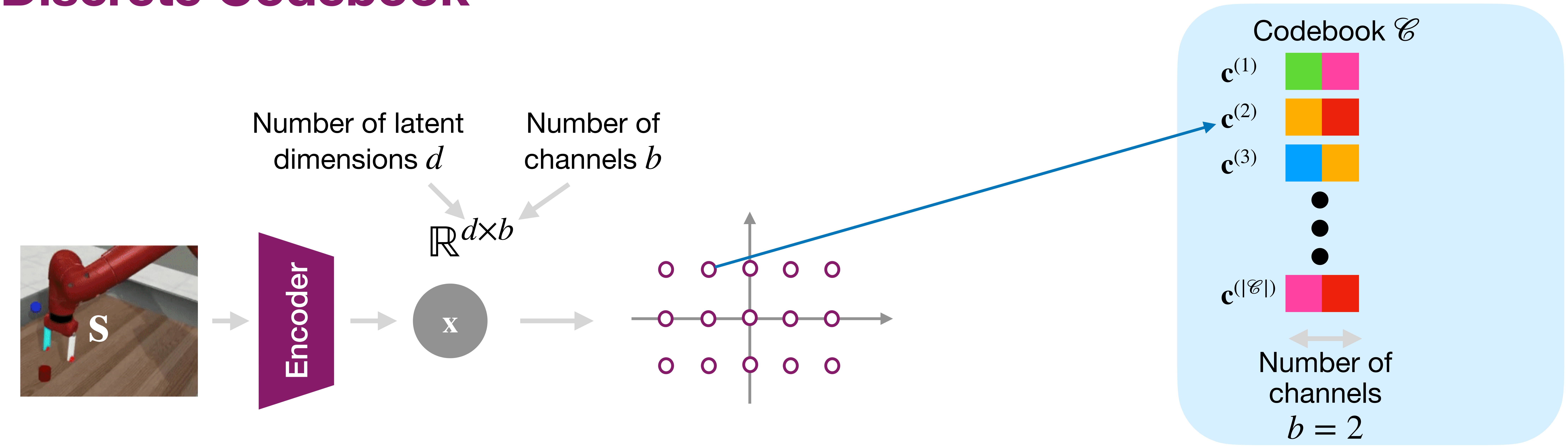
DCWM: Discrete Codebook World Model

Discrete Codebook



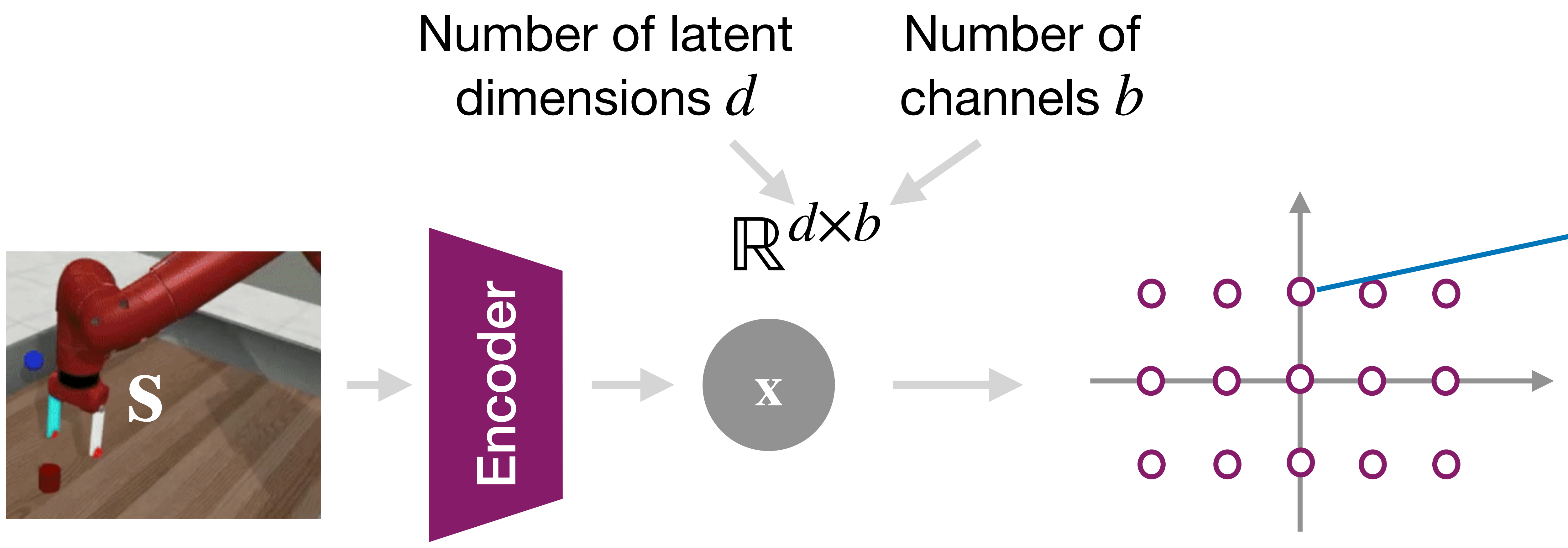
DCWM: Discrete Codebook World Model

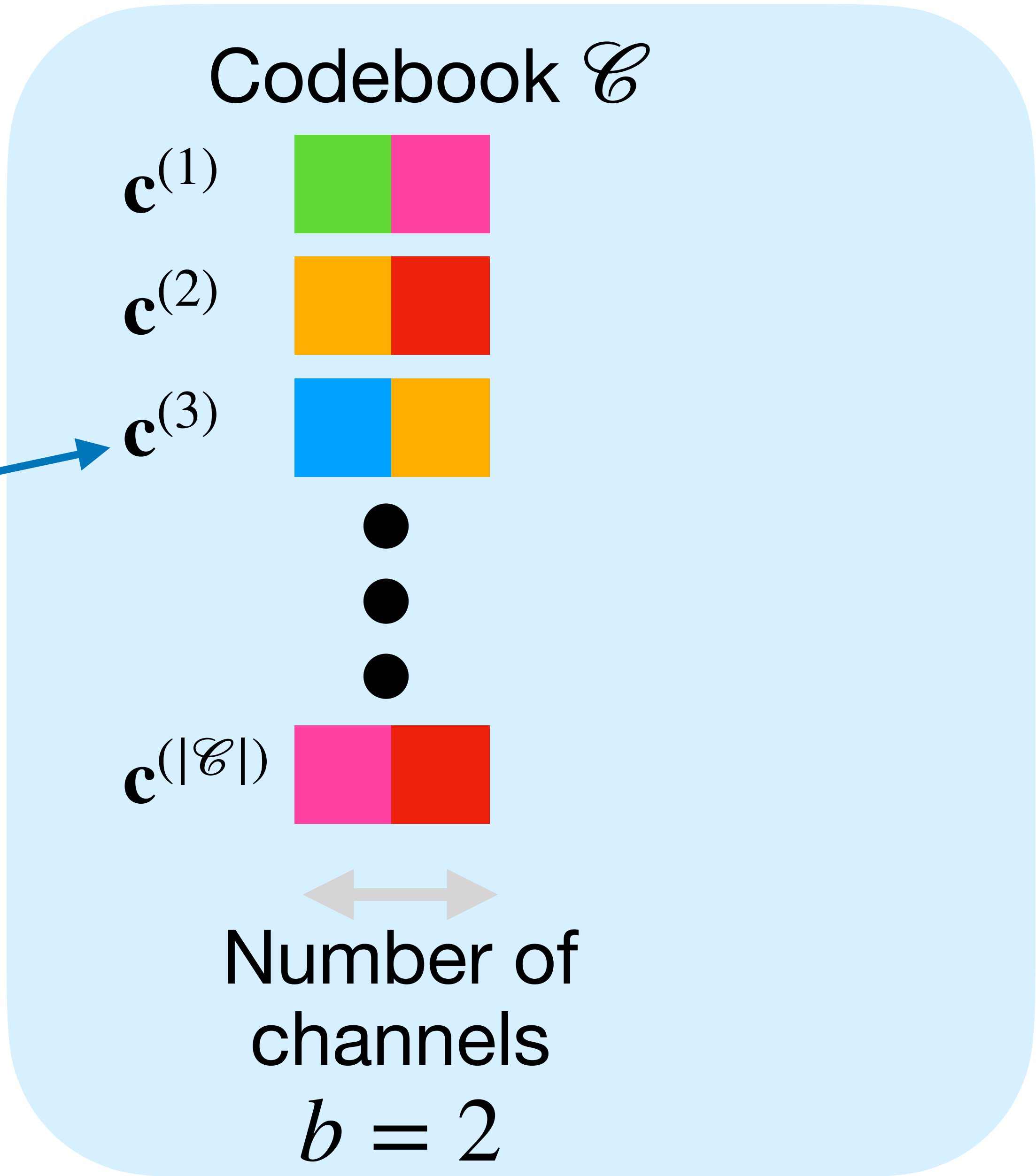
Discrete Codebook



DCWM: Discrete Codebook World Model

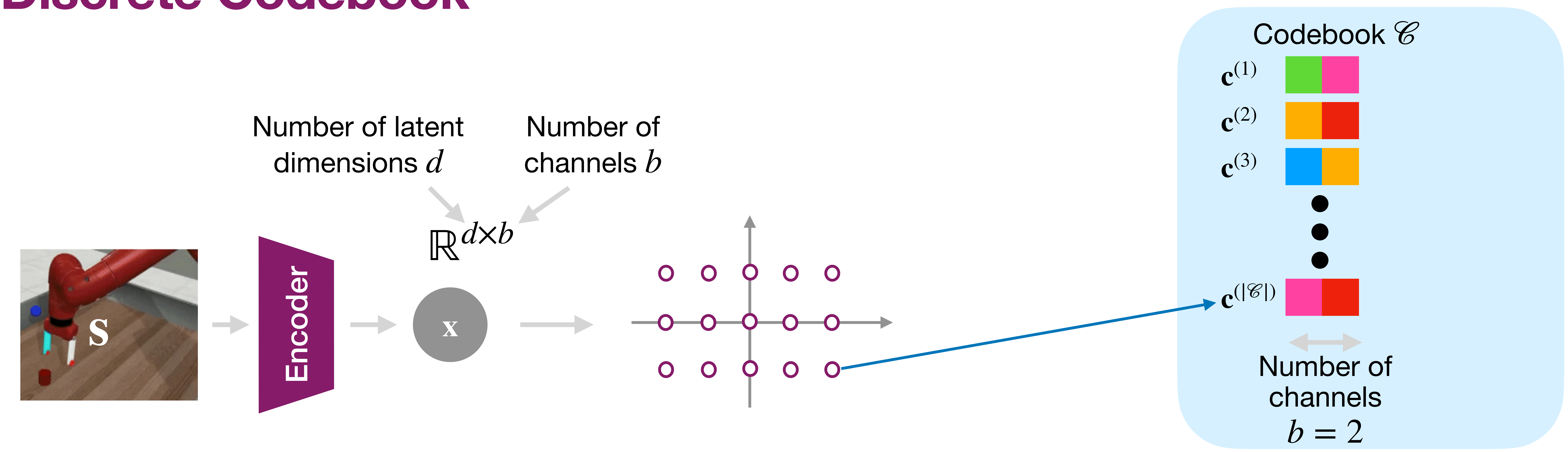
Discrete Codebook





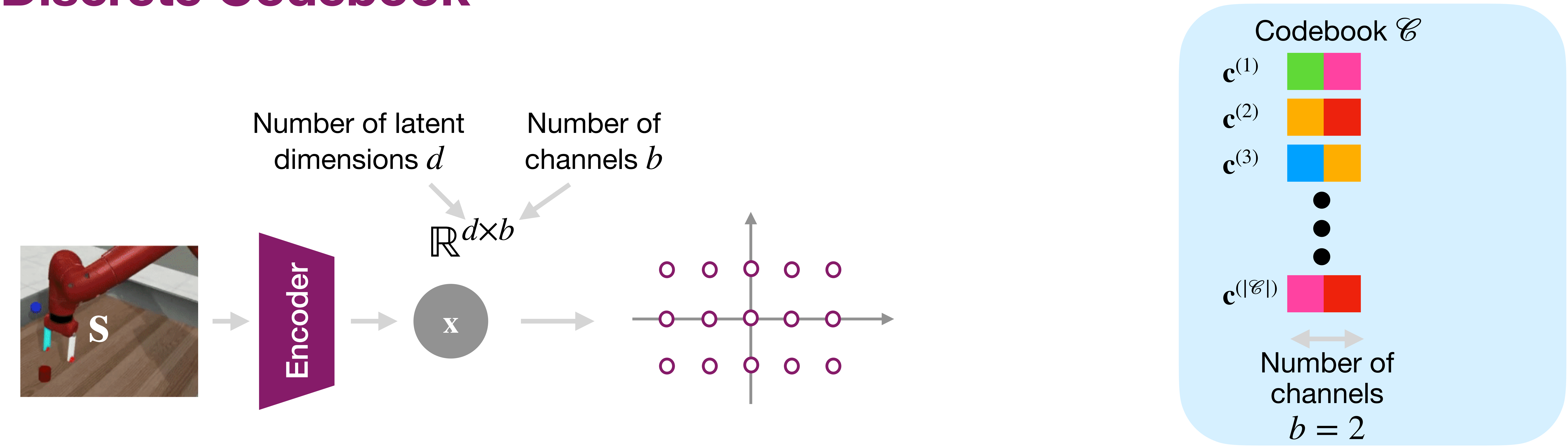
DCWM: Discrete Codebook World Model

Discrete Codebook



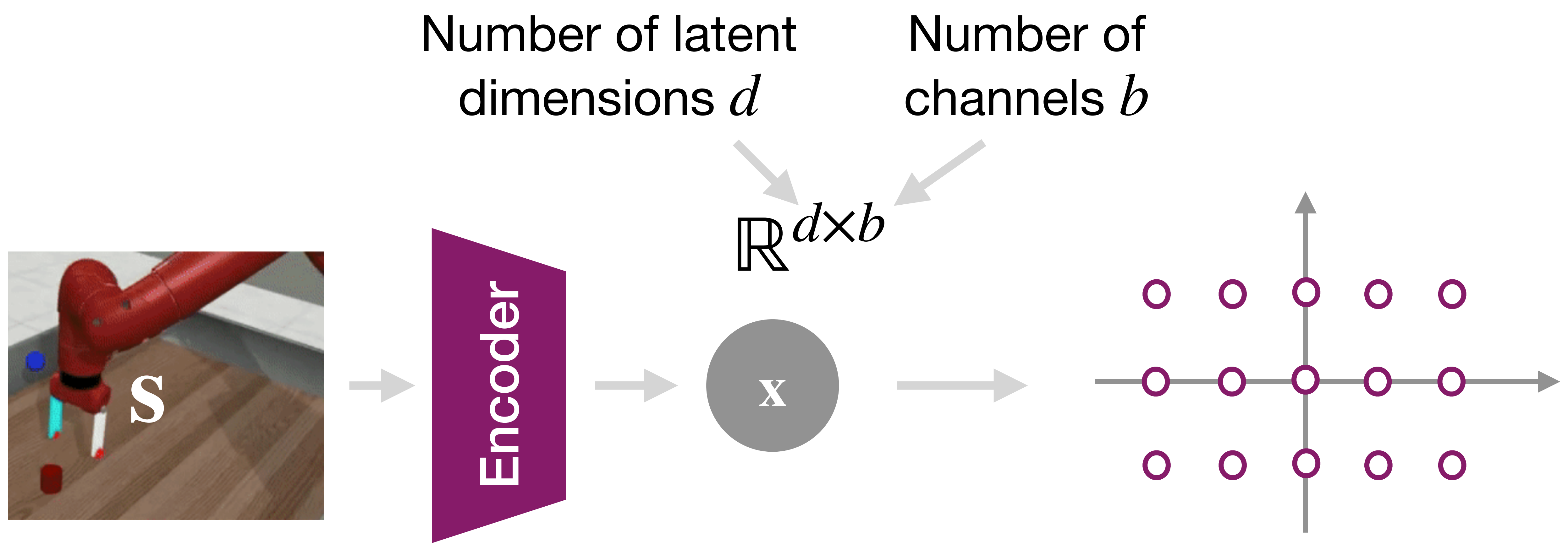
DCWM: Discrete Codebook World Model

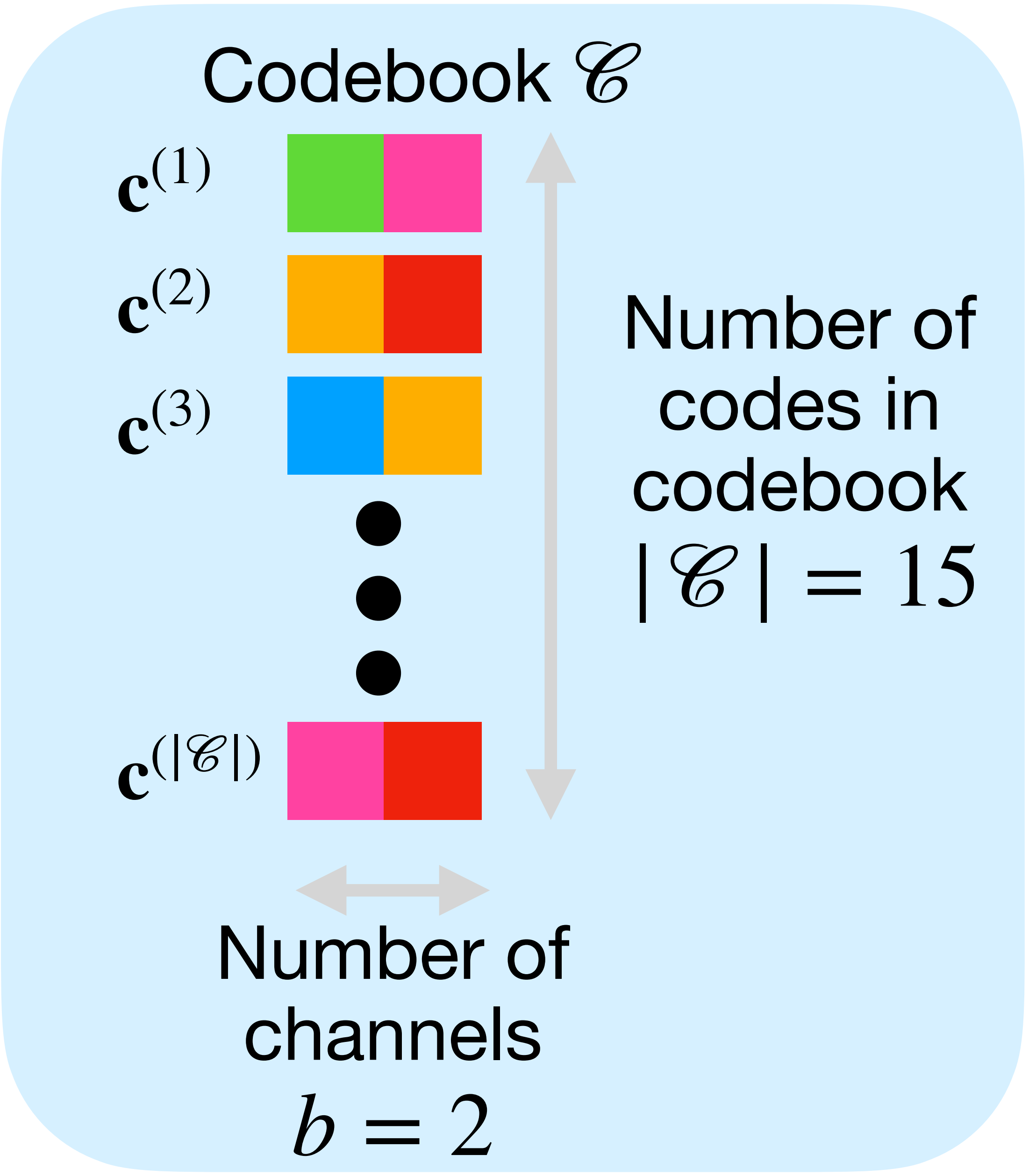
Discrete Codebook



DCWM: Discrete Codebook World Model

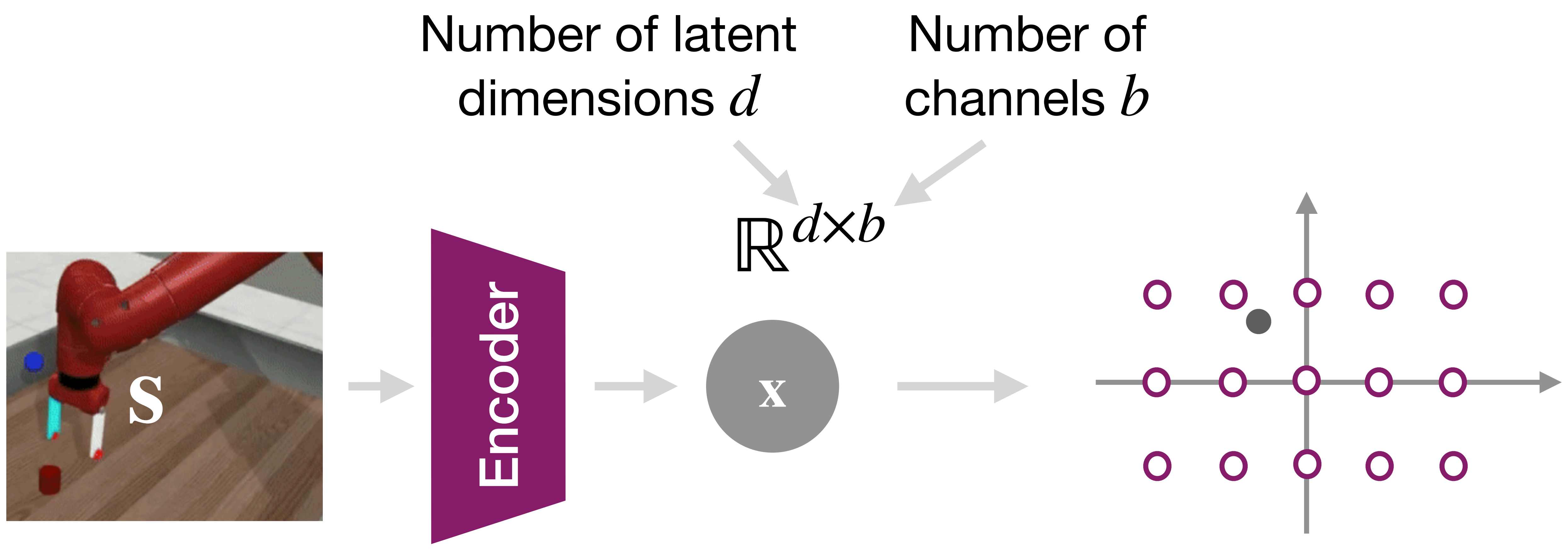
Discrete Codebook

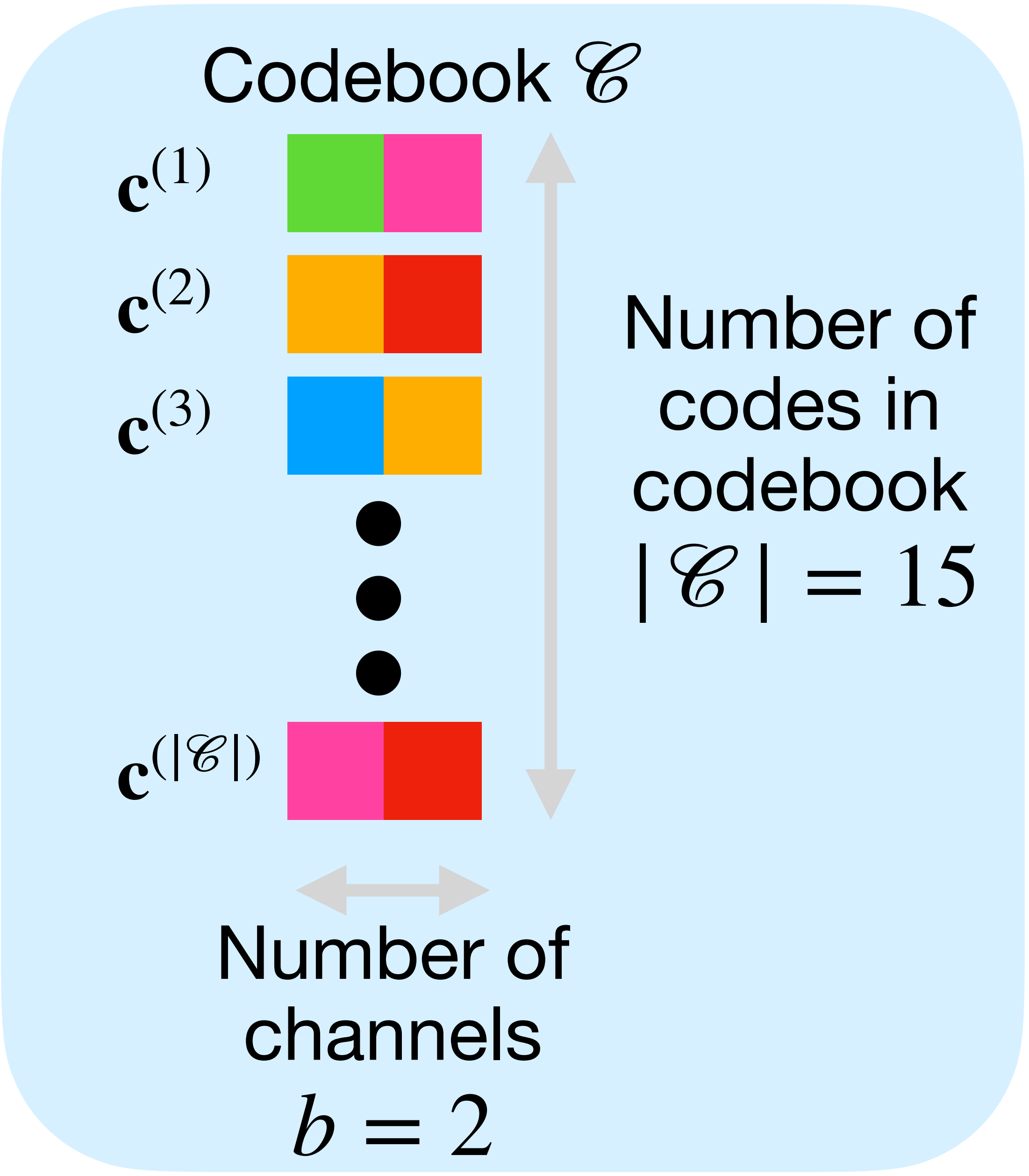




DCWM: Discrete Codebook World Model

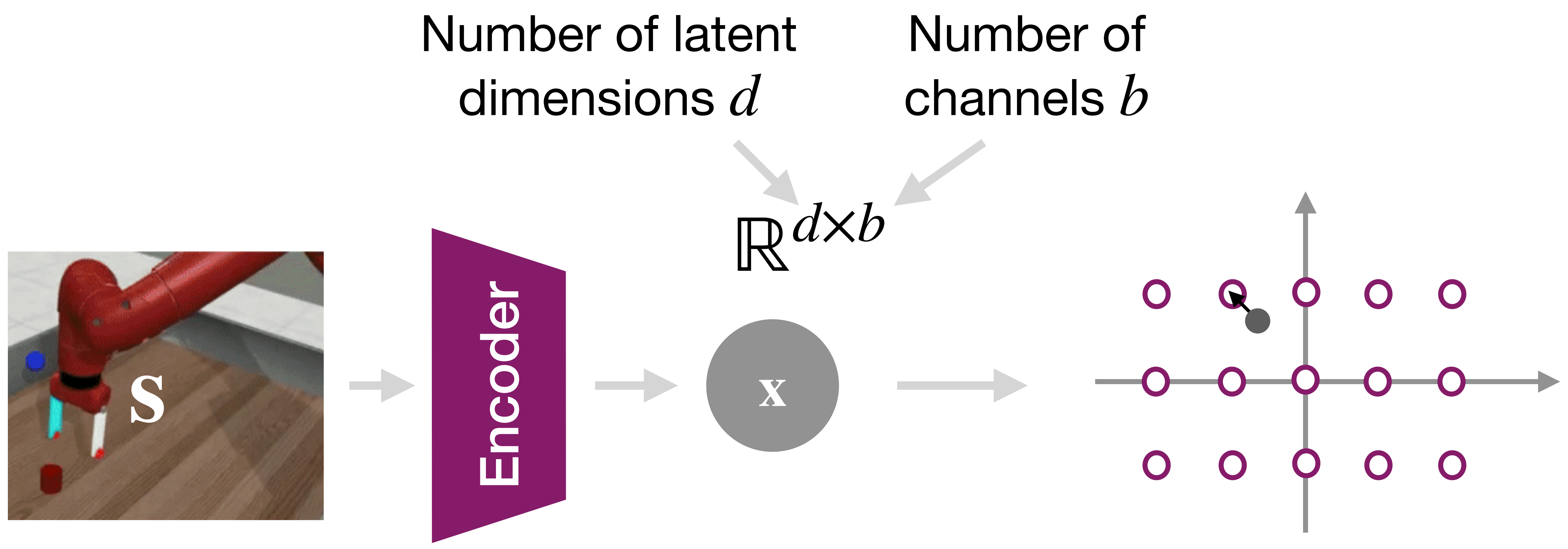
Discrete Codebook

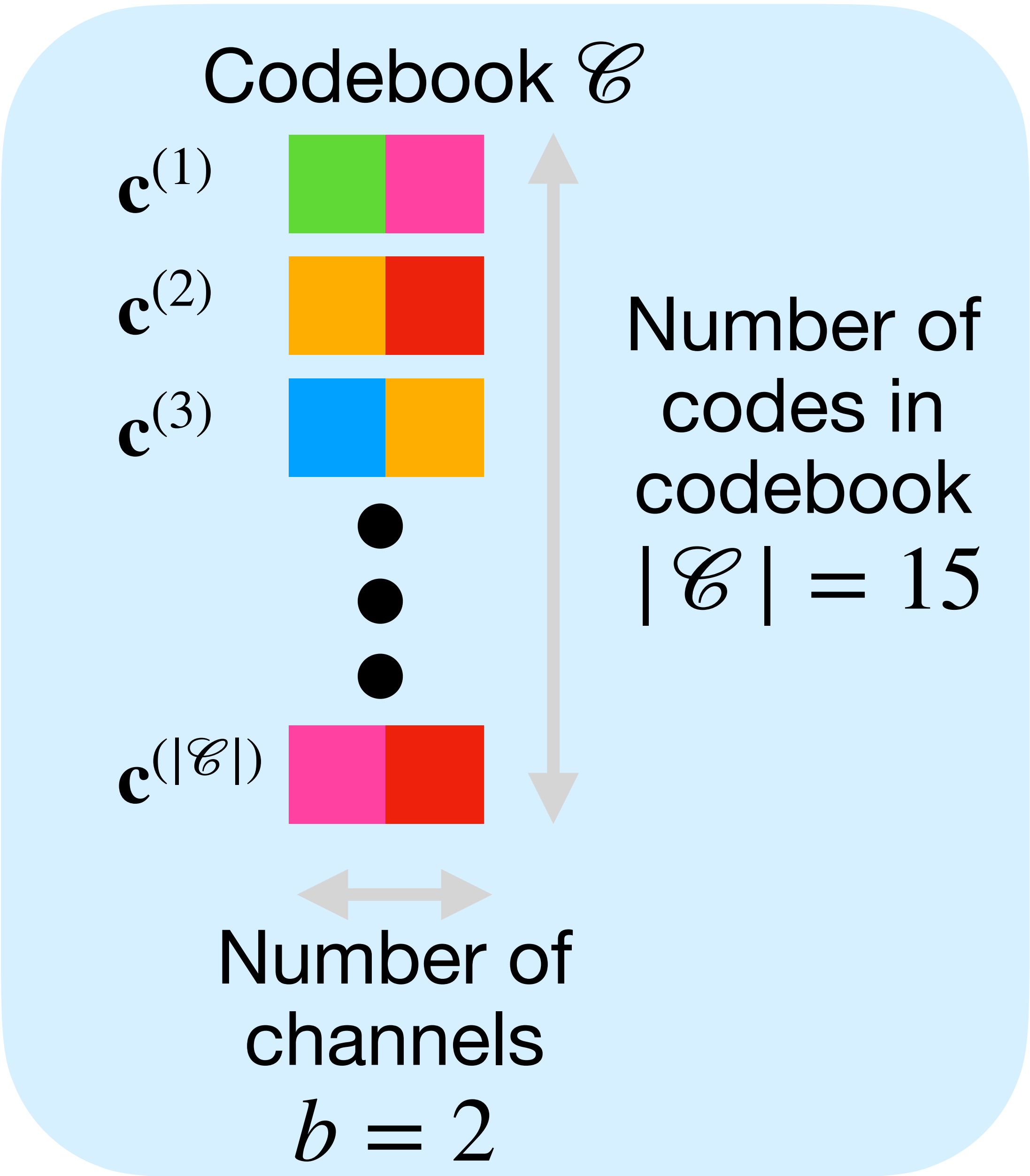




DCWM: Discrete Codebook World Model

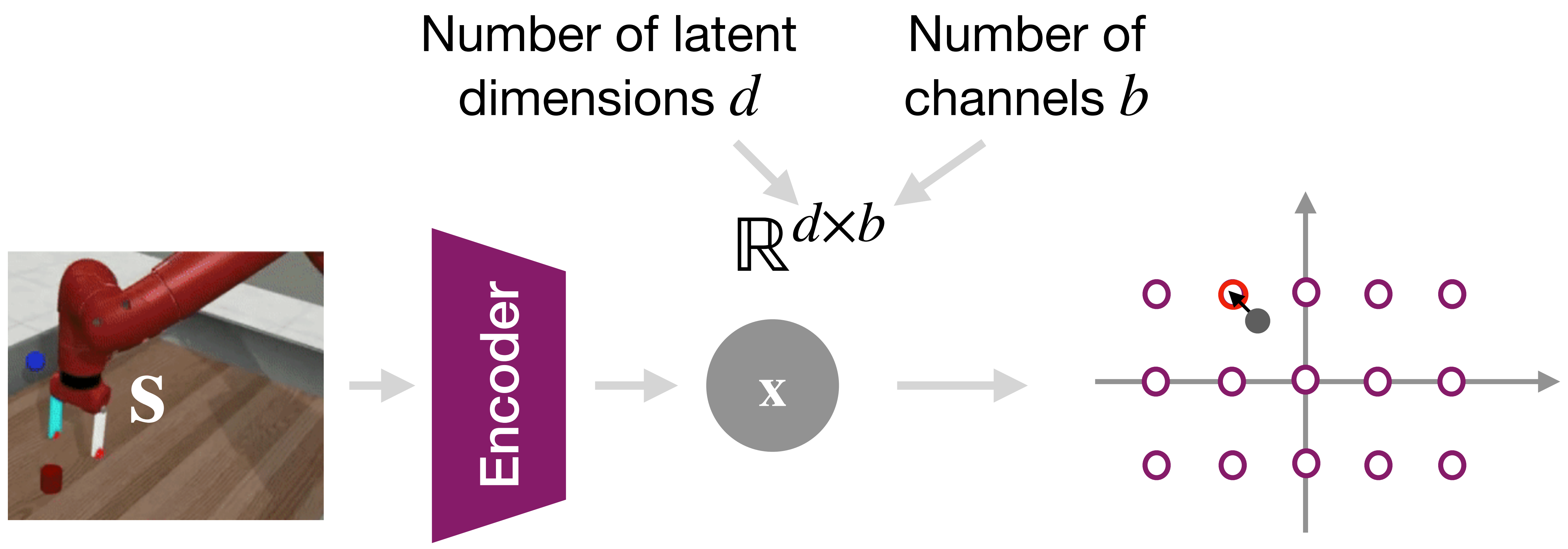
Discrete Codebook

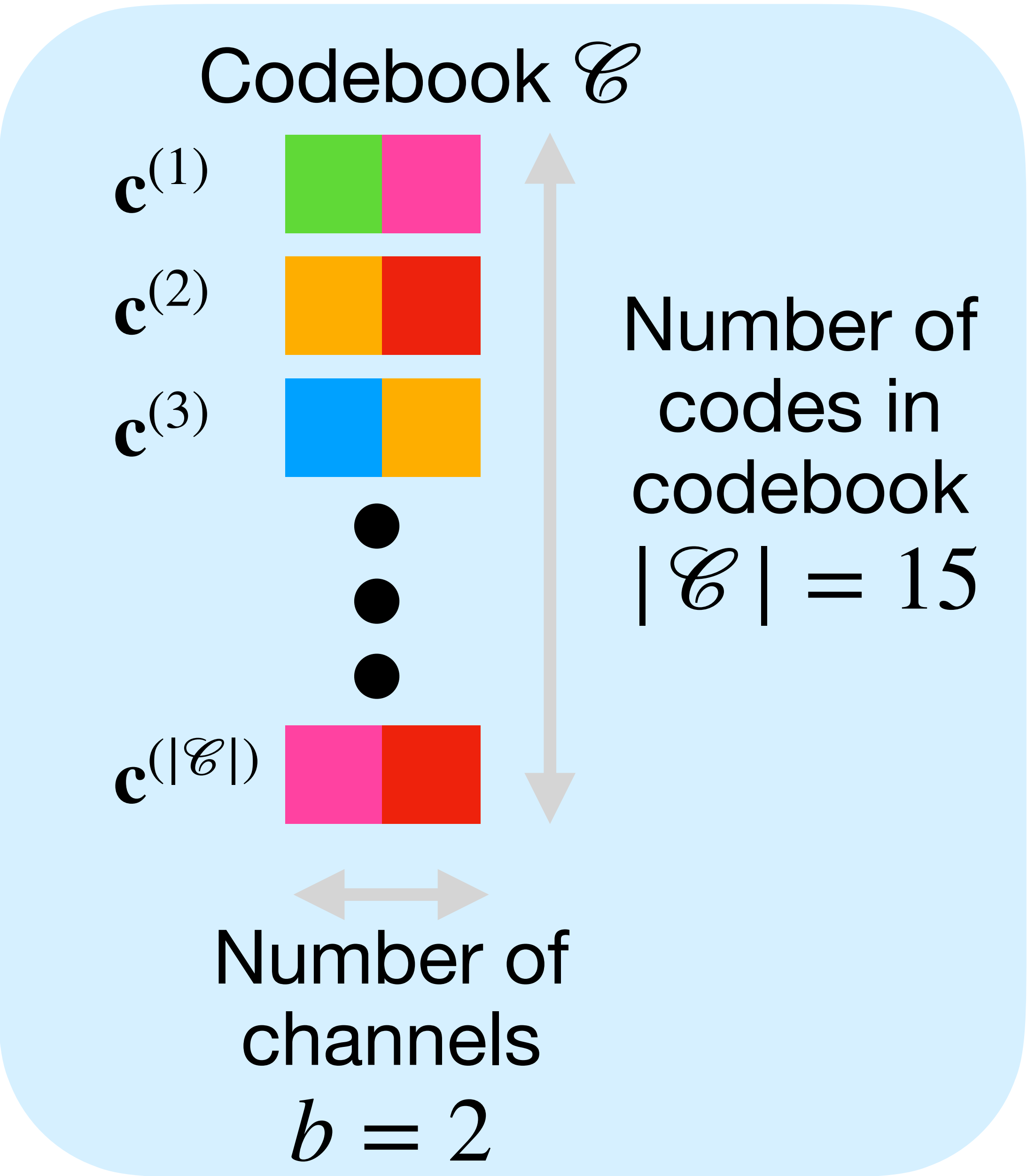




DCWM: Discrete Codebook World Model

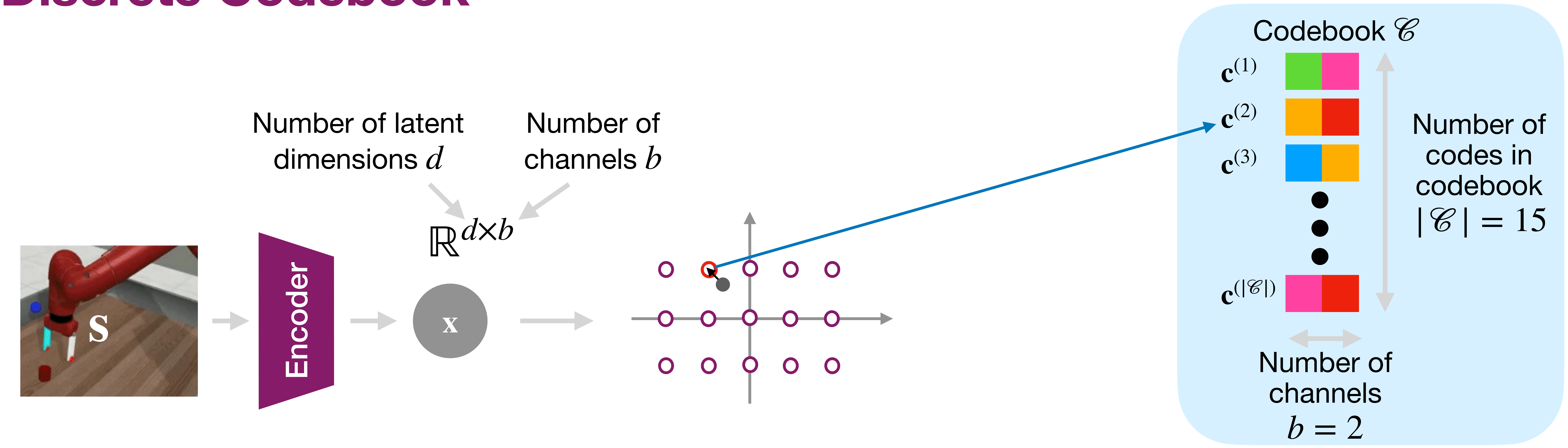
Discrete Codebook





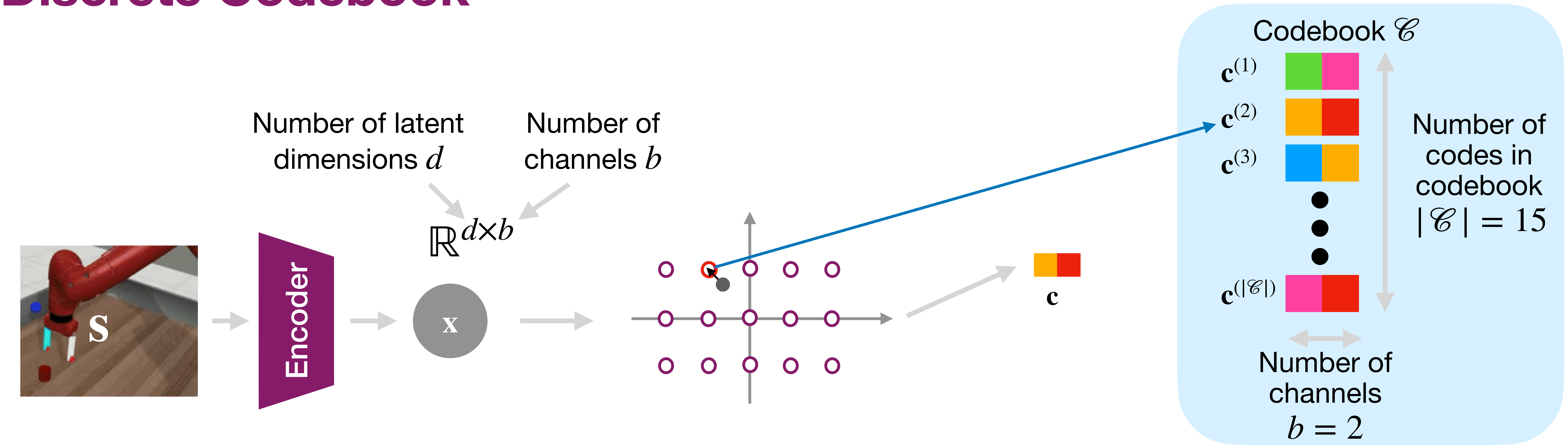
DCWM: Discrete Codebook World Model

Discrete Codebook



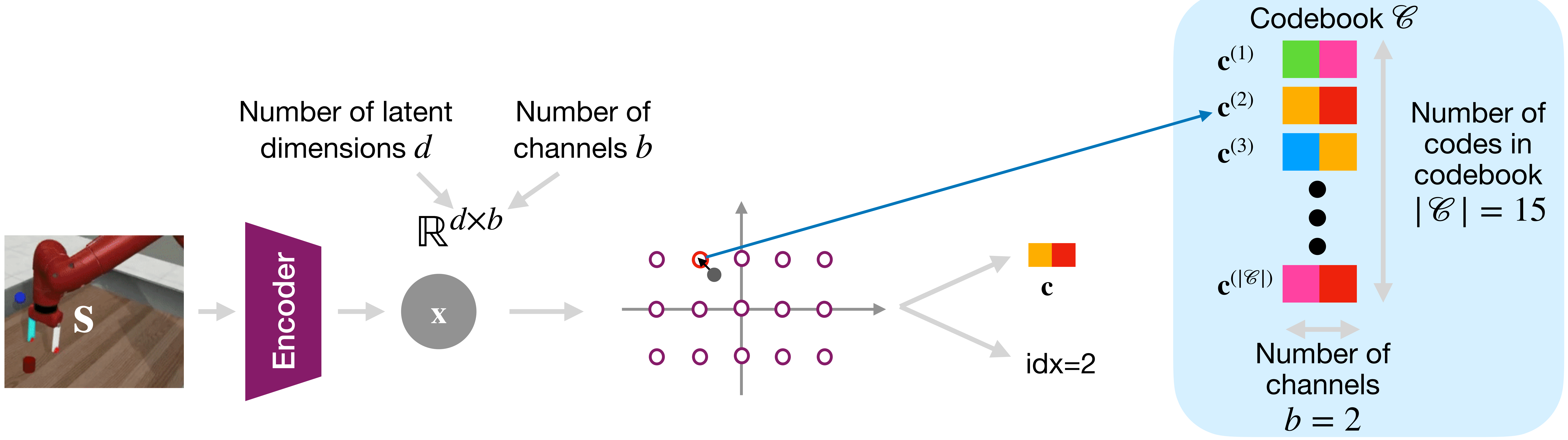
DCWM: Discrete Codebook World Model

Discrete Codebook



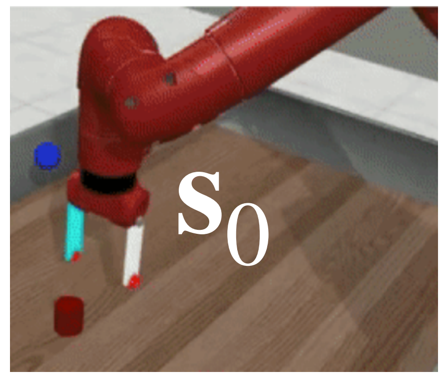
DCWM: Discrete Codebook World Model

Discrete Codebook

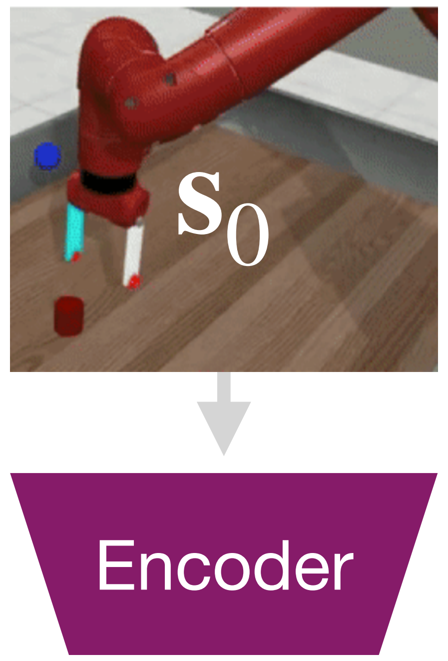


DCWM: World Model Training

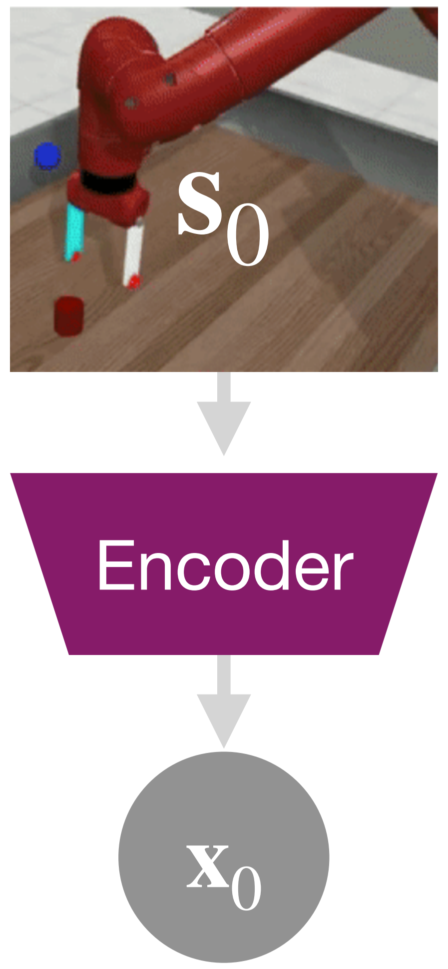
DCWM: World Model Training



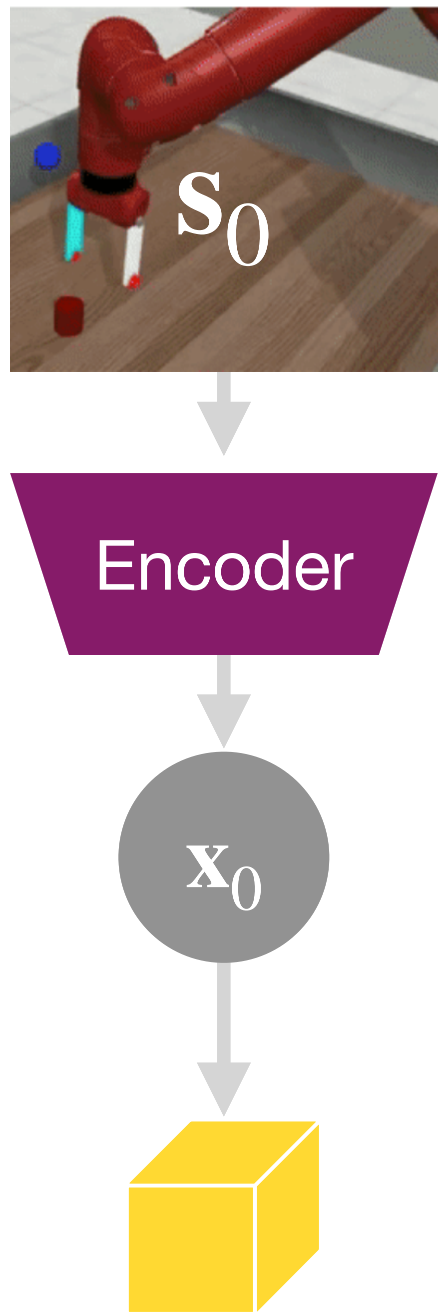
DCWM: World Model Training



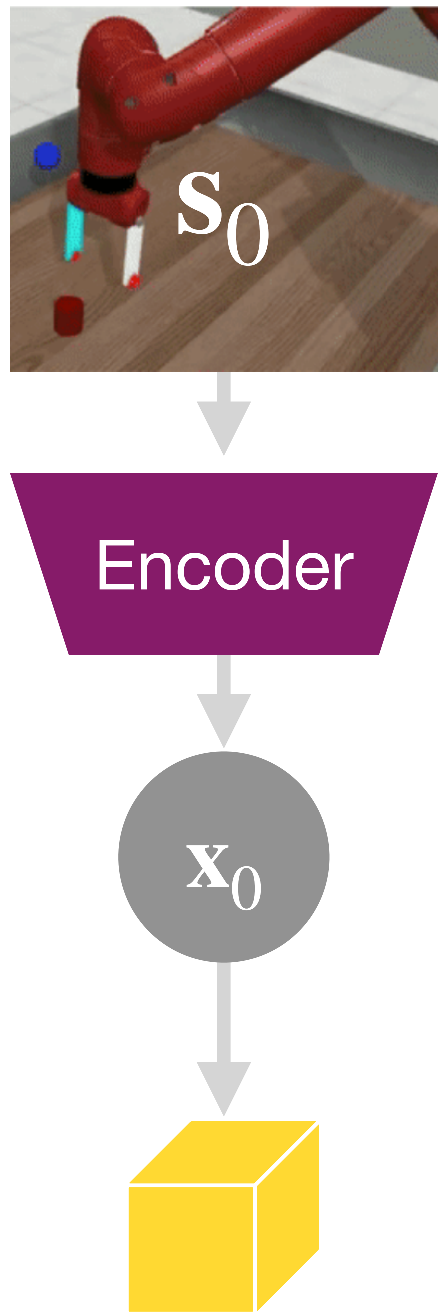
DCWM: World Model Training

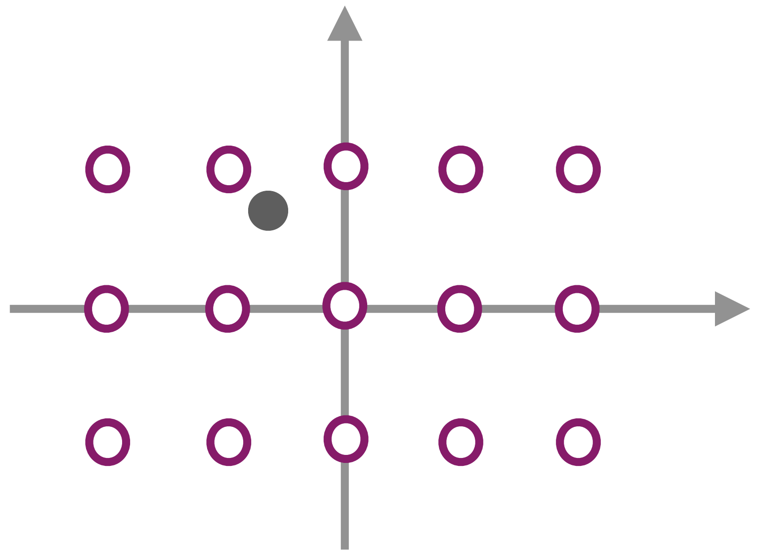


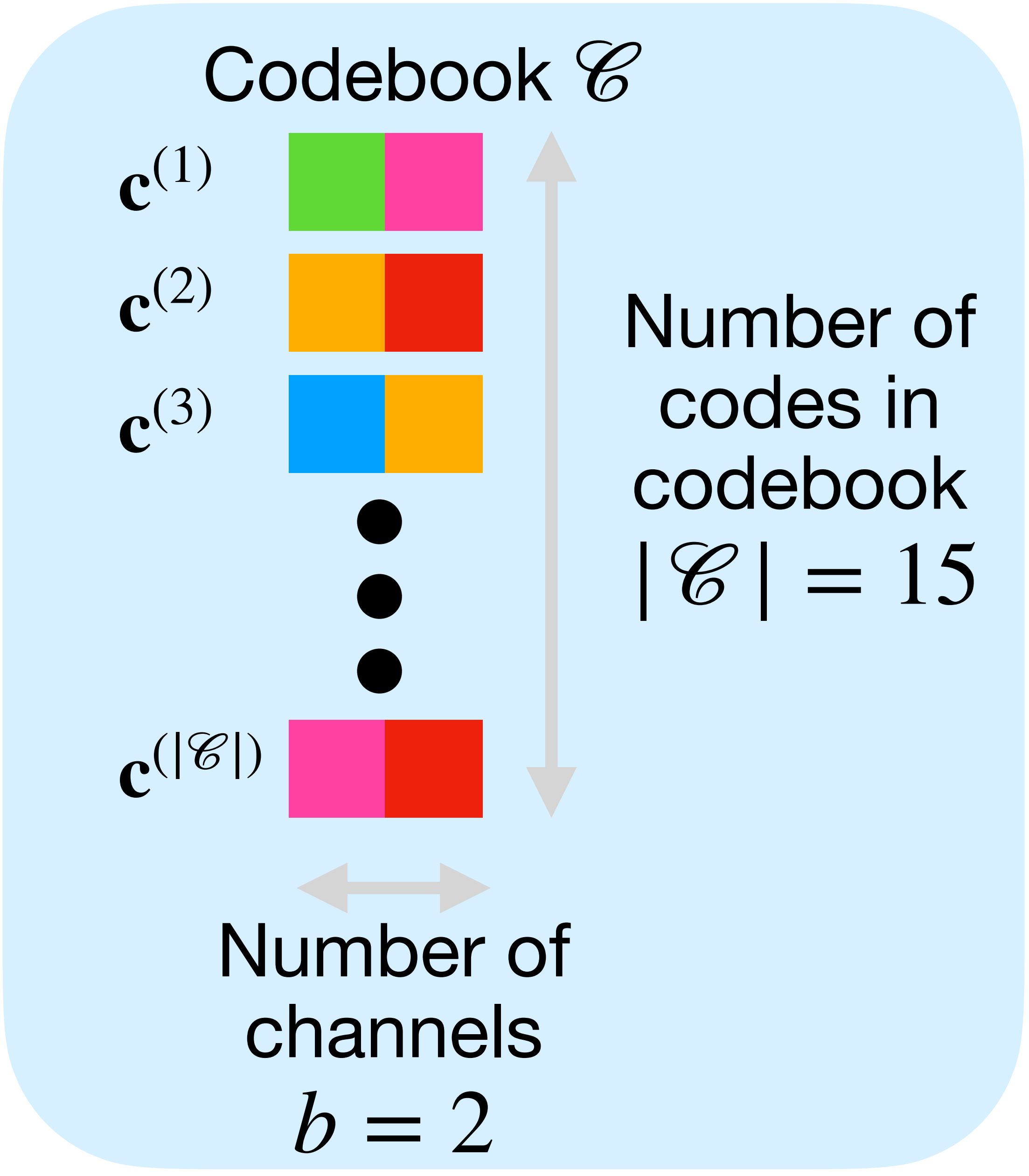
DCWM: World Model Training



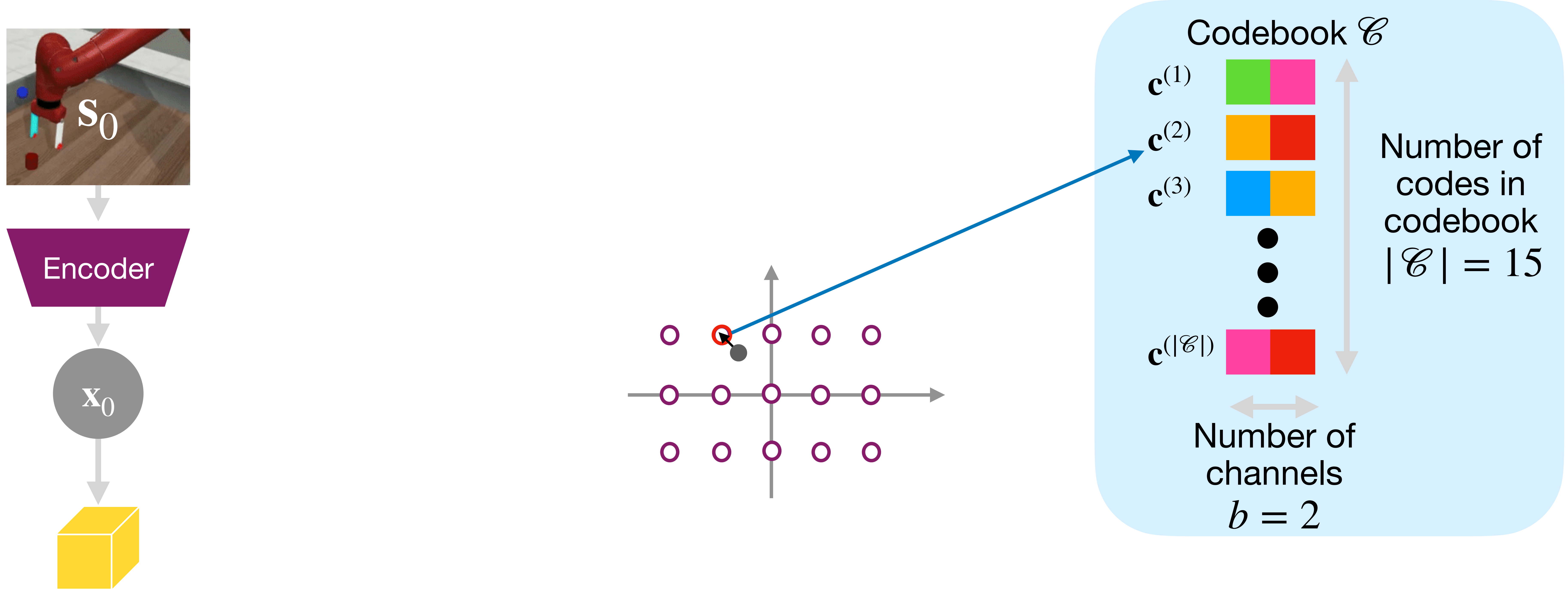
DCWM: World Model Training



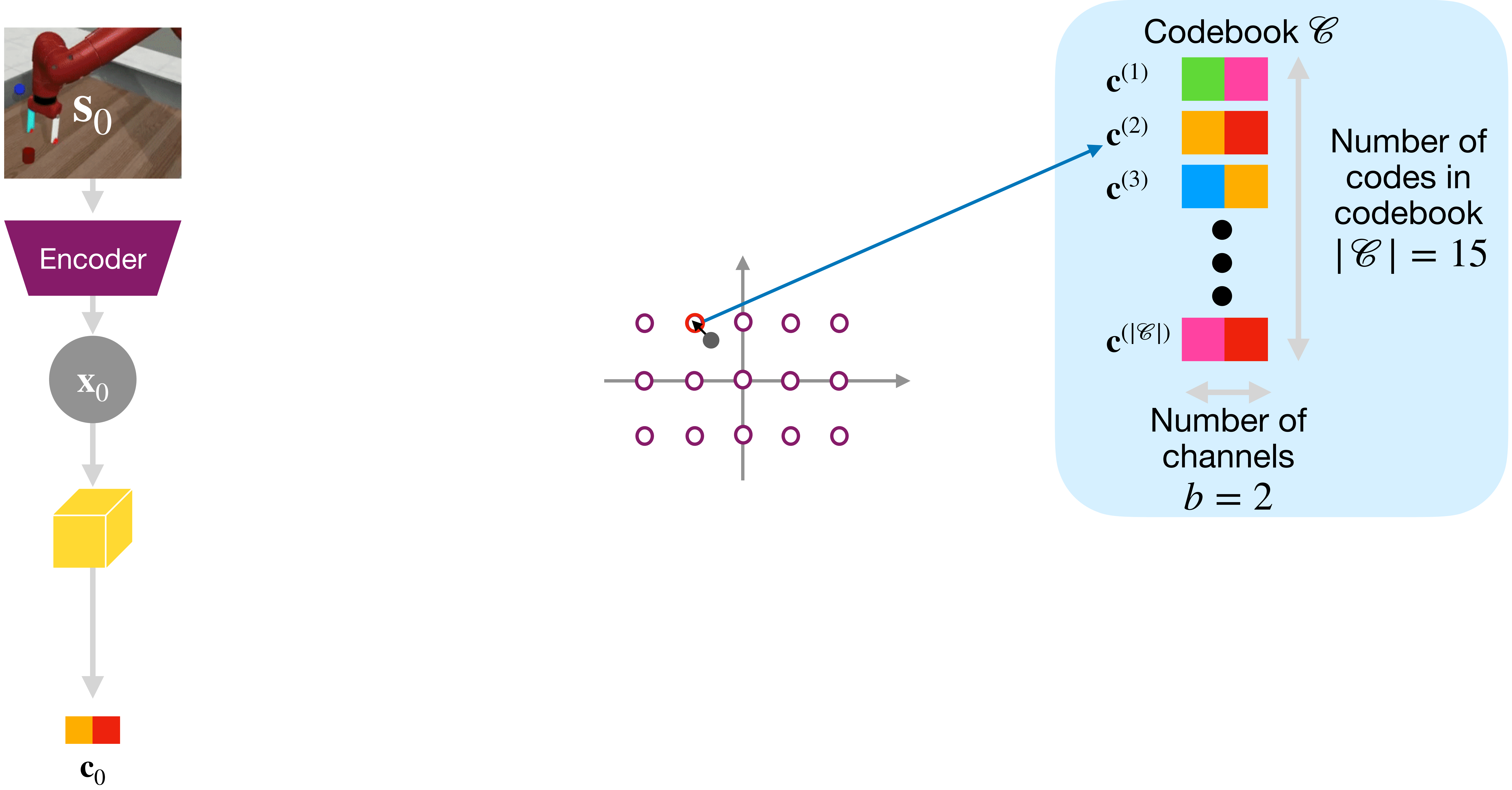




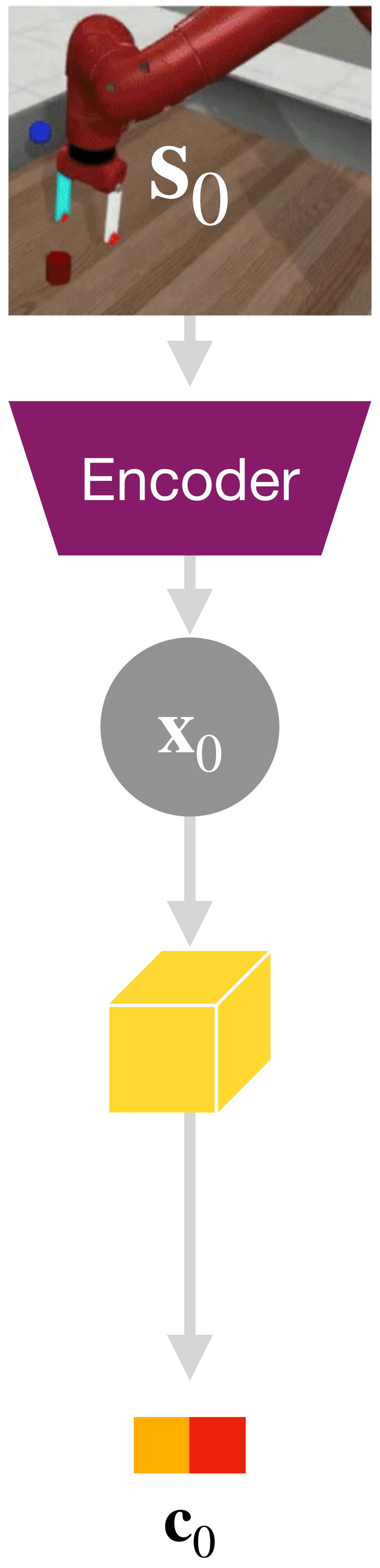
DCWM: World Model Training

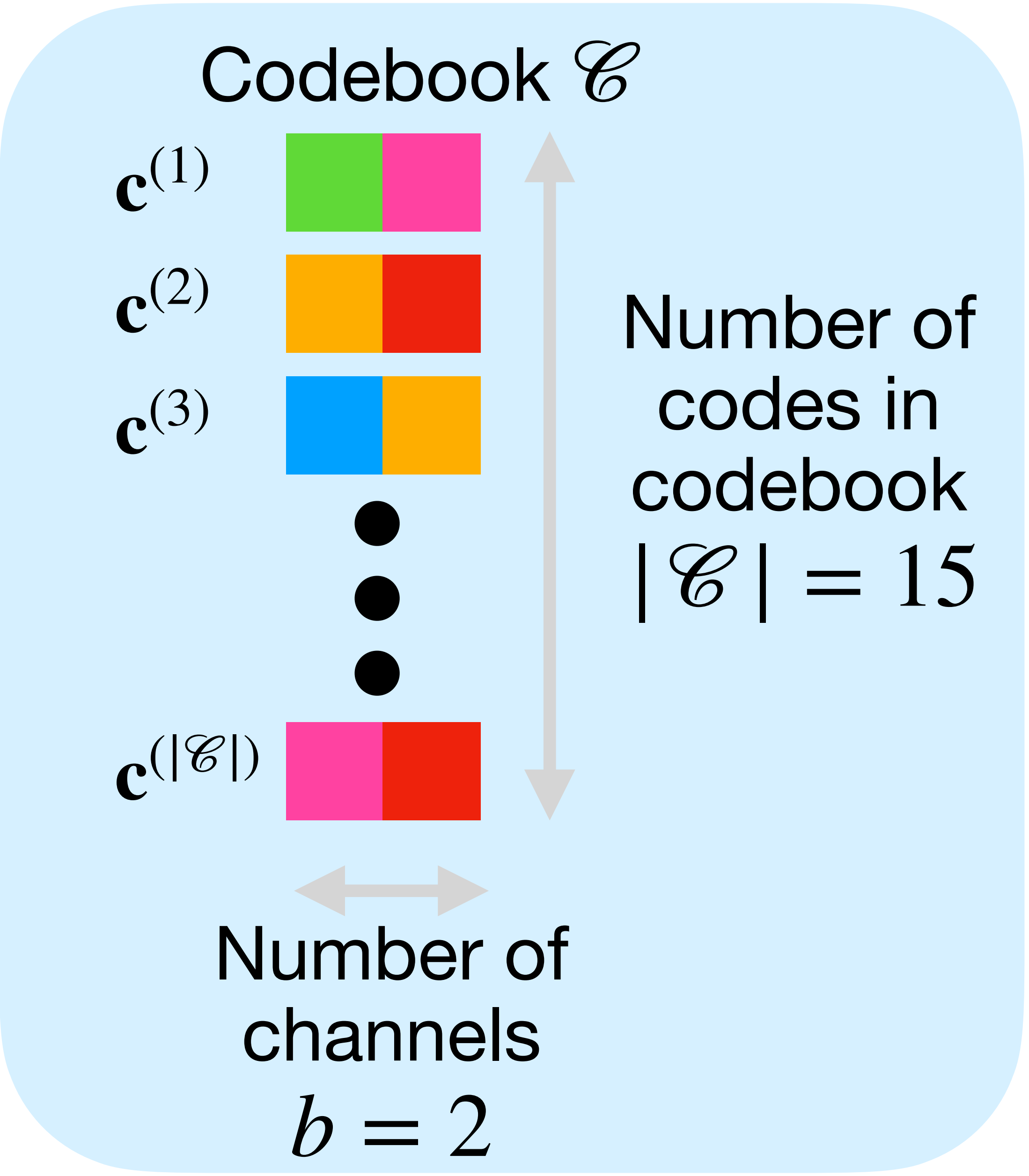


DCWM: World Model Training

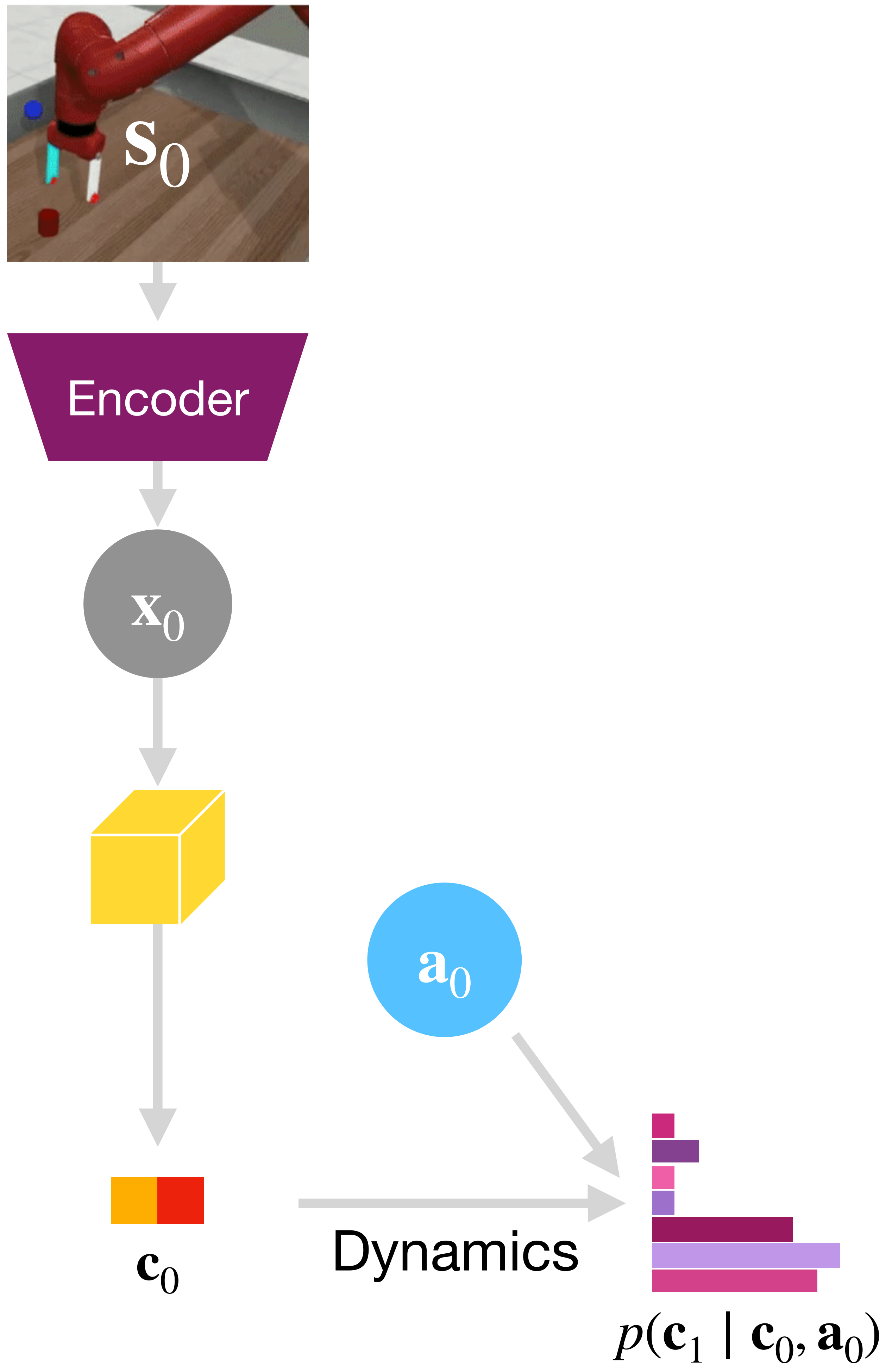


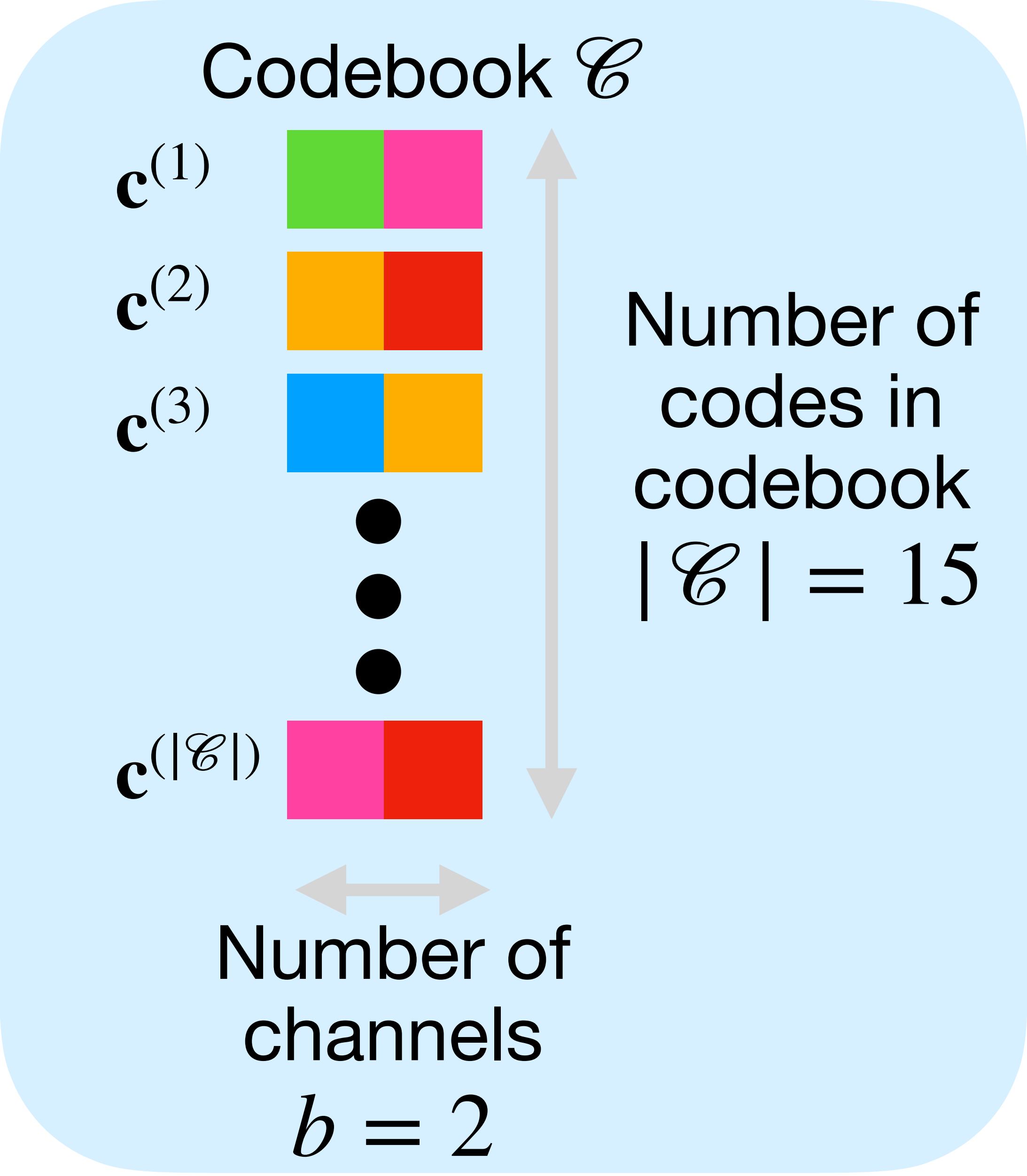
DCWM: World Model Training



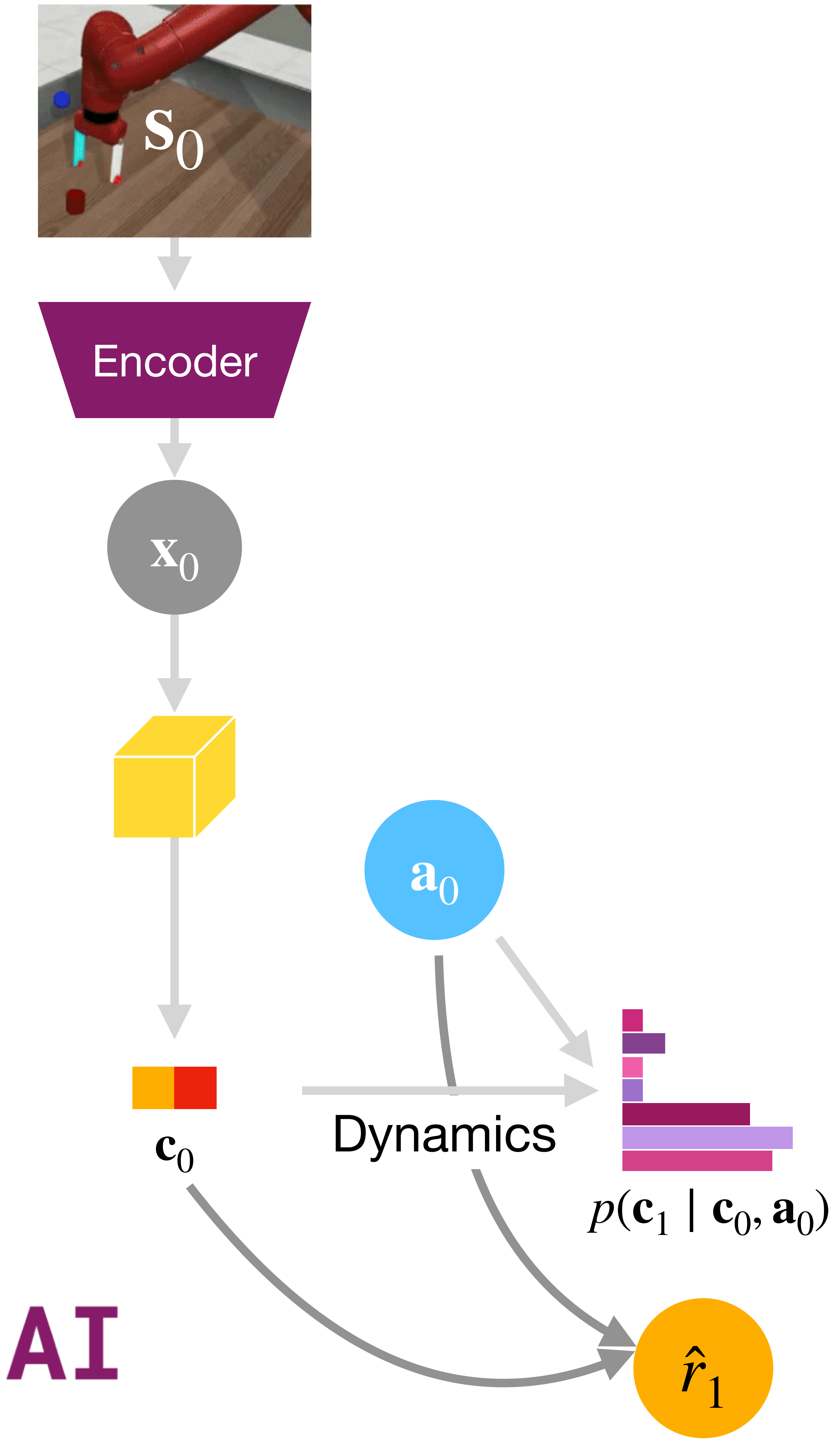


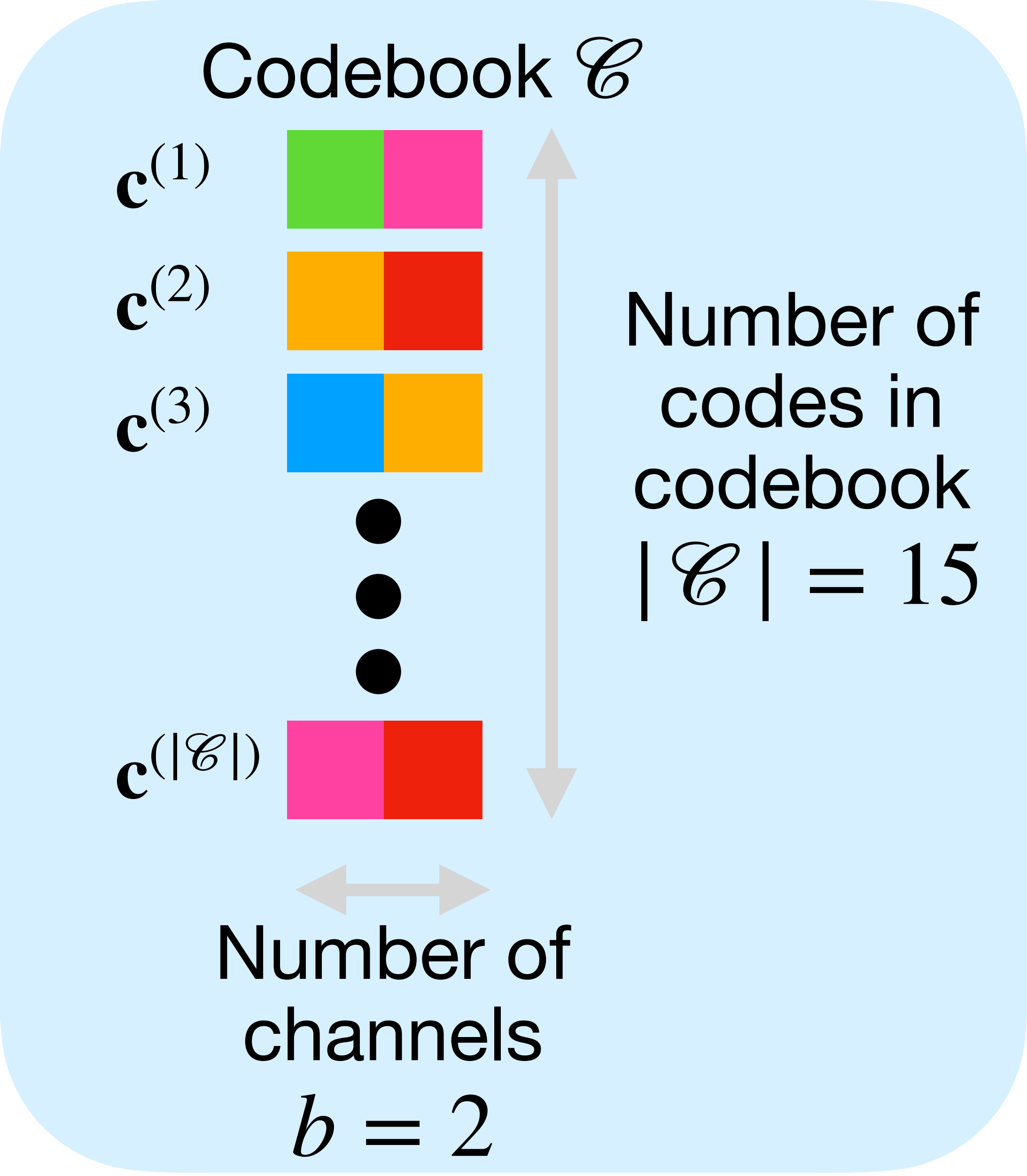
DCWM: World Model Training



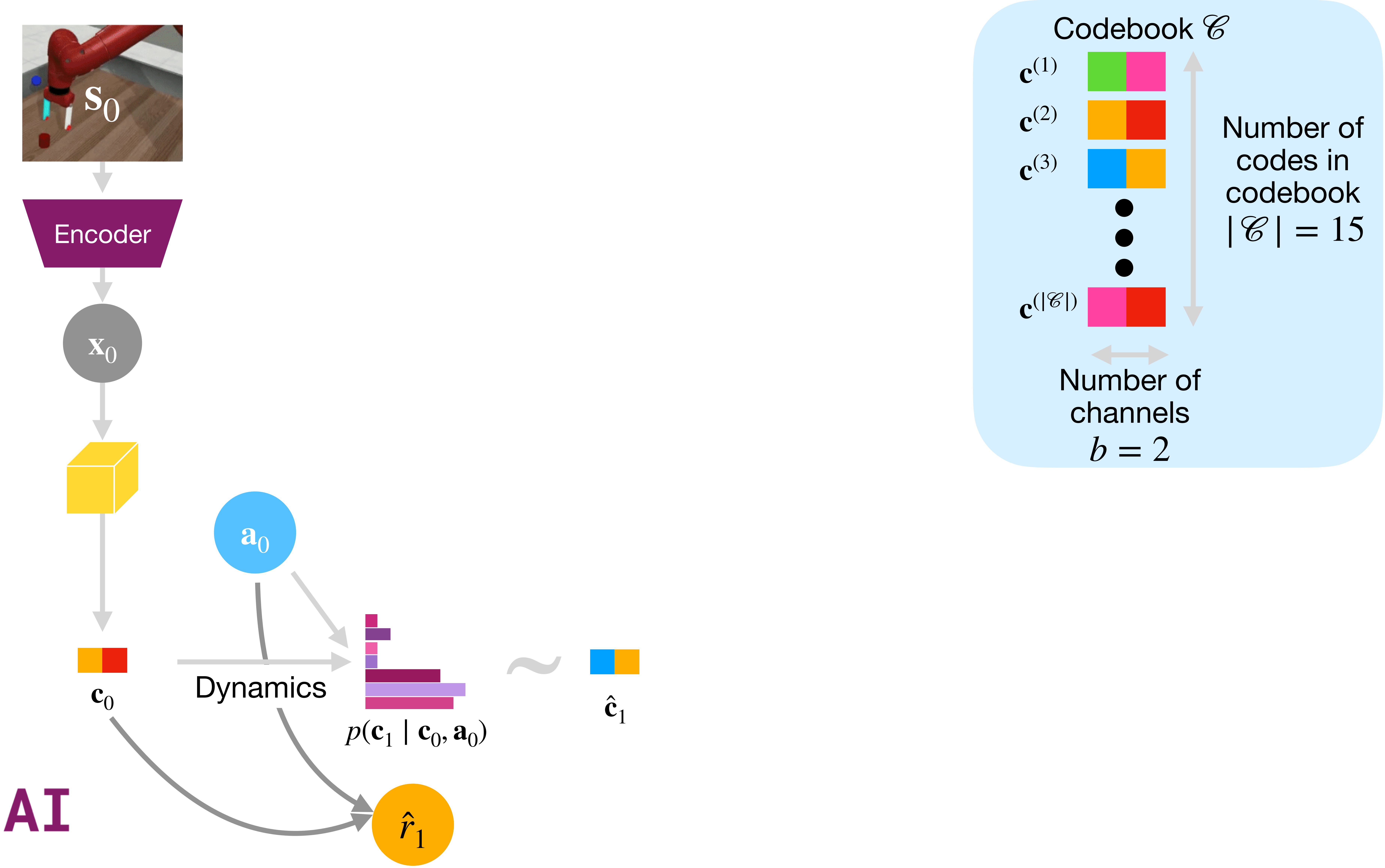


DCWM: World Model Training

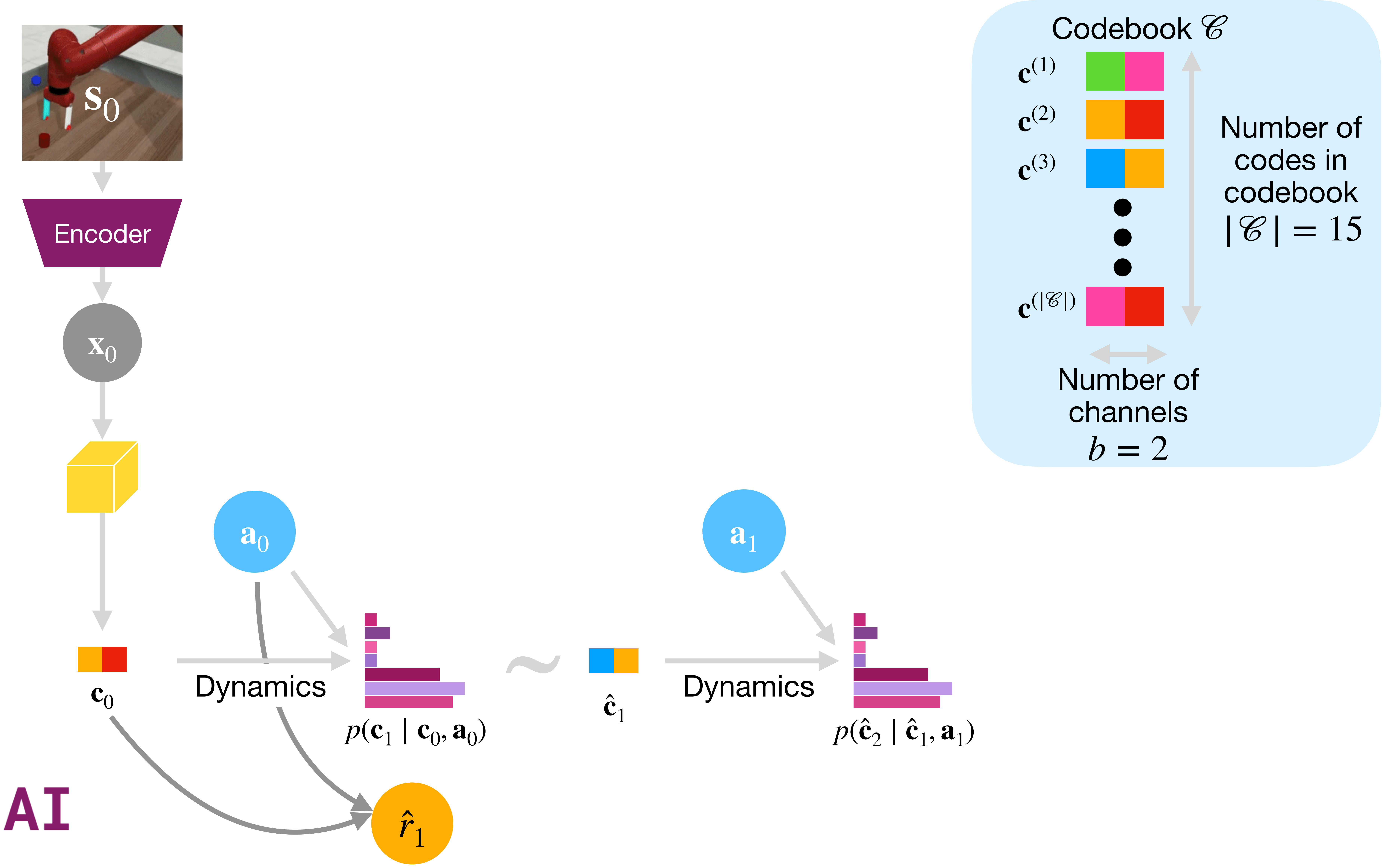




DCWM: World Model Training

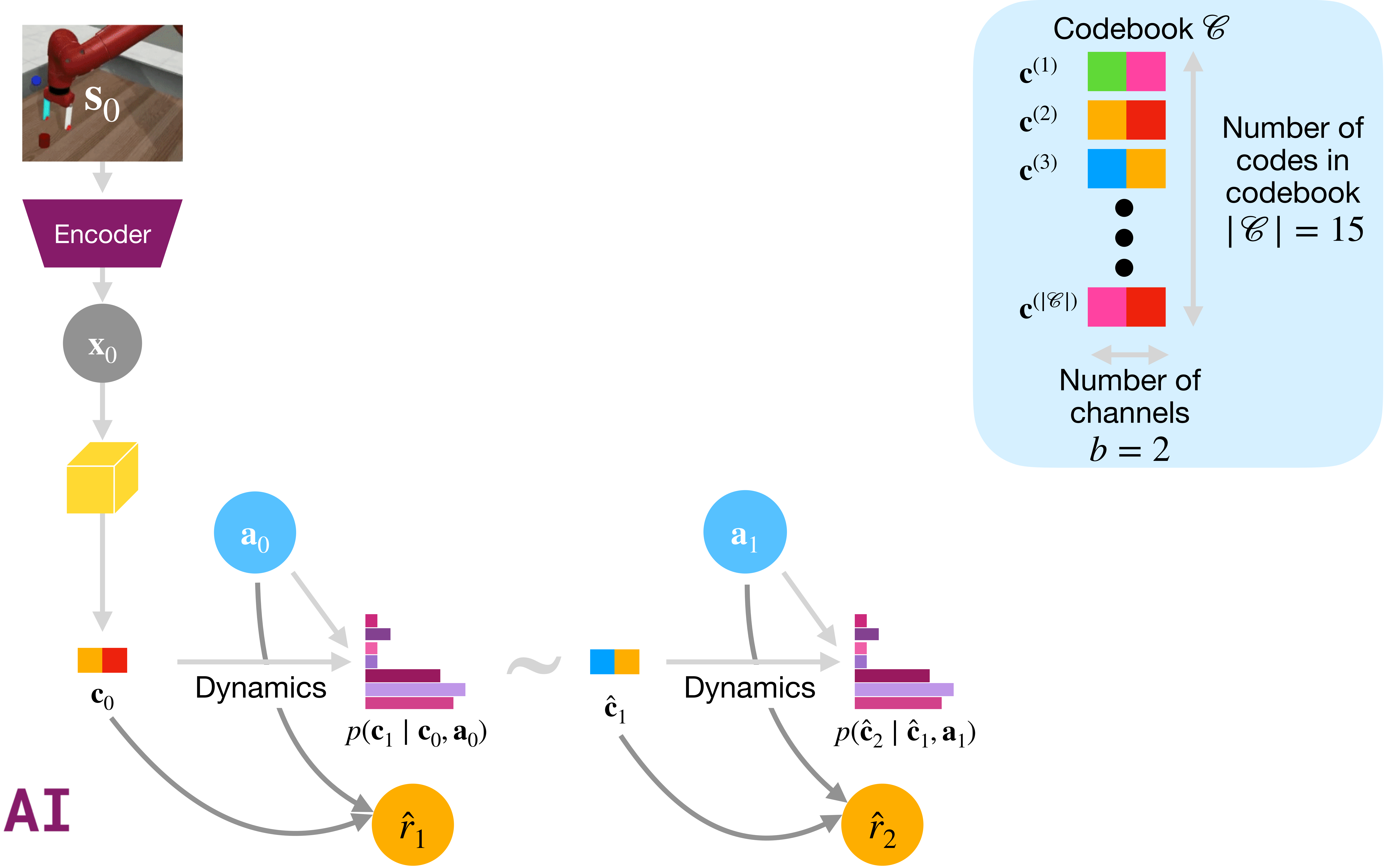


DCWM: World Model Training

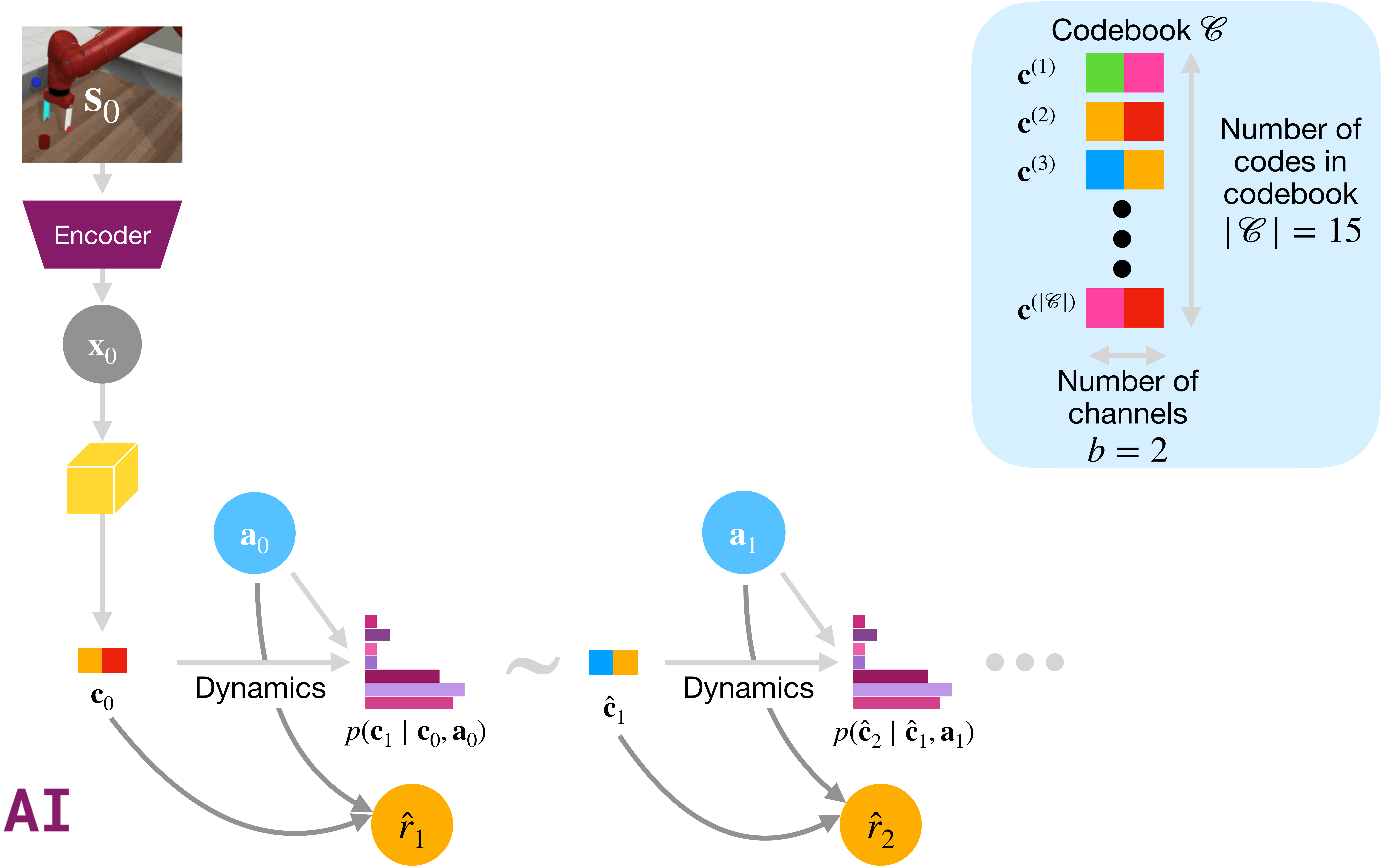


FCAI

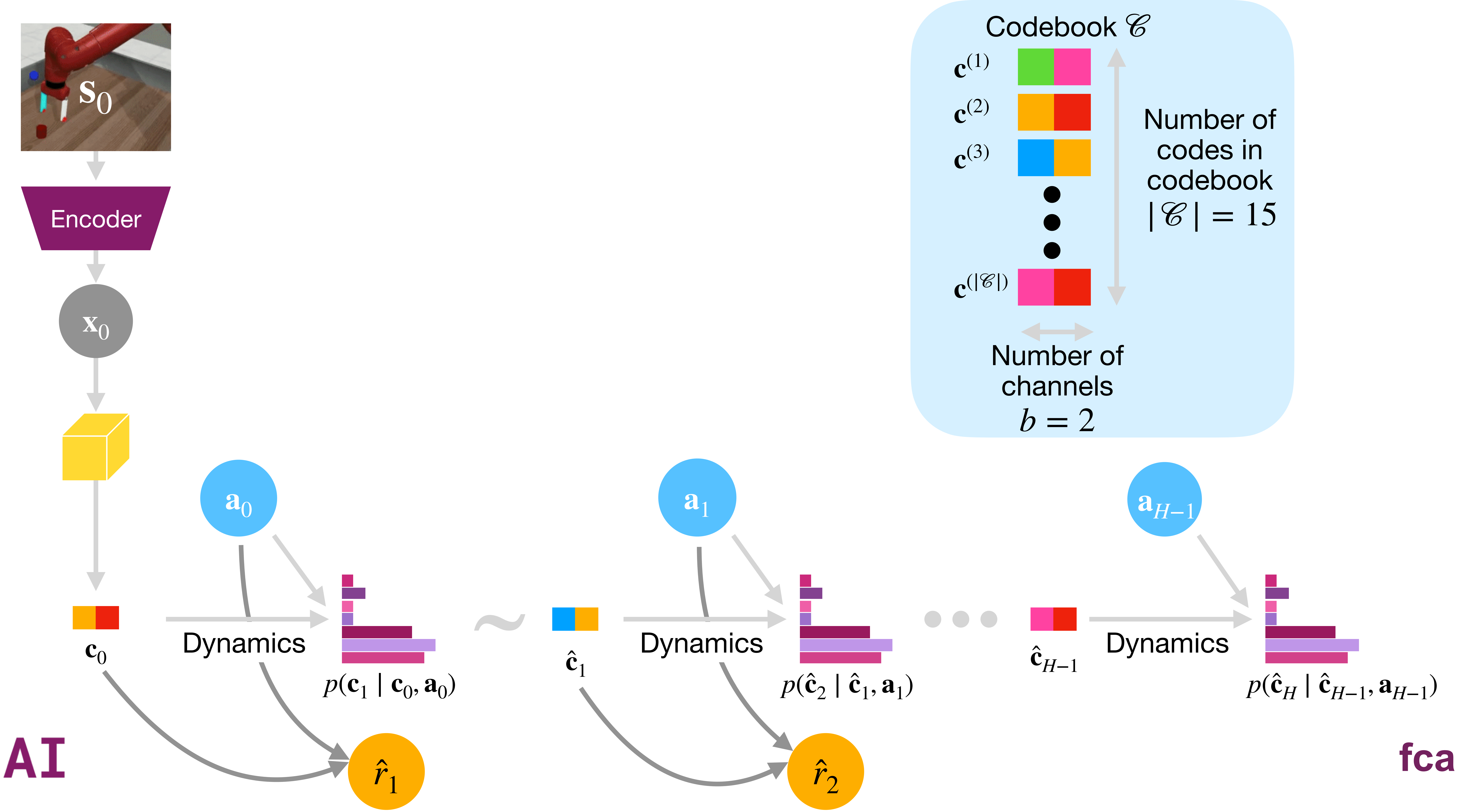
DCWM: World Model Training



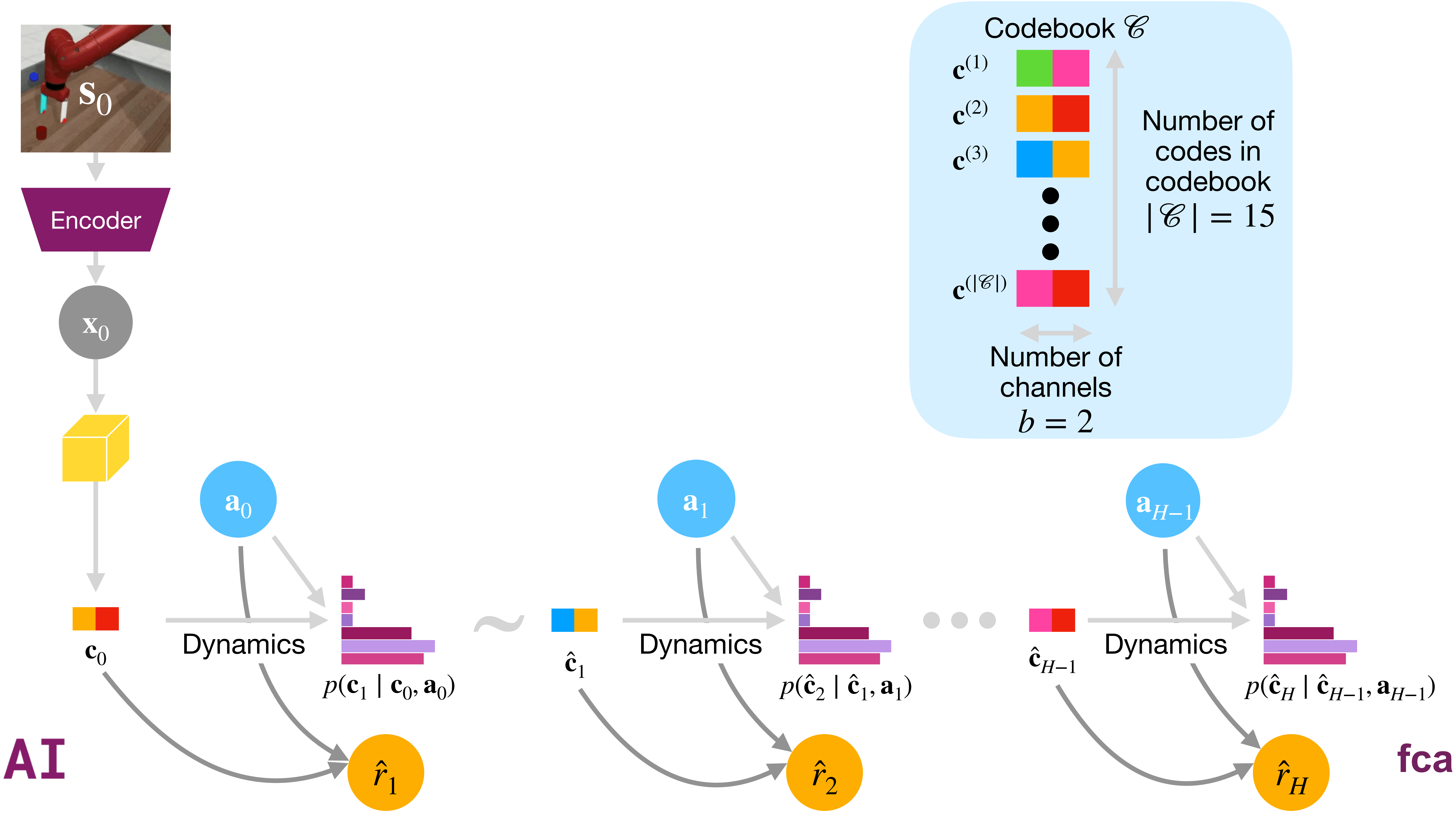
DCWM: World Model Training



DCWM: World Model Training



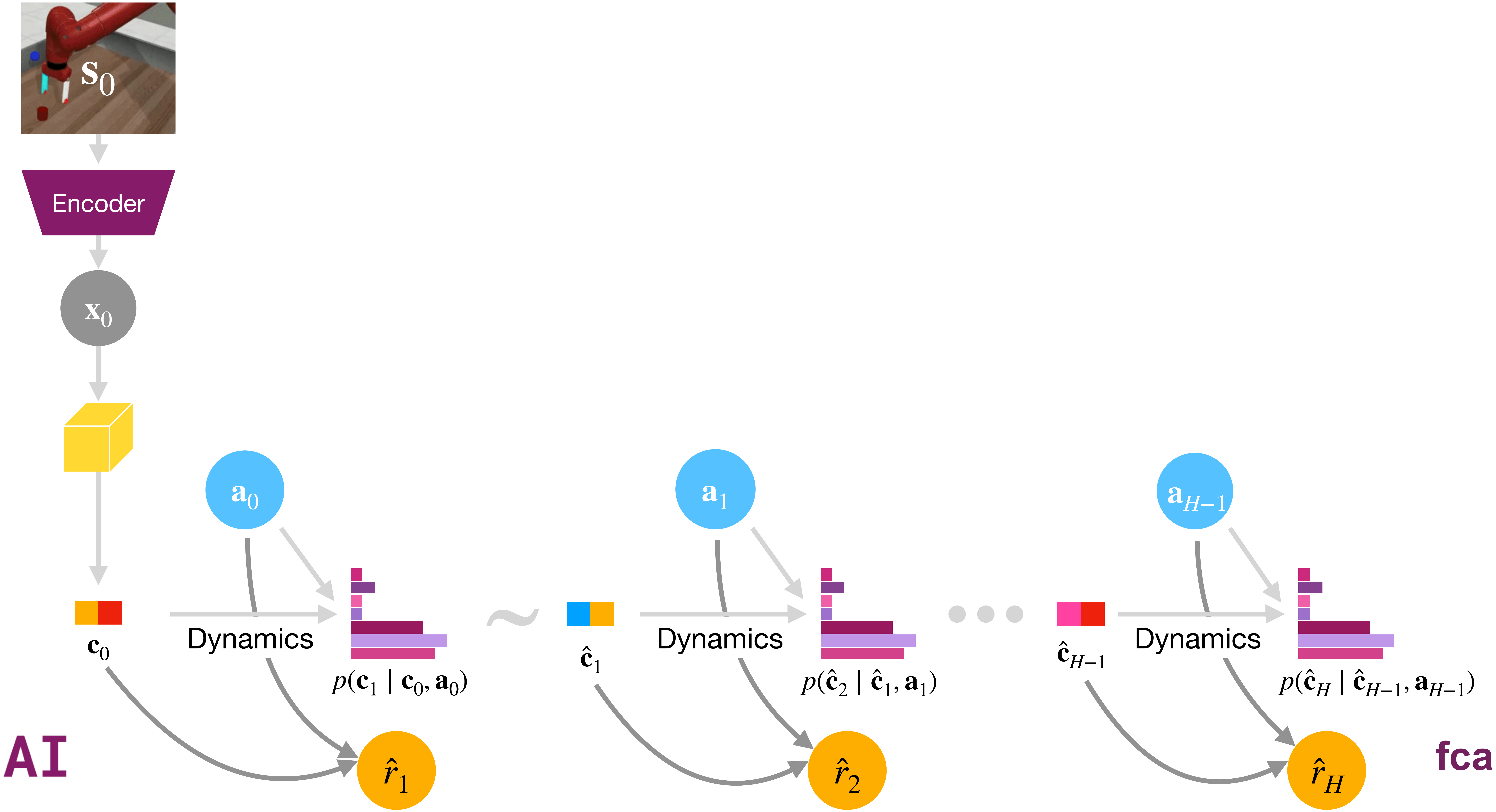
DCWM: World Model Training



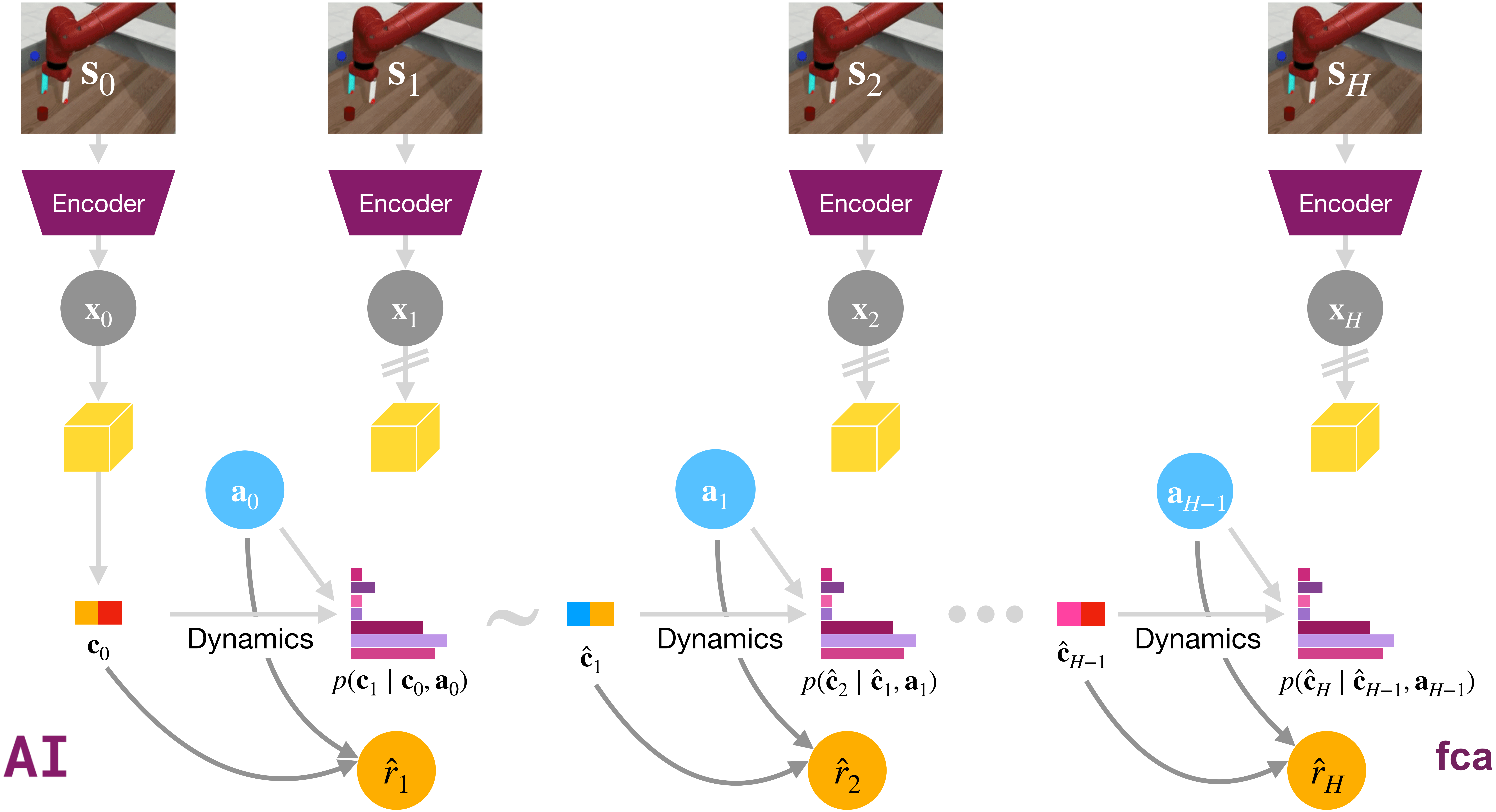
FCAI

fcai.fi

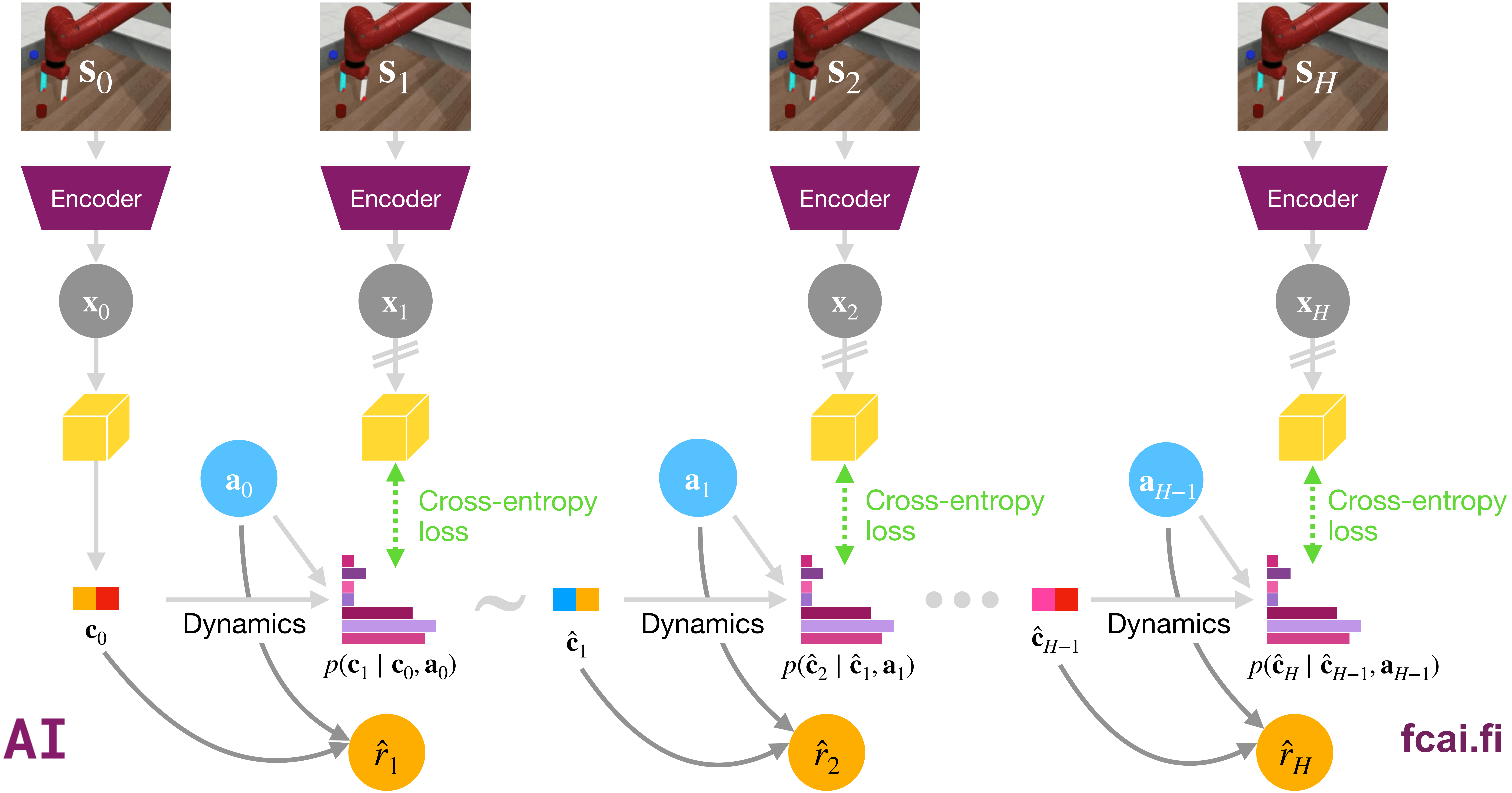
DCWM: World Model Training



DCWM: World Model Training

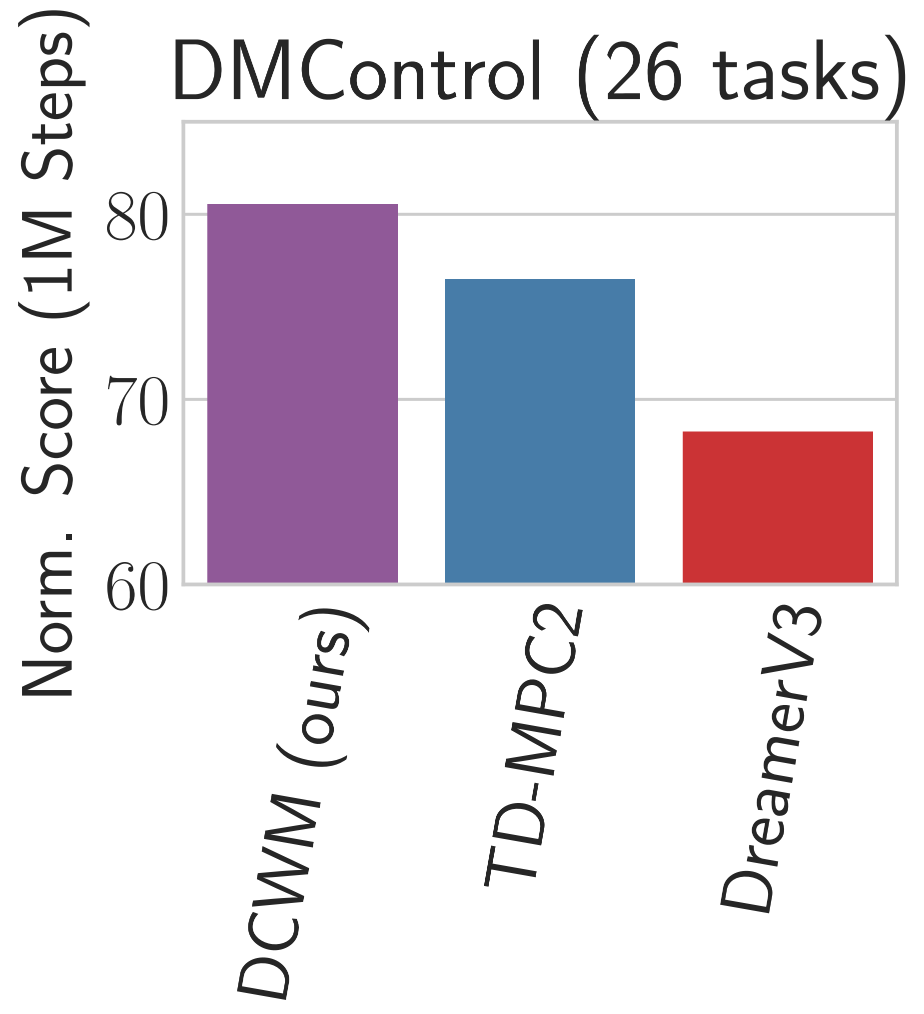


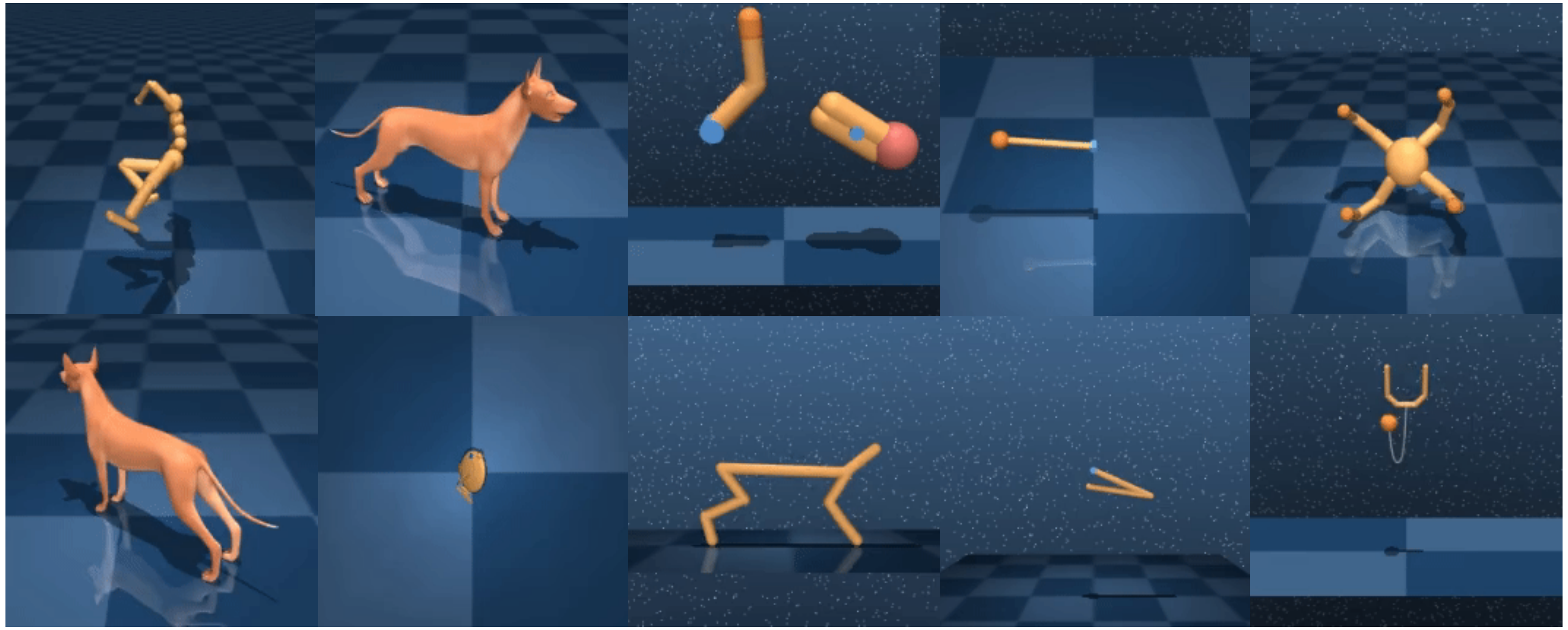
DCWM: World Model Training

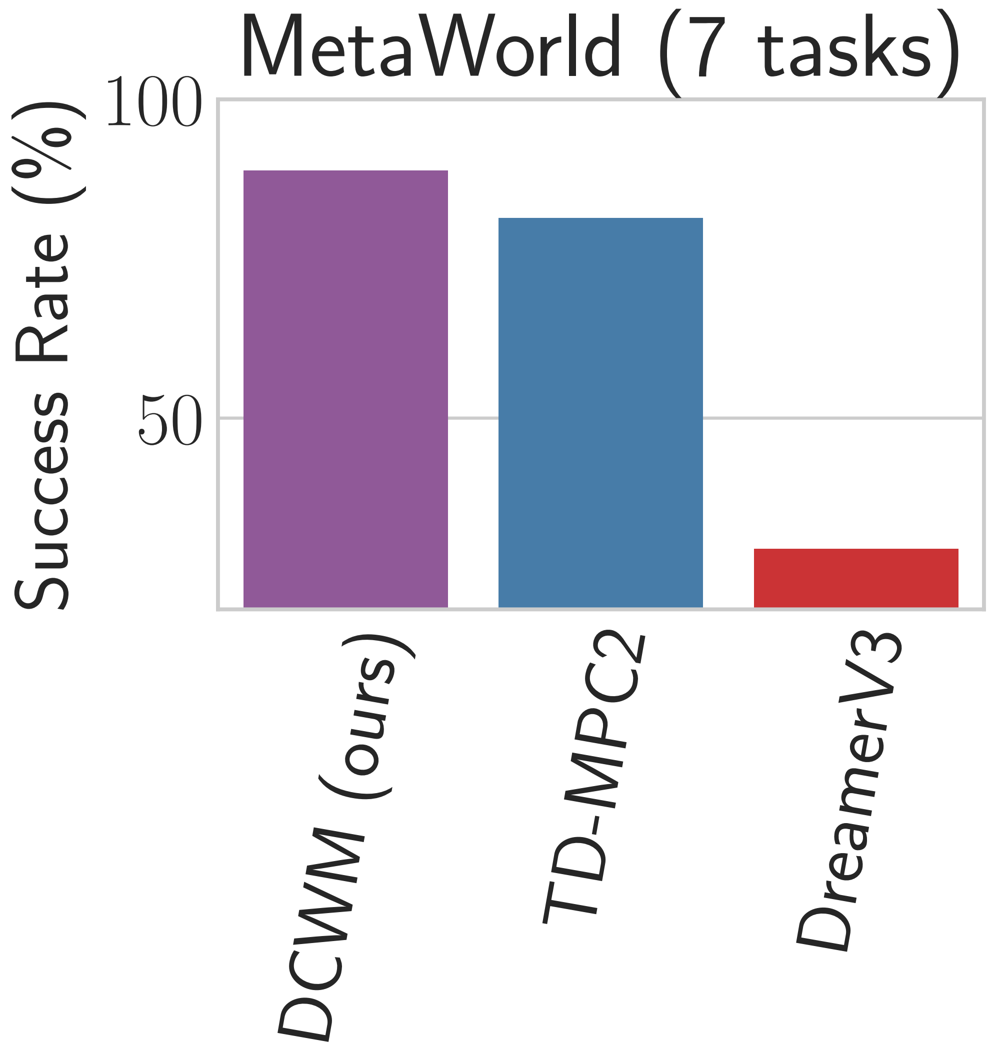


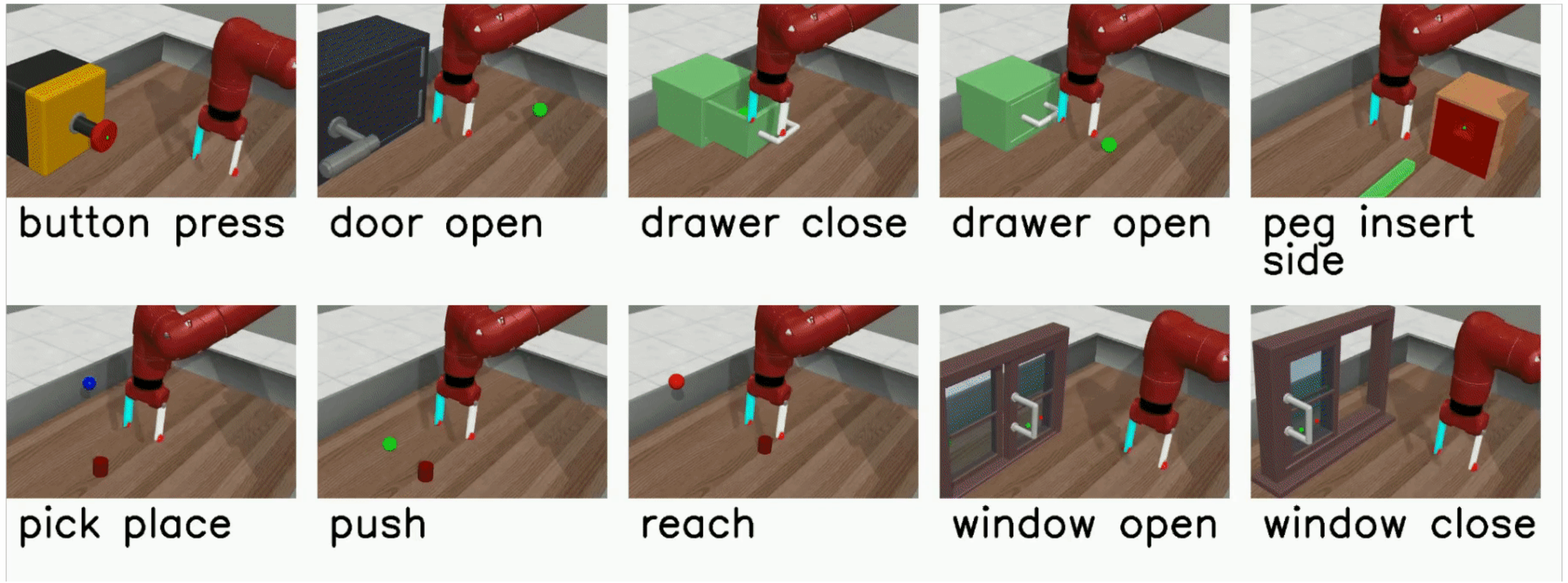
Results: Overview

Strong Performance in DMControl and MetaWorld Manipulation Tasks



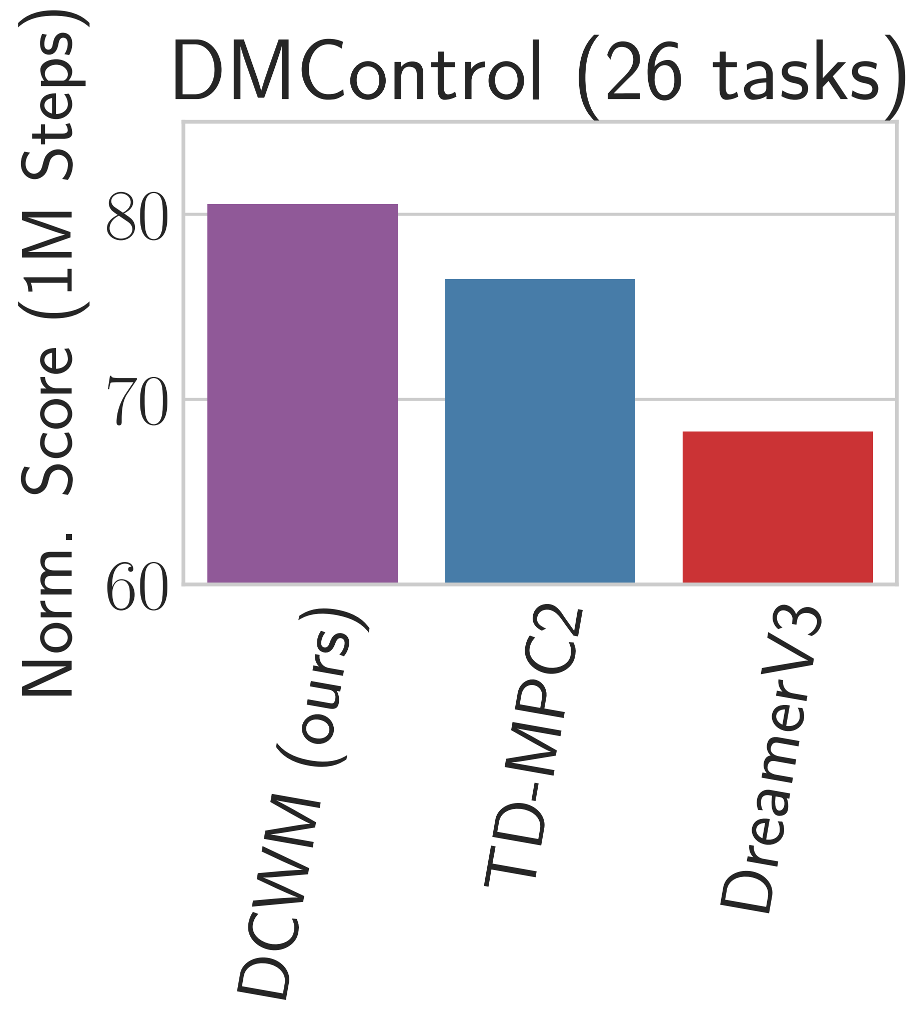


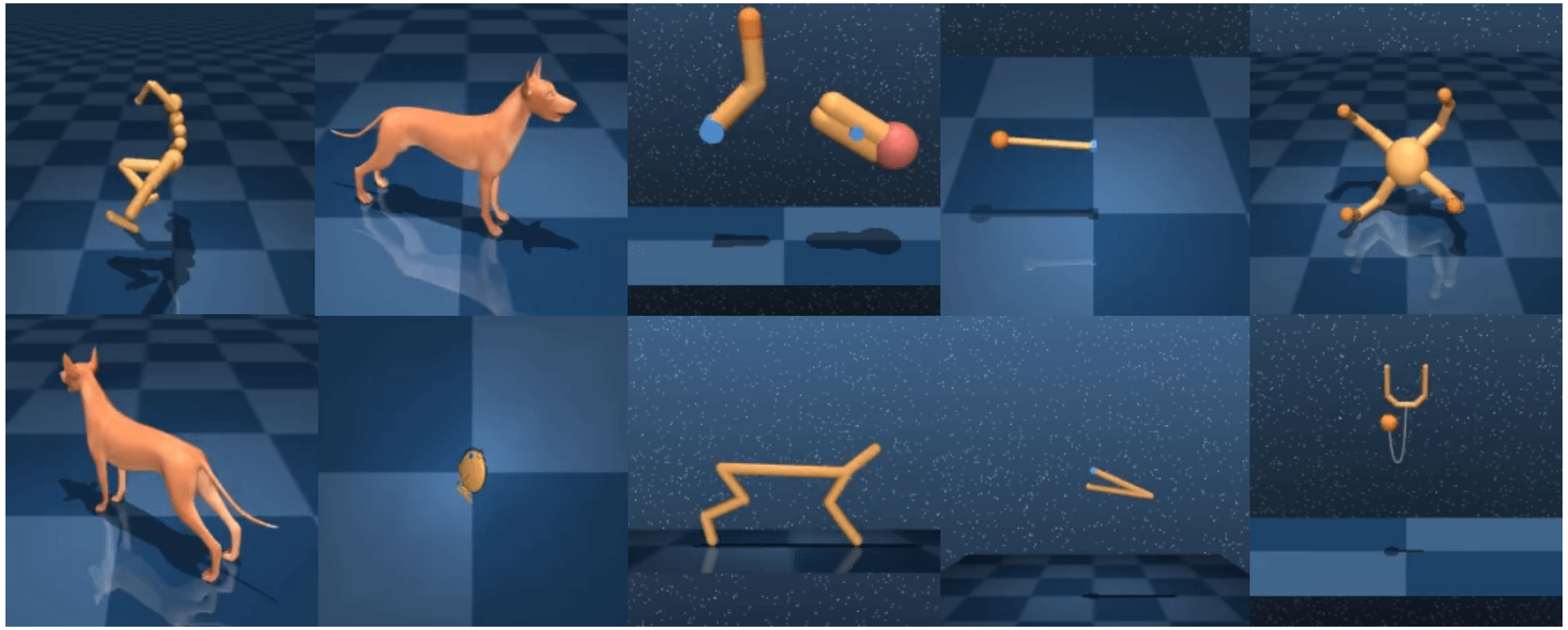


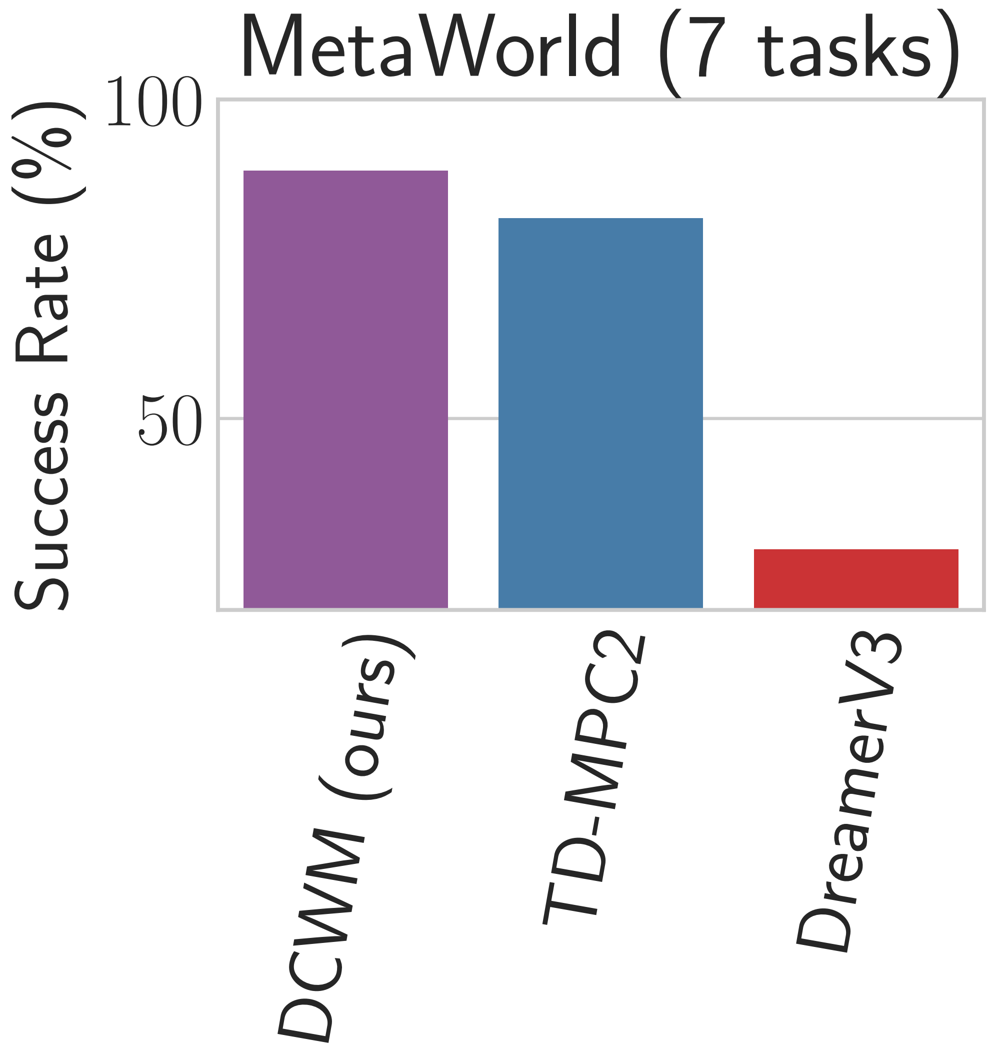


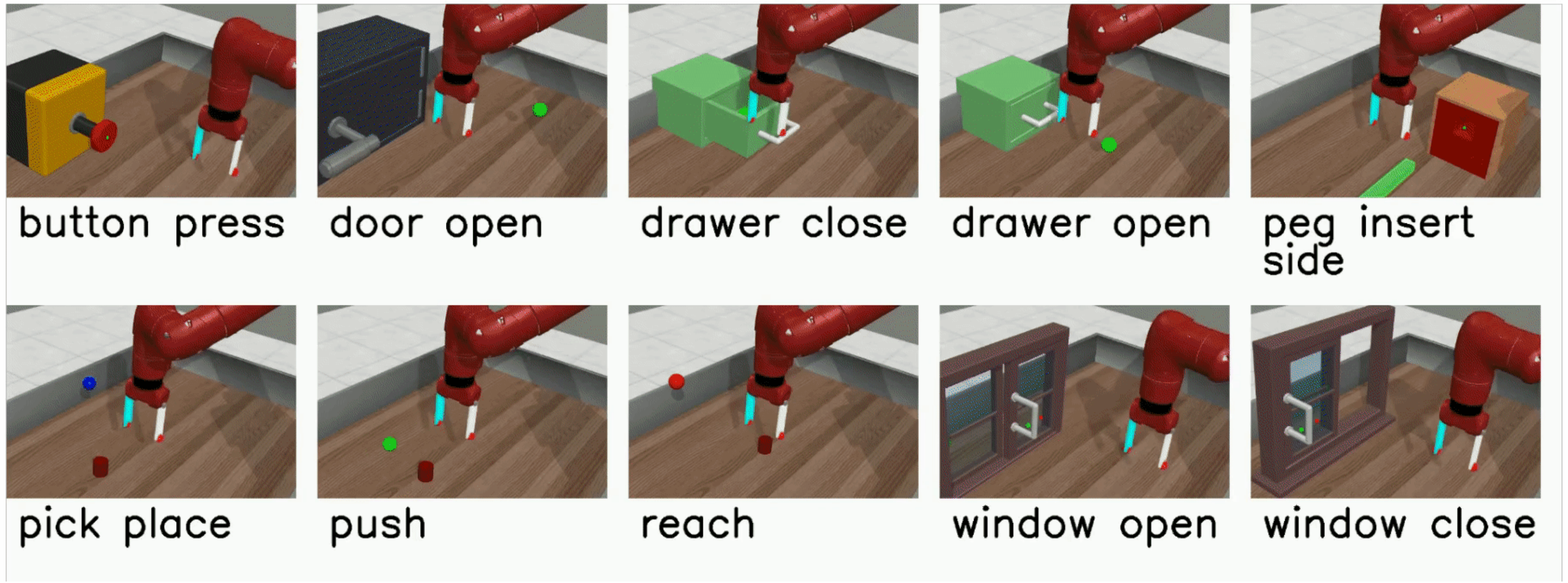
Results: Overview

Strong Performance in DMControl and MetaWorld Manipulation Tasks



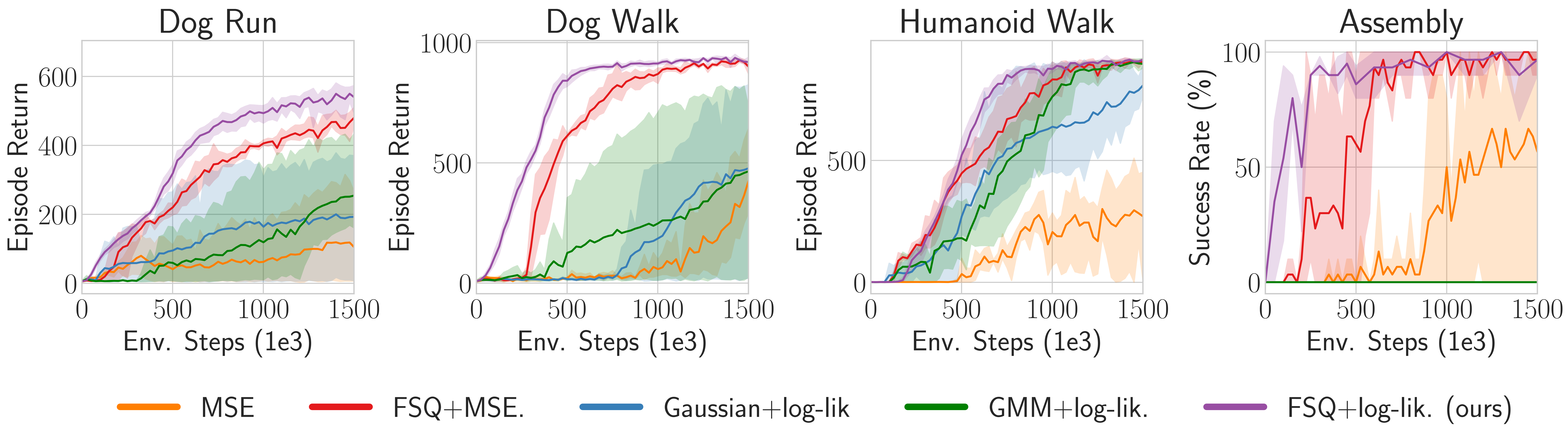






Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss

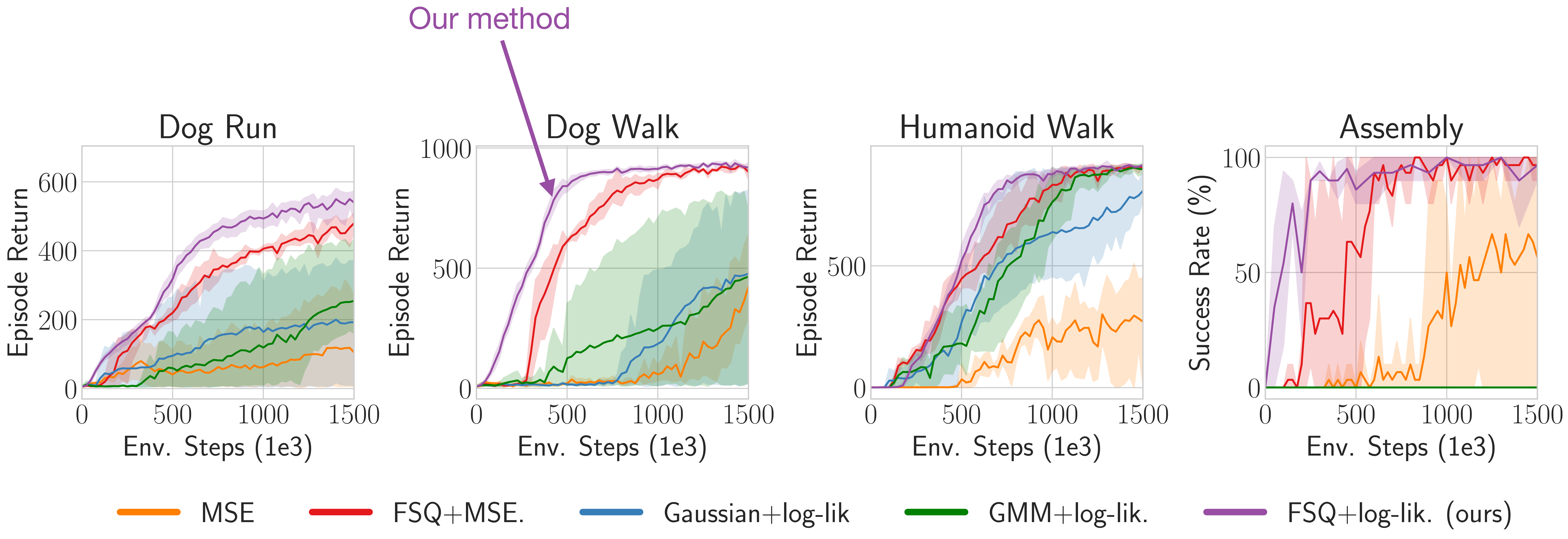


FCAI

fcai.fi

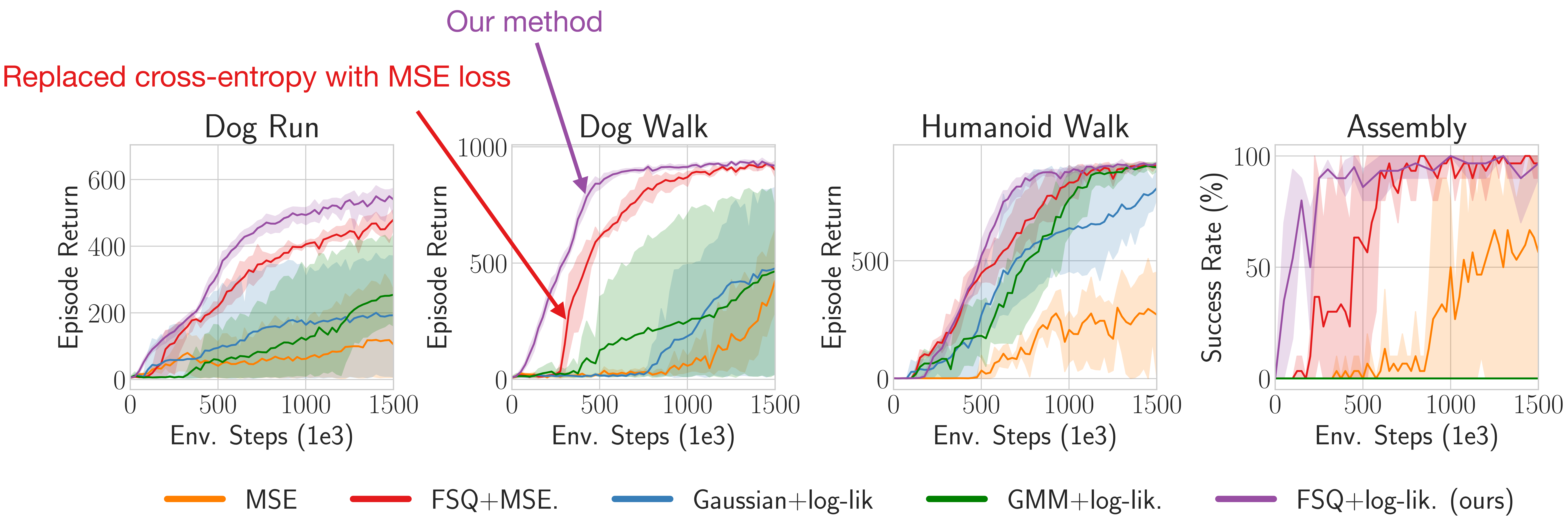
Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss



Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss

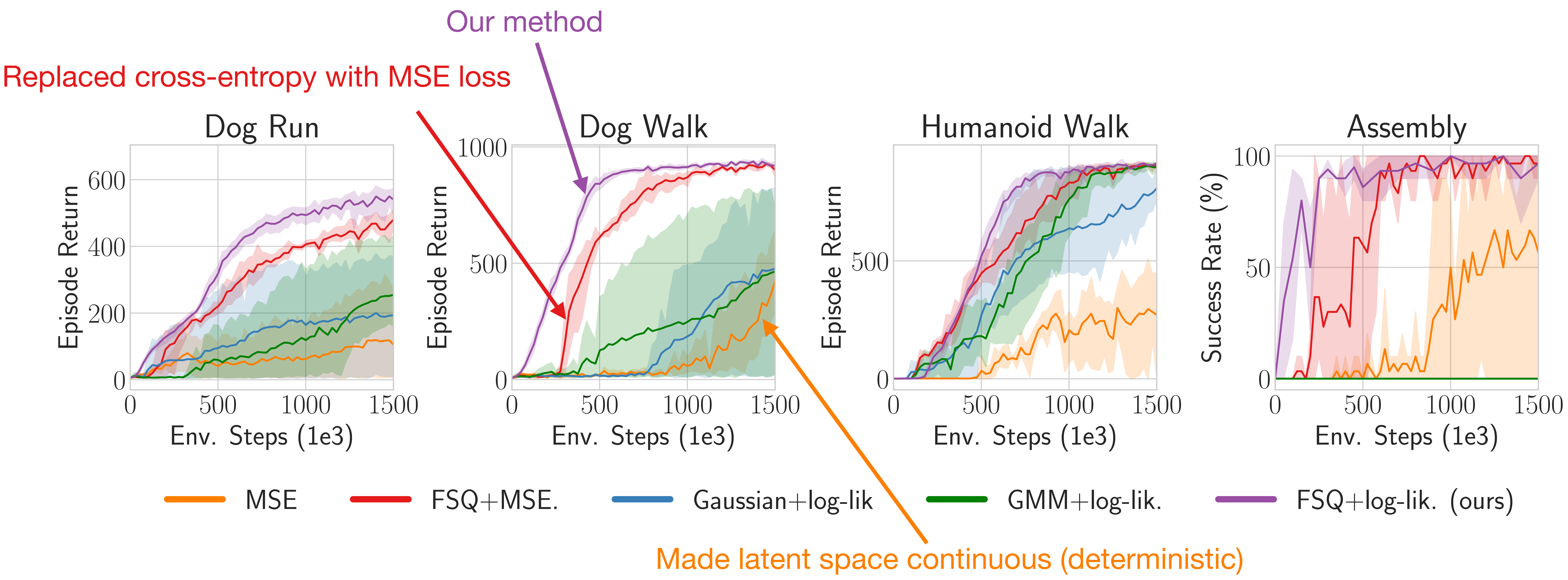


FCAI

fcai.fi

Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss

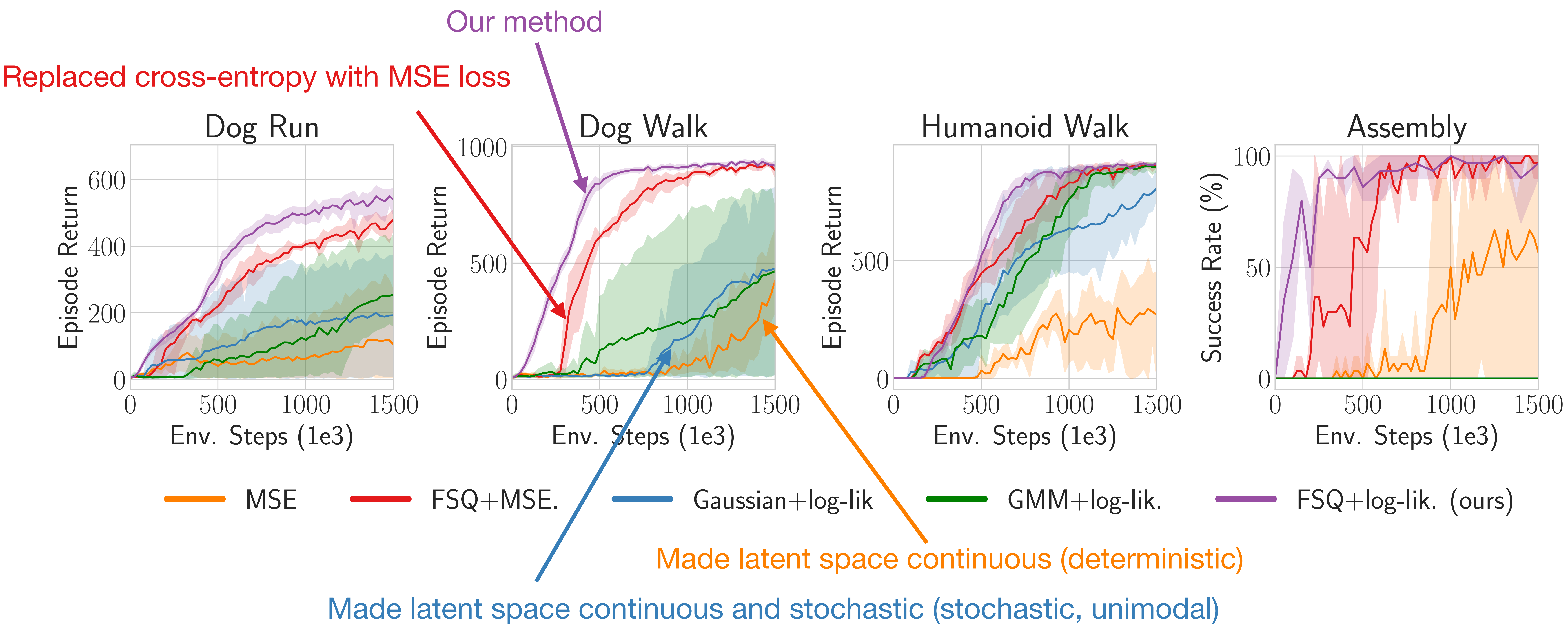


FCAI

fcai.fi

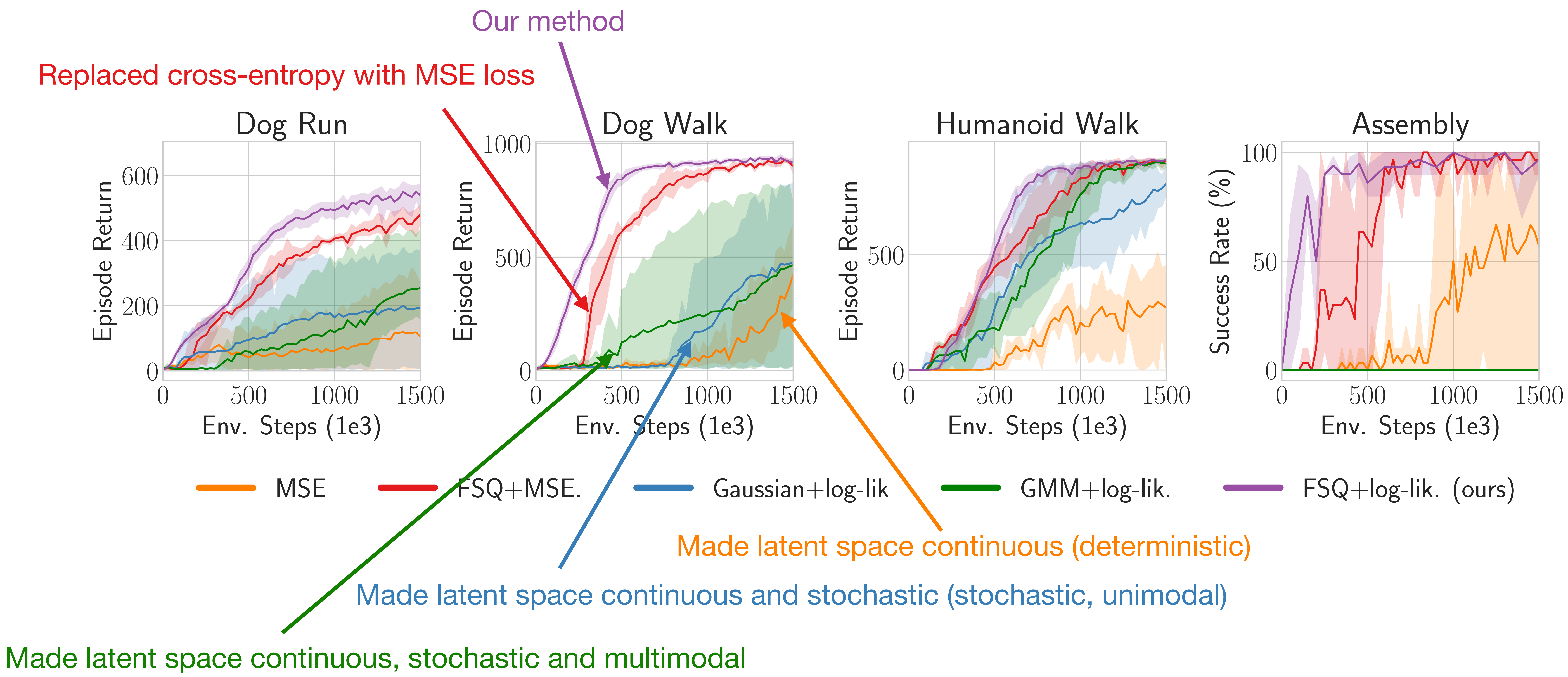
Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss



Why Does DCWM Work So Well?

Combination of Discrete Representation and Cross Entropy Loss



FCAI

fcai.fi

Main Takeaway:

Learning discrete codebook encodings with a self-supervised cross-entropy loss improves sample efficiency in continuous control tasks

Email: aidan.scannell@aalto.fi

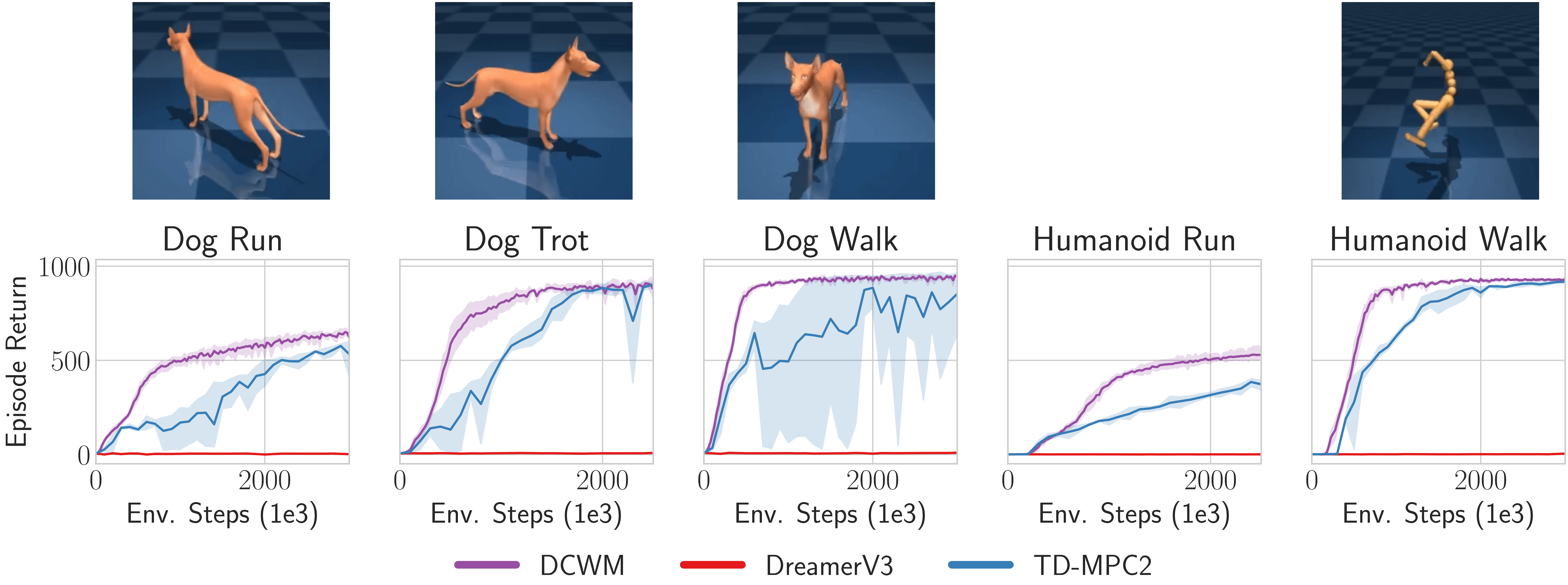
Website: www.aidanscannell.com

Main Takeaway:

Learning discrete codebook encodings with a self-supervised cross-entropy loss improves sample efficiency in continuous control tasks

Results: DeepMind Control Suite

Strong Performance in Hard DMControl Tasks

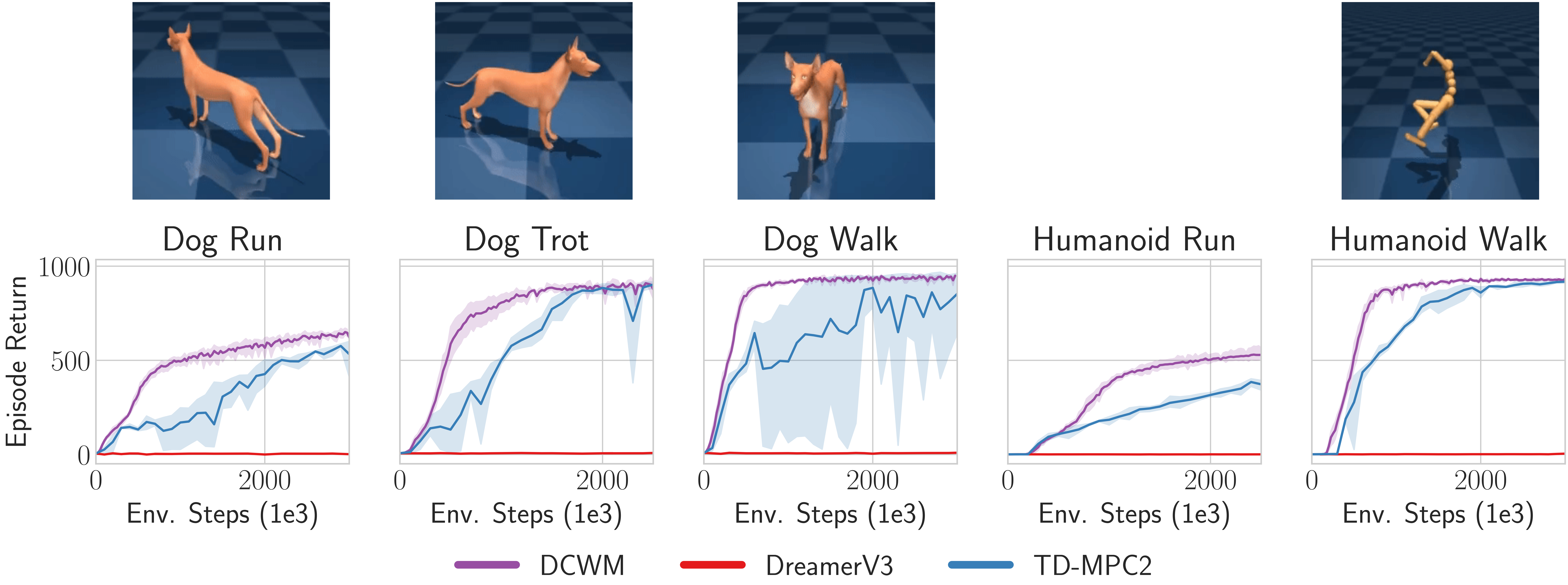


FCAI

fcai.fi

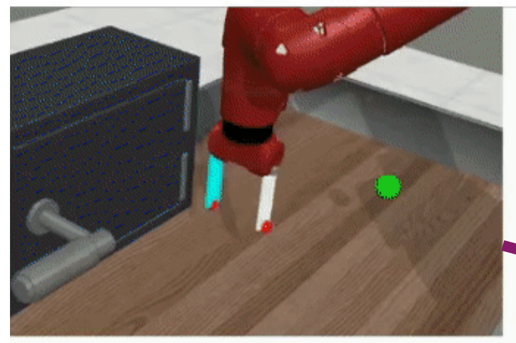
Results: DeepMind Control Suite

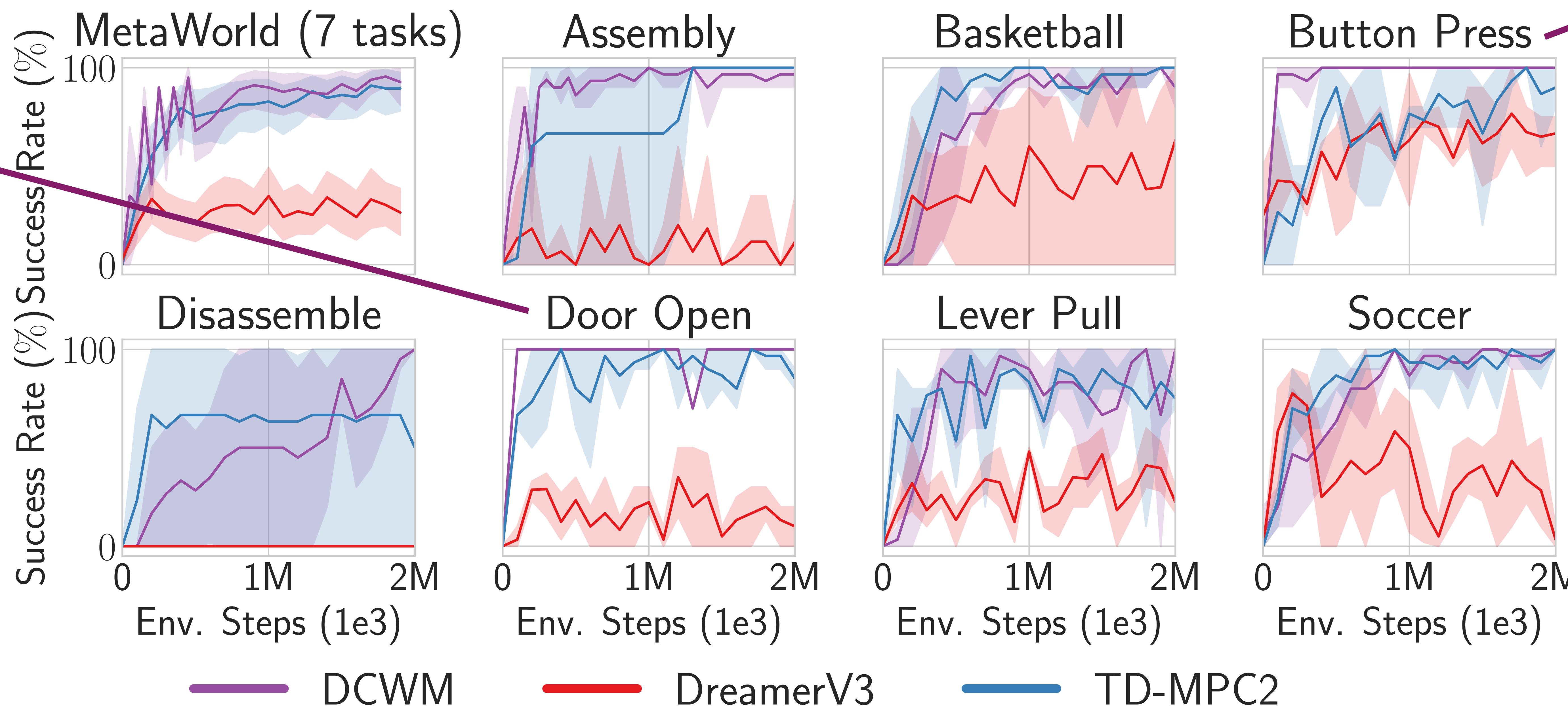
Strong Performance in Hard DMControl Tasks

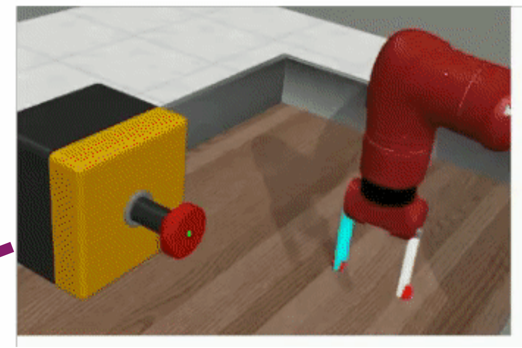


Results: MetaWorld

Competitive Performance in Robotic Manipulation

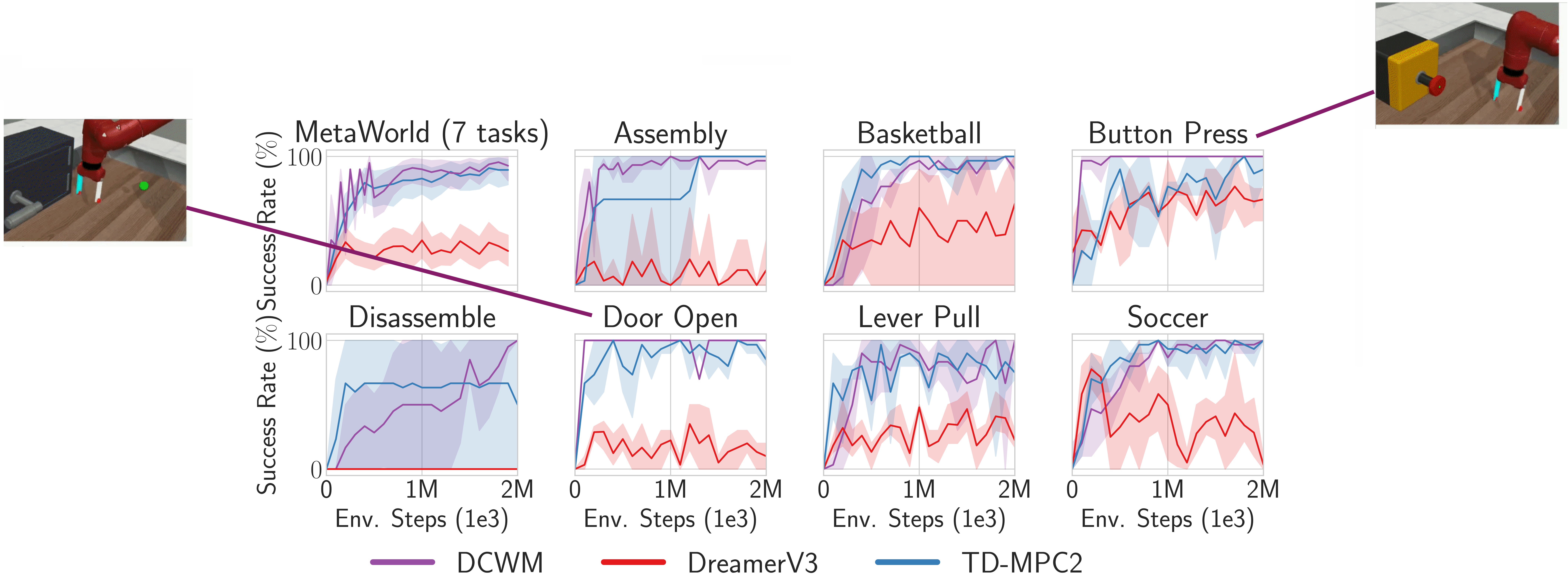






Results: MetaWorld

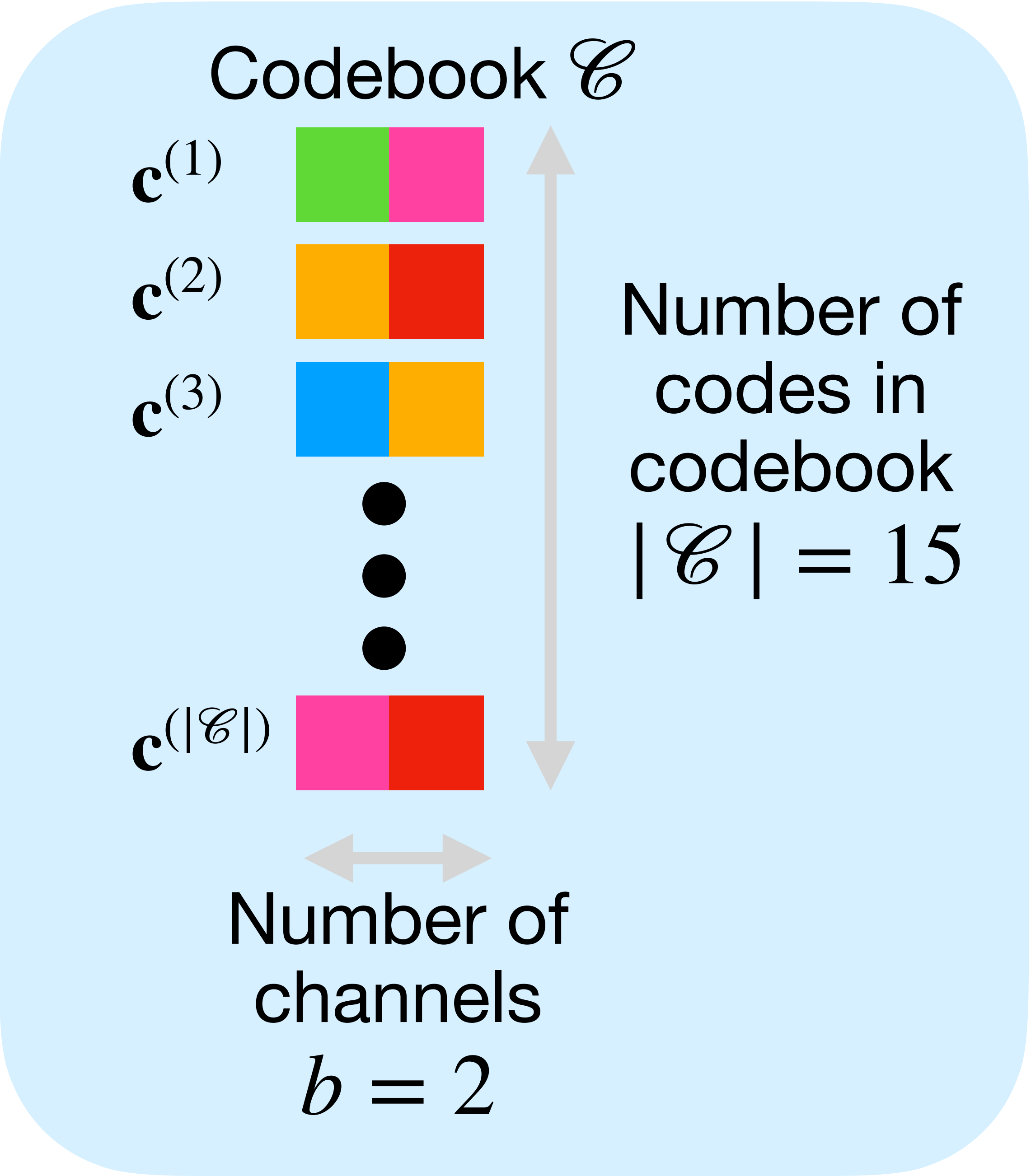
Competitive Performance in Robotic Manipulation



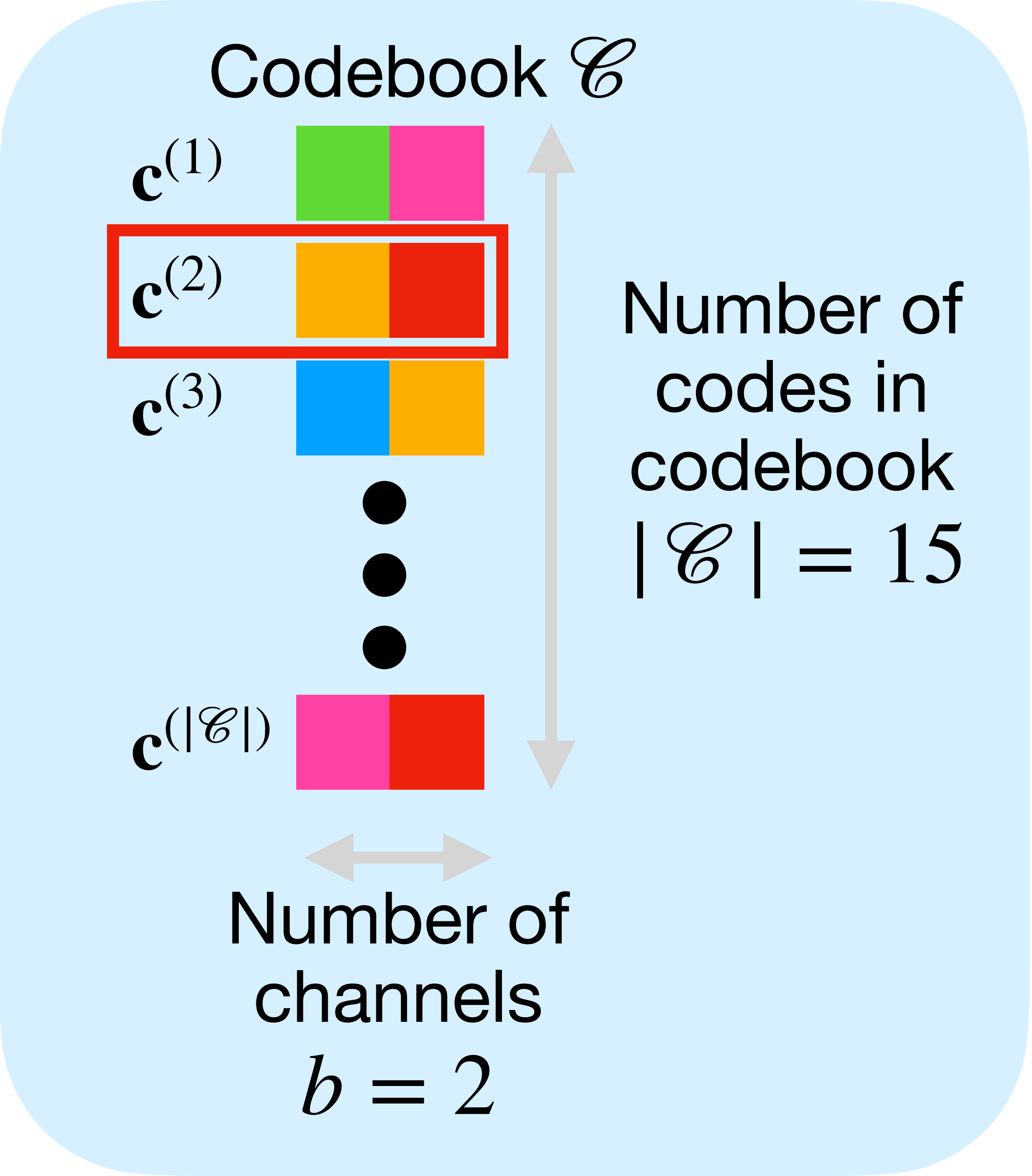
FCAI

fcai.fi

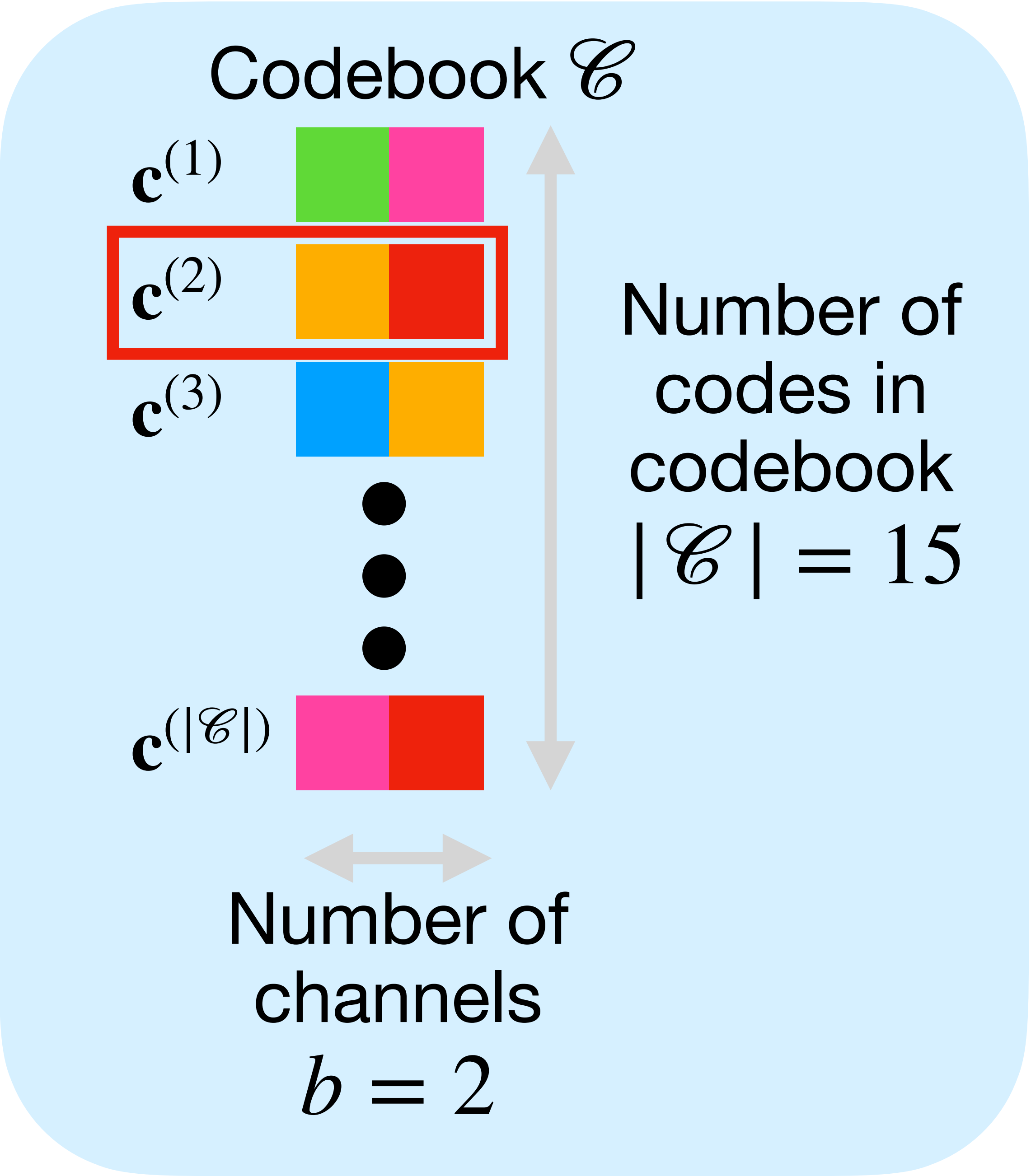
Comparison of Different Discrete Encodings



Comparison of Different Discrete Encodings

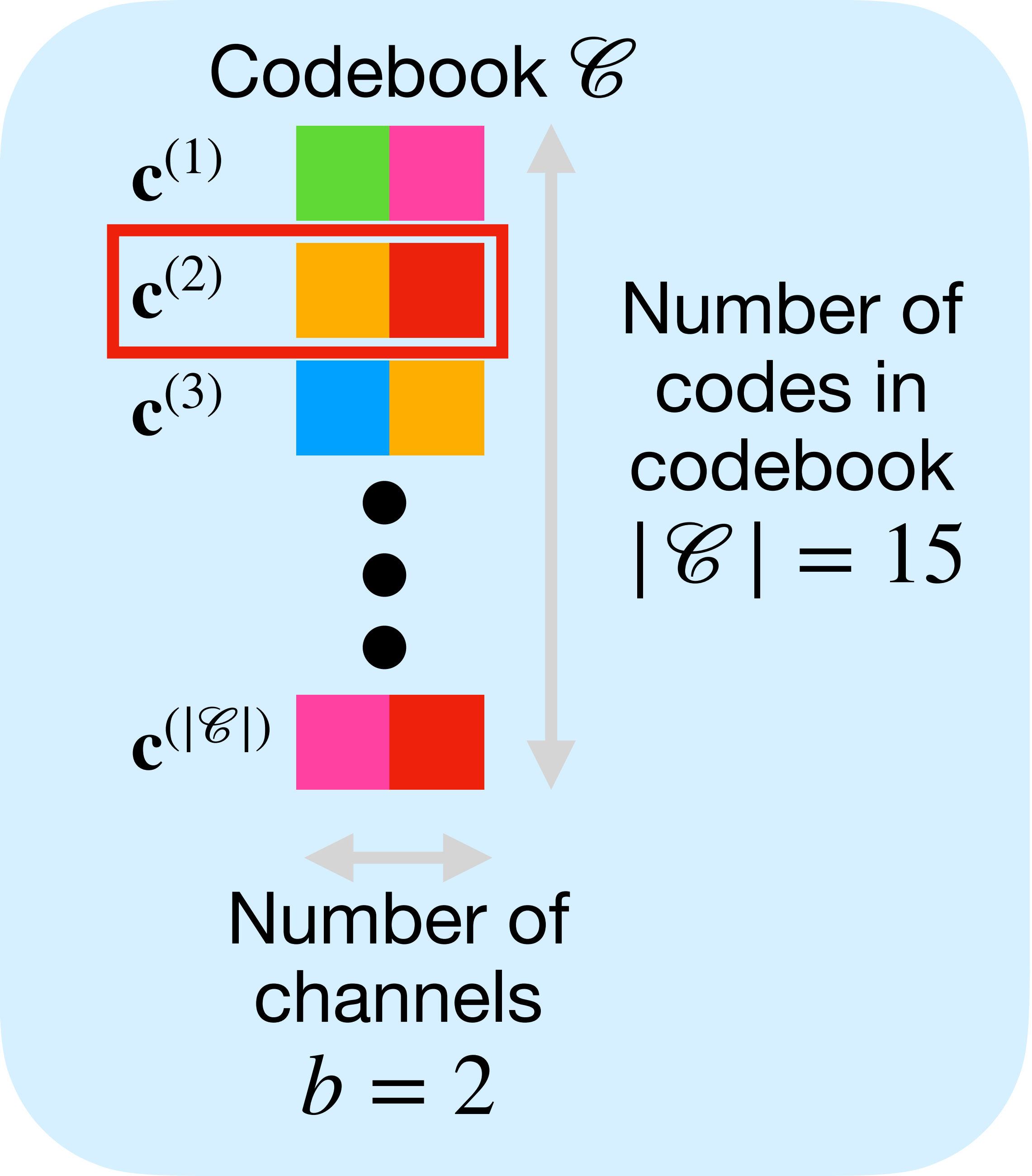


Comparison of Different Discrete Encodings



$$\mathbf{e}_{\text{code}} = \mathbf{c}^{(2)} = \{-0.5, 1\}$$

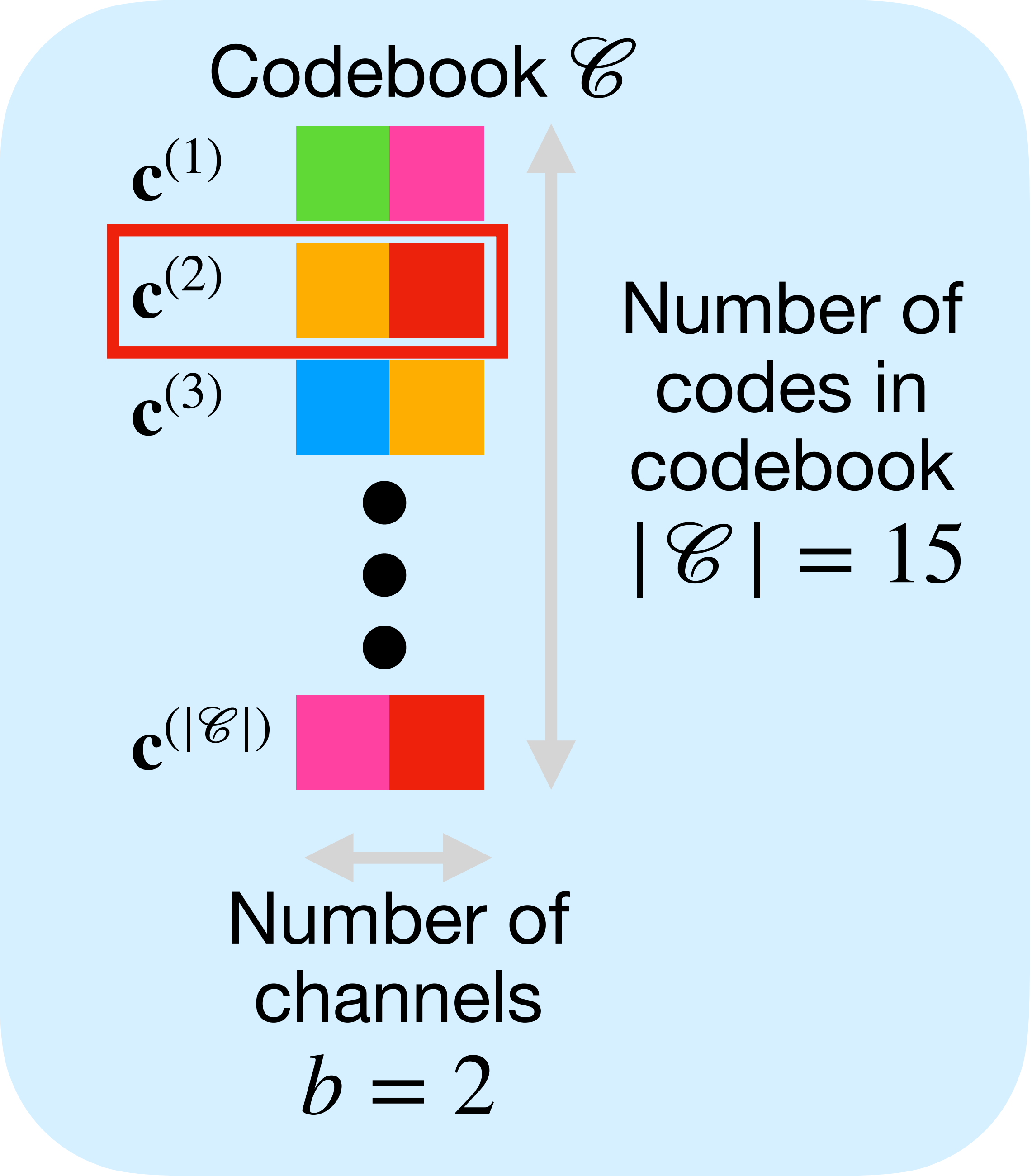
Comparison of Different Discrete Encodings



$\mathbf{e}_{\text{code}} = \mathbf{c}^{(2)} = \{-0.5, 1\}$

$\mathbf{e}_{\text{label}} = 2$

Comparison of Different Discrete Encodings

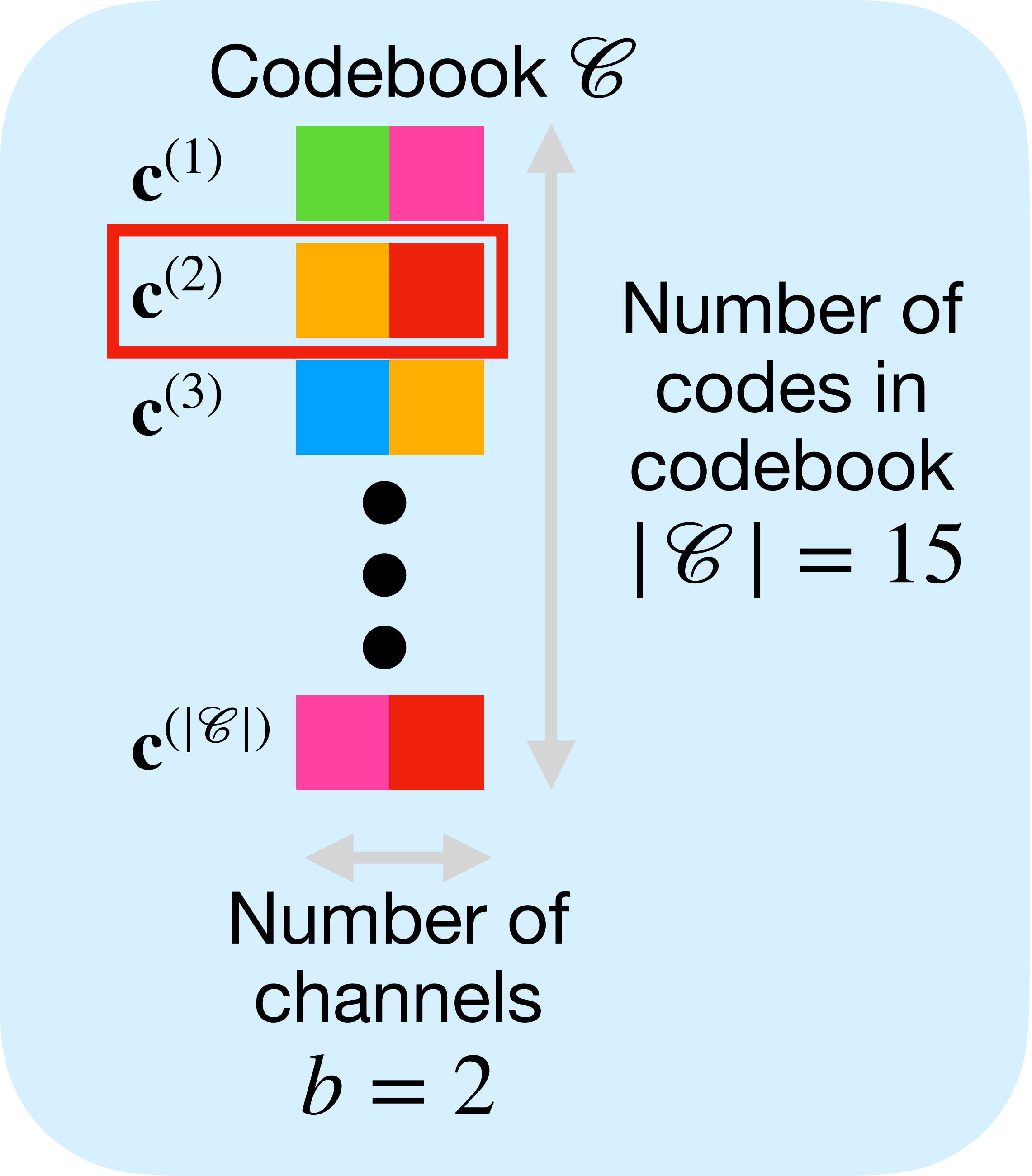


$$\mathbf{e}_{\text{code}} = \mathbf{c}^{(2)} = \{-0.5, 1\}$$

$$\mathbf{e}_{\text{label}} = 2$$

$$\mathbf{e}_{\text{one-hot}} = \{0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\}$$

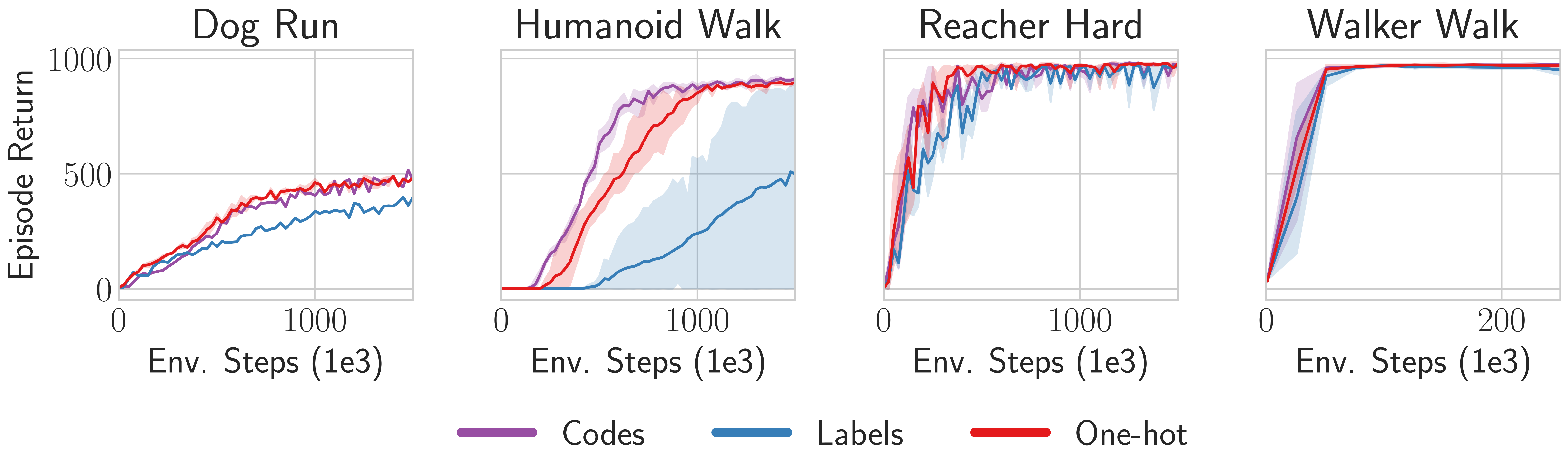
Comparison of Different Discrete Encodings



$$\mathbf{e}_{\text{code}} = \mathbf{c}^{(2)} = \{-0.5, 1\}$$

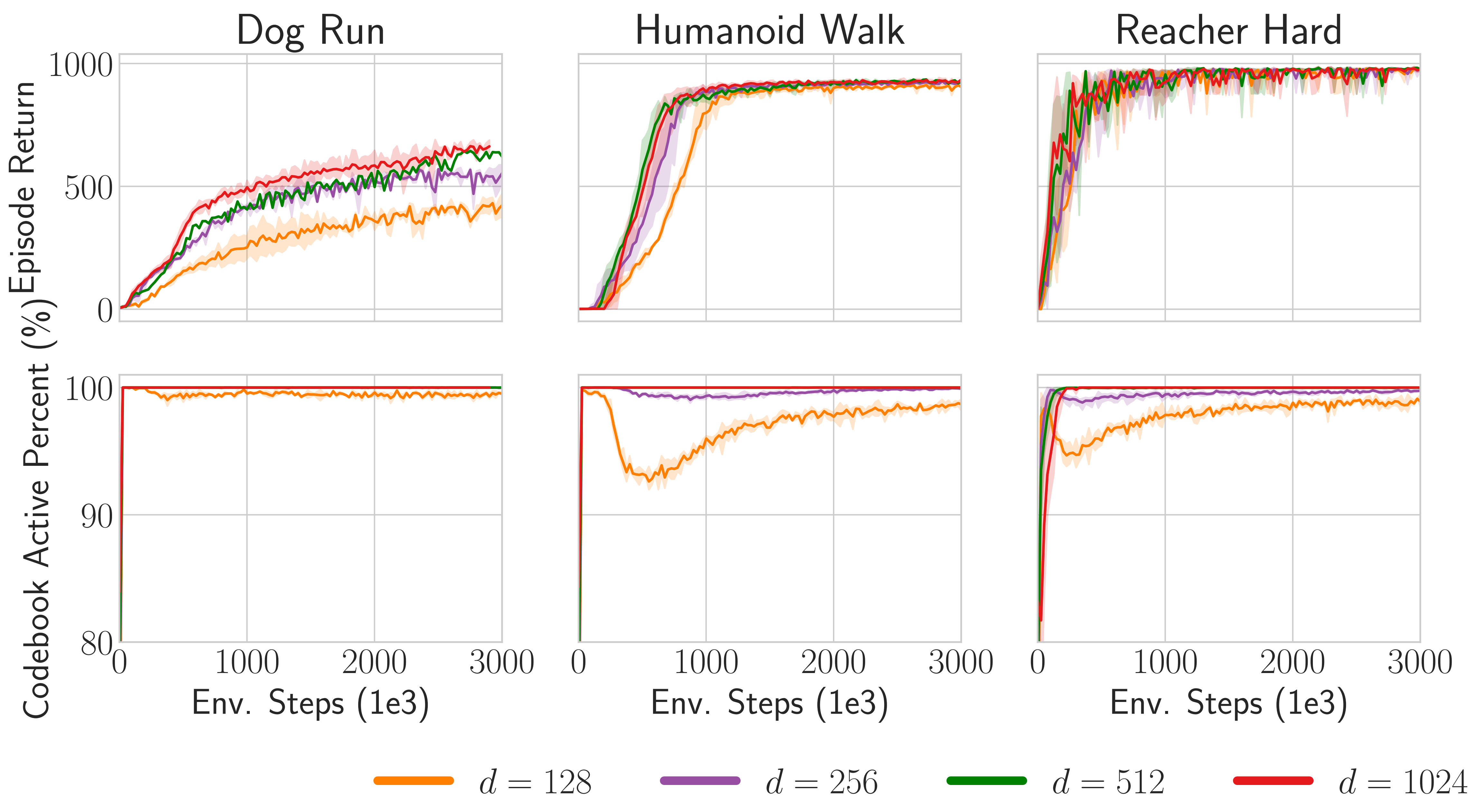
$$\mathbf{e}_{\text{label}} = 2$$

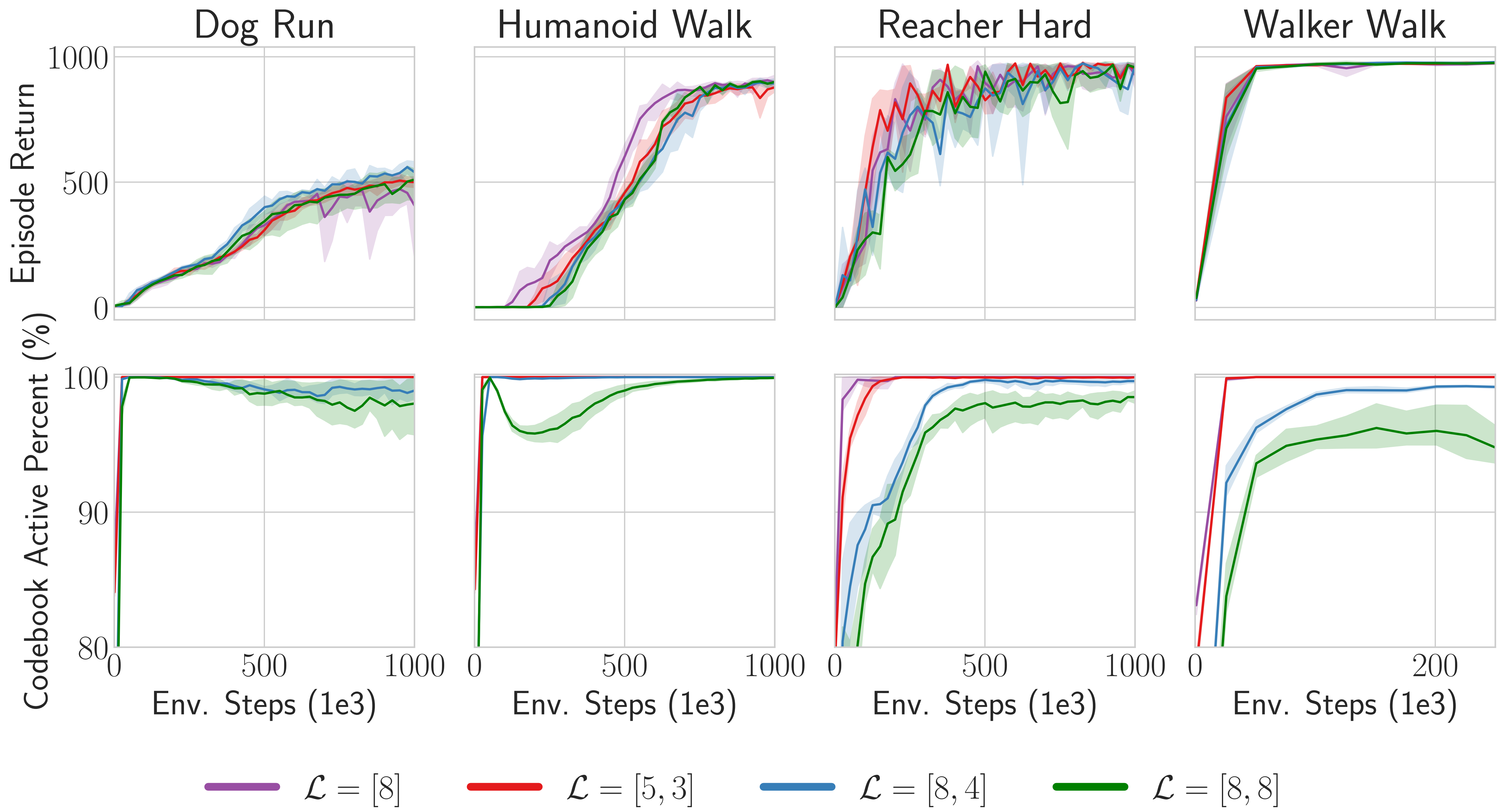
$$\mathbf{e}_{\text{one-hot}} = \{0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\}$$



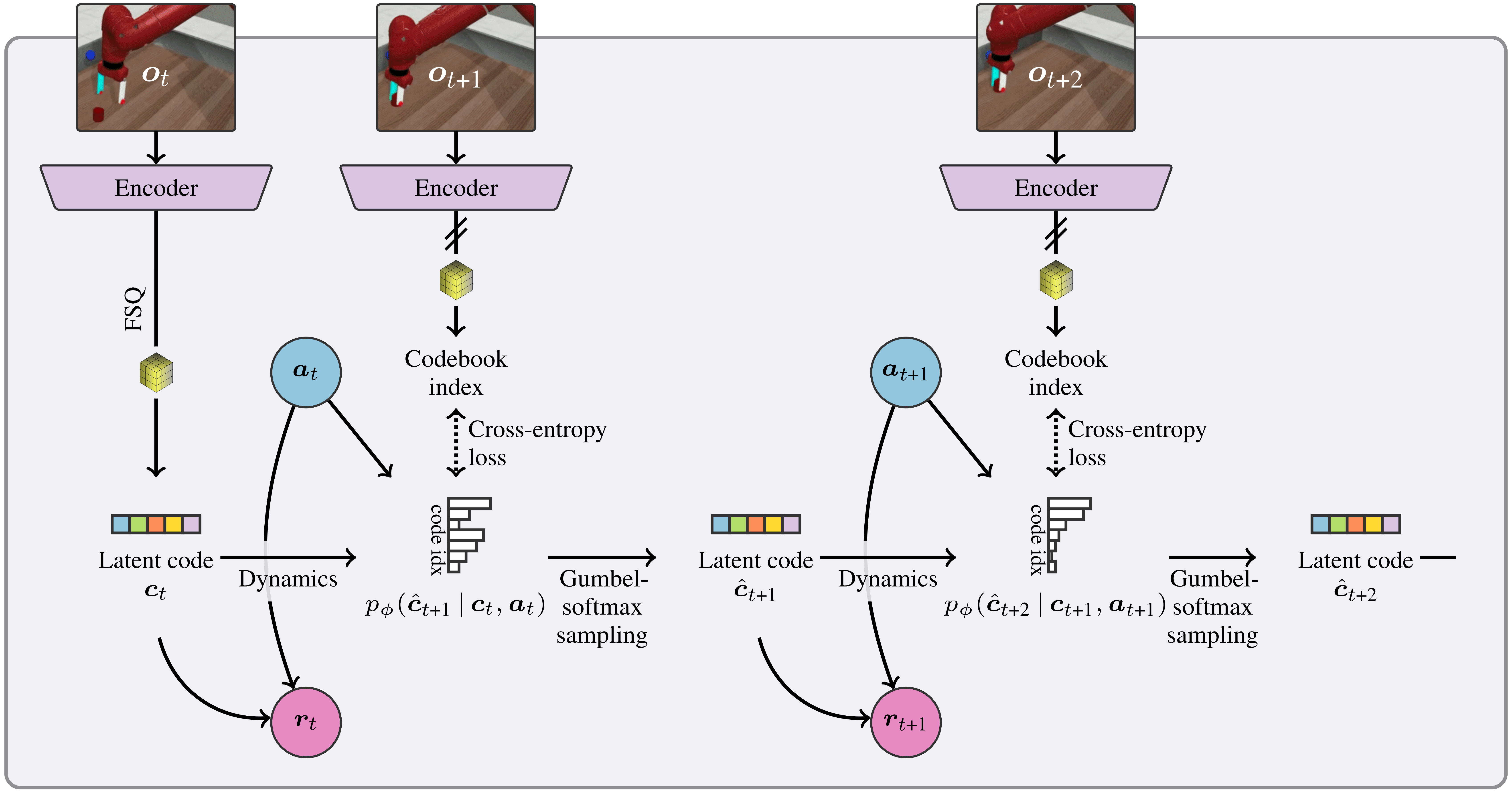
Results: Latent Space Size

DCWM is Fairly Robust to its Latent Space Size





DCWM: Discrete Codebook World Model



DCWM: Components

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

DCWM: Components

Encoder	$\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$
Latent quantization	$\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$
Dynamics	$\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{\lvert\mathscr{C}\rvert})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$
Reward	$\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

Reward $\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

World model loss

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t) \in \mathbb{R}^{d \times b}$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

Reward $\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

World model loss

$$\mathscr{L}(\theta, \phi, \xi; \mathscr{D}) = \mathbb{E}_{(\mathbf{o},\mathbf{a},\mathbf{o}',r)_{0:H} \sim \mathscr{D}} \left[\sum_{h=0}^{H-1} \gamma^h \left(\mathrm{CE}(p_\phi(\hat{\mathbf{c}}_{h+1} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h), \mathbf{c}_{h+1}) + \| R_\xi(\mathbf{c}_h, \mathbf{a}_h) - r_h \|_2^2 \right) \right]$$

FCAI fcai.fi

DCWM: Components

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \text{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

FCAI

fcai.fi

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

Reward $\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \ldots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

Reward $\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

Critic $q_t = Q_\psi(\mathbf{c}_t, \mathbf{a}_t)$

DCWM: Components

Encoder $\mathbf{x}_t = e_\theta(\mathbf{s}_t)$

Latent quantization $\mathbf{c}_t = f(\mathbf{x}_t) \in \mathscr{C}$

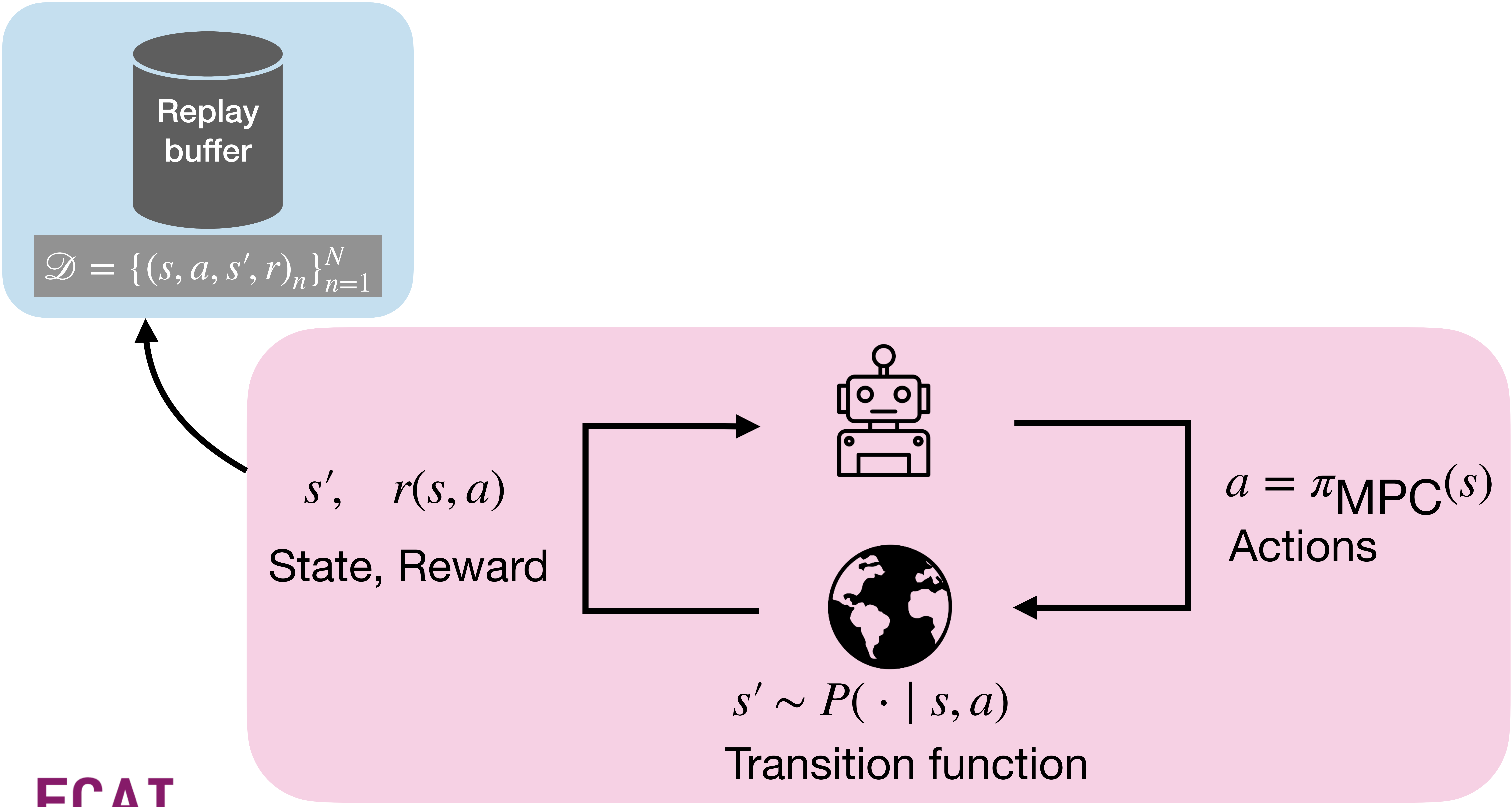
Dynamics $\hat{\mathbf{c}}_{t+1} \sim \mathrm{Categorical}(p_1, \dots, p_{|\mathscr{C}|})$ with $p_i = P_\phi(\mathbf{c}_{t+1} = \mathbf{c}^{(i)} \mid \mathbf{c}_t, \mathbf{a}_t)$

Reward $\hat{r}_{t+1} = R_\xi(\mathbf{c}_t, \mathbf{a}_t)$

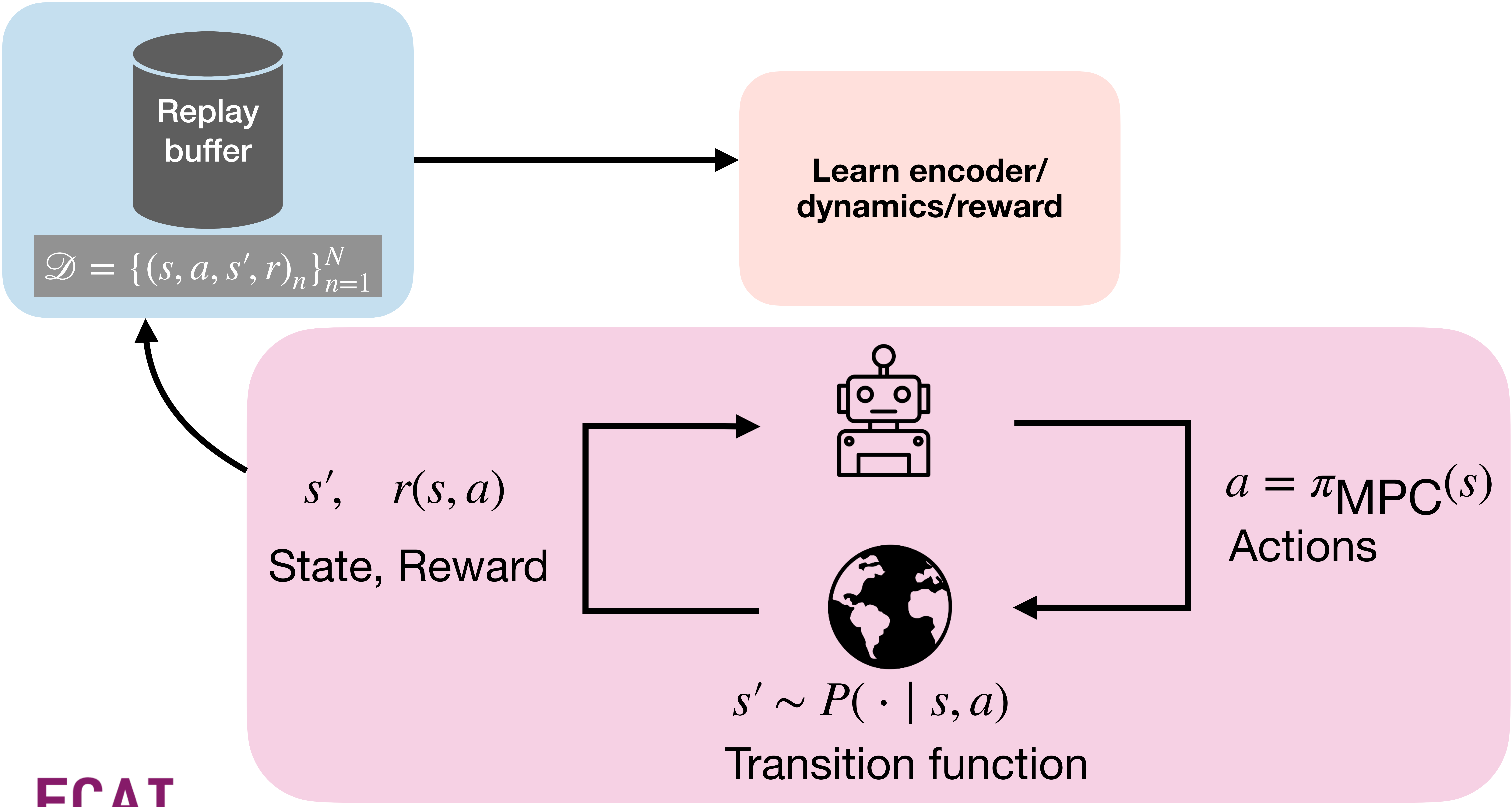
Critic $q_t = Q_\psi(\mathbf{c}_t, \mathbf{a}_t)$

Prior Policy $\mathbf{a}_t \sim \pi_\eta(\mathbf{a}_t \mid \mathbf{c}_t)$

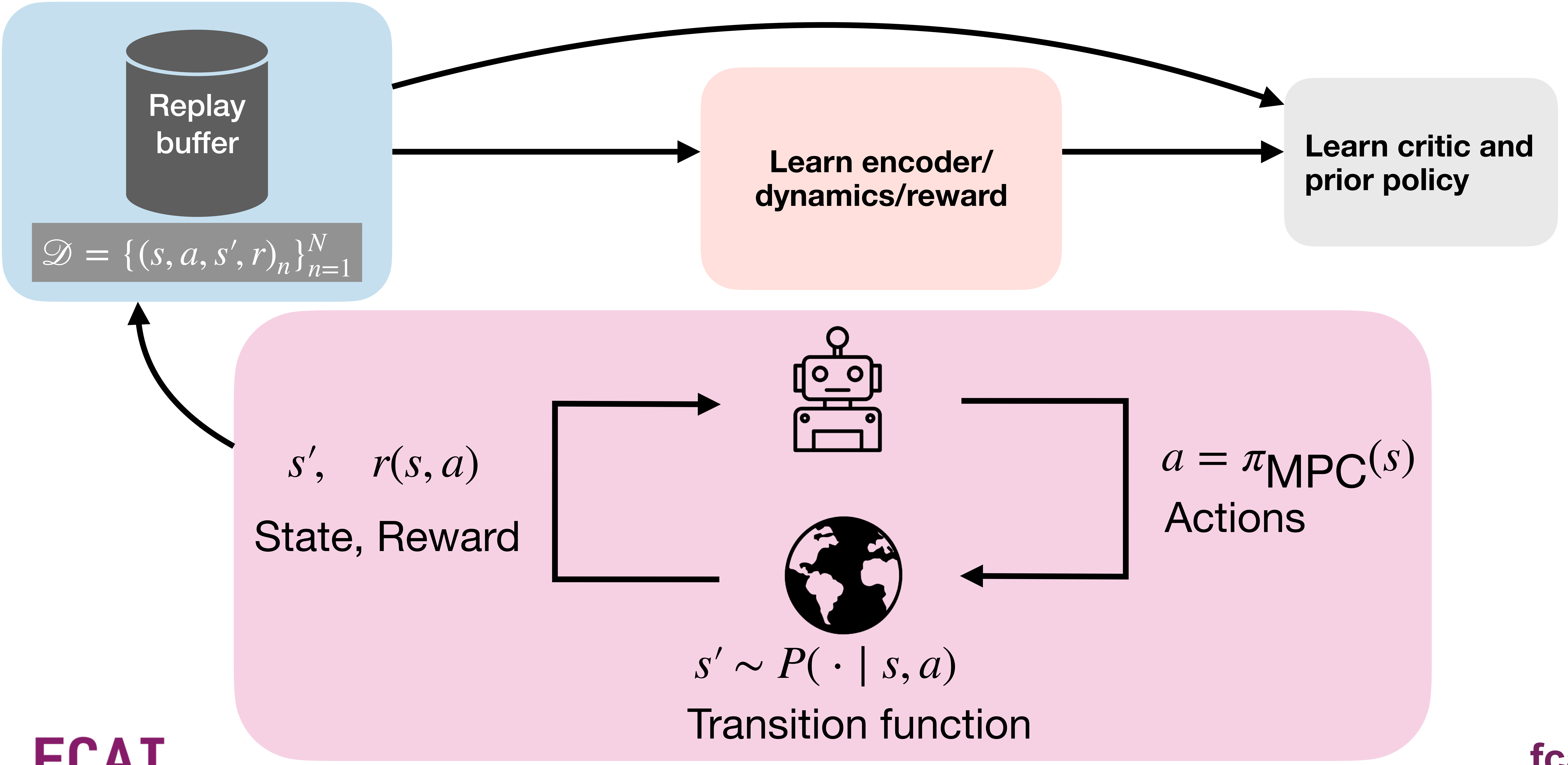
Model-based Reinforcement Learning



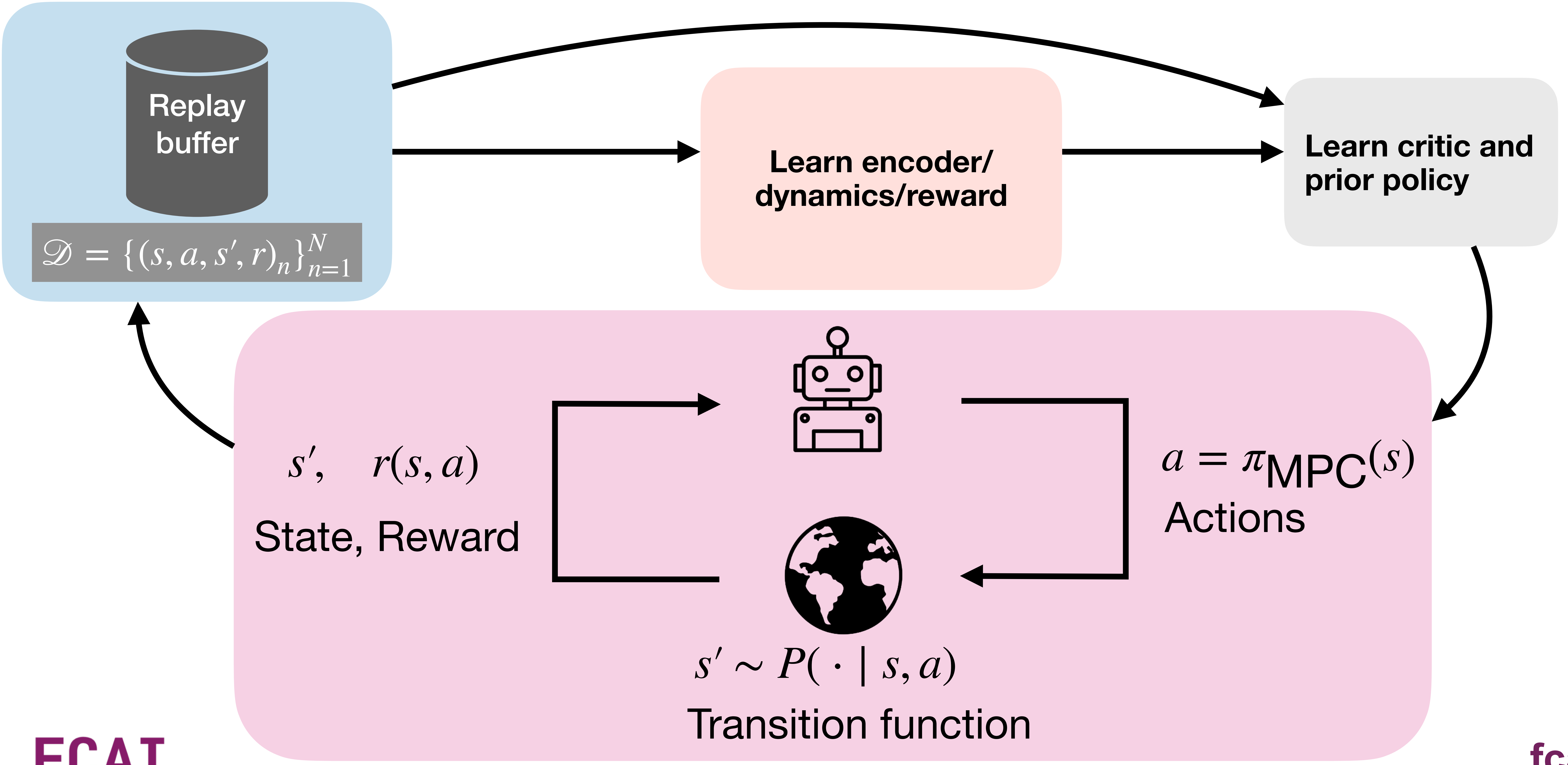
Model-based Reinforcement Learning



Model-based Reinforcement Learning



Model-based Reinforcement Learning



DCWM

Algorithm

DCWM

Algorithm

i. For i in number of episodes

DCWM

Algorithm

i. For i in number of episodes

    i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$

DCWM

Algorithm

i. For i in number of episodes

   i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$

   ii. Add trajectory to replay buffer $\mathscr{D} \leftarrow \mathscr{D} \cup \tau_i$

DCWM

Algorithm

i. For i in number of episodes

   i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$

   ii. Add trajectory to replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \tau_i$

   iii. Perform T updates to world model

DCWM

Algorithm

i. For i in number of episodes

   i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$

   ii. Add trajectory to replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \tau_i$

   iii. Perform T updates to world model

      i. Sample batch from replay buffer $\mathcal{D}$

DCWM

Algorithm

i. For i in number of episodes

   i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$

   ii. Add trajectory to replay buffer $\mathscr{D} \leftarrow \mathscr{D} \cup \tau_i$

   iii. Perform T updates to world model

      i. Sample batch from replay buffer $\mathscr{D}$

      ii. Update encoder, dynamics and reward

DCWM

Algorithm

i. For i in number of episodes
   i. Collect trajectory $\tau_i = \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}_{t=0}^{T}$
   ii. Add trajectory to replay buffer $\mathscr{D} \leftarrow \mathscr{D} \cup \tau_i$
   iii. Perform T updates to world model
      i. Sample batch from replay buffer $\mathscr{D}$
      ii. Update encoder, dynamics and reward
      iii. Update actor and critic

Decision-time Planning

Model Predictive Control (MPC)

Decision-time Planning

Model Predictive Control (MPC)

For each environment step

★

Decision-time Planning

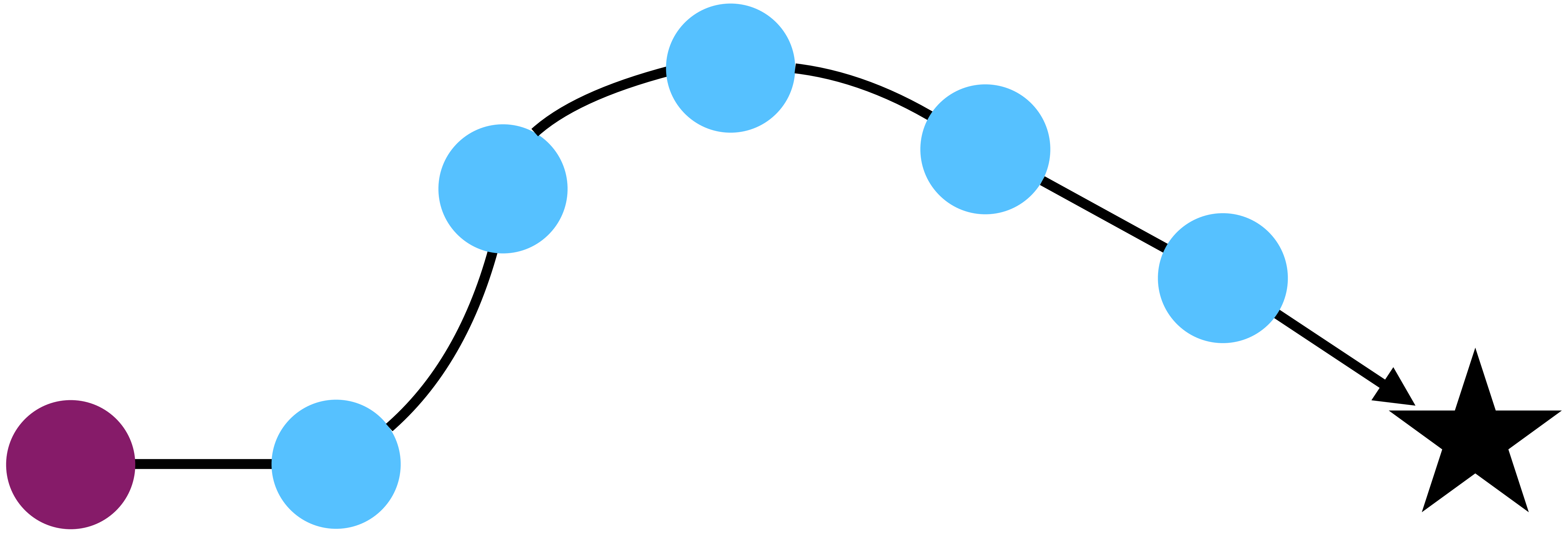
Model Predictive Control (MPC)

For each environment step

Observe state s

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

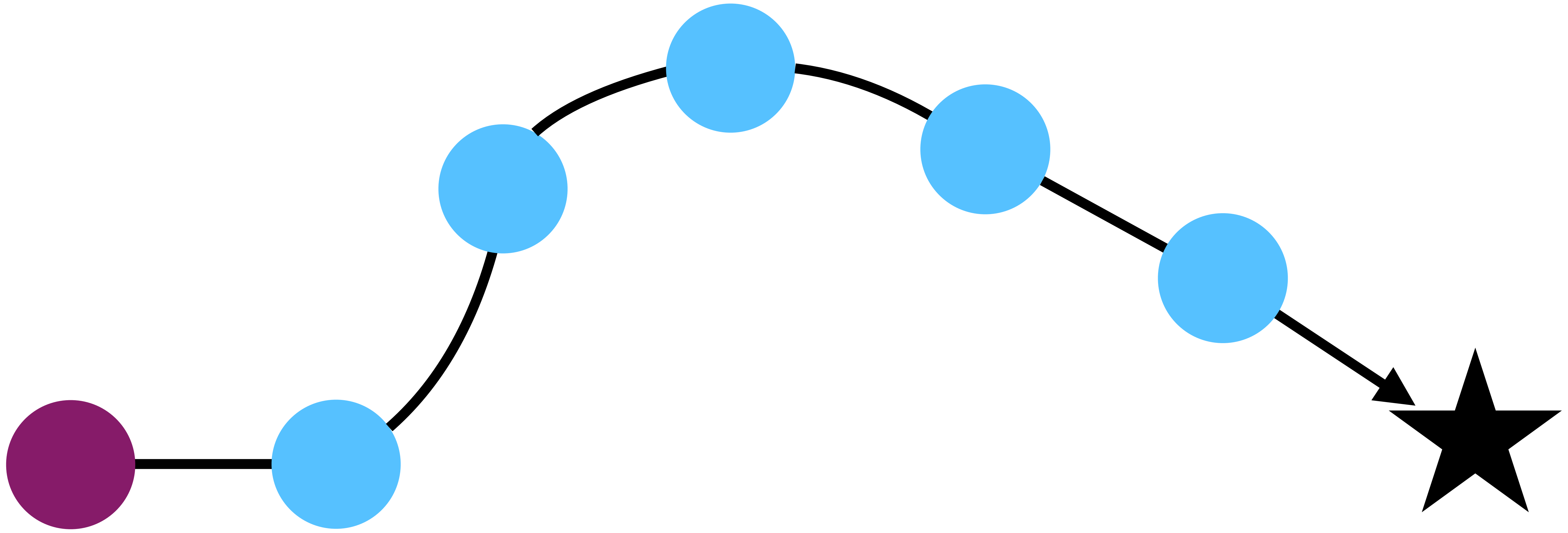
Plan $a_{0:H}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

FCAI

fcai.fi

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

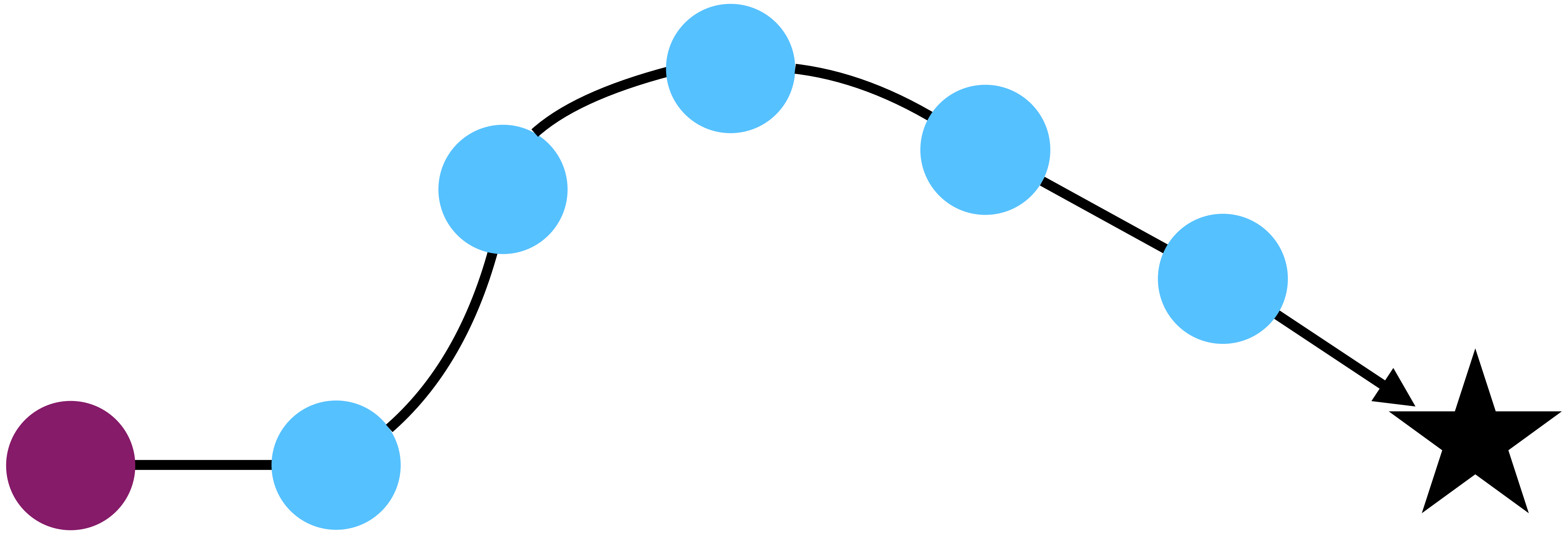
Observe state s

Model Predictive Path Integral Control (MPPI)

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

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

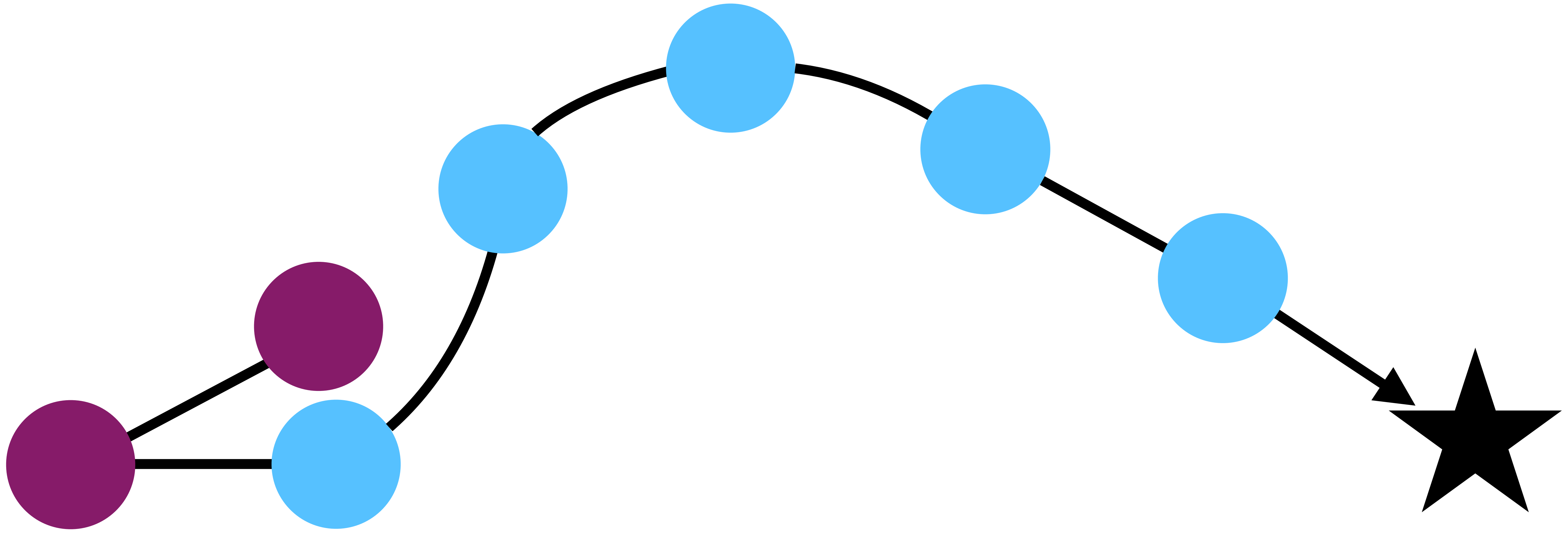
Model Predictive Path Integral Control (MPPI)

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

Model Predictive Path Integral Control (MPPI)

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

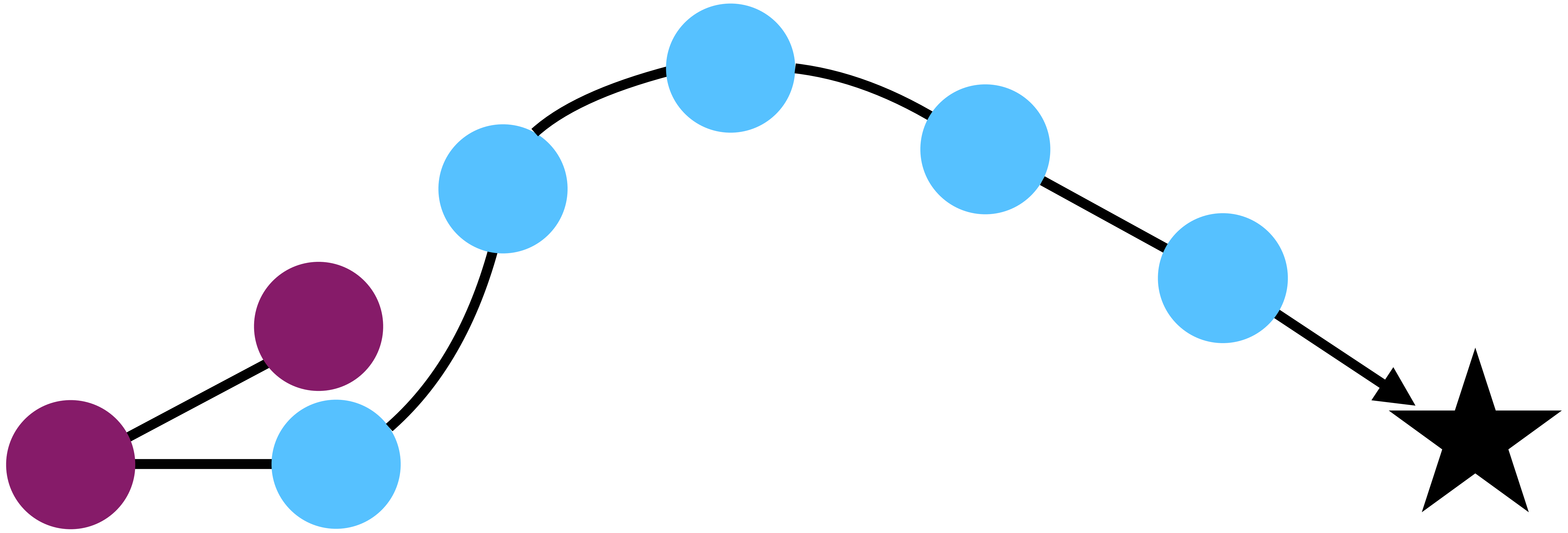
Execute a_0 and discard $a_1, \ldots, a_H$

FCAI

fcai.fi

Decision-time Planning

Model Predictive Control (MPC)



Diverged from planned trajectory…

For each environment step

Observe state s

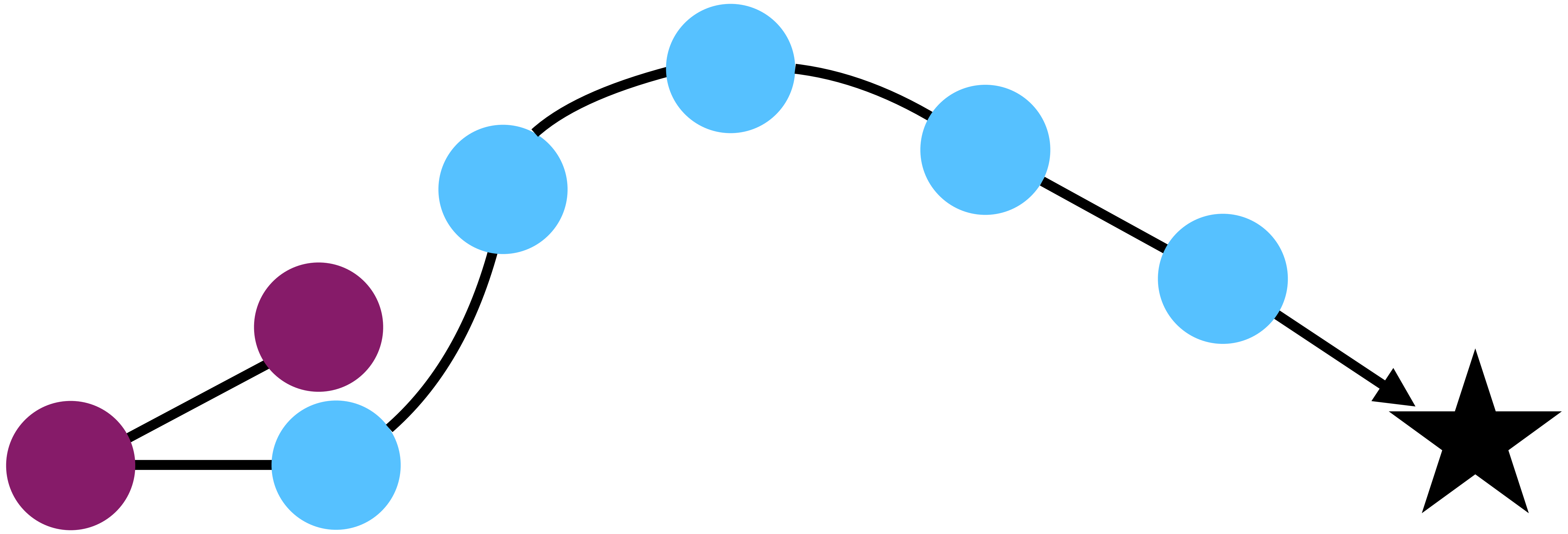
Model Predictive Path Integral Control (MPPI)

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

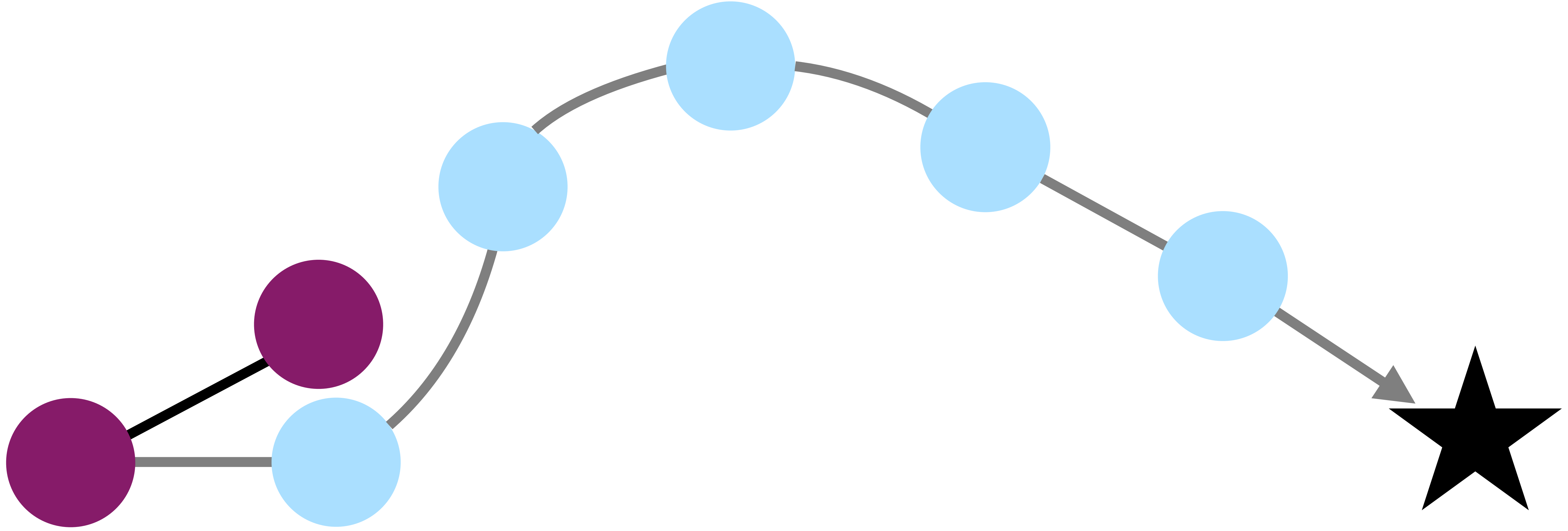
Model Predictive Path Integral Control (MPPI)

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

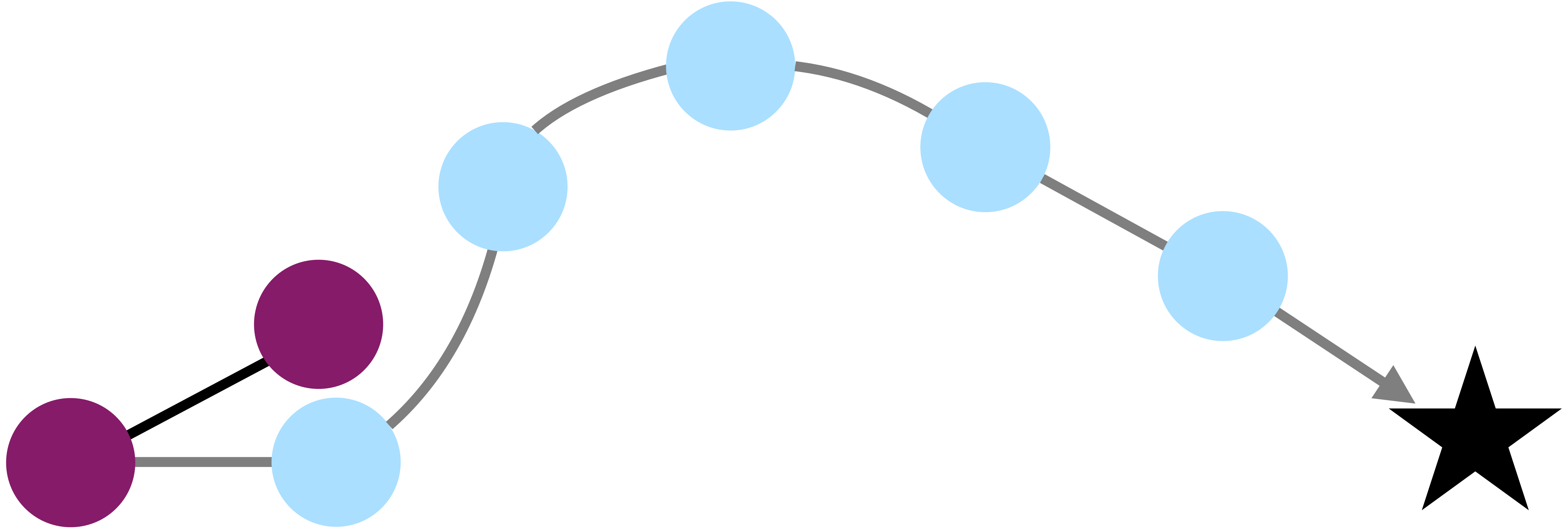
Model Predictive Path Integral Control (MPPI)

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

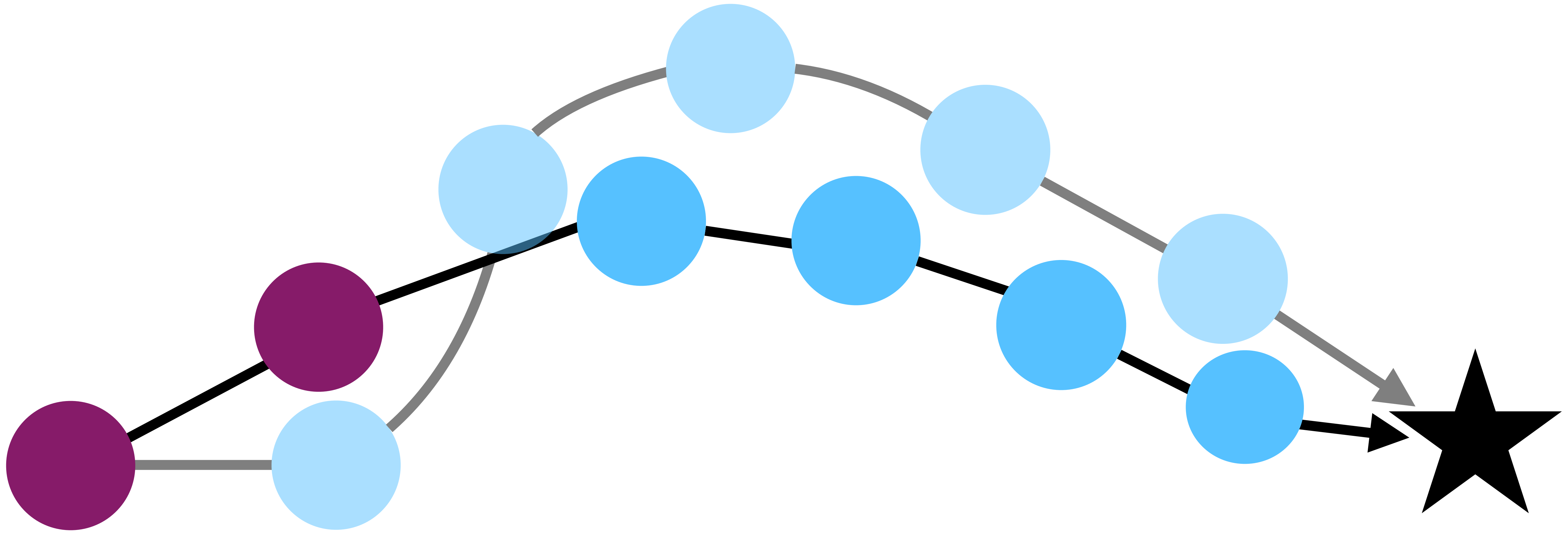
Model Predictive Path Integral Control (MPPI)

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

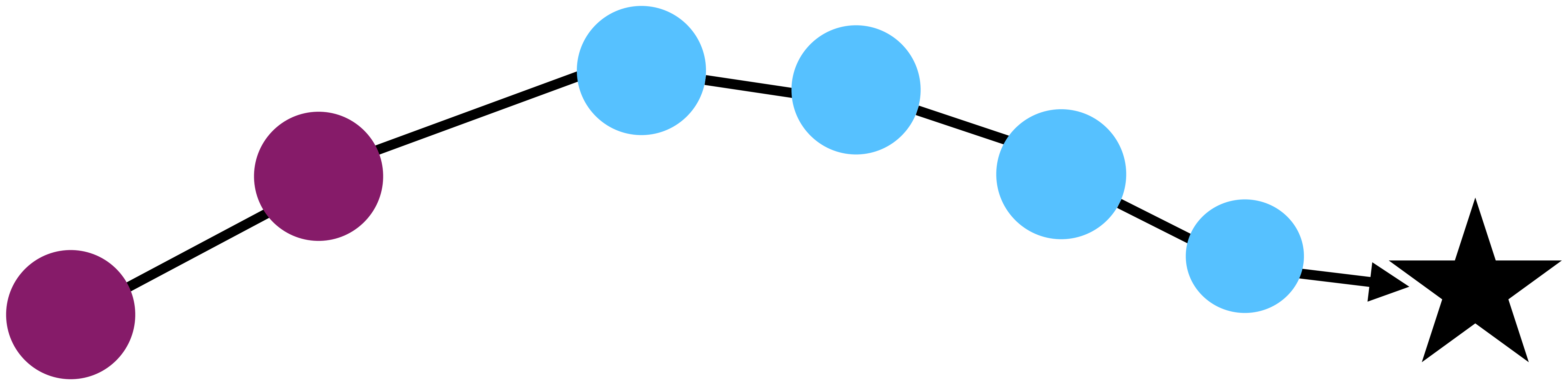
Model Predictive Path Integral Control (MPPI)

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

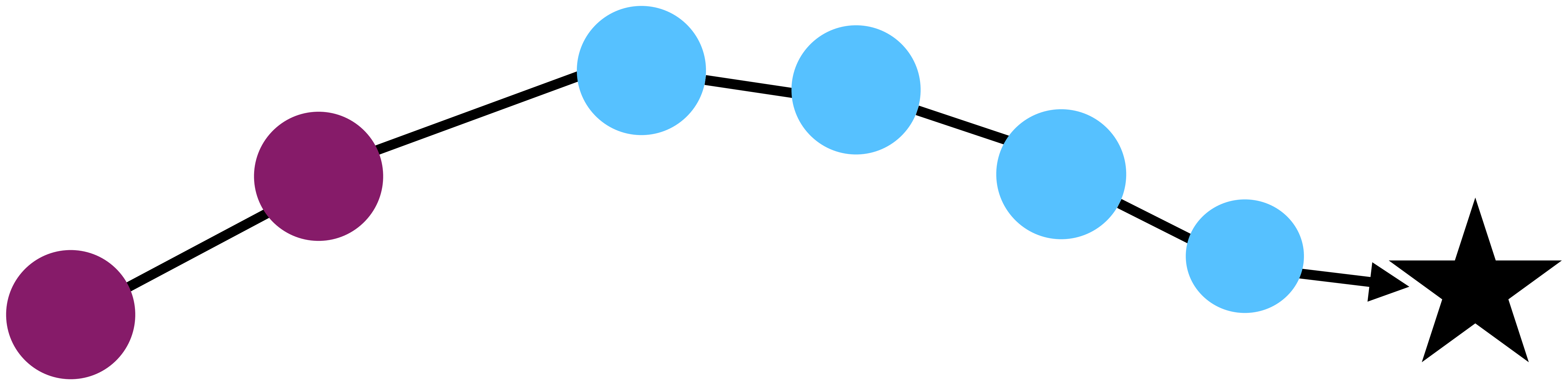
Model Predictive Path Integral Control (MPPI)

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

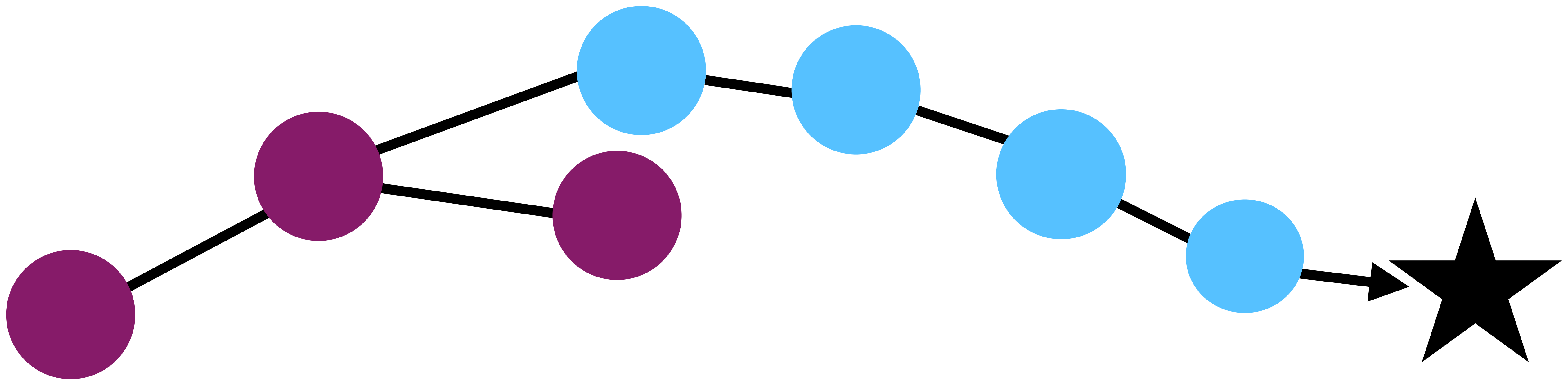
Model Predictive Path Integral Control (MPPI)

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

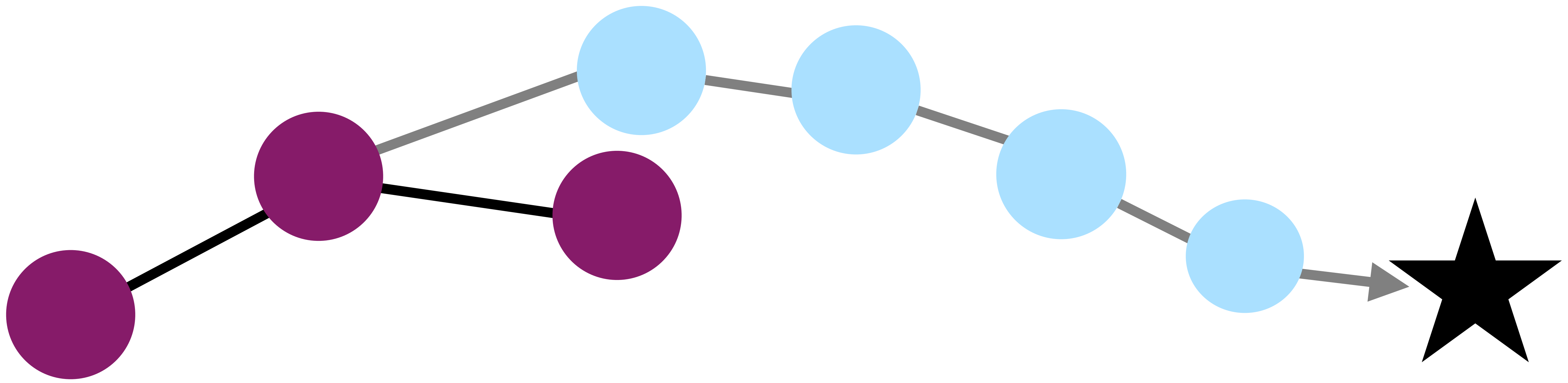
Model Predictive Path Integral Control (MPPI)

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

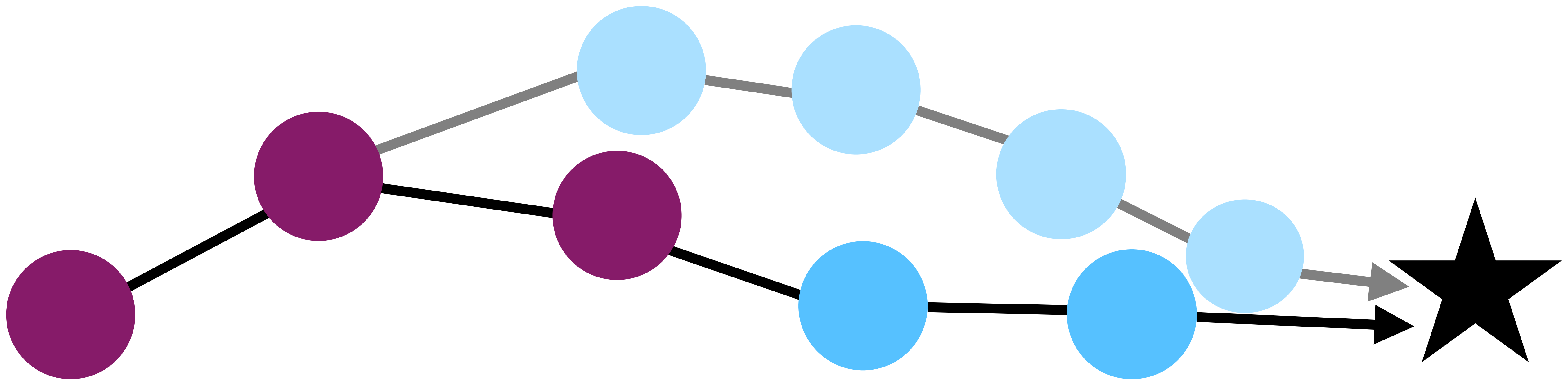
Model Predictive Path Integral Control (MPPI)

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 $a_1, \dots, a_H$

Decision-time Planning

Model Predictive Control (MPC)



For each environment step

Observe state s

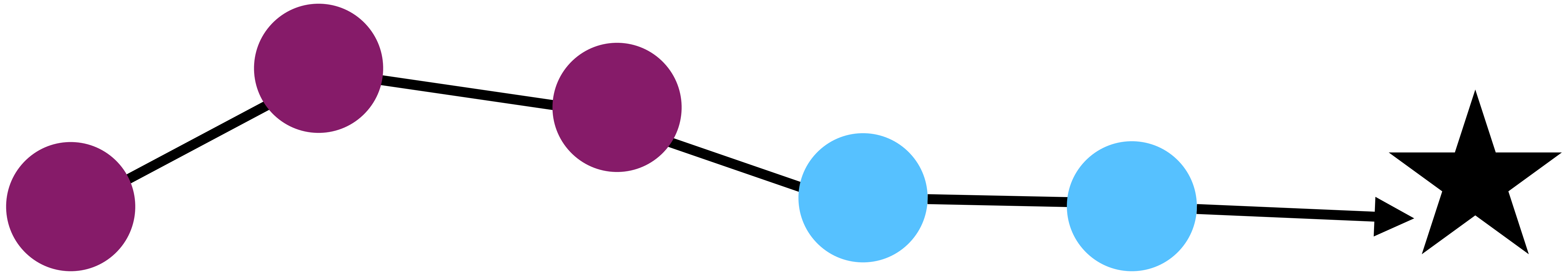
Model Predictive Path Integral Control (MPPI)

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

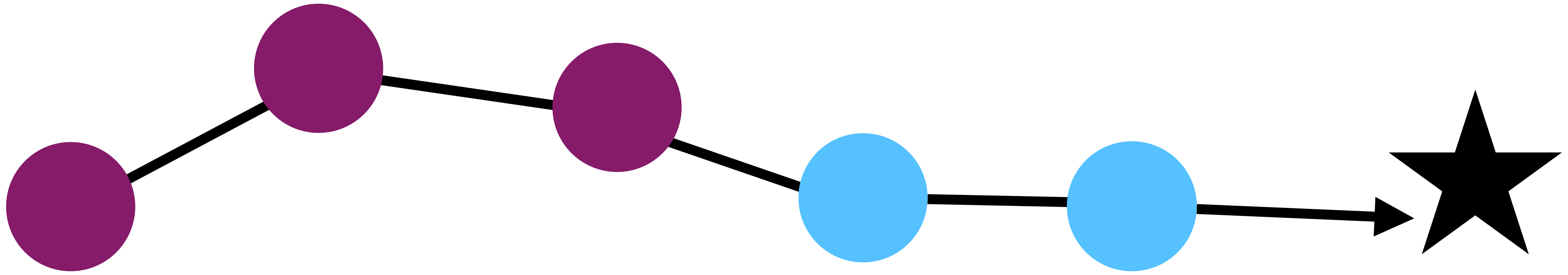
Model Predictive Path Integral Control (MPPI)

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

Model Predictive Path Integral Control (MPPI)

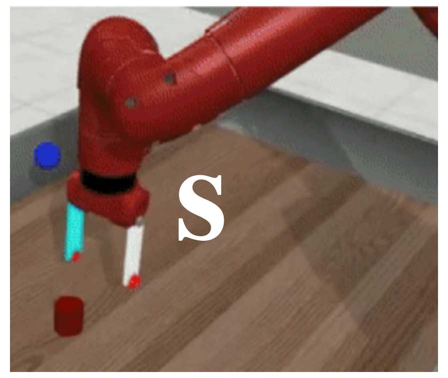
Plan $\boxed{a_{0:H}}$ to maximise return $\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)$

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

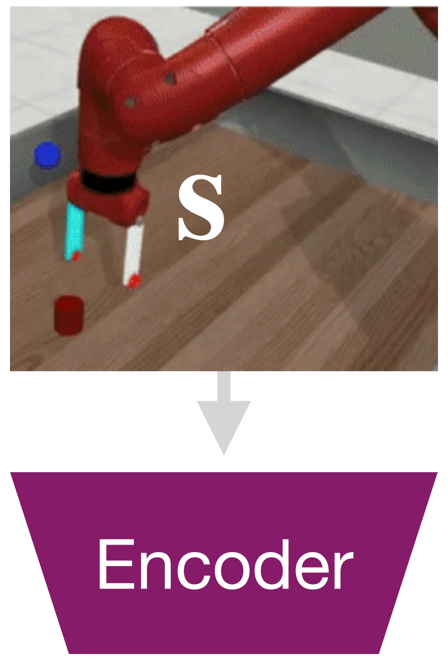
And so on...

DCWM: Decision-time Planning

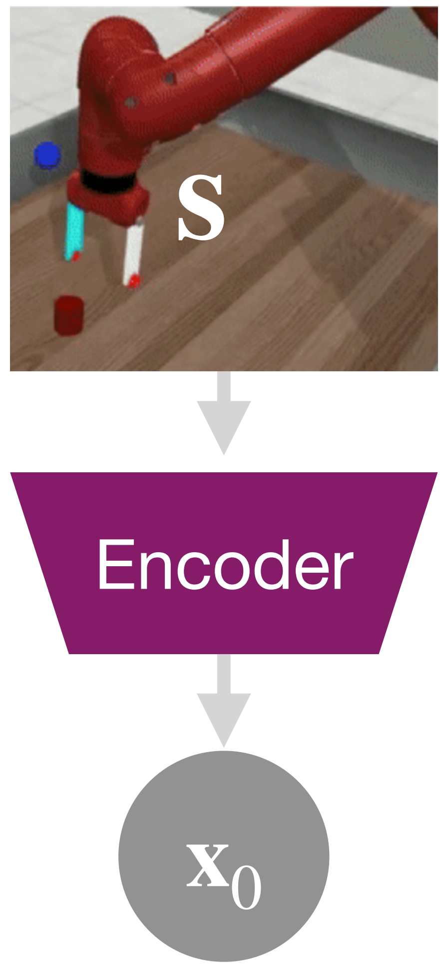
DCWM: Decision-time Planning



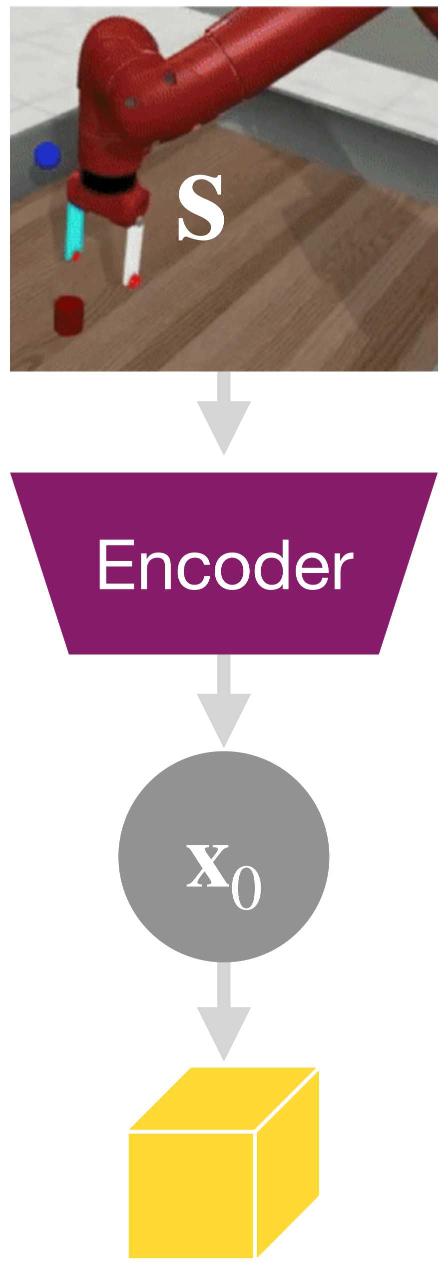
DCWM: Decision-time Planning



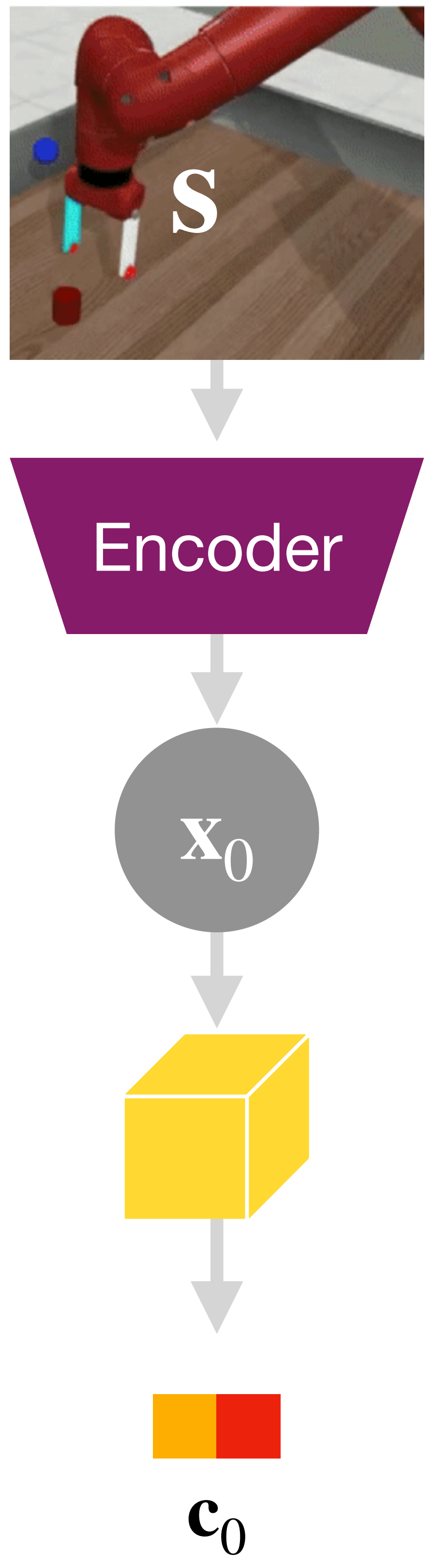
DCWM: Decision-time Planning



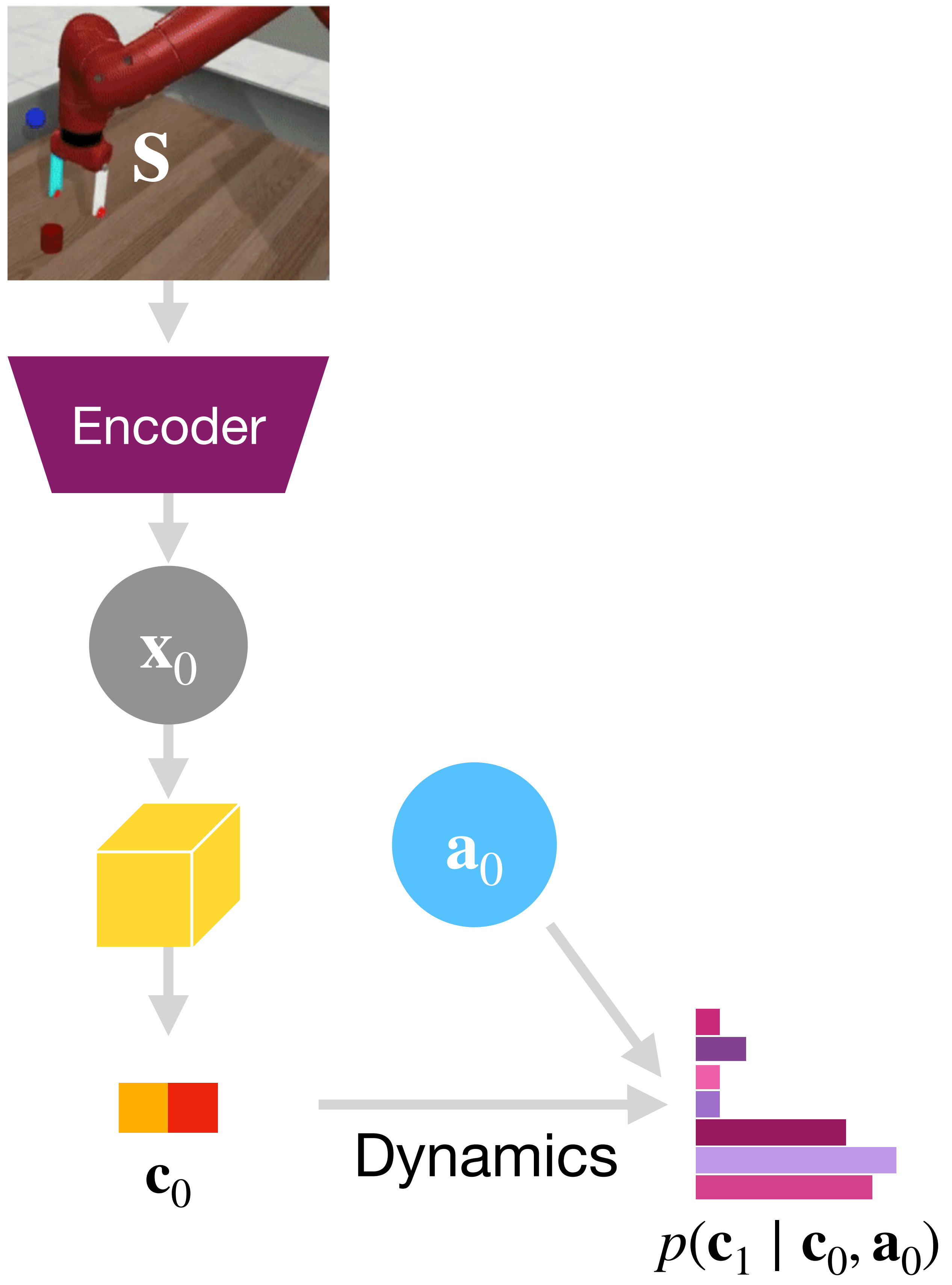
DCWM: Decision-time Planning



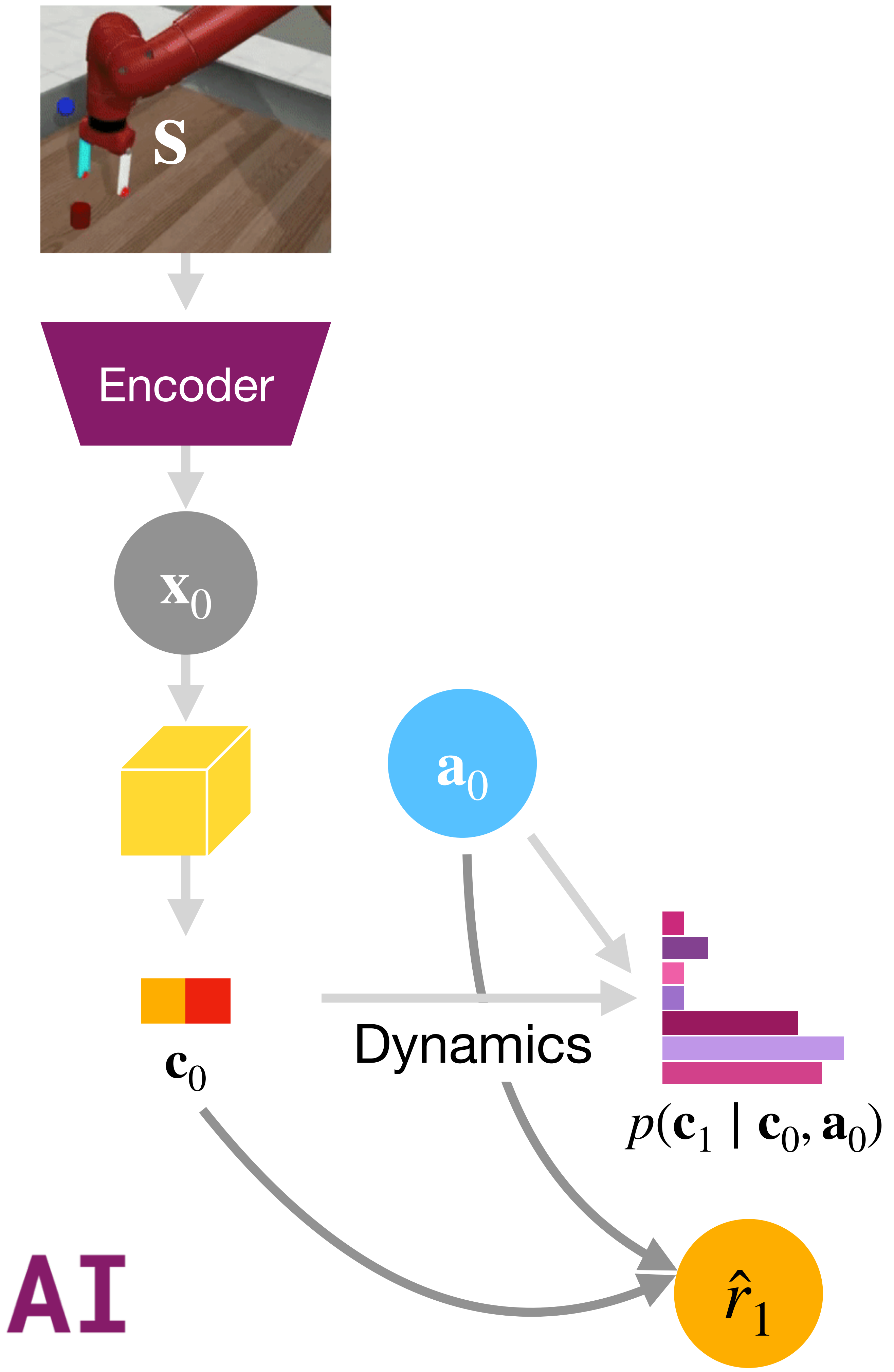
DCWM: Decision-time Planning



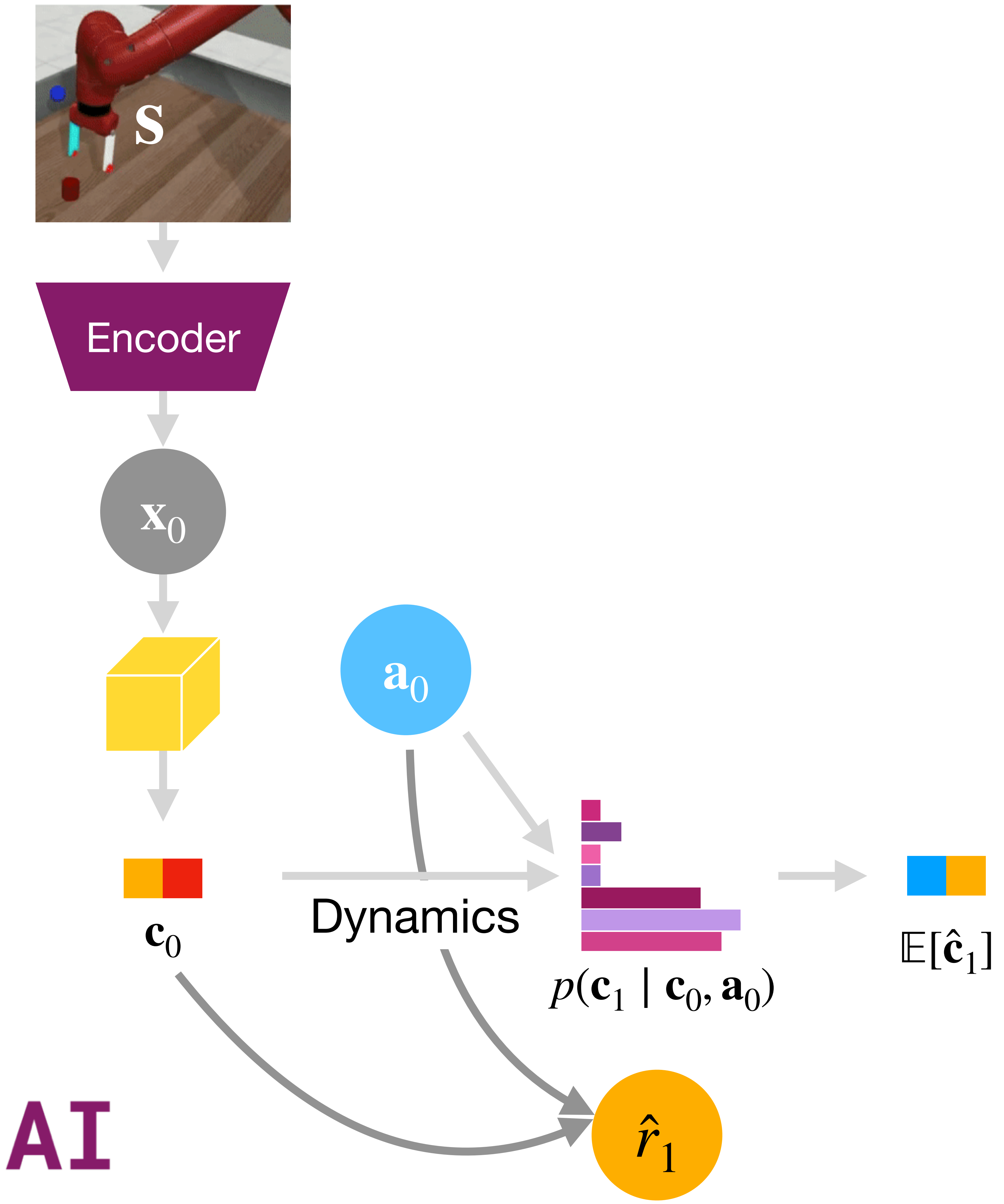
DCWM: Decision-time Planning



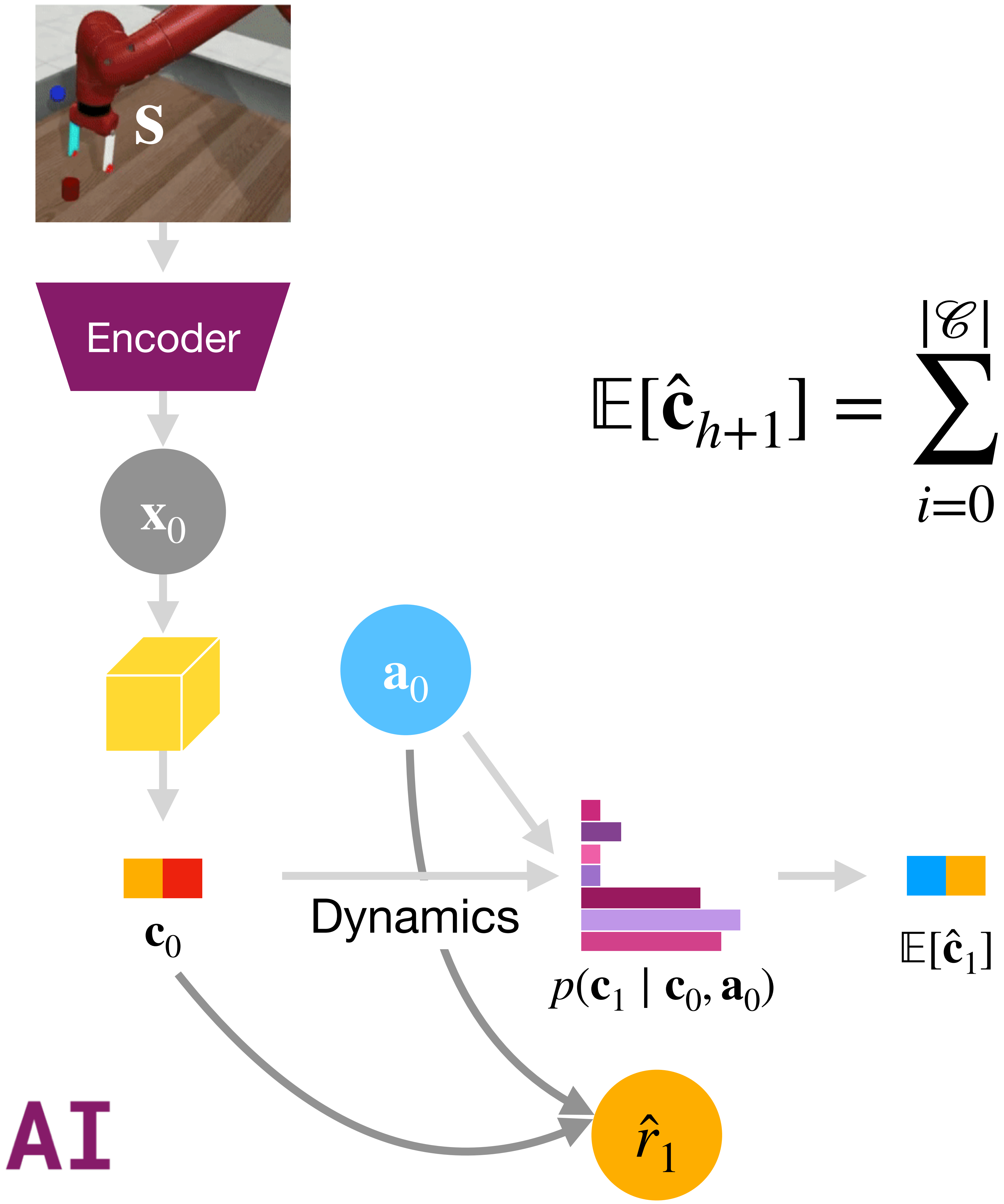
DCWM: Decision-time Planning



DCWM: Decision-time Planning

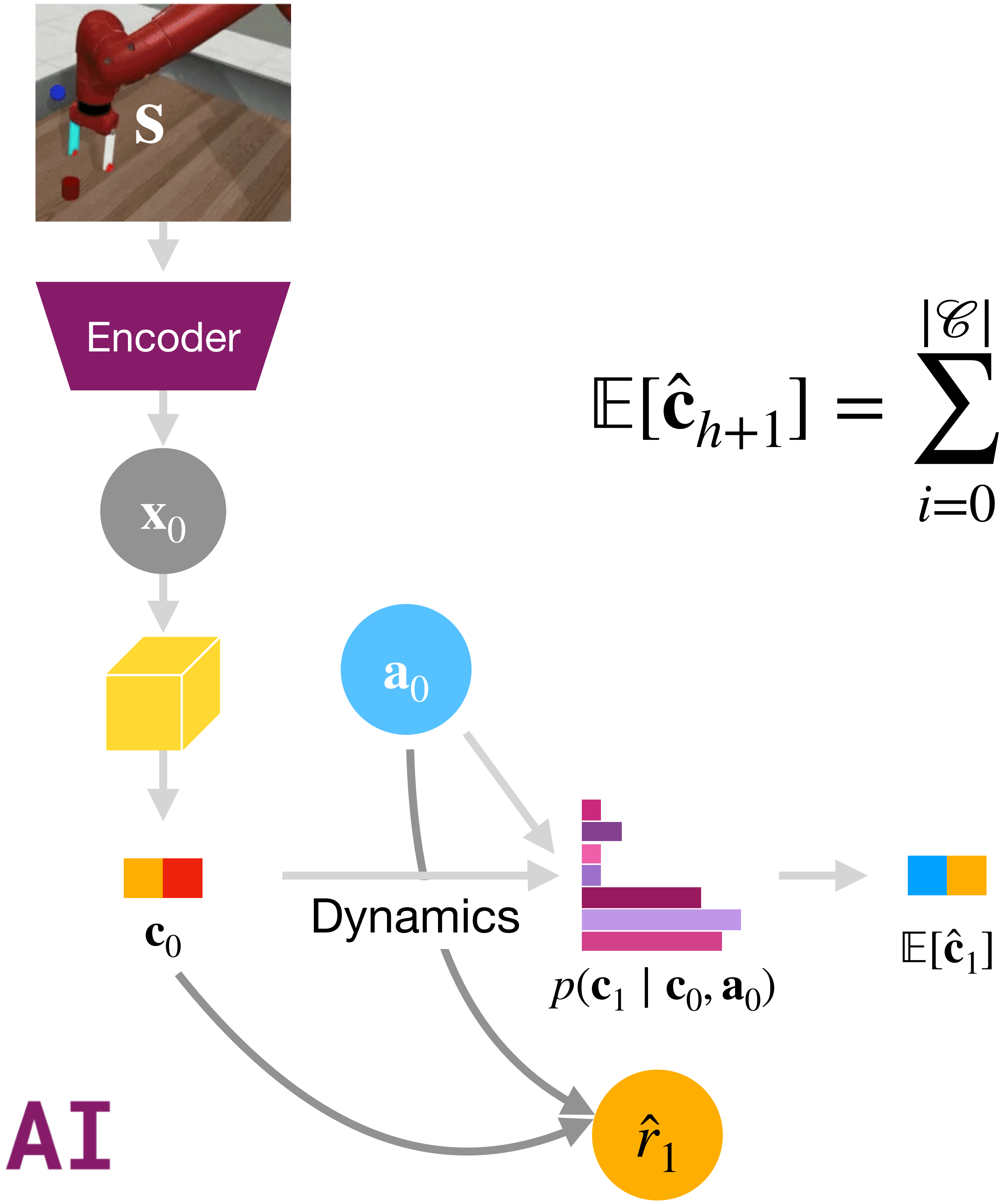


DCWM: Decision-time Planning



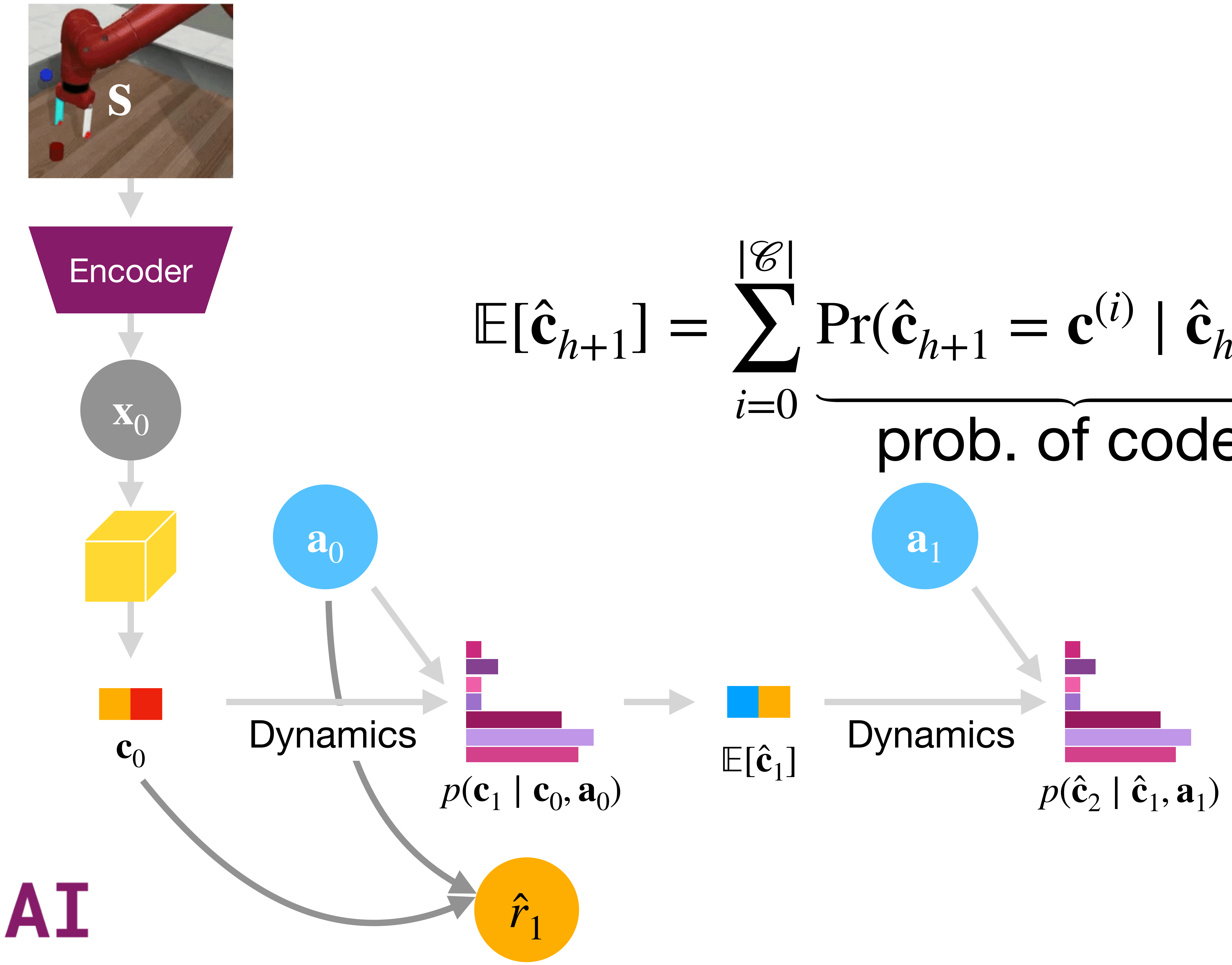
$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$

DCWM: Decision-time Planning



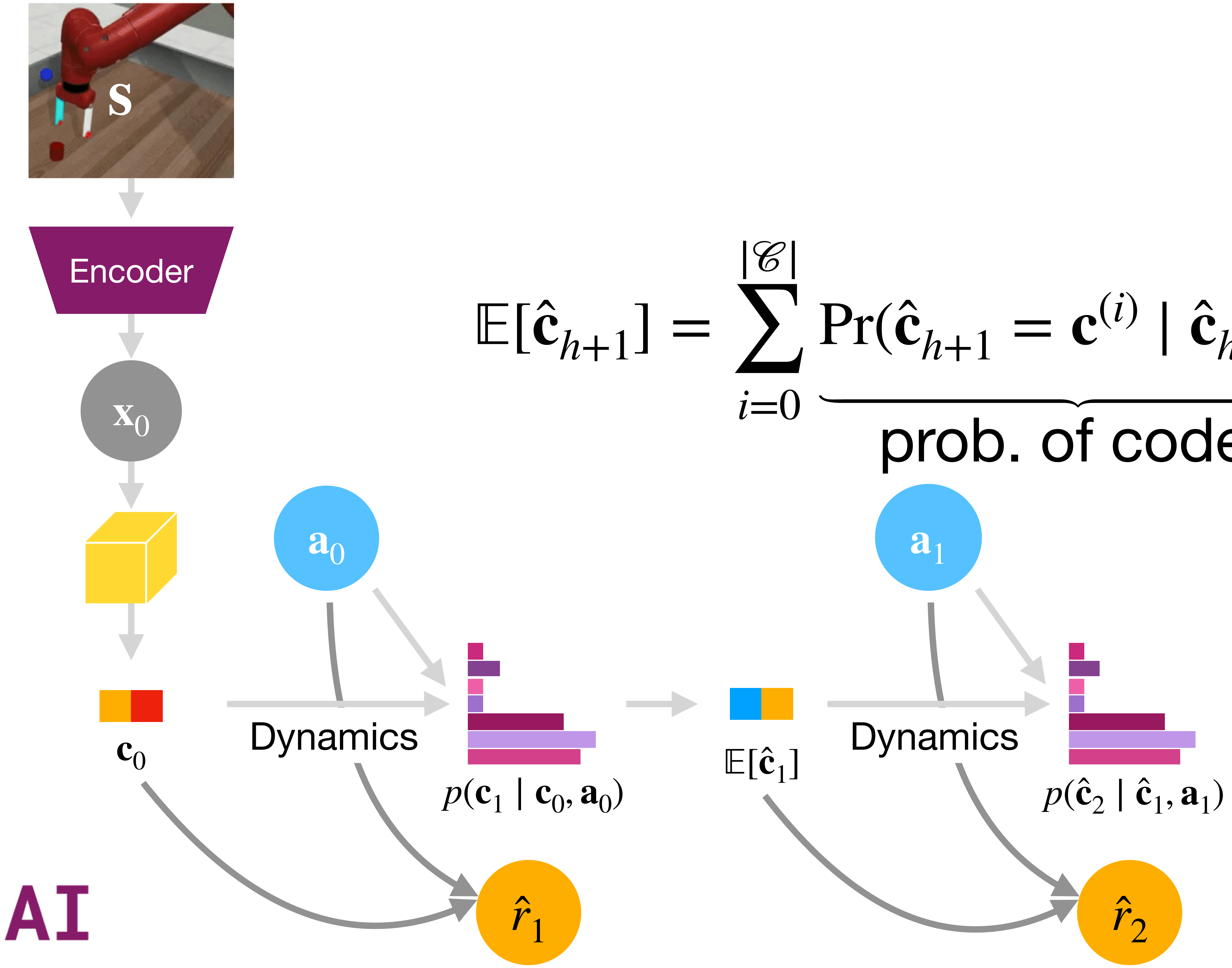
$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$

DCWM: Decision-time Planning



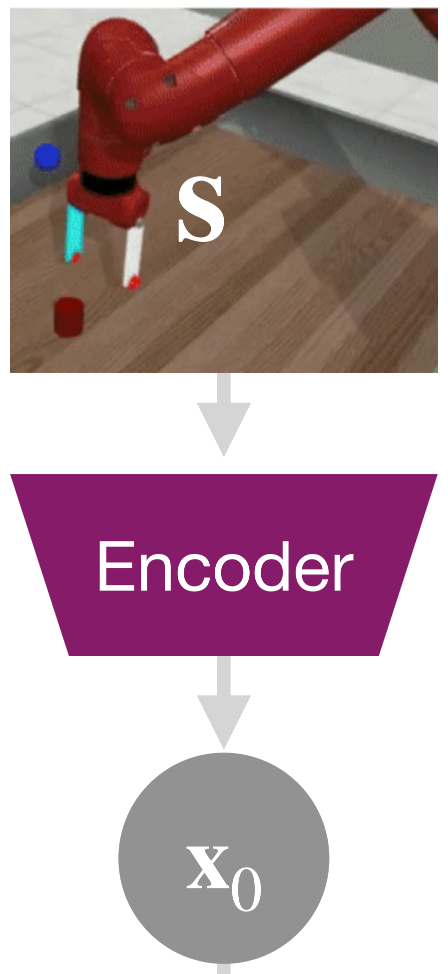
$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$

DCWM: Decision-time Planning

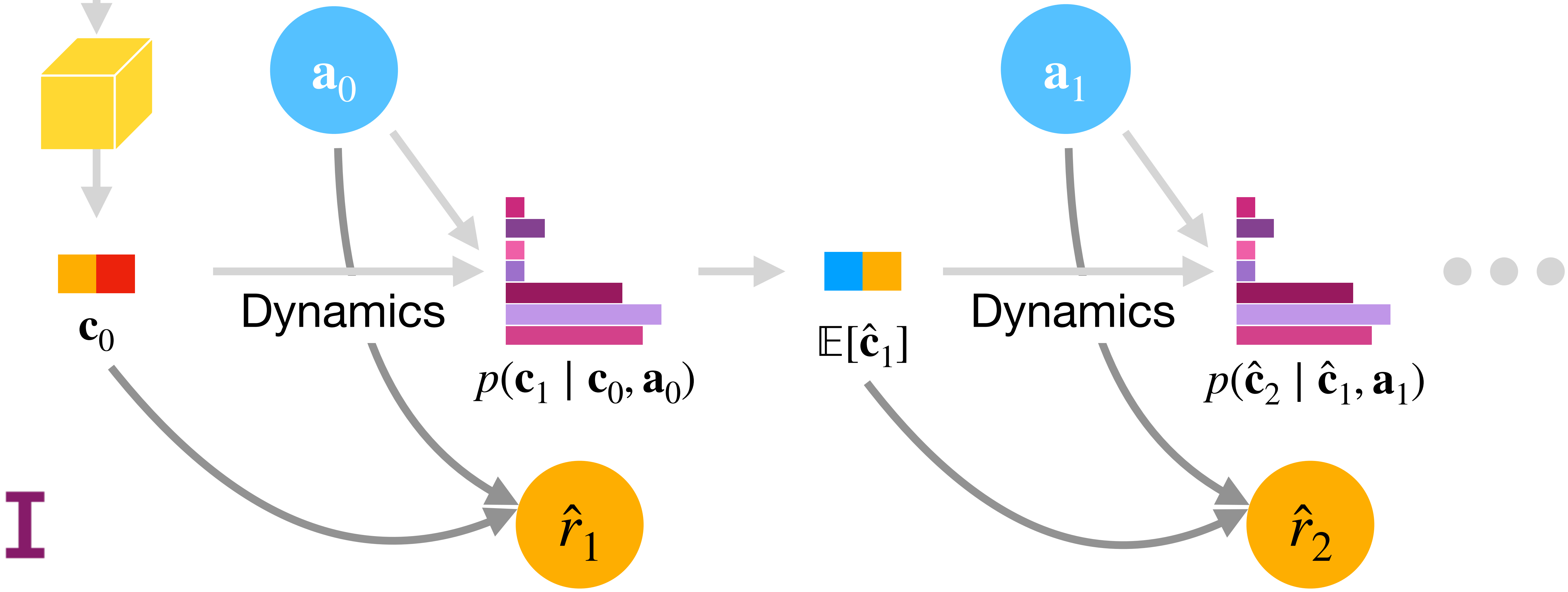


$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \; \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$

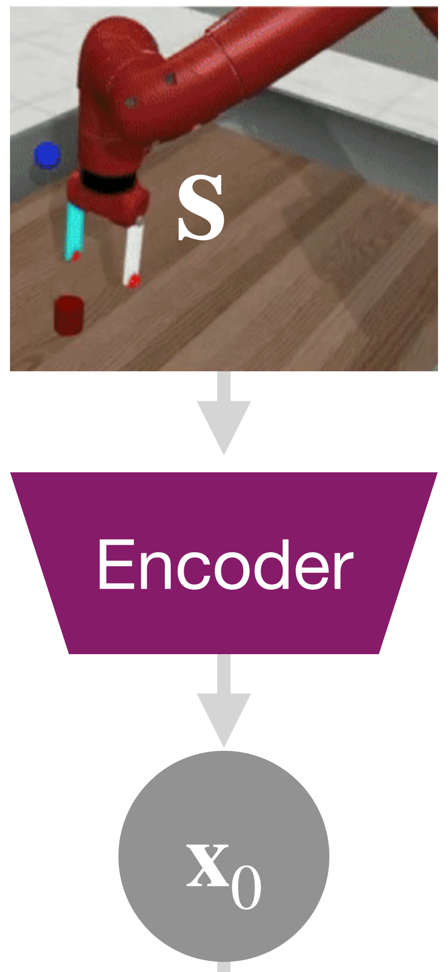
DCWM: Decision-time Planning



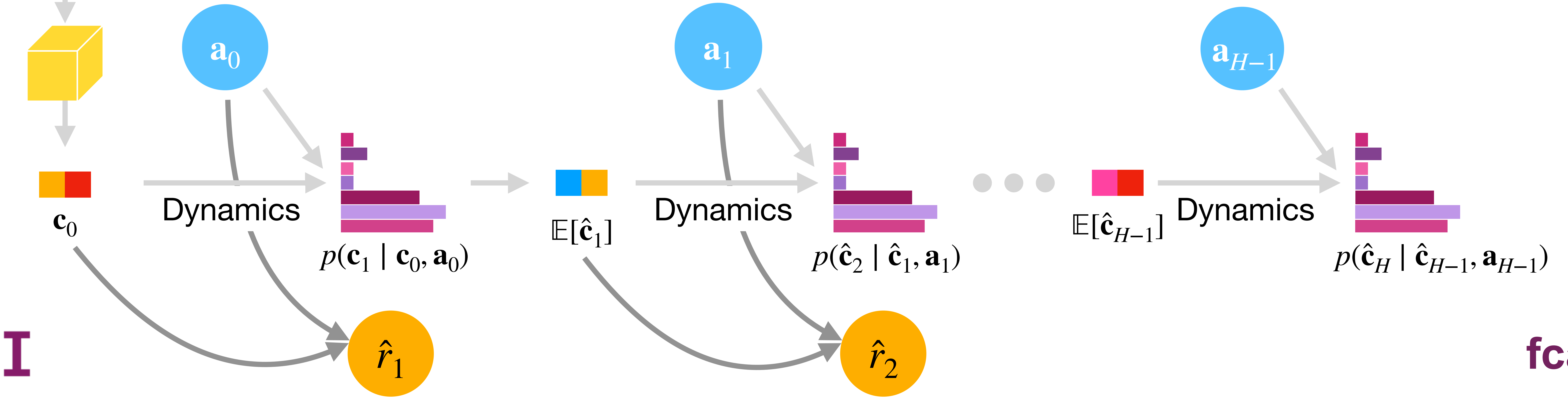
$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \; \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$



DCWM: Decision-time Planning

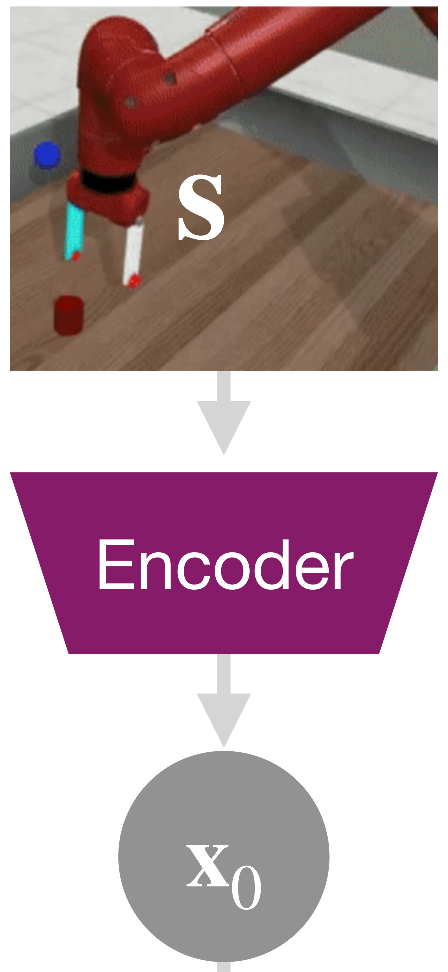


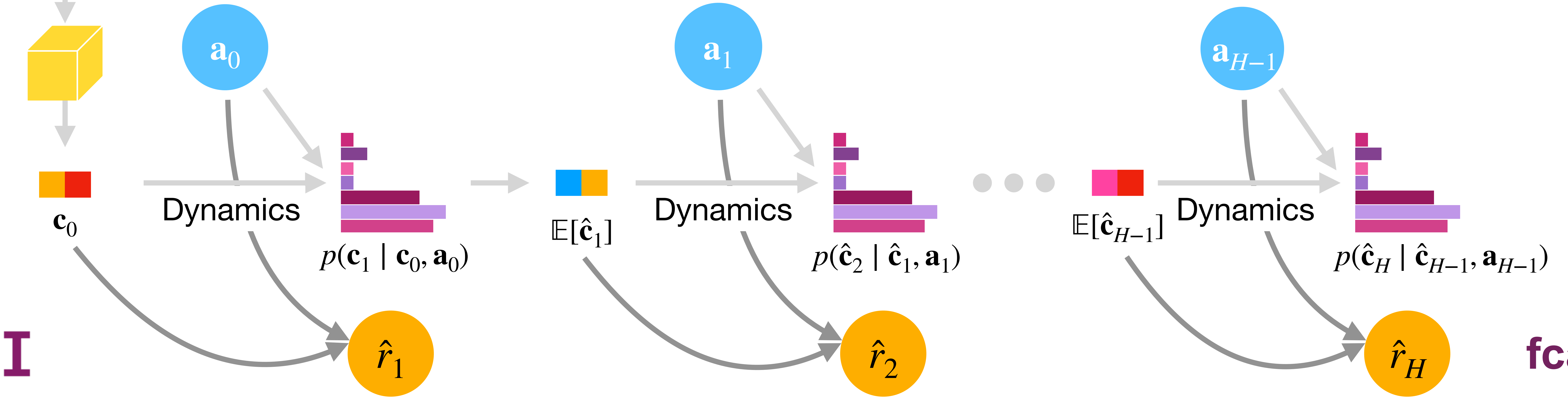
$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$



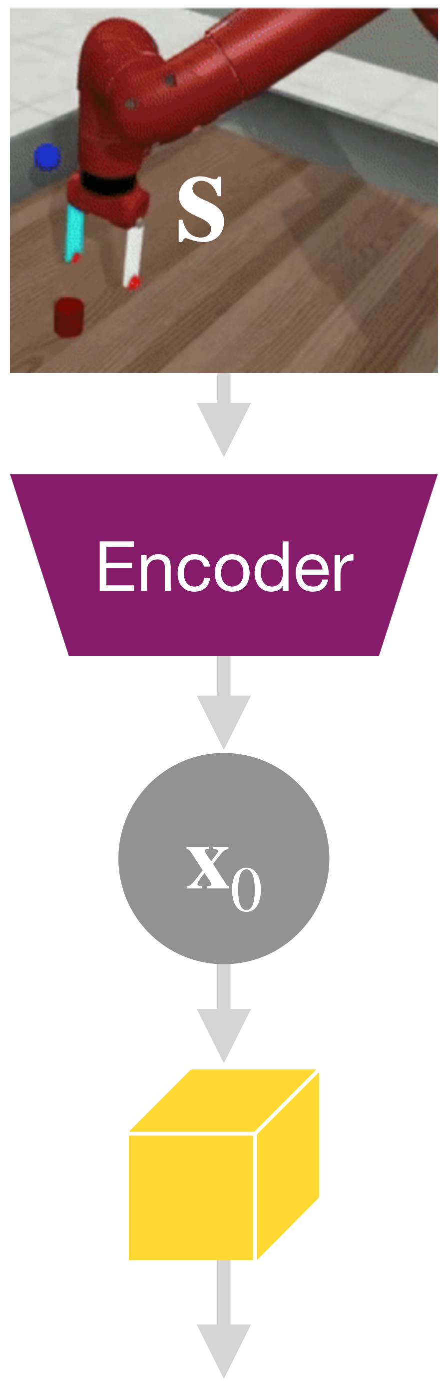
DCWM: Decision-time Planning

$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$



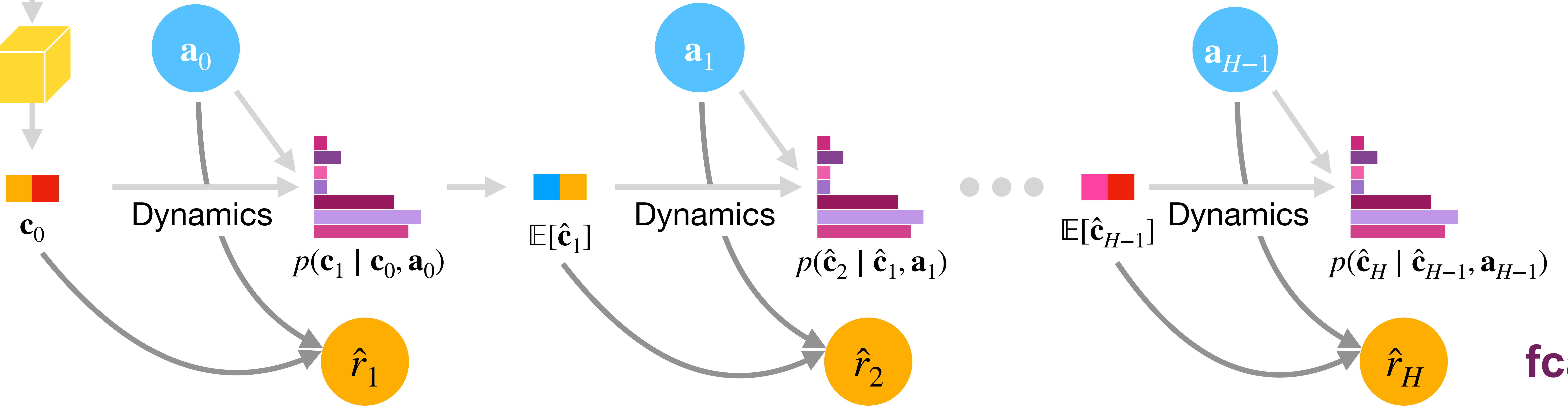


DCWM: Decision-time Planning

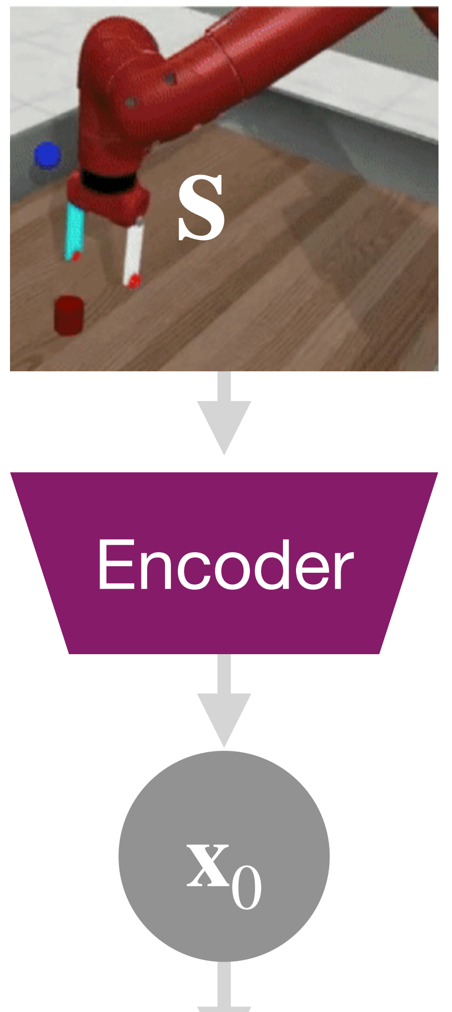


$$J(\mathbf{a}_{0:H}, \mathbf{s}) = \gamma^H Q_\psi(\hat{\mathbf{c}}_H, \mathbf{a}_H) + \sum_{h=0}^{H-1} \gamma^h R_\xi(\hat{\mathbf{c}}_h, \mathbf{a}_h)$$

$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$



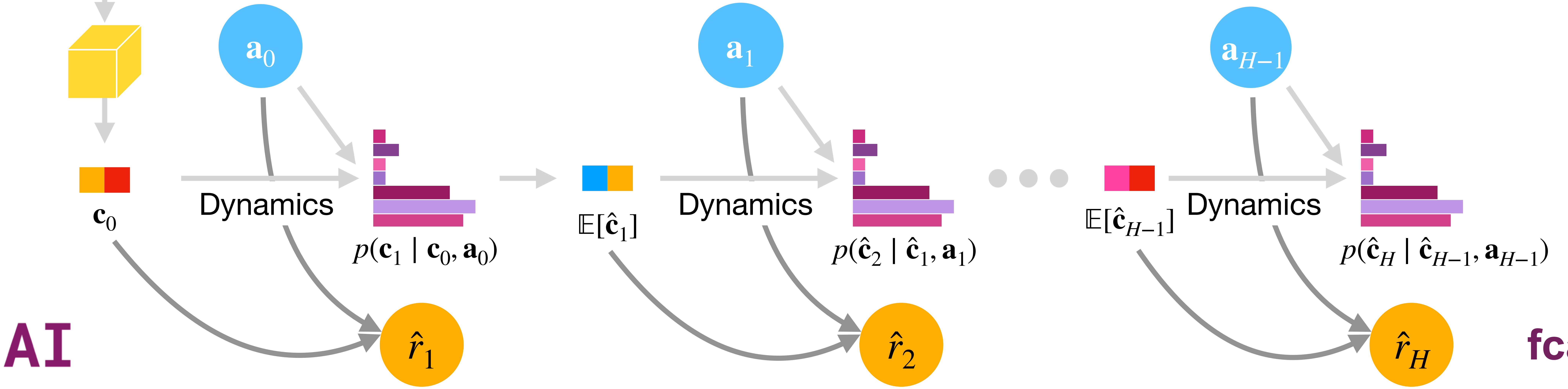
DCWM: Decision-time Planning



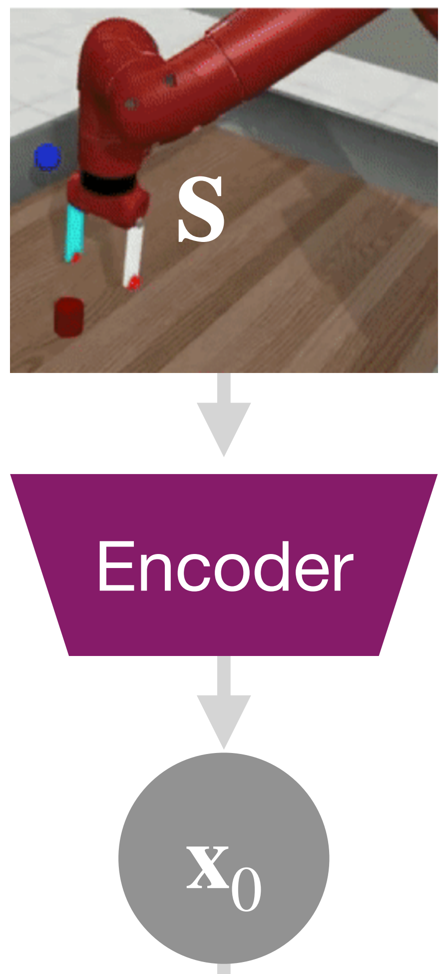
Reward func.

$$J(\mathbf{a}_{0:H}, \mathbf{s}) = \gamma^H Q_\psi(\hat{\mathbf{c}}_H, \mathbf{a}_H) + \boxed{\sum_{h=0}^{H-1} \gamma^h R_\xi(\hat{\mathbf{c}}_h, \mathbf{a}_h)}$$

$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \; \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$

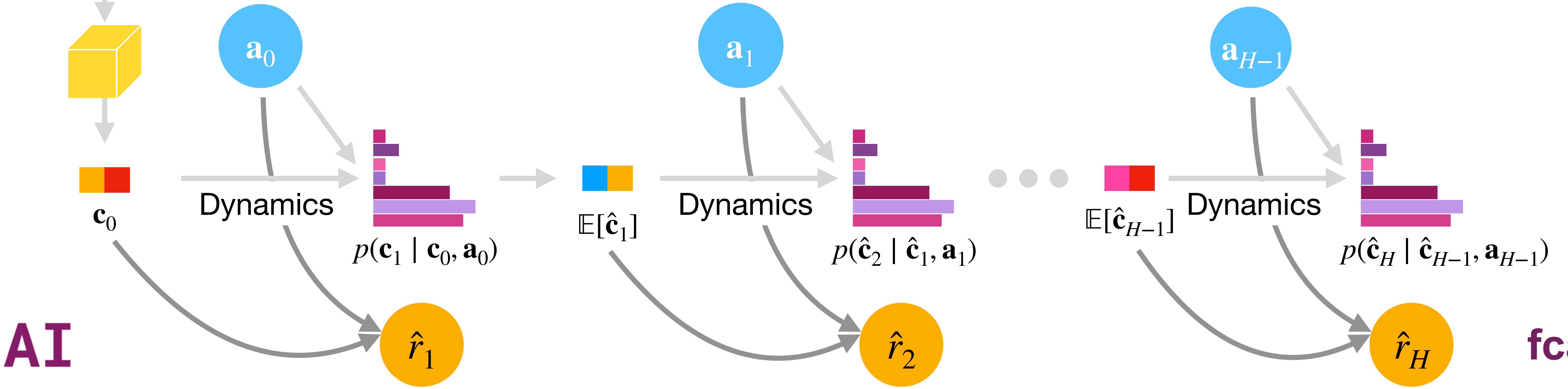


DCWM: Decision-time Planning



$$J(\mathbf{a}_{0:H}, \mathbf{s}) = \underbrace{\boxed{\gamma^H Q_\psi(\hat{\mathbf{c}}_H, \mathbf{a}_H)}}_{\text{Bootstrap with action-value}} + \underbrace{\boxed{\sum_{h=0}^{H-1} \gamma^h R_\xi(\hat{\mathbf{c}}_h, \mathbf{a}_h)}}_{\text{Reward func.}}$$

$$\mathbb{E}[\hat{\mathbf{c}}_{h+1}] = \sum_{i=0}^{|\mathscr{C}|} \underbrace{\Pr(\hat{\mathbf{c}}_{h+1} = \mathbf{c}^{(i)} \mid \hat{\mathbf{c}}_h, \mathbf{a}_h)}_{\text{prob. of code } i} \underbrace{\mathbf{c}^{(i)}}_{\text{code}}$$



DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1

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

For each iteration

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1

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

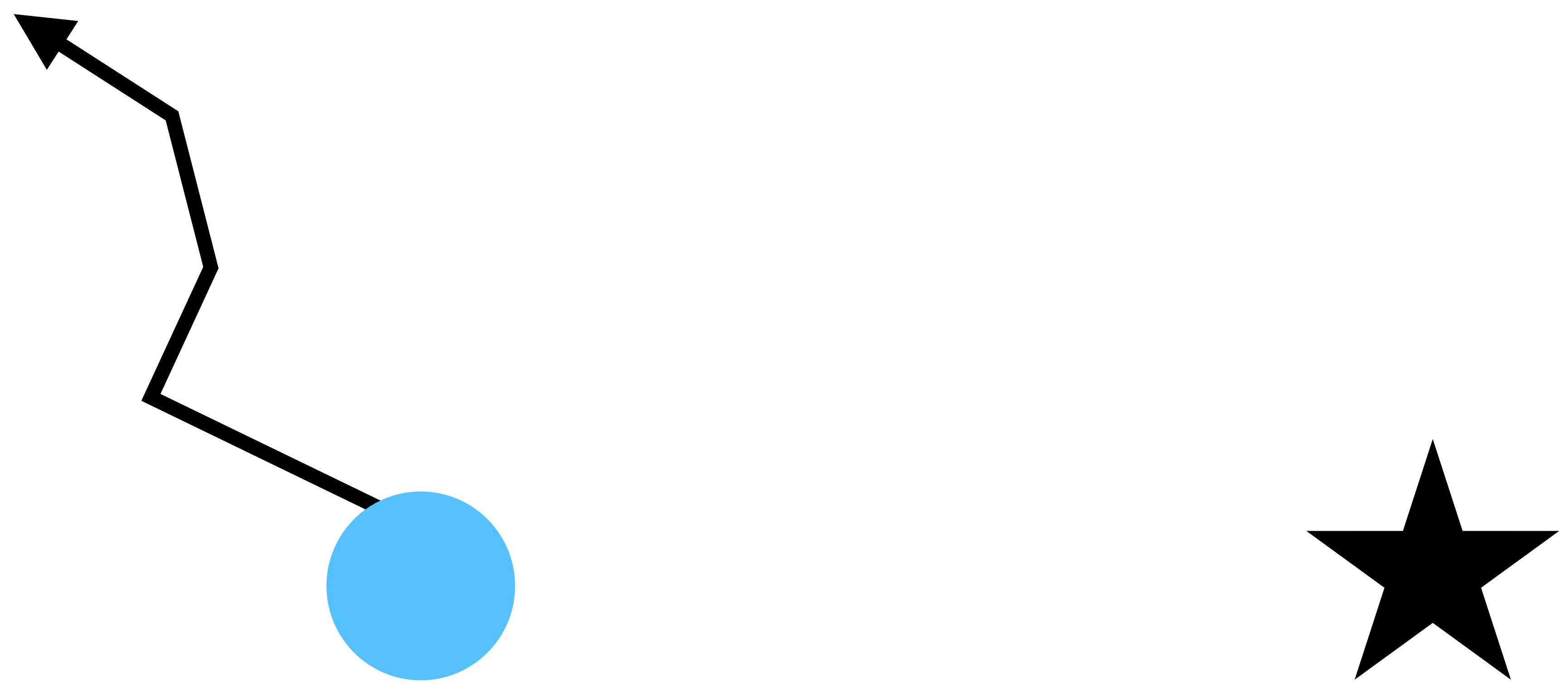
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

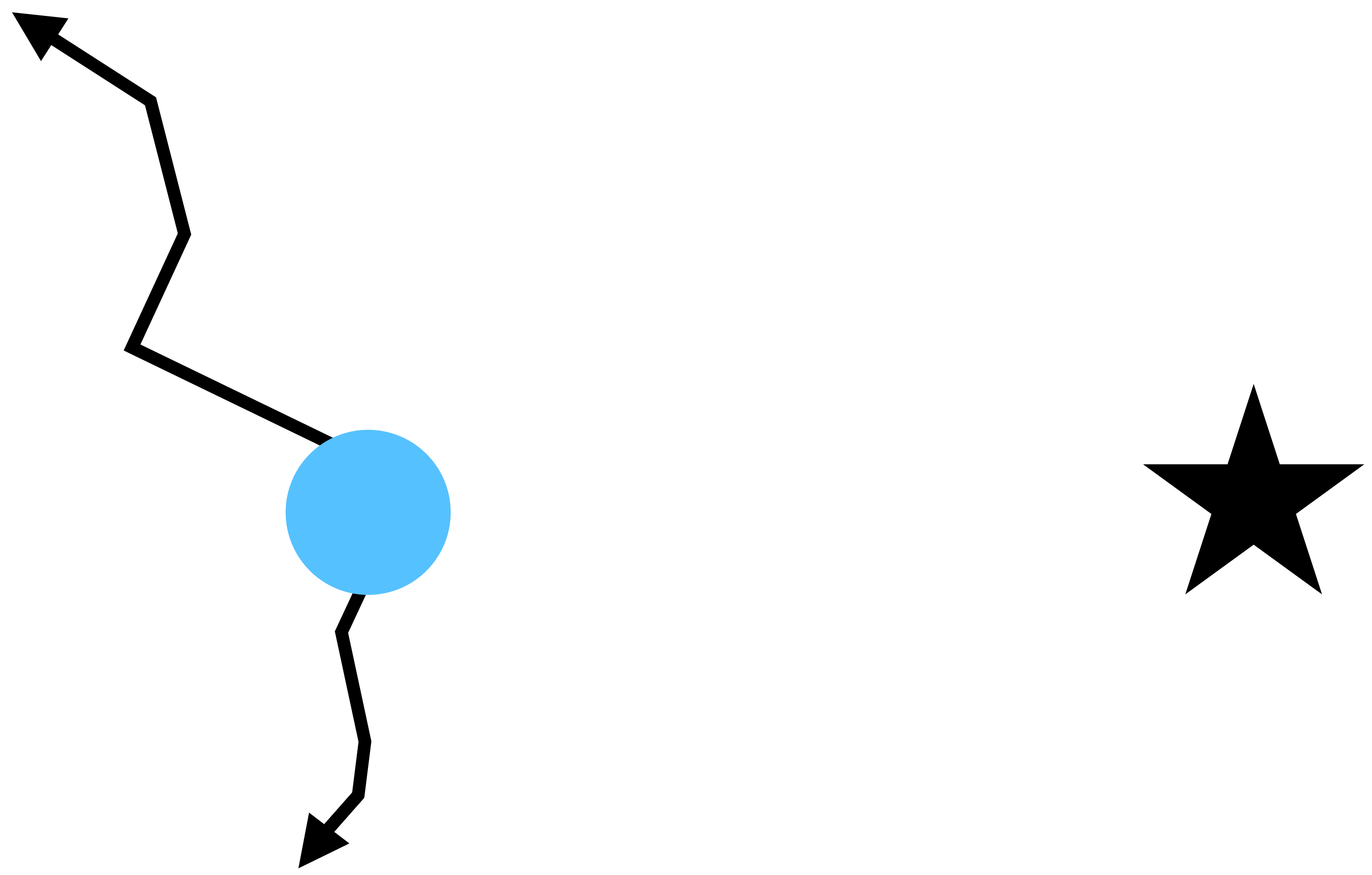
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

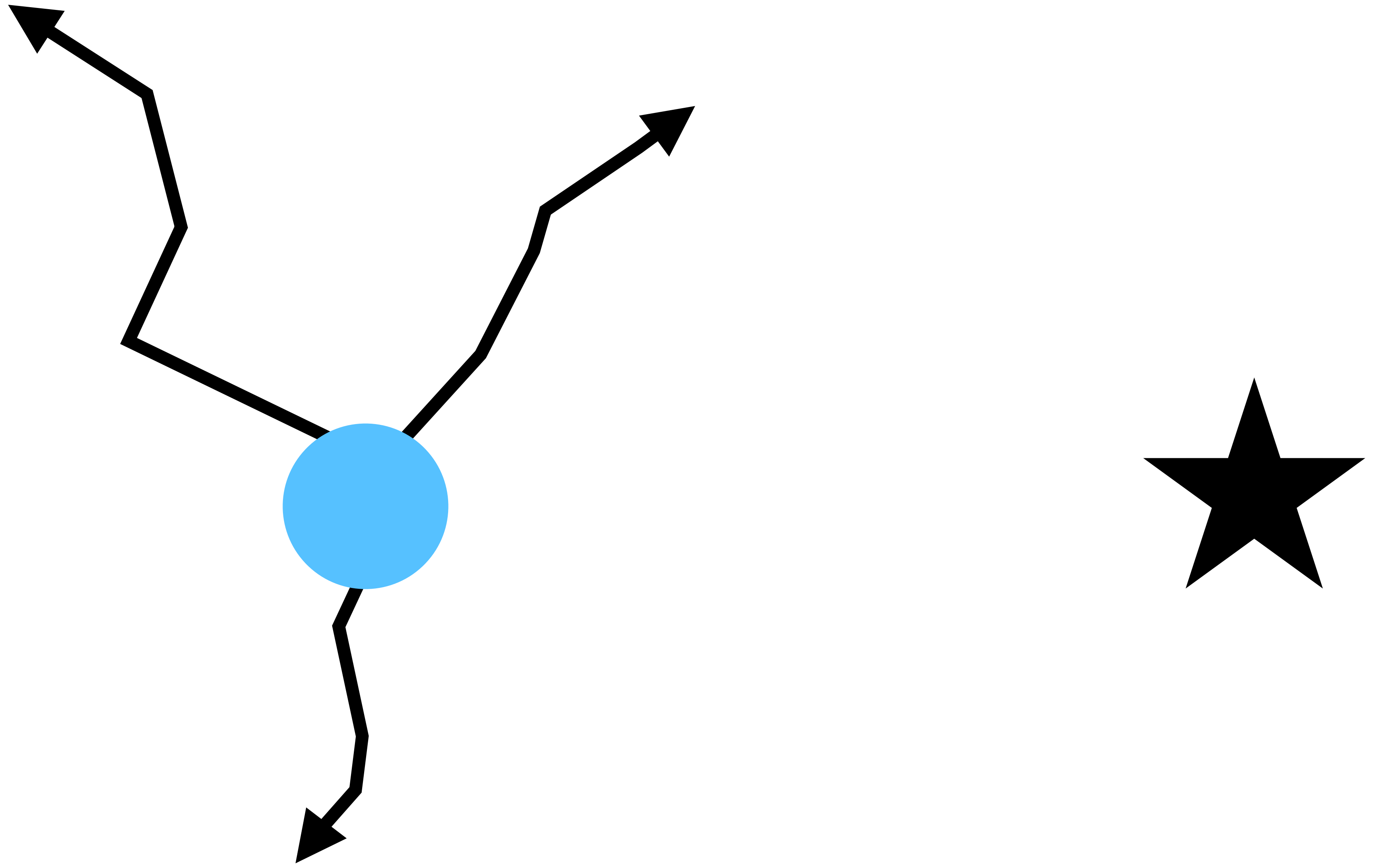
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

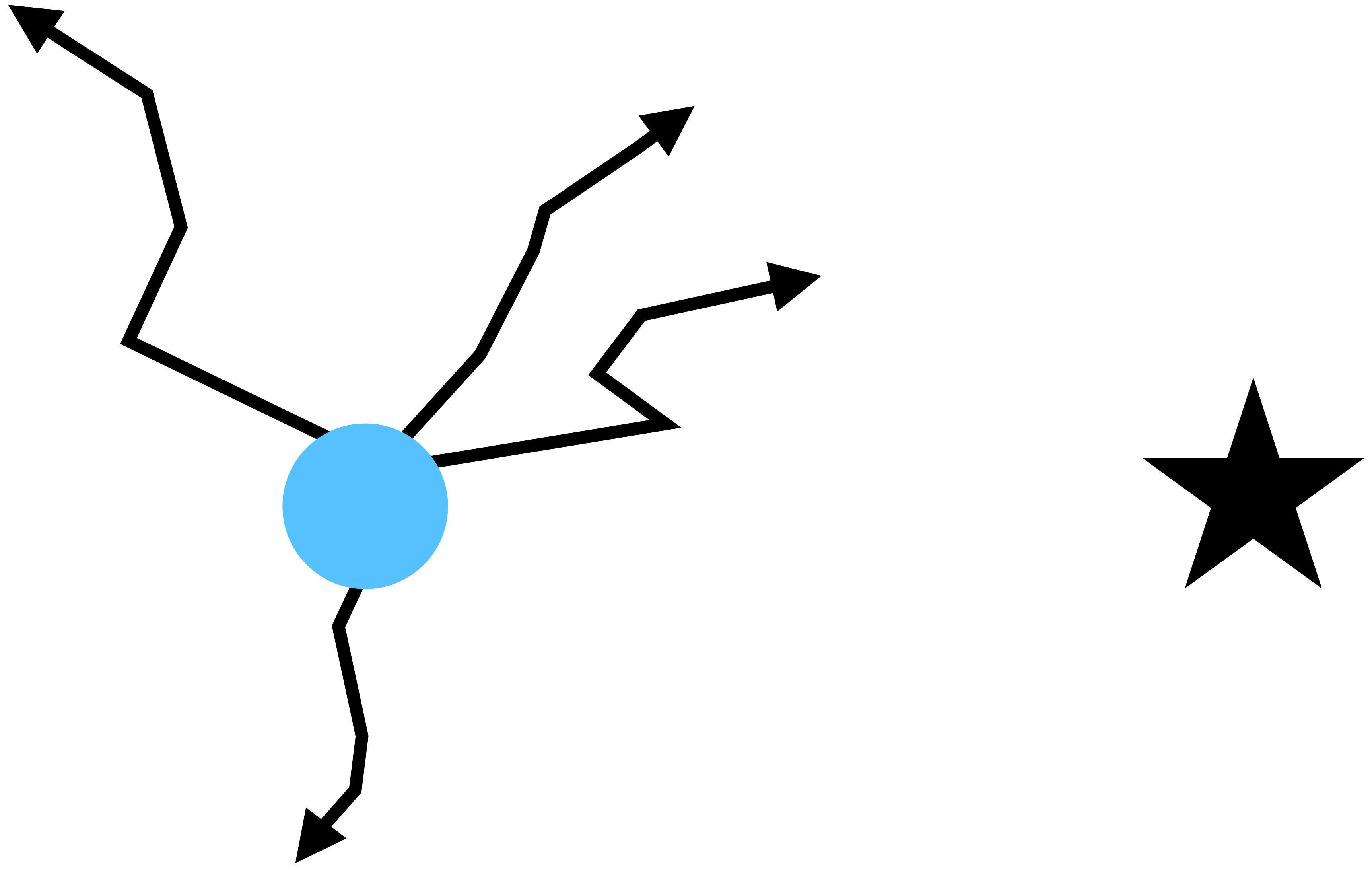
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

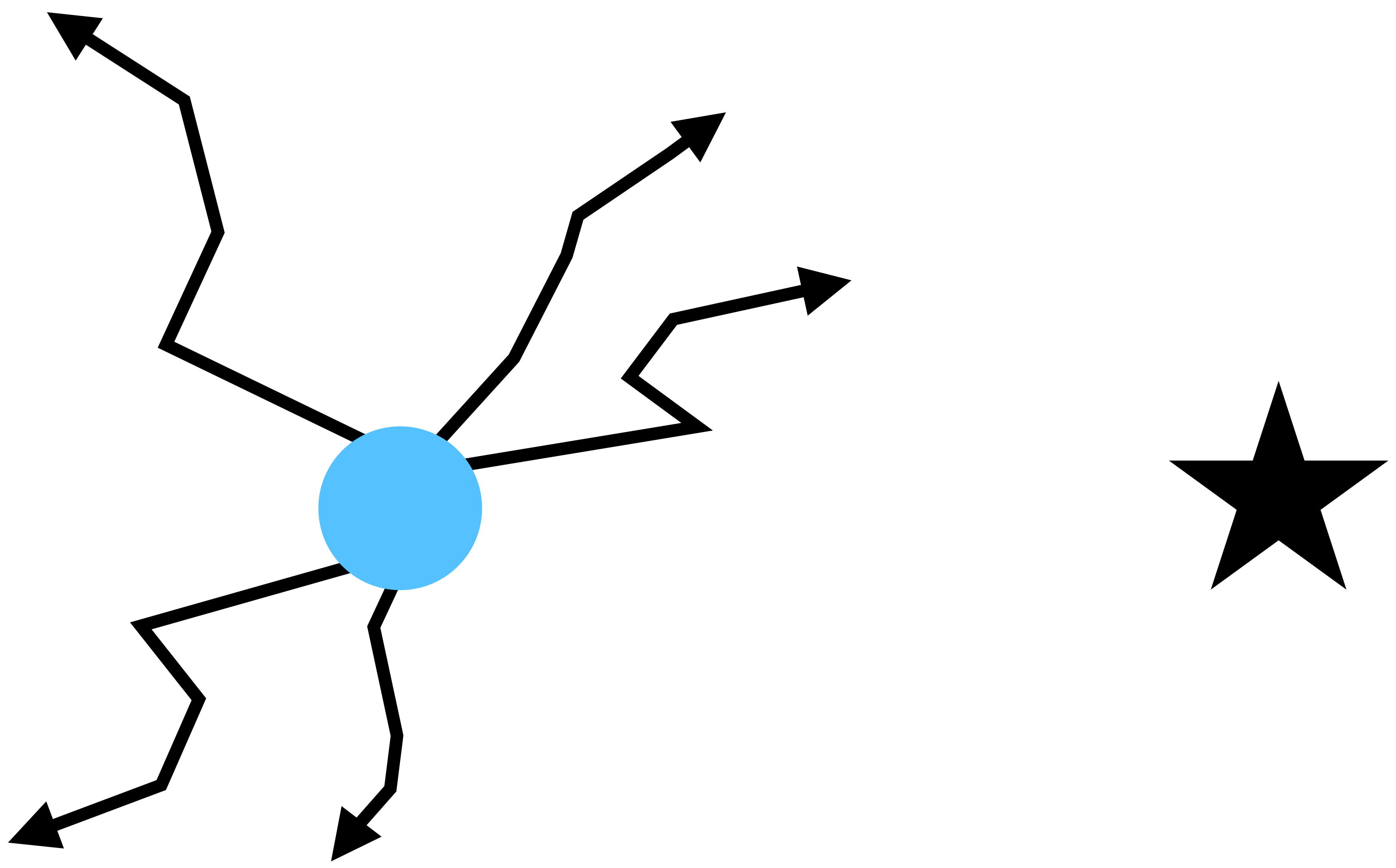
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

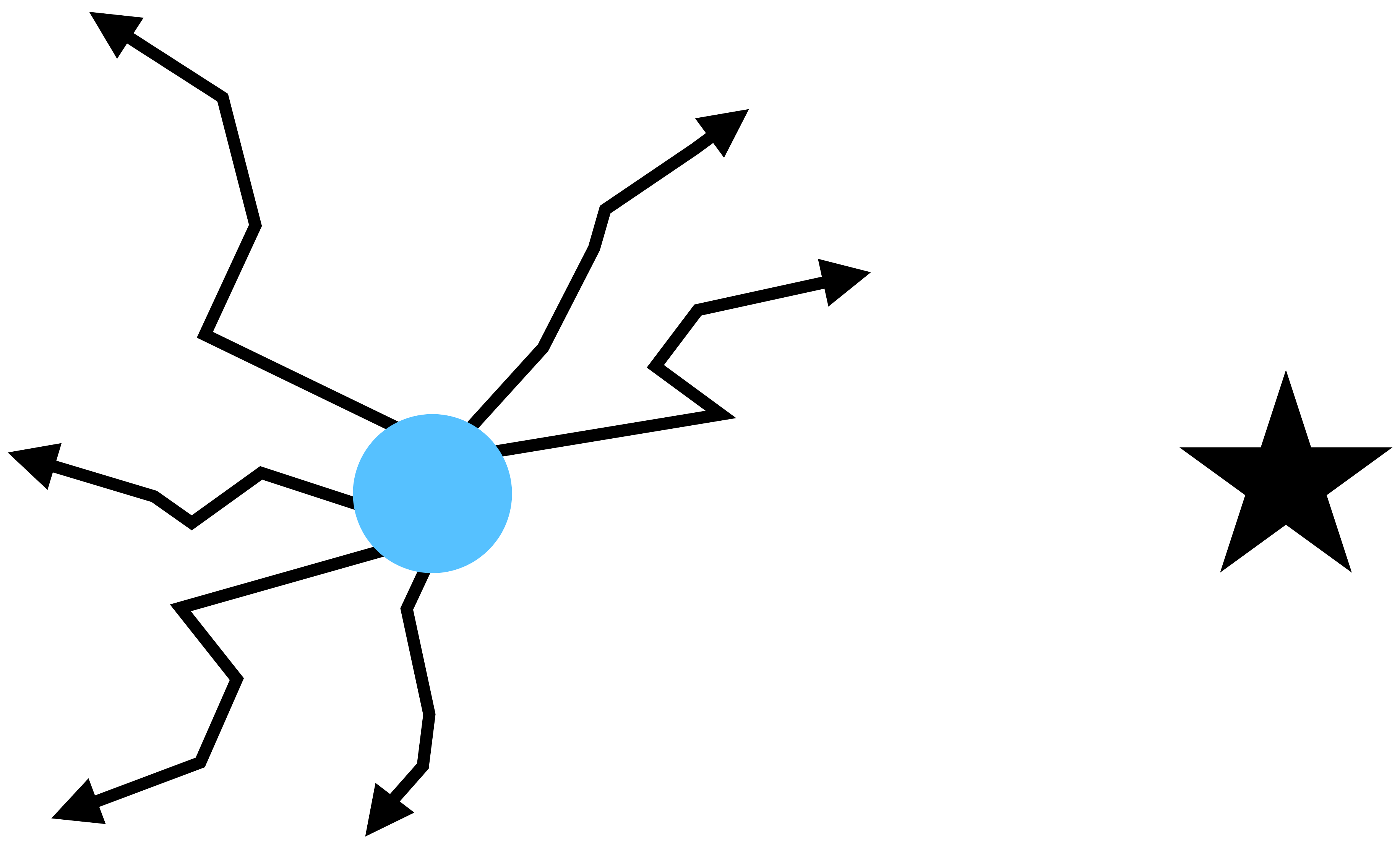
For each iteration

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

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

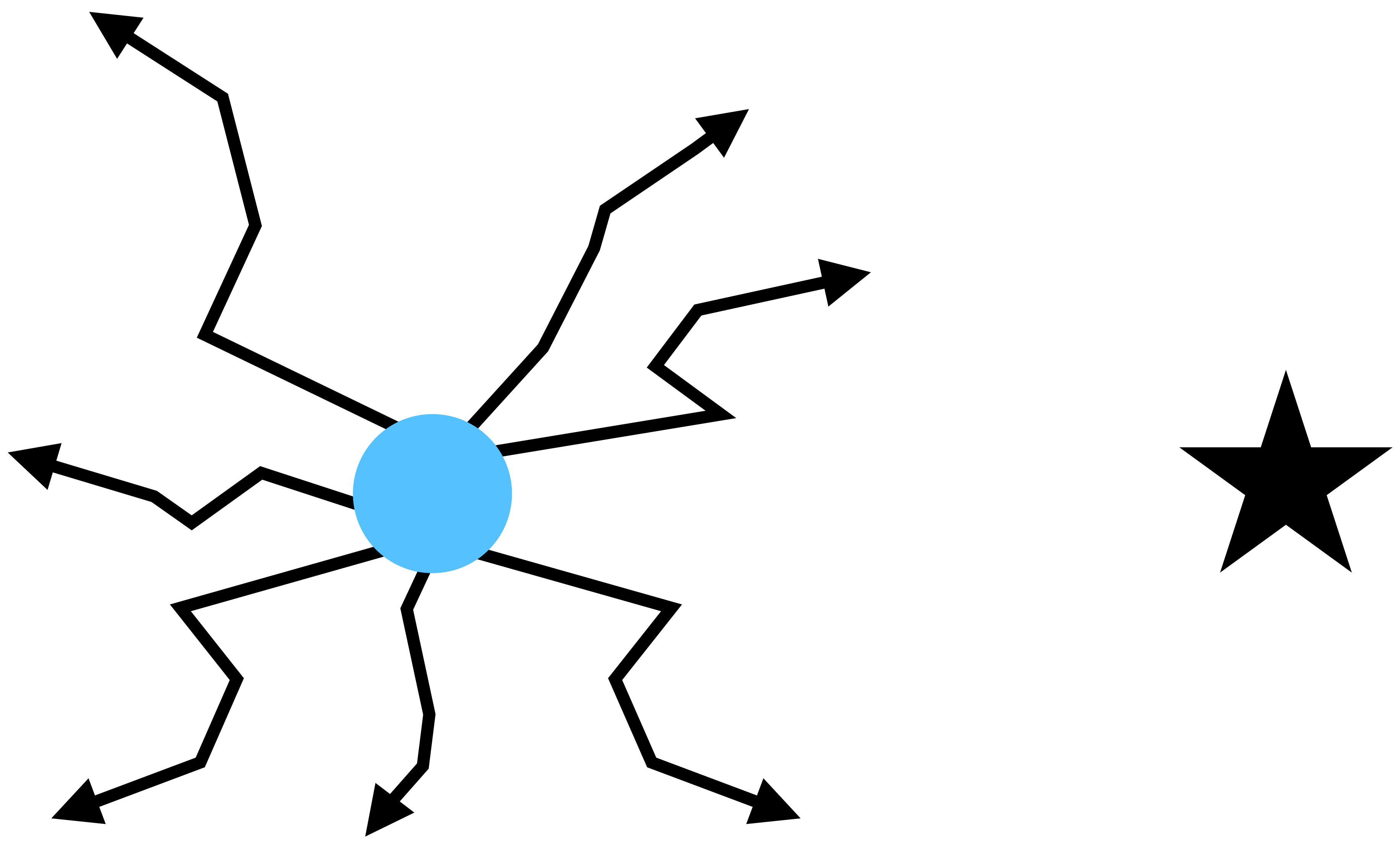
FCAI

fcai.fi

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

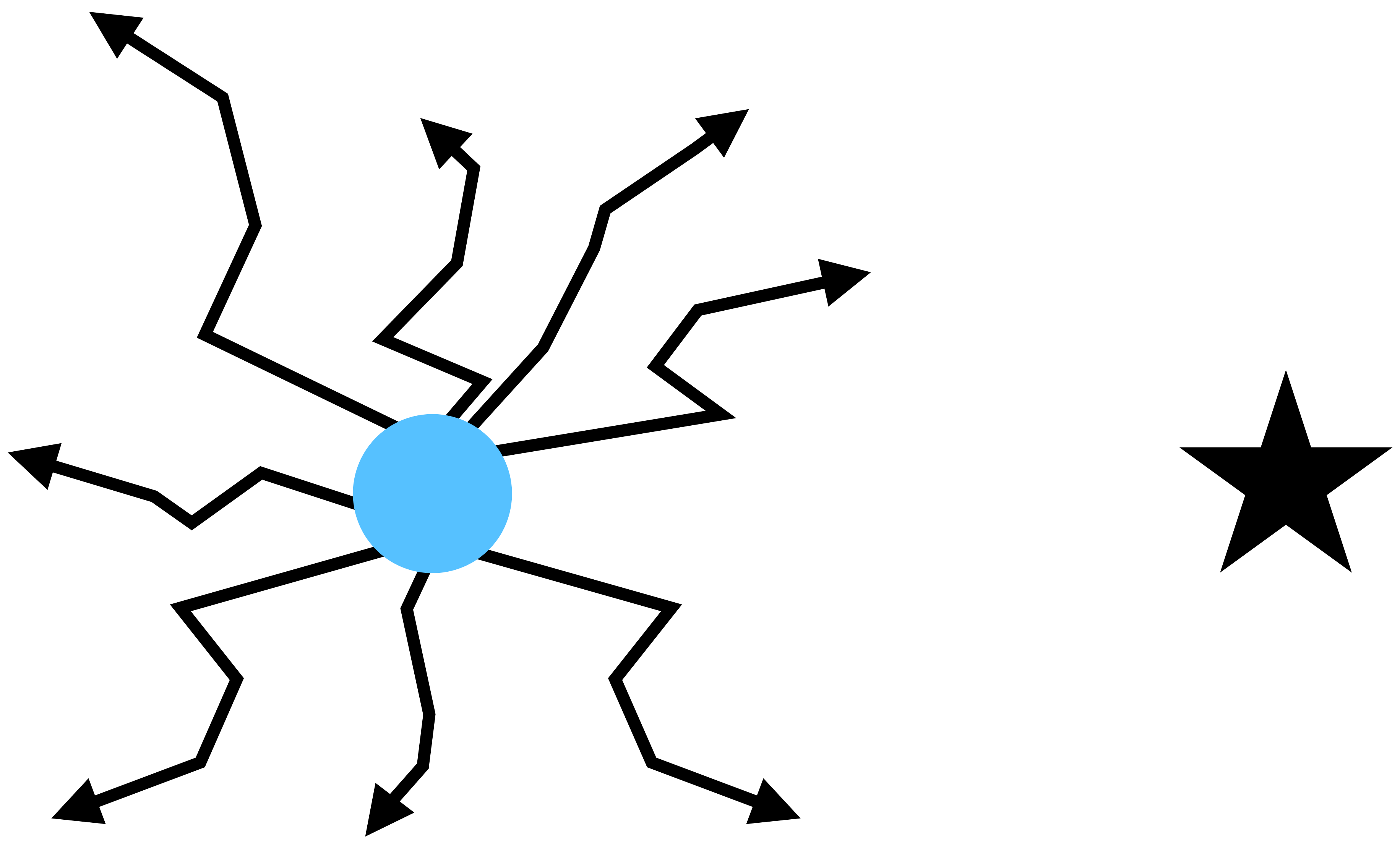
FCAI

fcai.fi

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

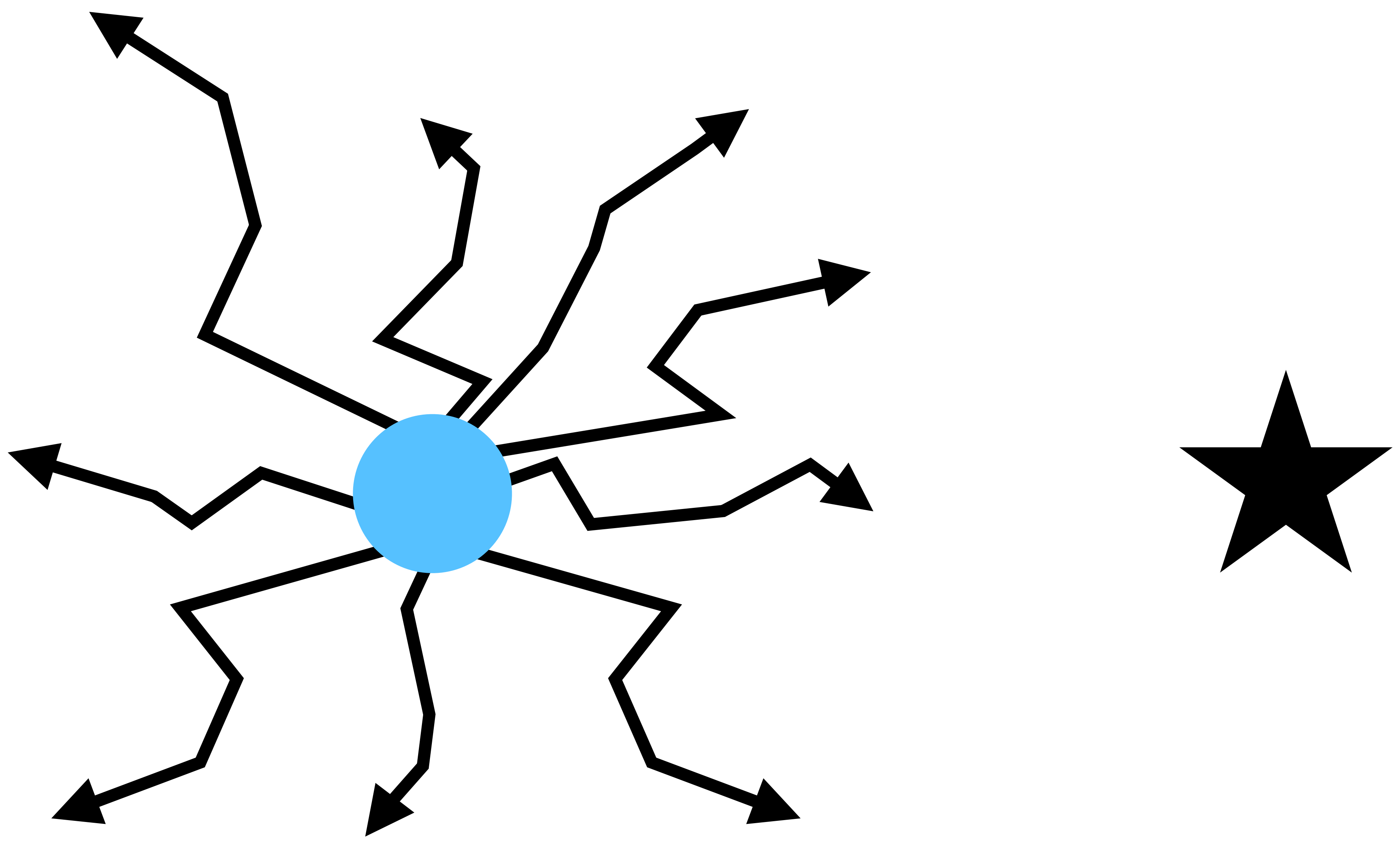
FCAI

fcai.fi

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

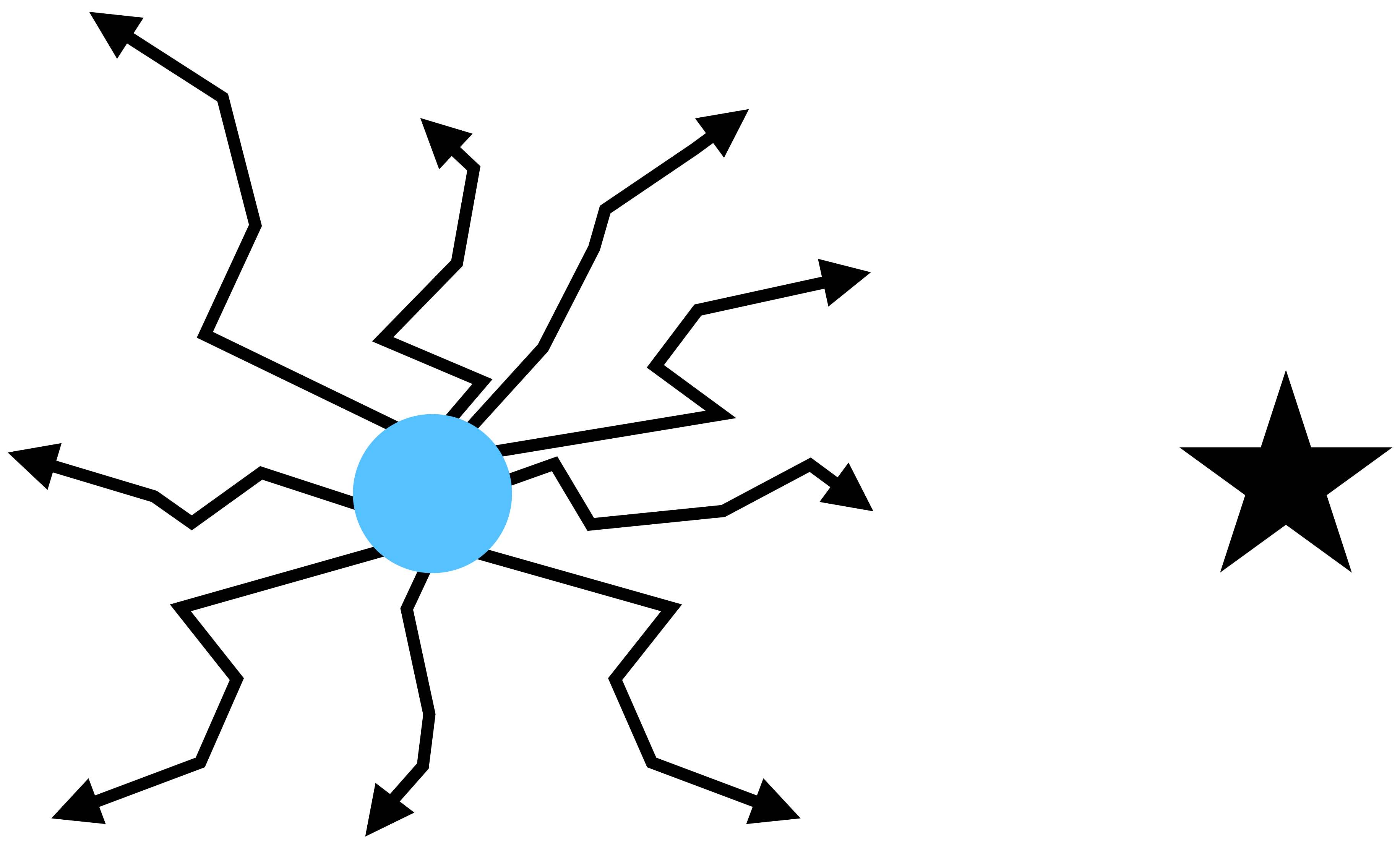
FCAI

fcai.fi

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

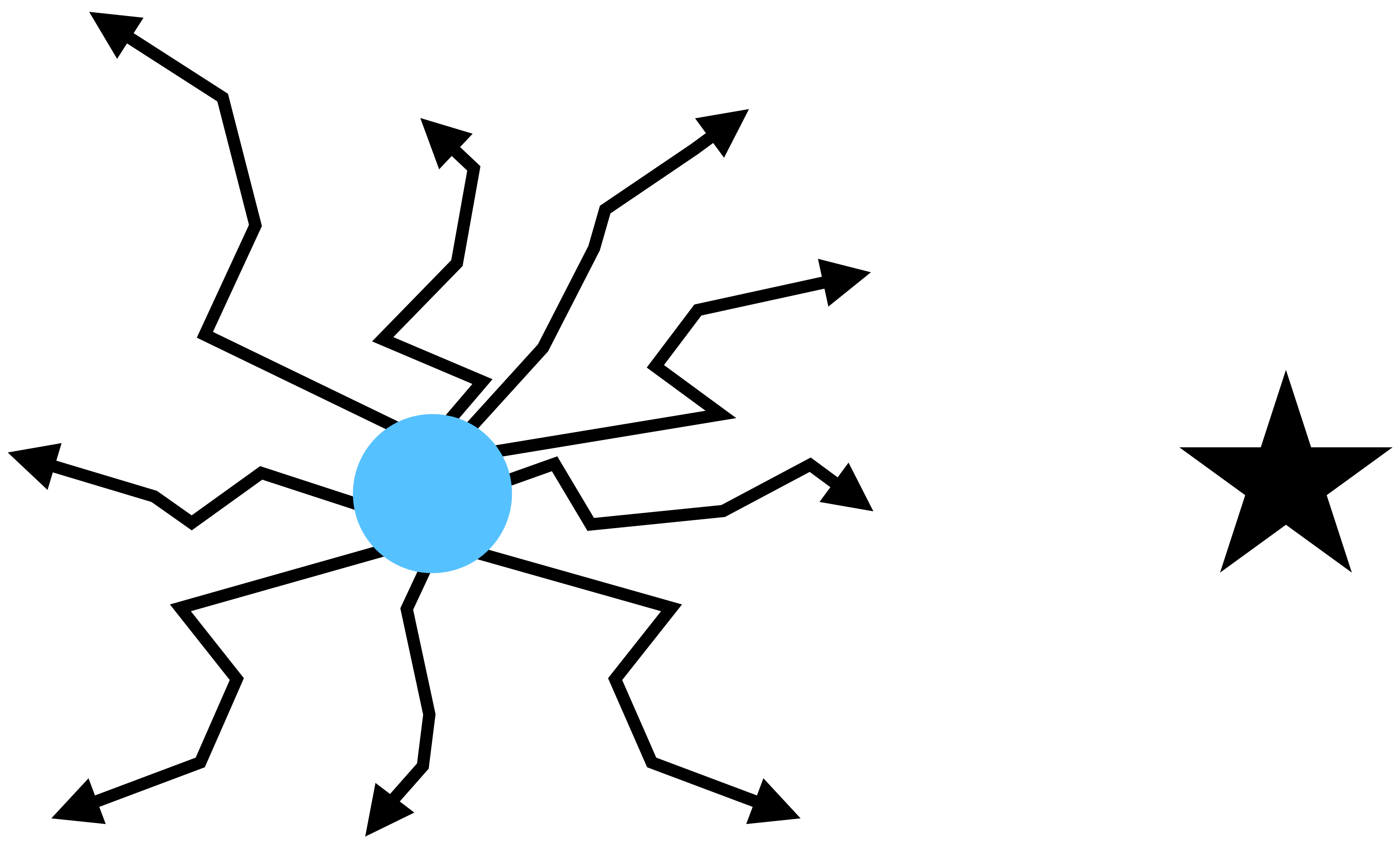
FCAI

fcai.fi

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

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

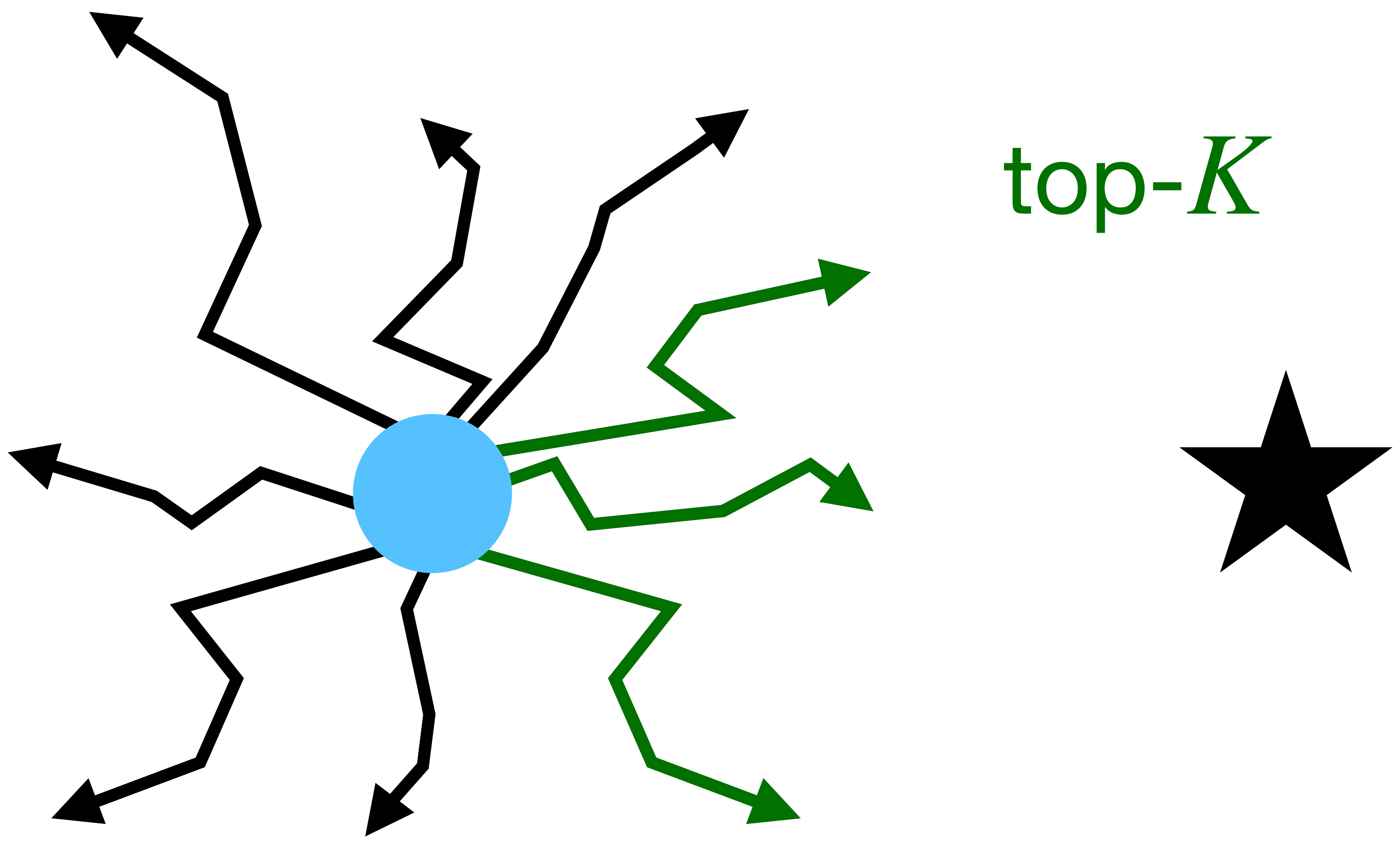
Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

Select top K performing samples

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 1



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

For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2

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

For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2

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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^{i}, \mathbf{s})$ for each sample

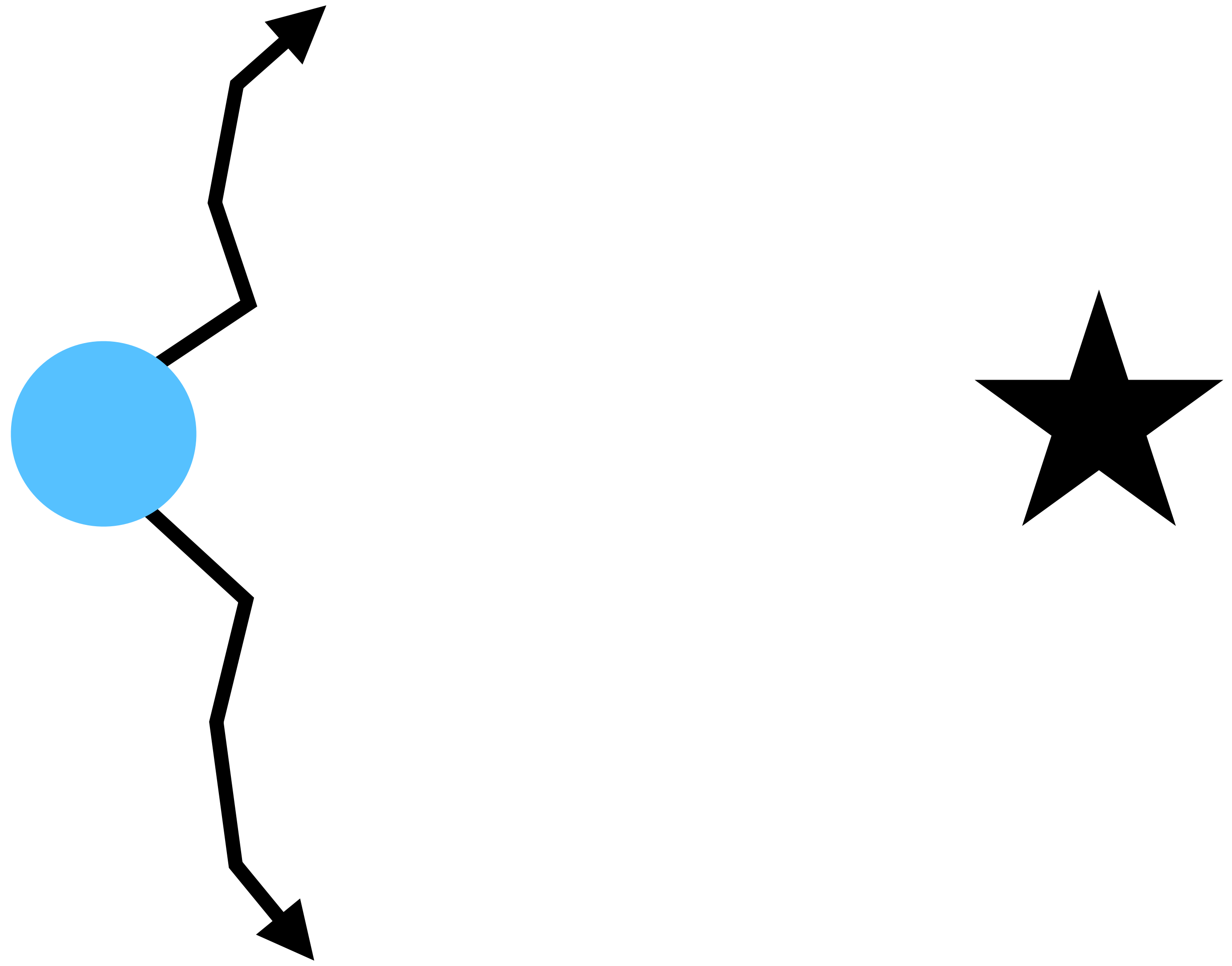
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2



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

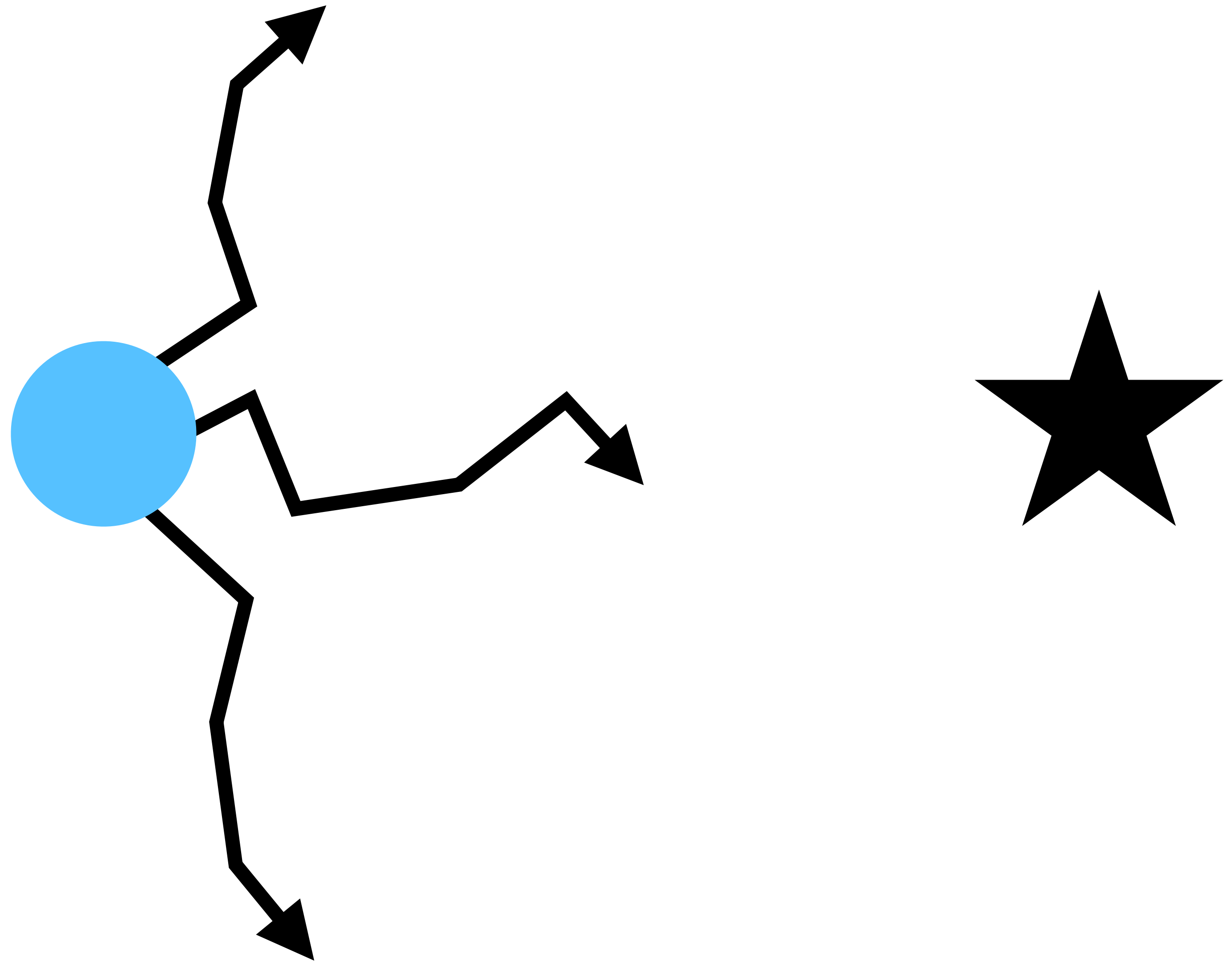
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

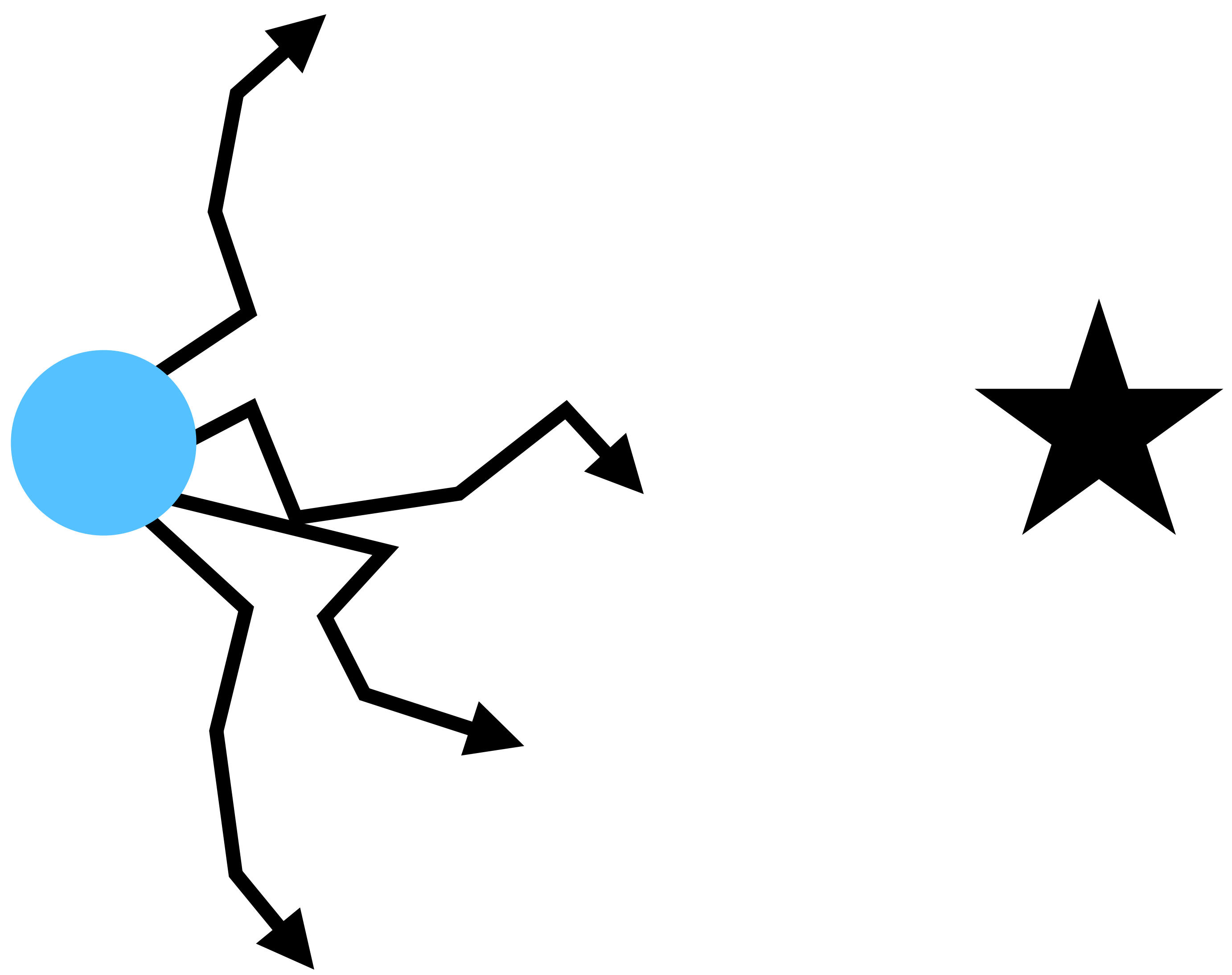
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

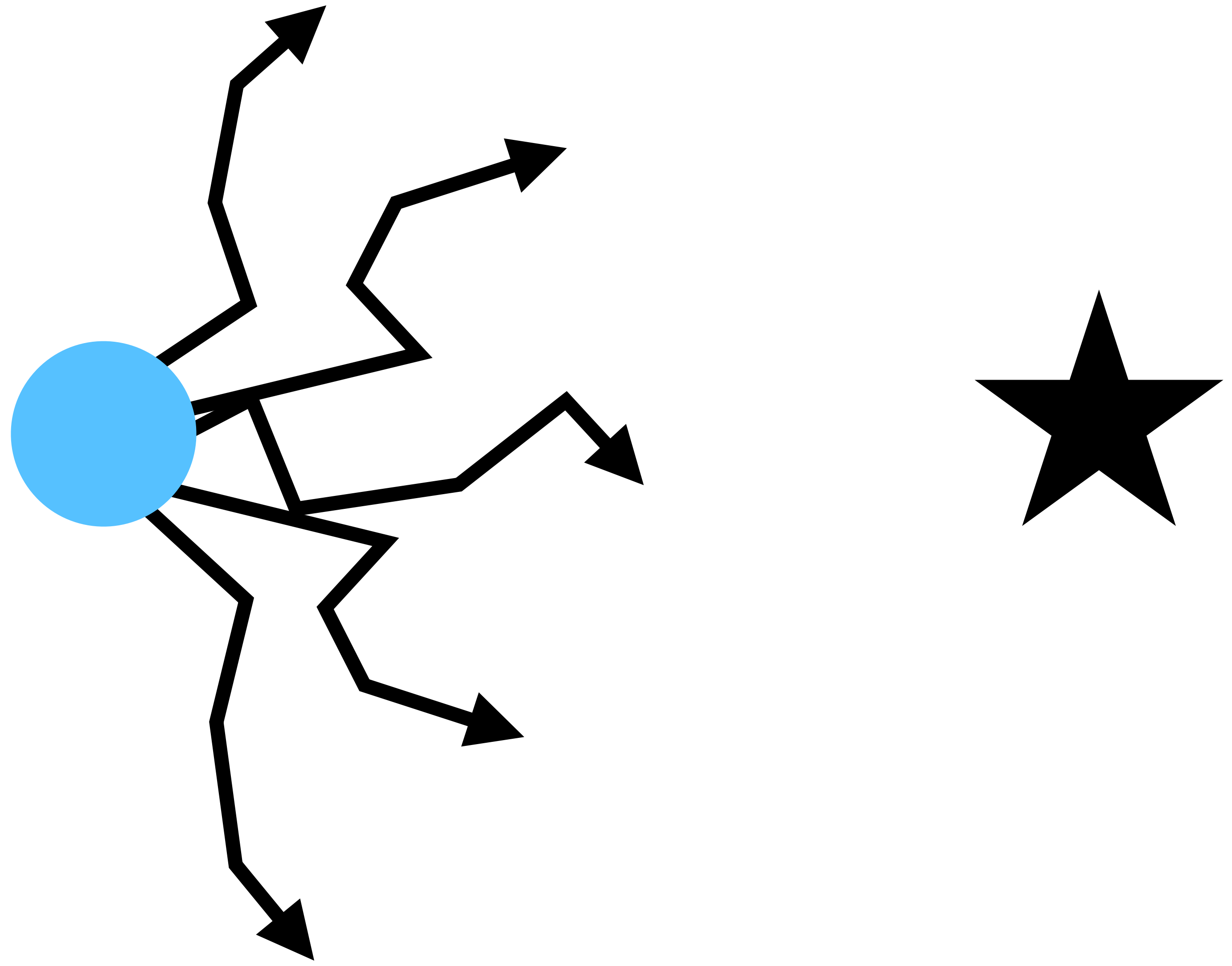
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

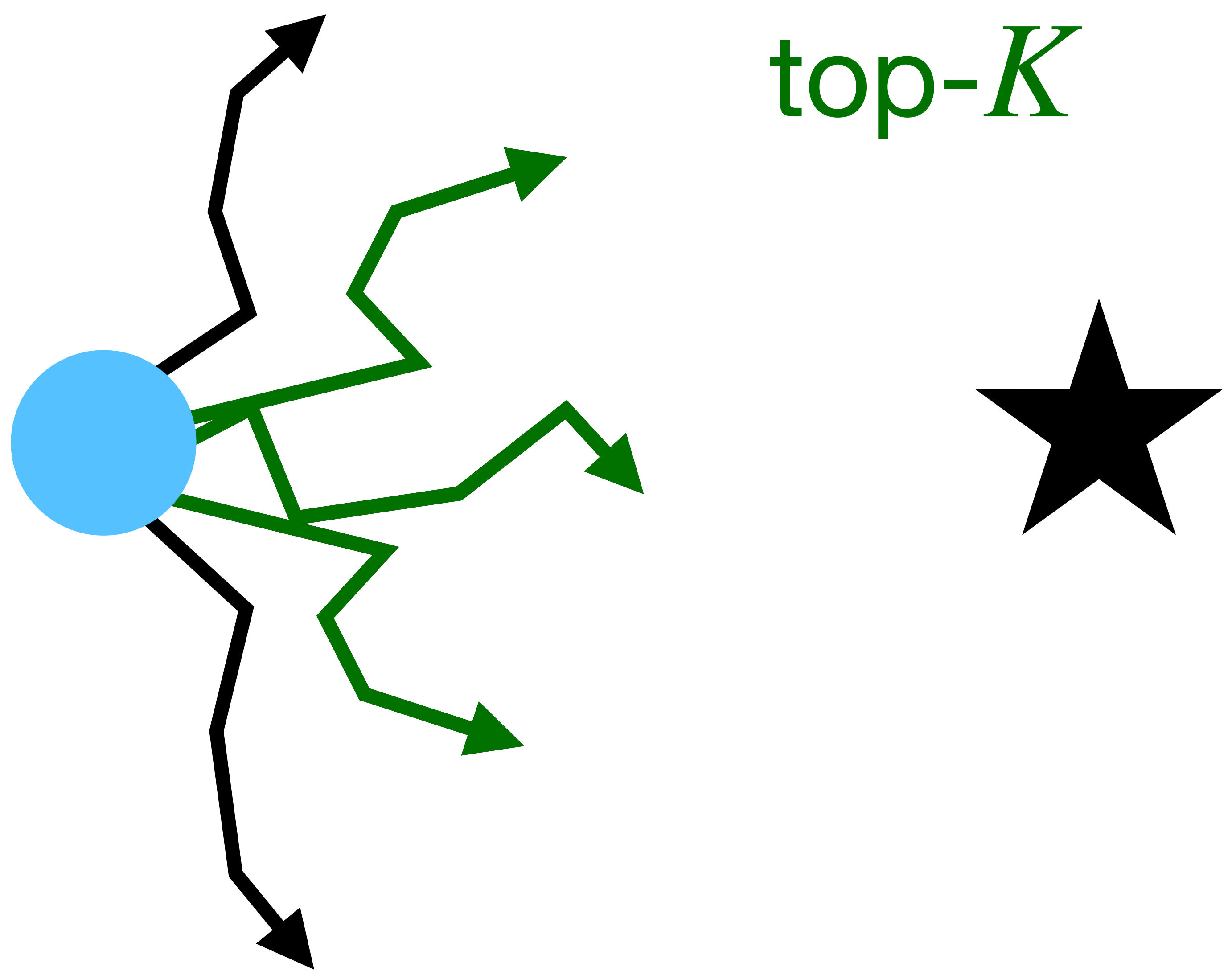
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 2



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3

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

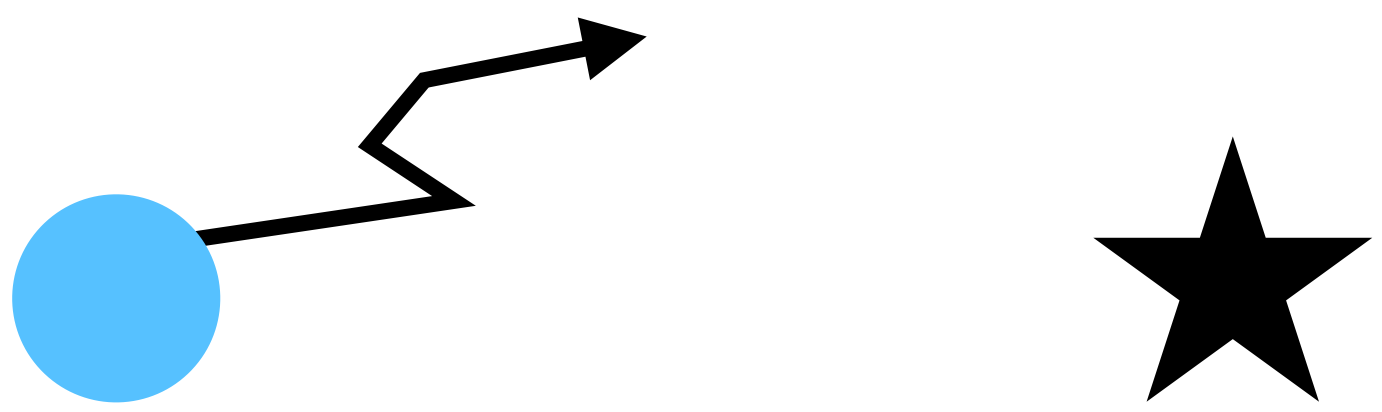
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^{i}, \mathbf{s})$ for each sample

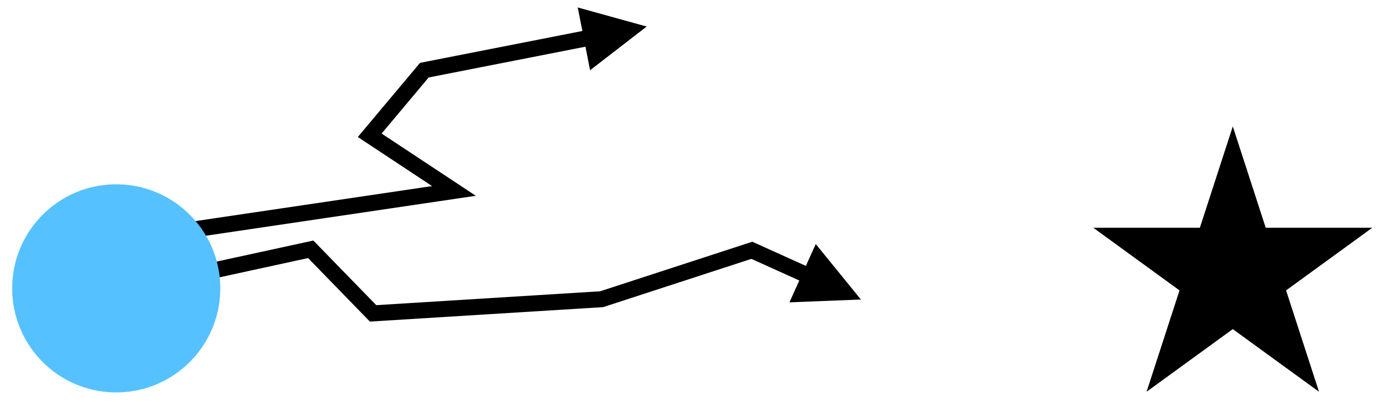
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

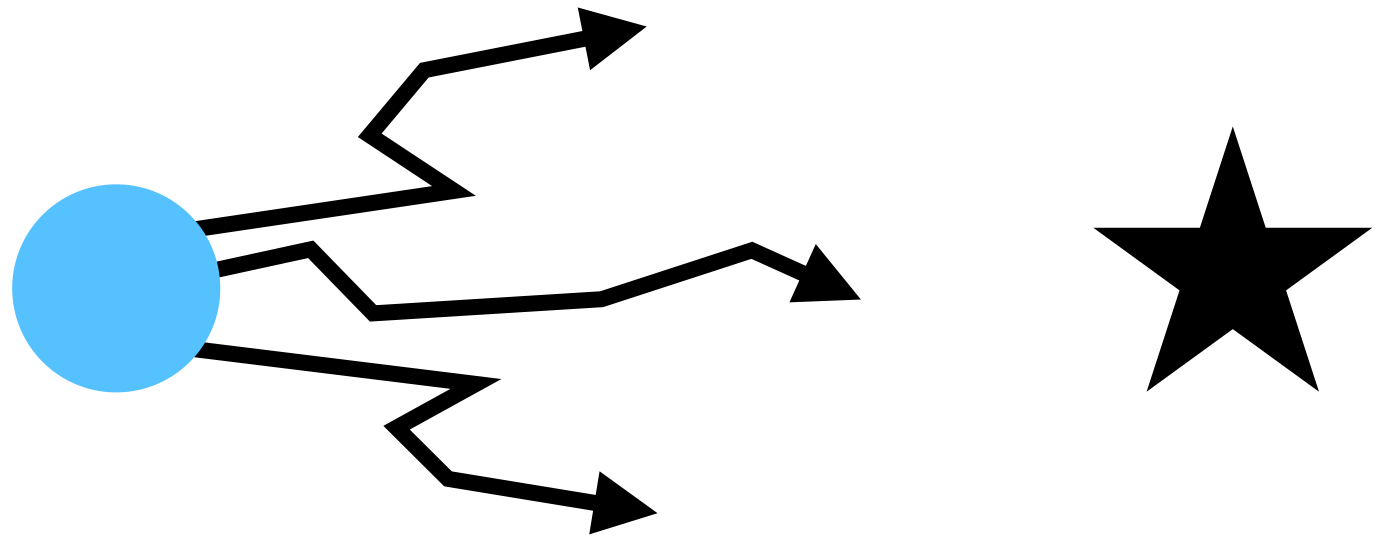
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

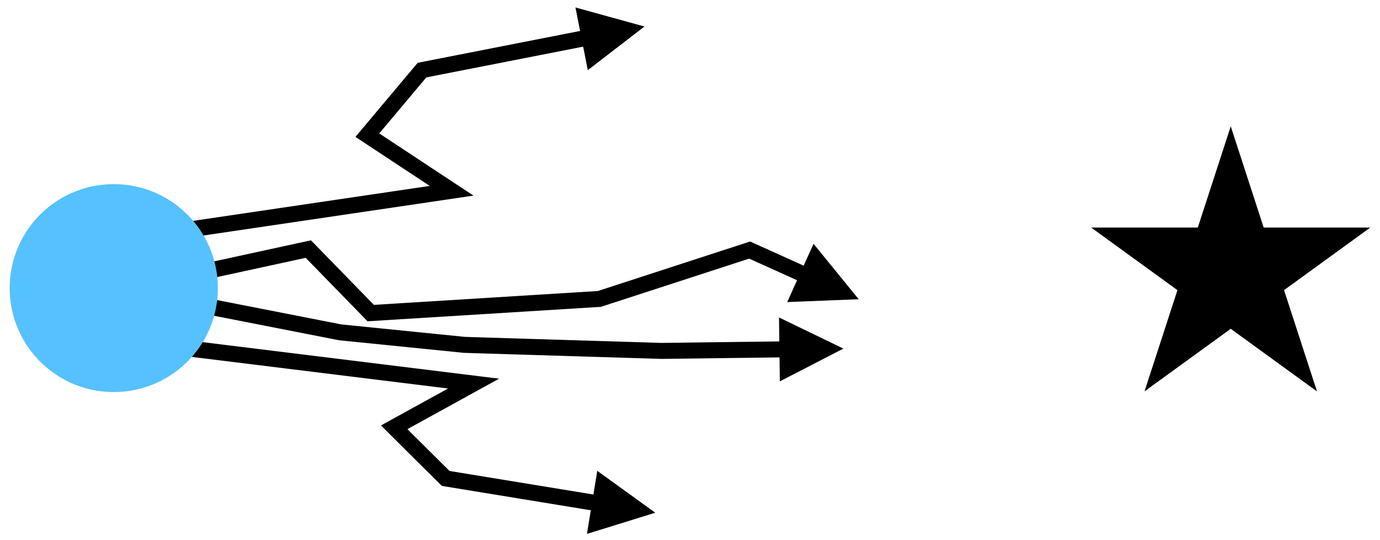
For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^N$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^H$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

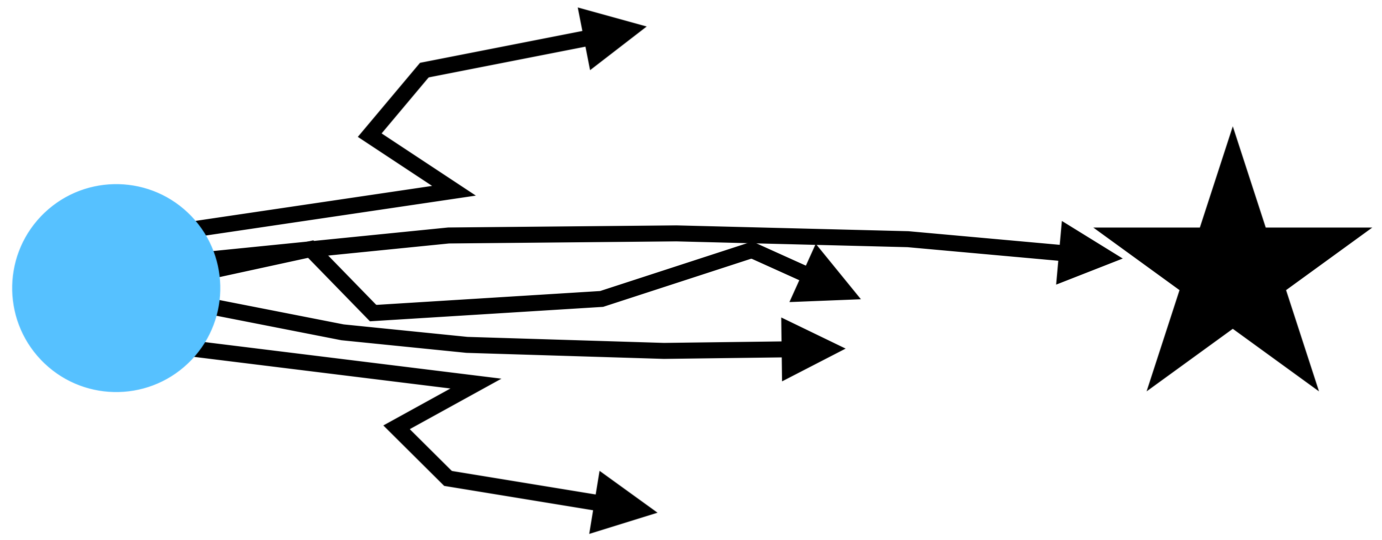
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

For each iteration

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

Evaluate objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample

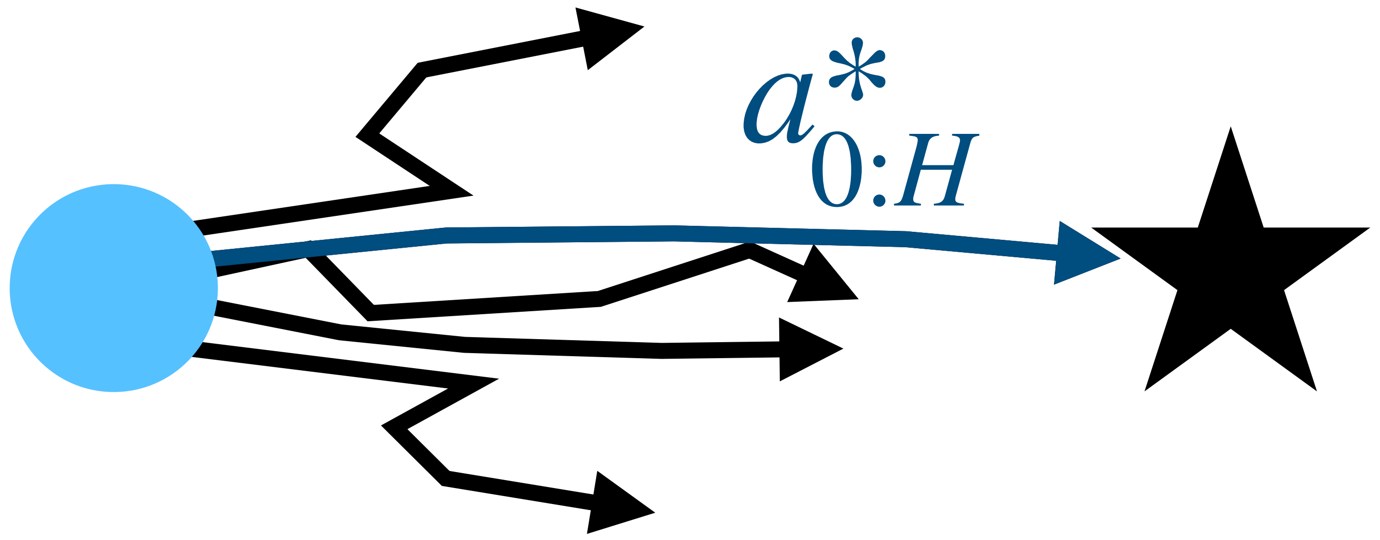
Select top K performing samples

Update action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

DCWM: Decision-time Planning

Model Predictive Path Integral Control (MPPI)

Iteration 3



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

For each iteration

- **Sample** N action sequences $\{a_{0:H}^i\}_{i=1}^{N}$
- **Evaluate** objective $J(\mathbf{a}_{0:H}^i, \mathbf{s})$ for each sample
- **Select** top K performing samples
- **Update** action distribution parameters $\{\mu_t, \sigma_t^2\}_{t=0}^{H}$

FCAI

fcai.fi