Model-based Reinforcement Learning (Part 2)

Sanjiban Choudhury

Cornell Bowers CIS Computer Science

Overall Course Plan

✅ Foundations (up until last class)

☐ Advanced Algorithms and Applications (till end of course)

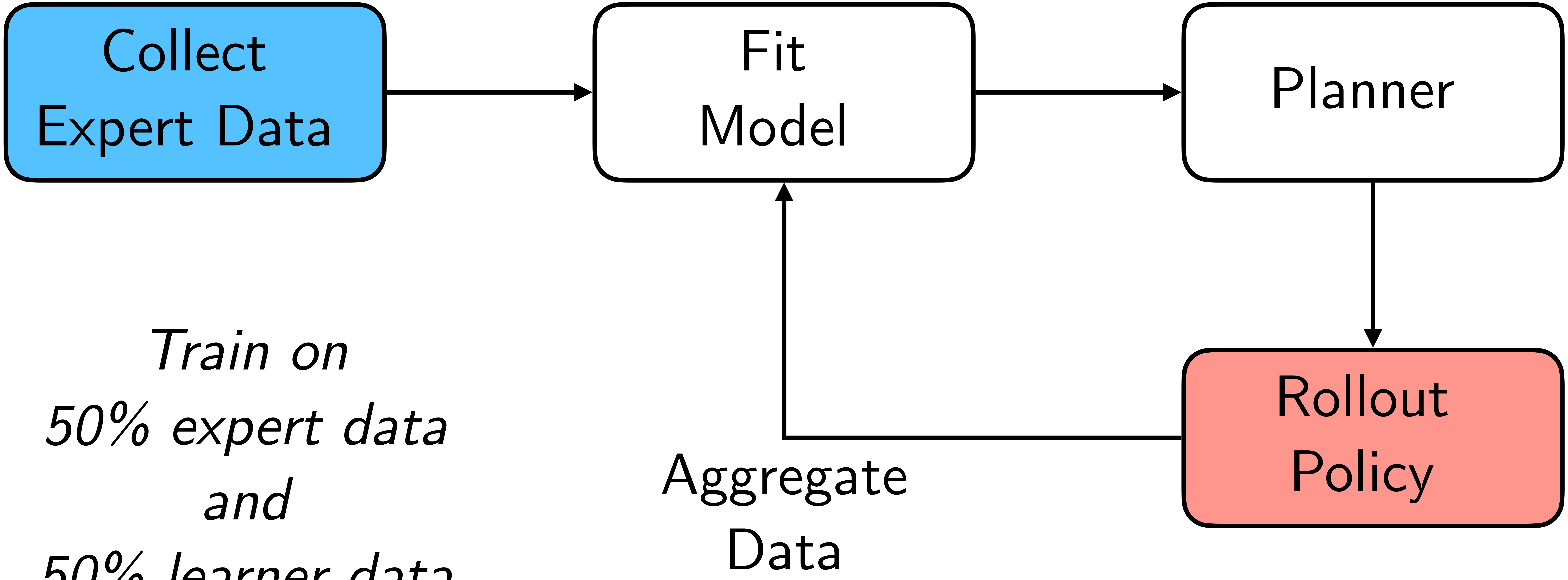
Topics: Generative world models, Offline RL, Visual Representations, RLHF, Human motion forecasting, ...
Lecturers: Sanjiban, Tapo, Killian Weinberger, Kuan Fang, Tapo, Lerrel Pinto, Pulkit Agarwal

Today's class

- ☐ Deriving MBRL loss
- ☐ Practical MBRL
- ☐ The DREAMER algorithm

Model Learning with Planner in Loop

(Ross & Bagnell, 2012)



Train on
50% expert data
and
50% learner data

Model Learning with Planner in Loop

Collect data from an expert $\mathscr{D}_{\text{expert}} = \{(s, a, s')\}$

Fit a model $\hat{M}_1$. Compute a policy $\hat{\pi}_1$ in the model via planning

Initialize empty data buffer $\mathscr{D}_{\text{learner}} \leftarrow \{\}$

For $i = 1, \ldots, N$

Execute policy $\hat{\pi}_i$ in the real world and collect data

$$\mathscr{D}_i = \{(s, a, s')\}$$

Aggregate data $\mathscr{D}_{\text{learner}} \leftarrow \mathscr{D}_{\text{learner}} \cup \mathscr{D}_i$

Train a new model on 50% expert + 50% learner data

$$\hat{M}_{i+1} \leftarrow \text{Train}(0.5 * \mathscr{D}_{\text{expert}} + 0.5 * \mathscr{D}_{\text{learner}})$$

Train a new policy $\hat{\pi}_{i+1}$ in the model $\hat{M}_{i+1}$

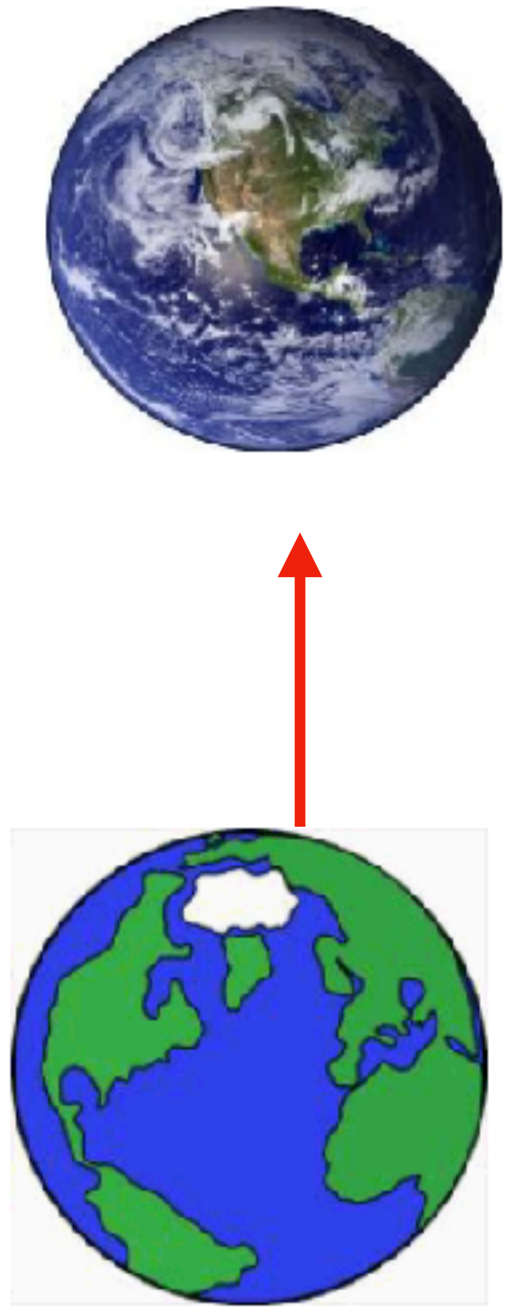
Select the best policy in $\hat{\pi}_{1:N+1}$

How do we derive this algorithm?

What is the goal of learning models?

Is it to perfectly approximate the world?

World M^*



Model $\hat{M}$

What is the goal of learning models?

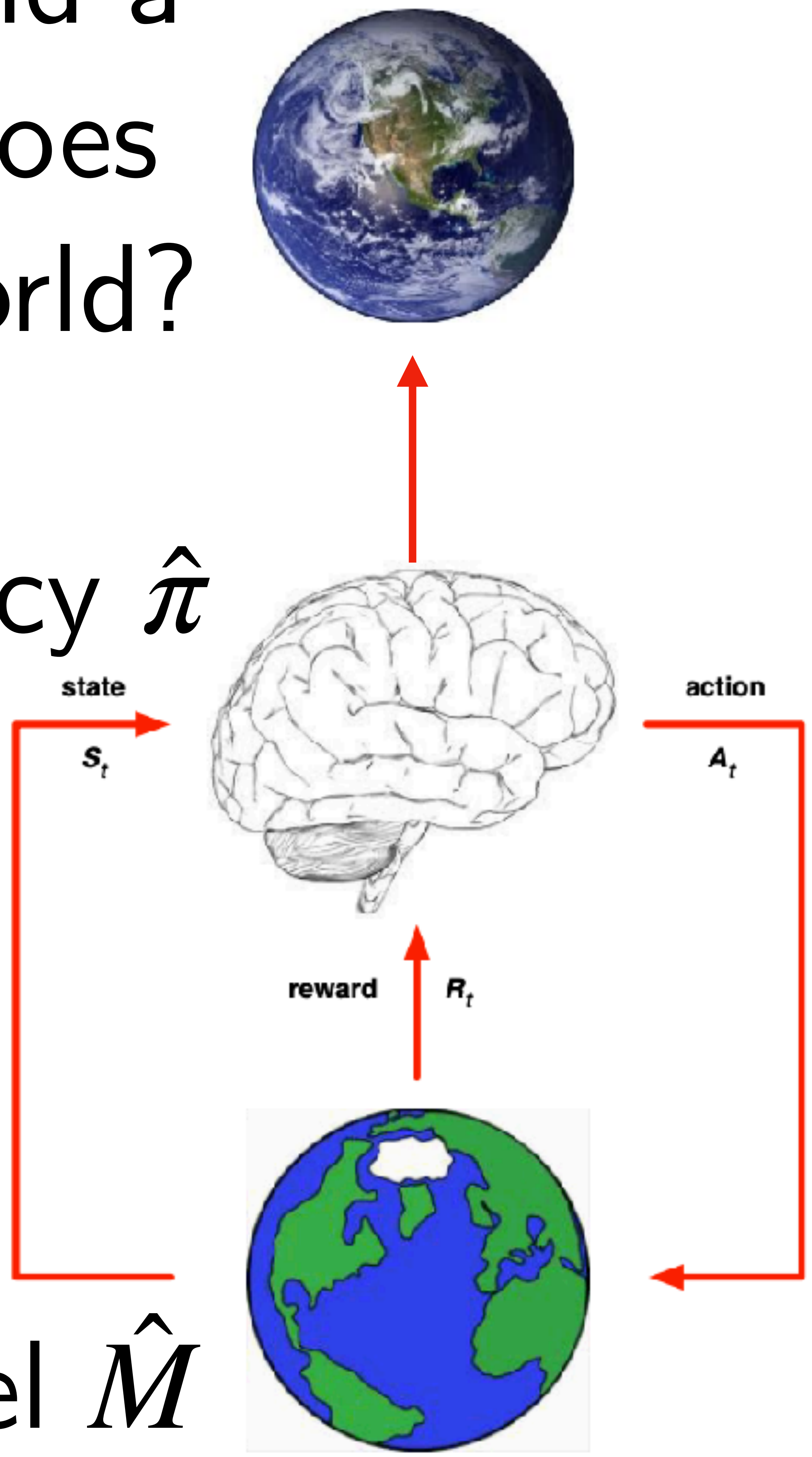
Is it to perfectly approximate the world?

World M^*

Model $\hat{M}$

Or … is to find a policy that does well in the world?

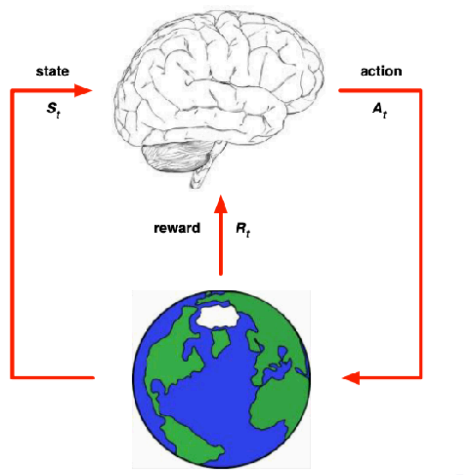
Policy $\hat{\pi}$



Model $\hat{M}$

Goal: Find model-based policy that bounds performance difference to the optimal policy in the real world

Optimal Policy



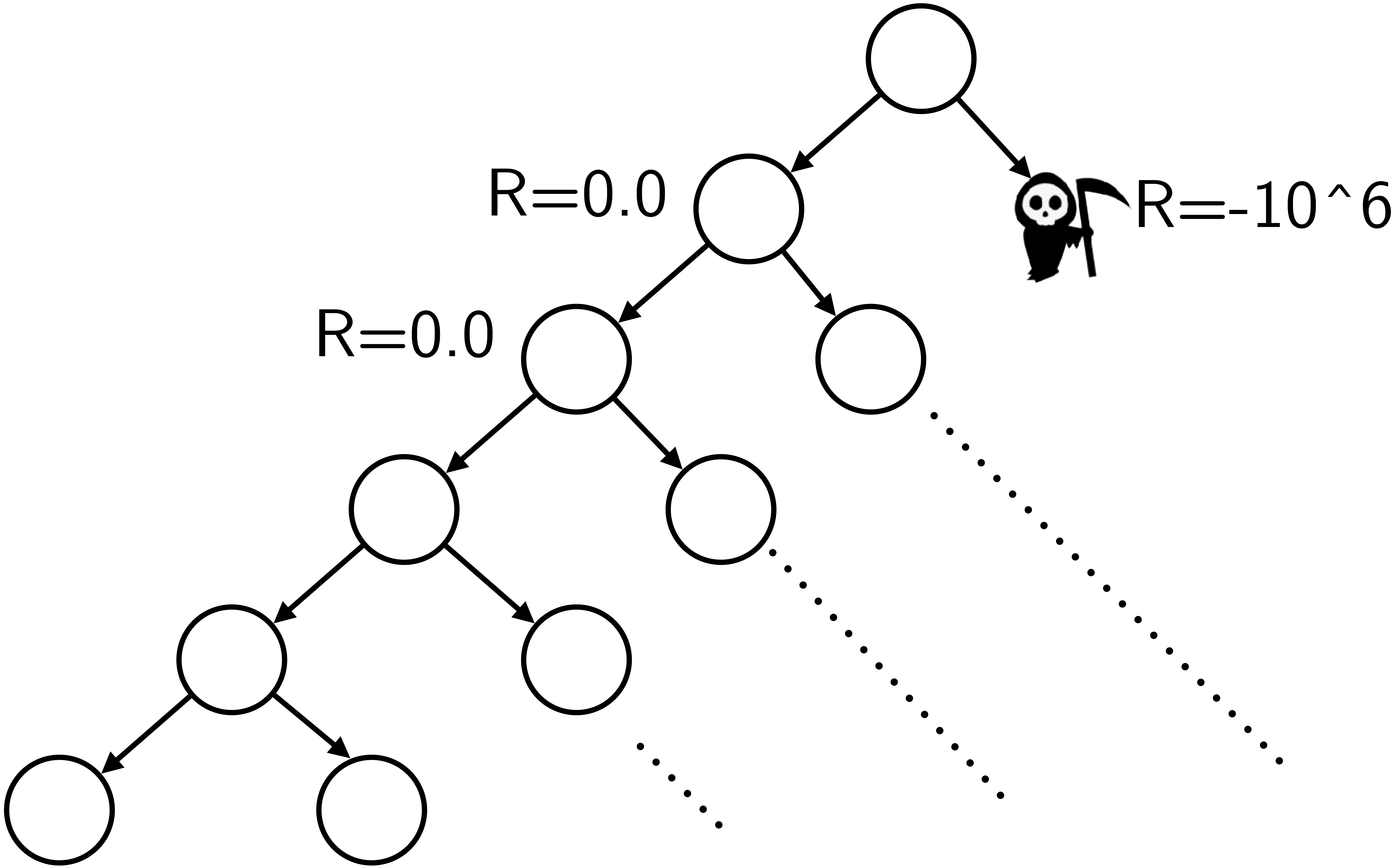
Model-based policy

$$V^{\pi^\star}_{M^*}(s_0) - V^{\hat{\pi}}_{M^*}(s_0)$$

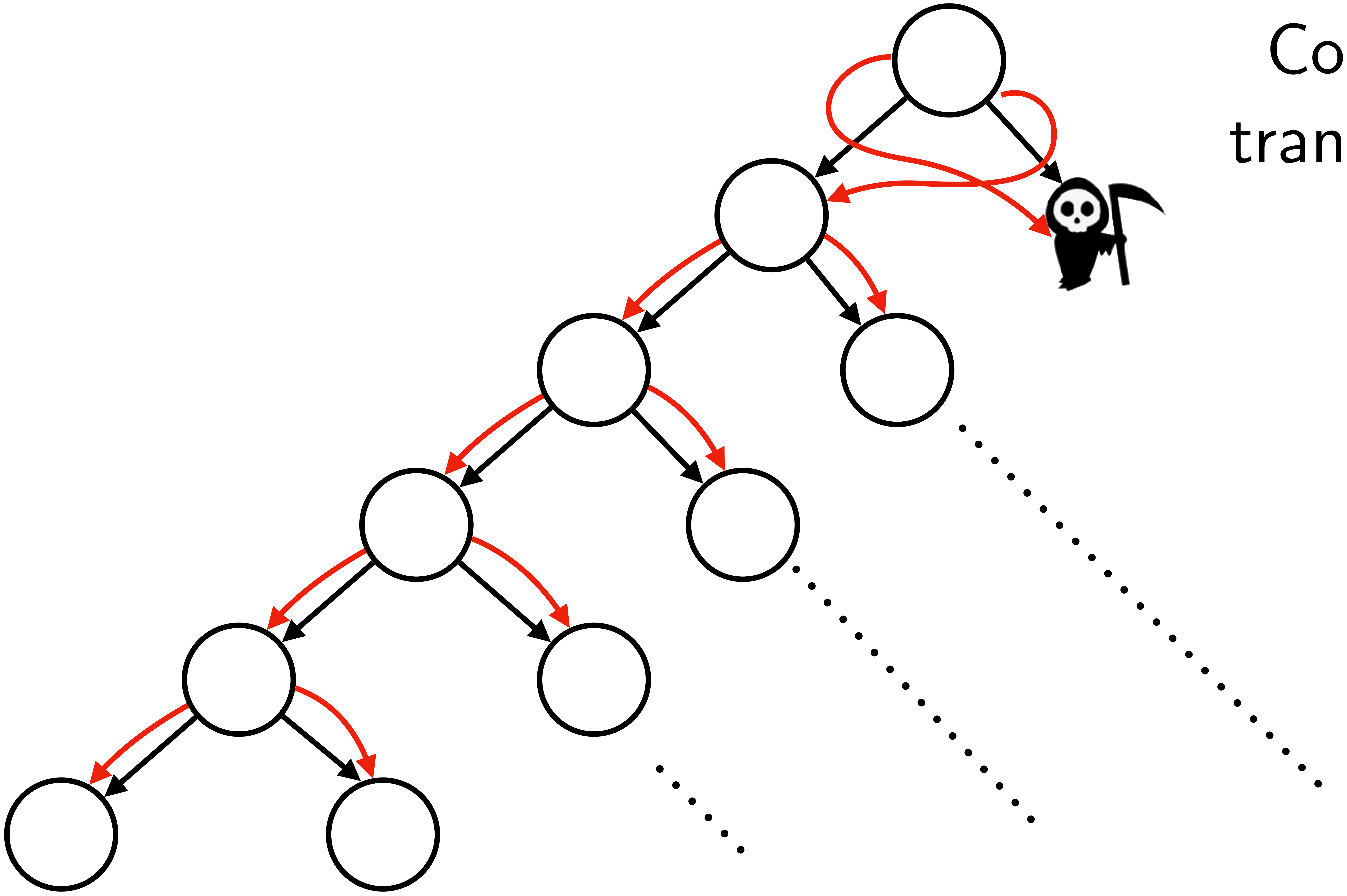
Performance Difference via Planning in Model Lemma

ft. Simulation Lemma

Let's say the following is the true MDP

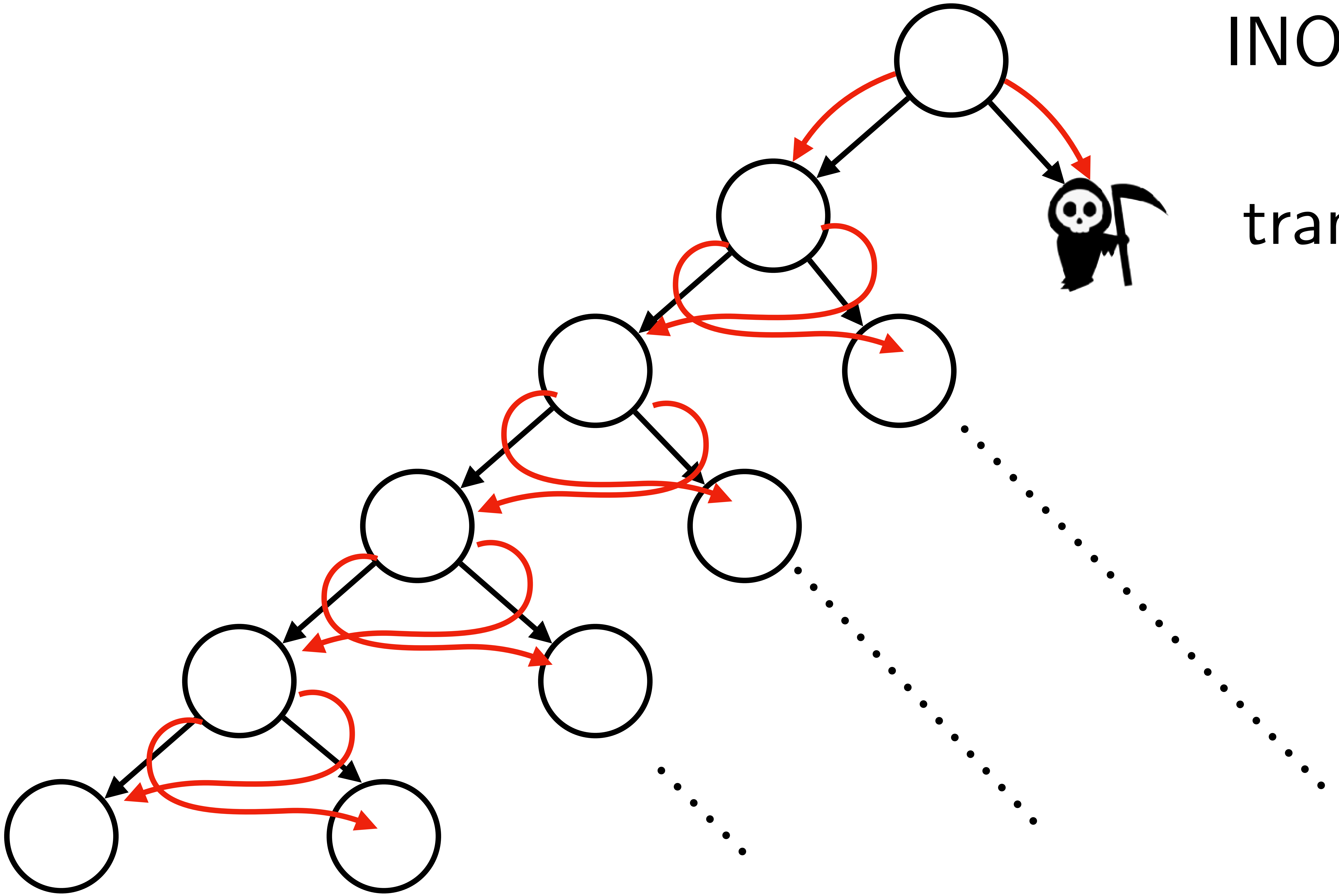


Candidate Model A



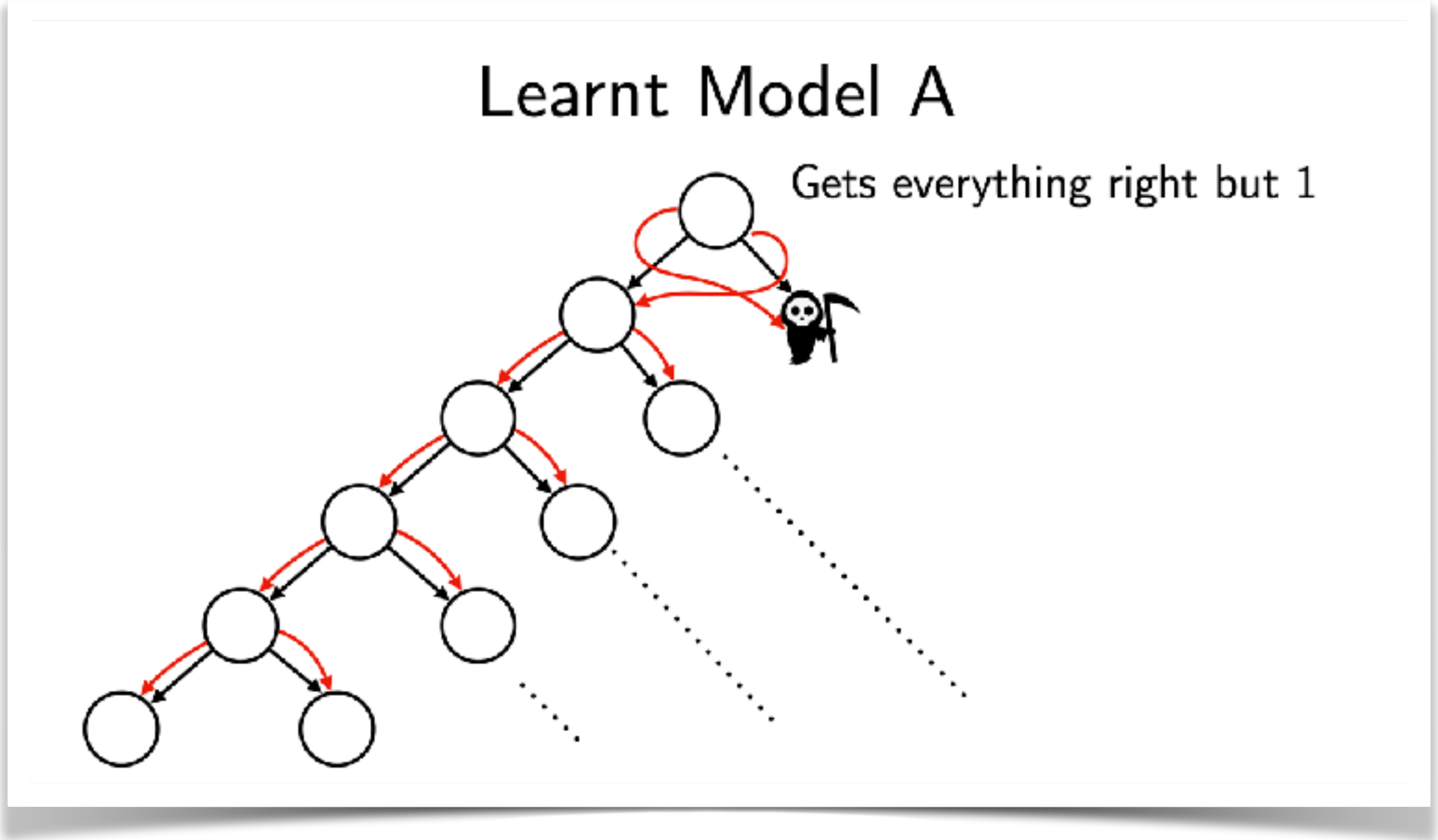
Correctly predicts all transitions but the first

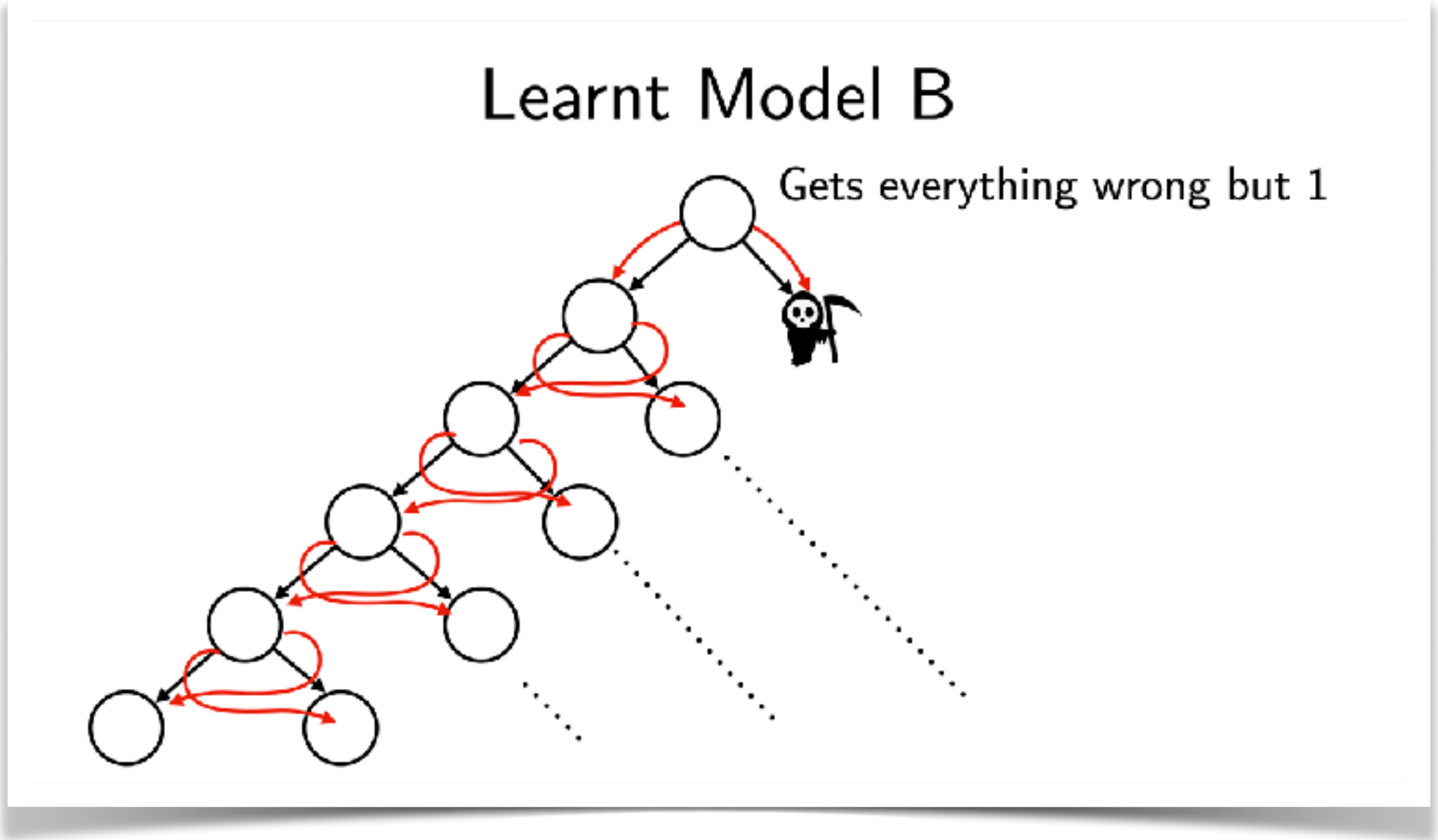
Candidate Model B



INOCRRECTLY predicts all transitions but gets the first right

Which model is better? What does MBRL learn?





When poll is active respond at **PollEv.com/sc2582**

Send **sc2582** to **22333**

Today's class

- ☑ Deriving MBRL loss

  (Sim. lemma, PD via PM lemma)

- ☐ Practical MBRL

- ☐ The DREAMER algorithm

The story so far ...

Robots have to act in the world

Hence, we learned various algorithms for decision making

But we assumed that we can observe the "state"

The story so far ...

But in the real world, no one tells you the "state"

All you see are observations

How do we learn from observations?

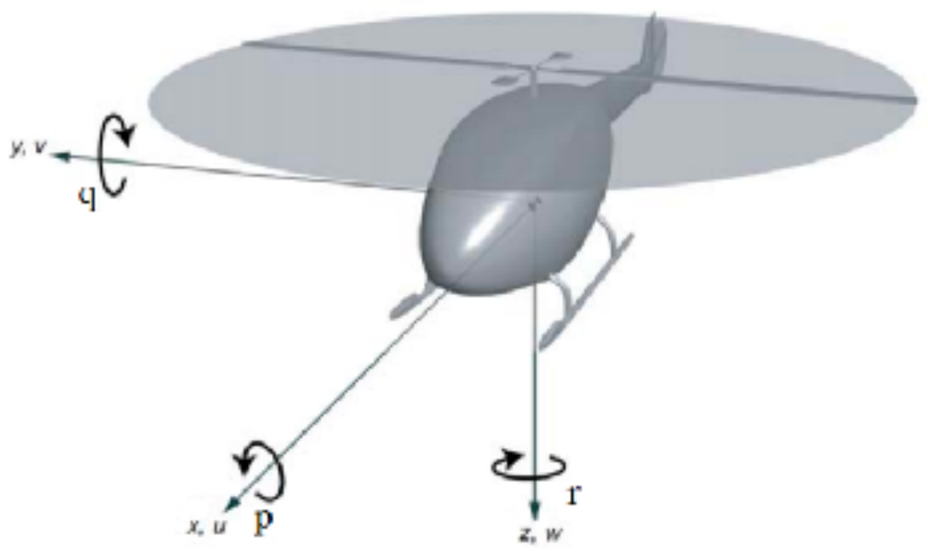
Models.

Models: From Simple to Complex

Simple

Complex

Models: From Simple to Complex



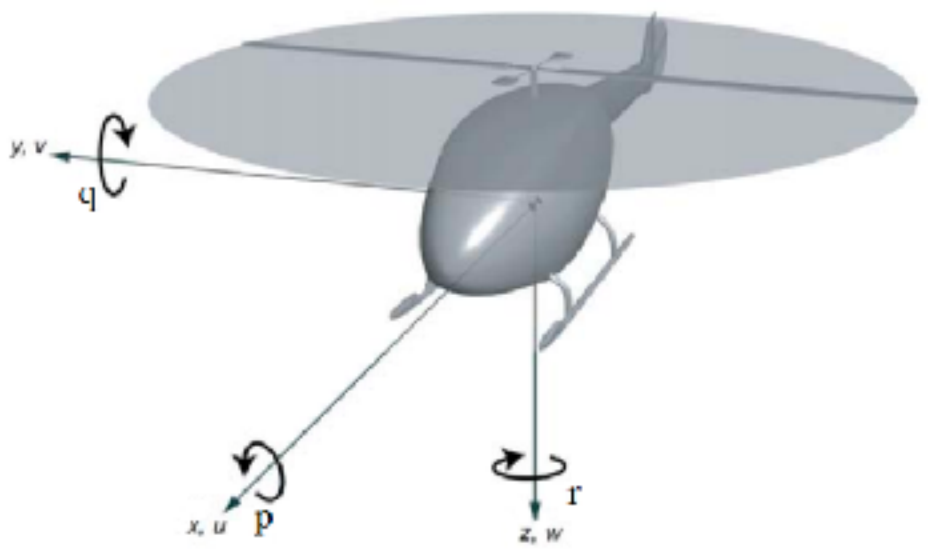
Physics Models

Simple

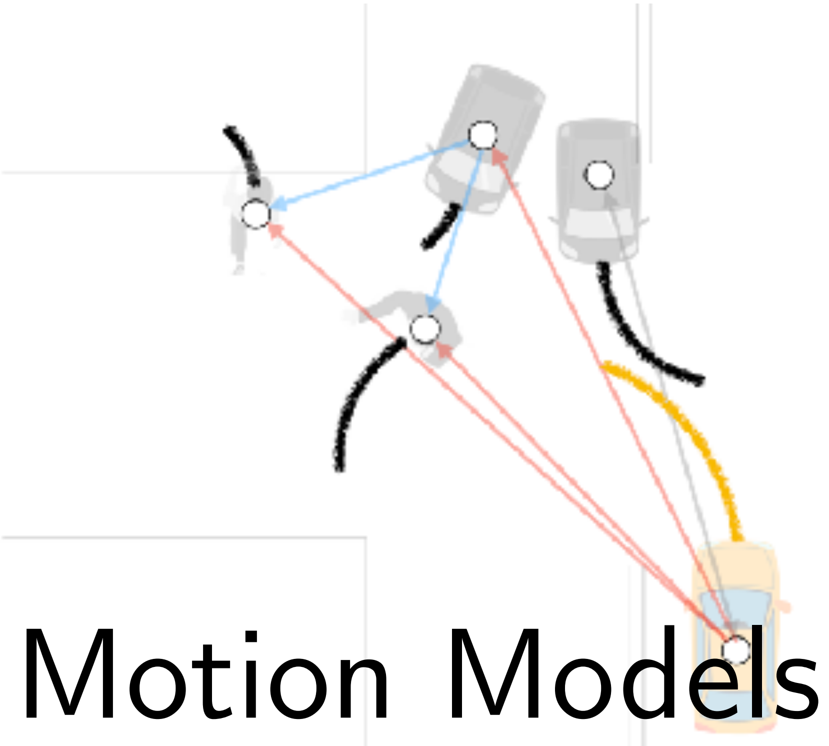
Known state

Strong prior
on dynamics

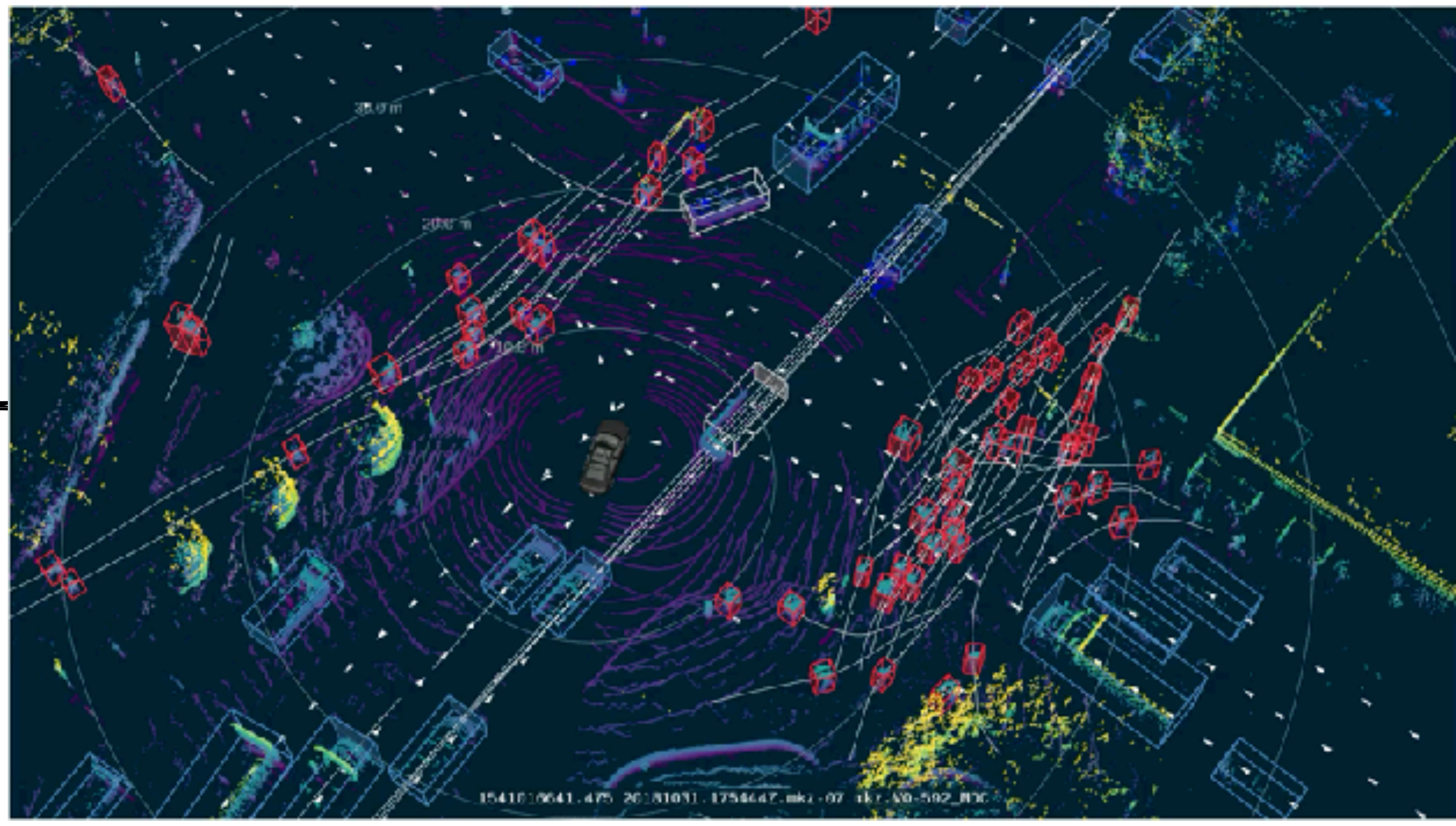
Models: From Simple to Complex



Physics Models



Motion Models



Simple

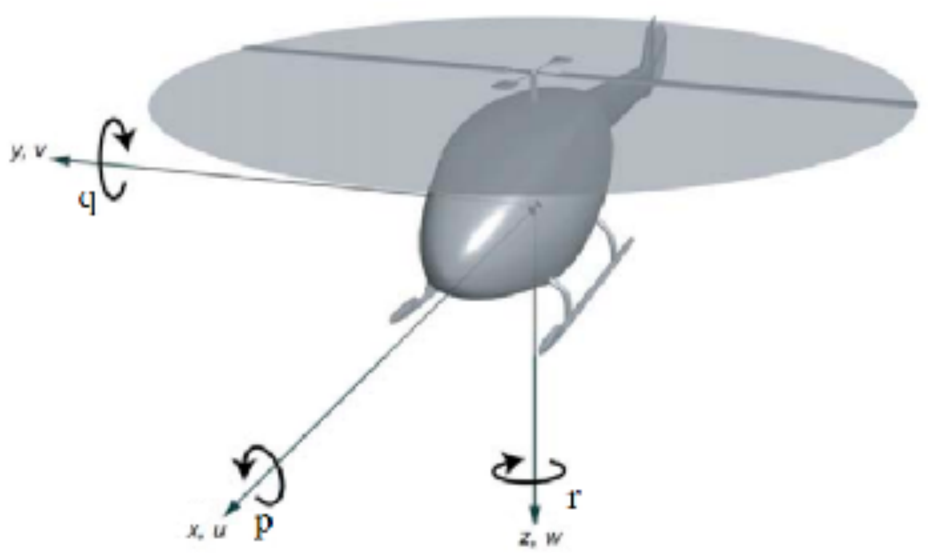
Known state

Strong prior
on dynamics

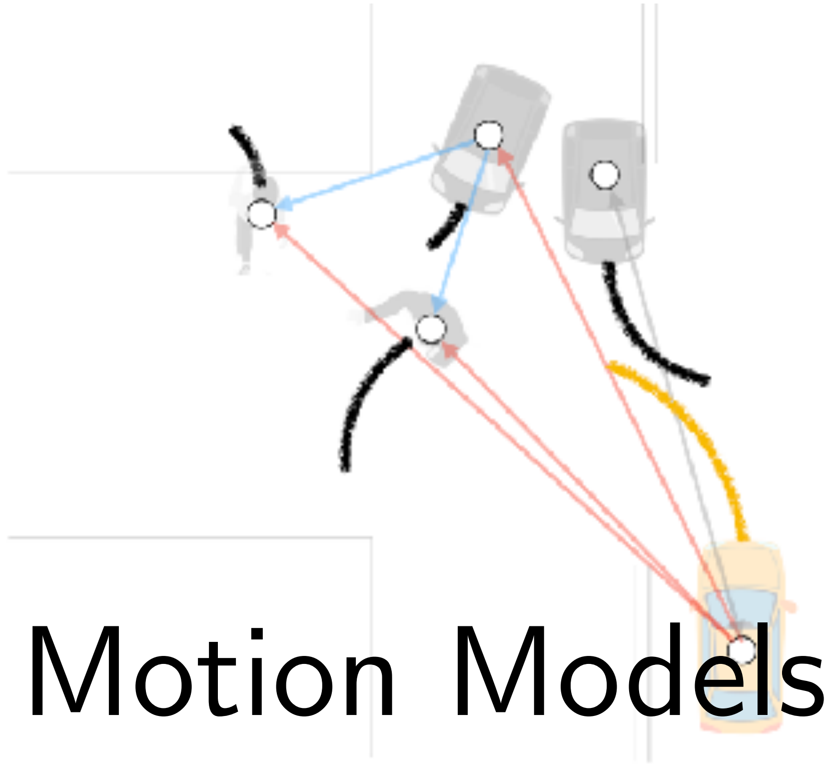
Known state

Unknown
dynamics

Models: From Simple to Complex



Physics Models



Motion Models

Open World Models

Simple — Complex

Physics Models	Motion Models	Open World Models
Known state	Known state	Unknown state
Strong prior on dynamics	Unknown dynamics	Unknown dynamics

Activity!

Modelling Tamago Sushi

Think-Pair-Share!

Think (30 sec): How would you model making tamago sushi?

Pair: Find a partner

Share (45 sec): Partners exchange ideas

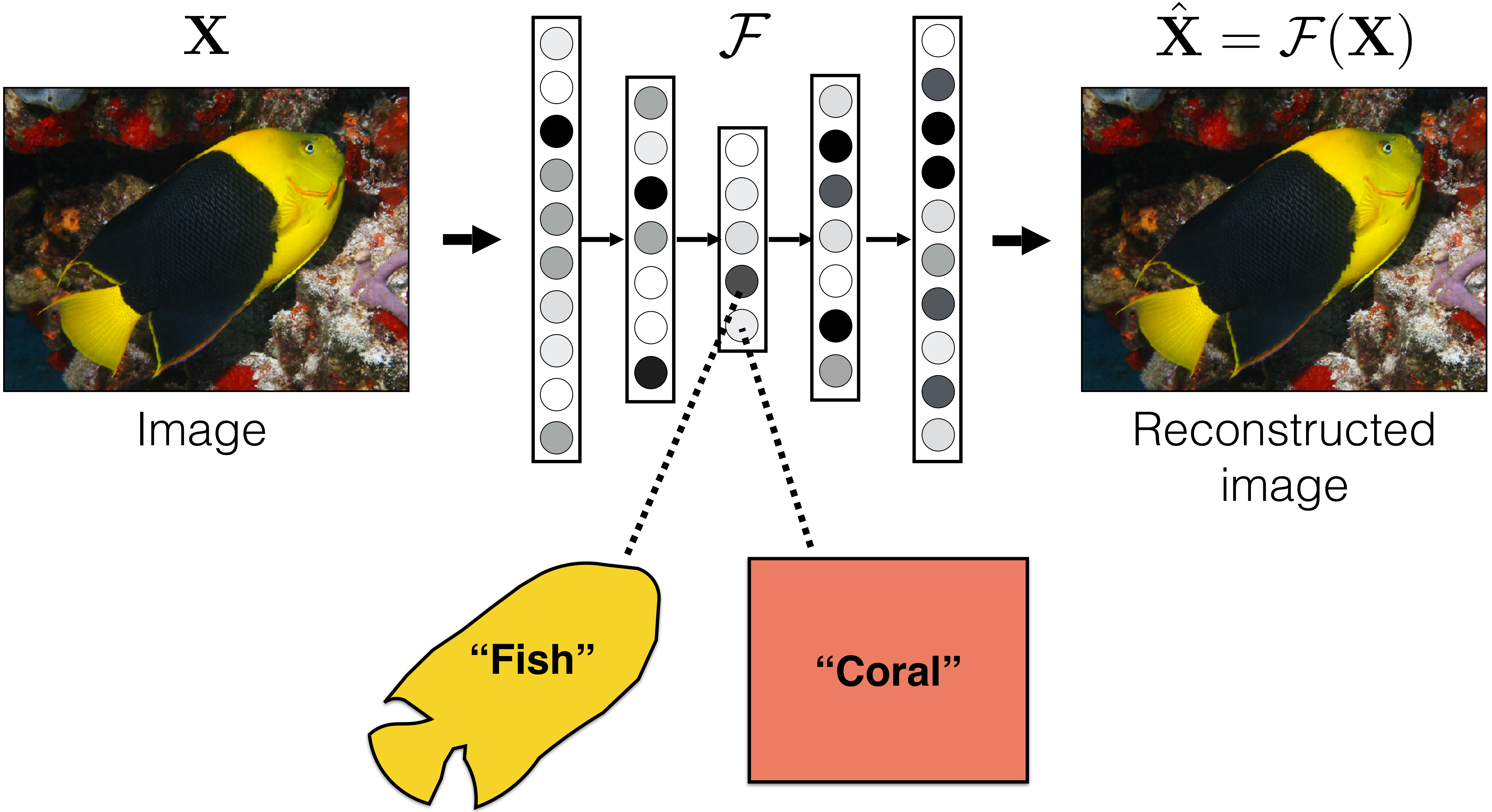
Challenges with learning complex models

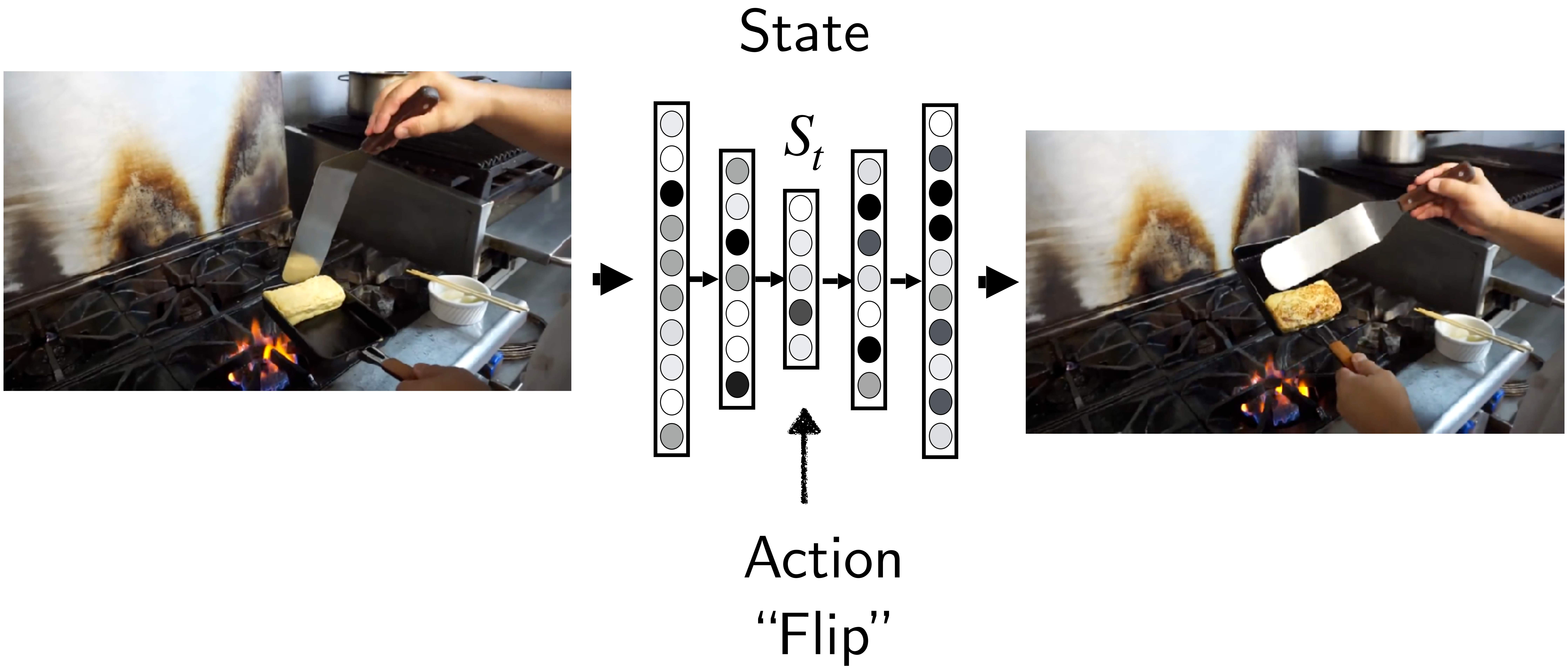
Challenge 1: Can't see state, only get high-dimensional observations

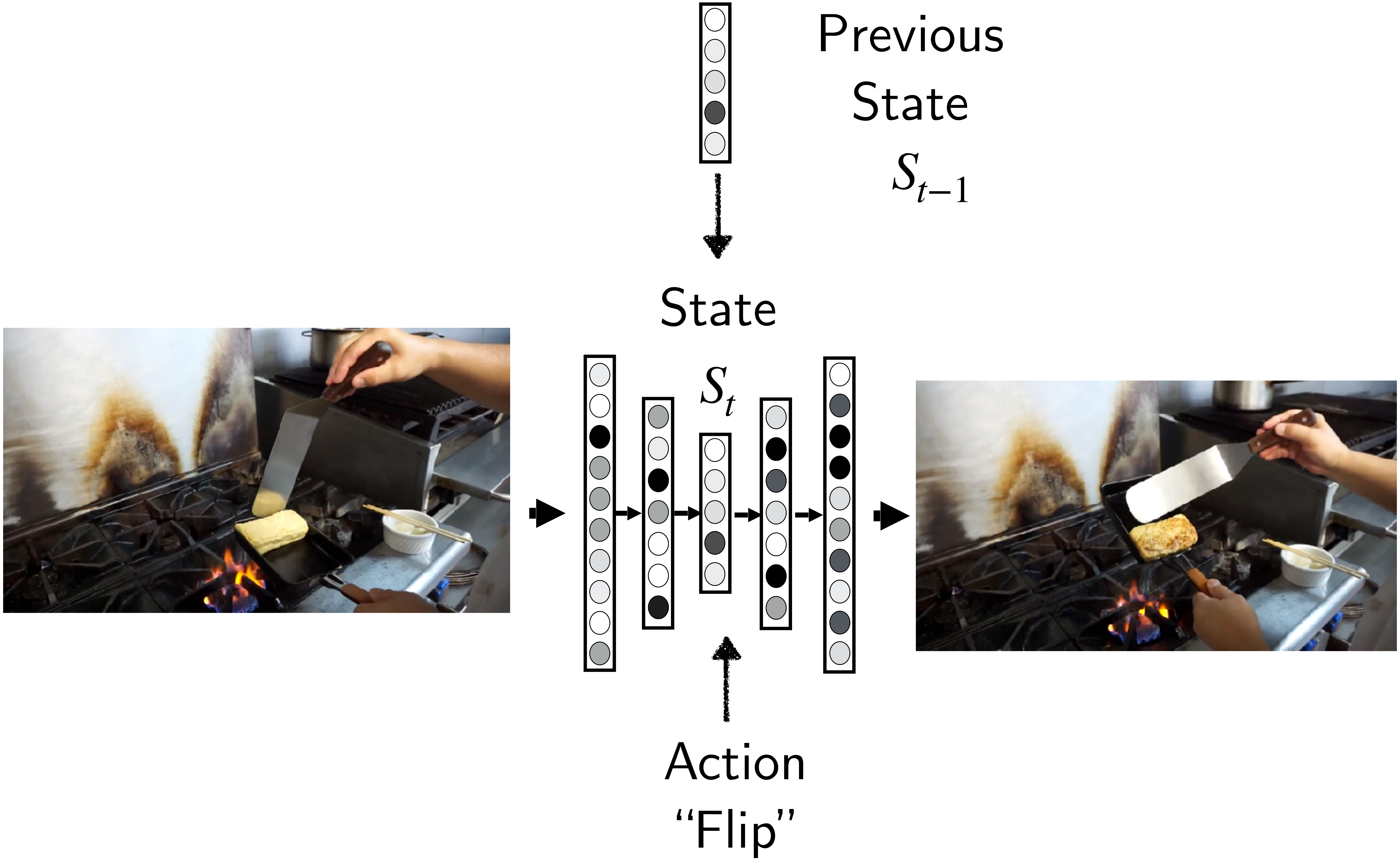
Challenge 2: Planning with complex dynamics

How can we learn latent low-dimensional state from high-dimensional observations?

Idea: Use “auto-encoder” trick from computer vision


$\mathbf{X}$
$\mathcal{F}$
$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$
Image
Reconstructed image
"Fish"
"Coral"


State
S_t
Action
“Flip”


Previous
State
S_{t-1}
State
S_t
Action
“Flip”

Today's class

- ☑ Deriving MBRL loss

  (Sim. lemma, PD via PM lemma)

- ☑ Practical MBRL

  (Only observations, complex dynamics)

- ☐ The DREAMER algorithm

The DREAMER Algorithms

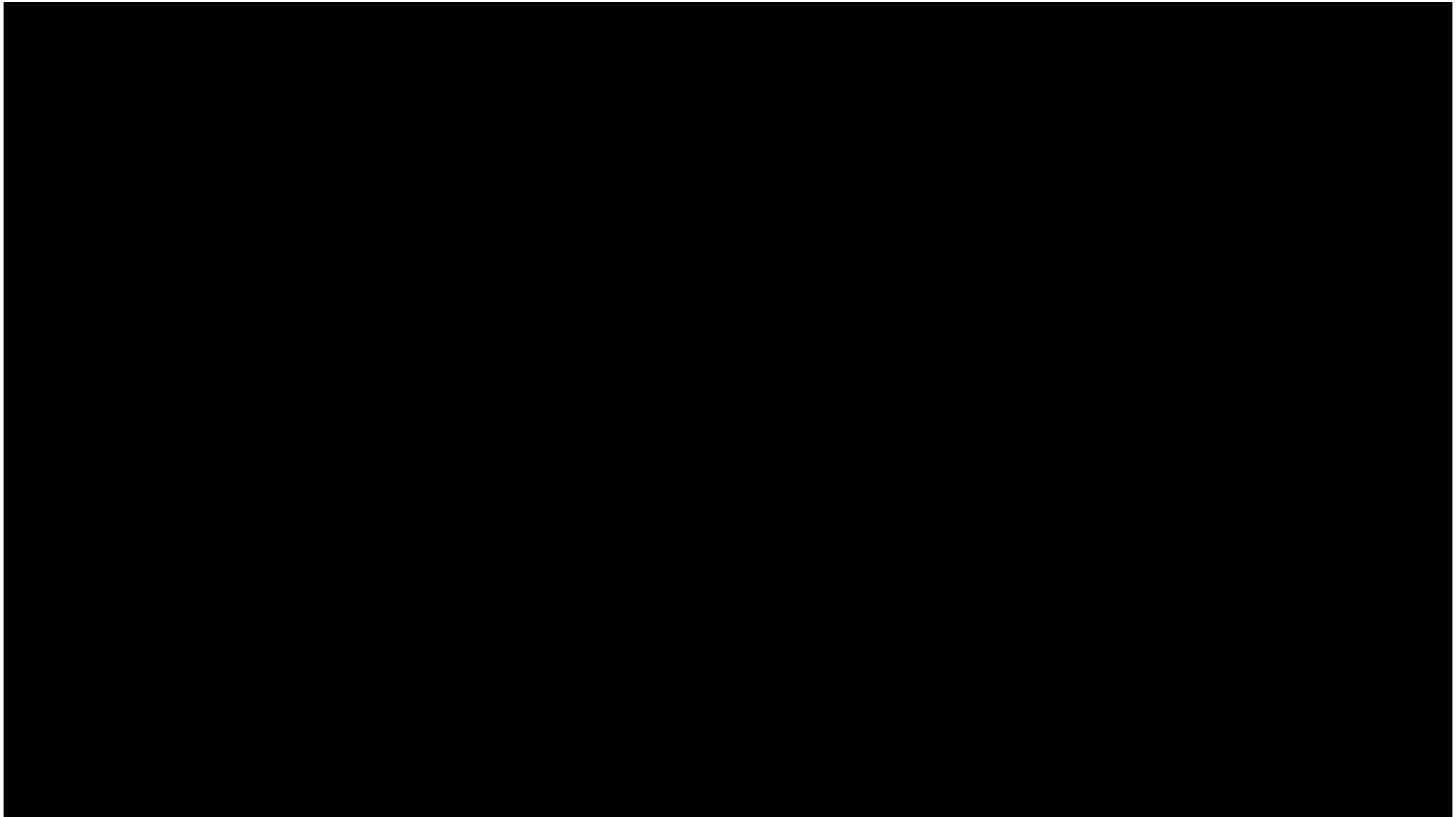
2023

Mastering Diverse Domains through World Models

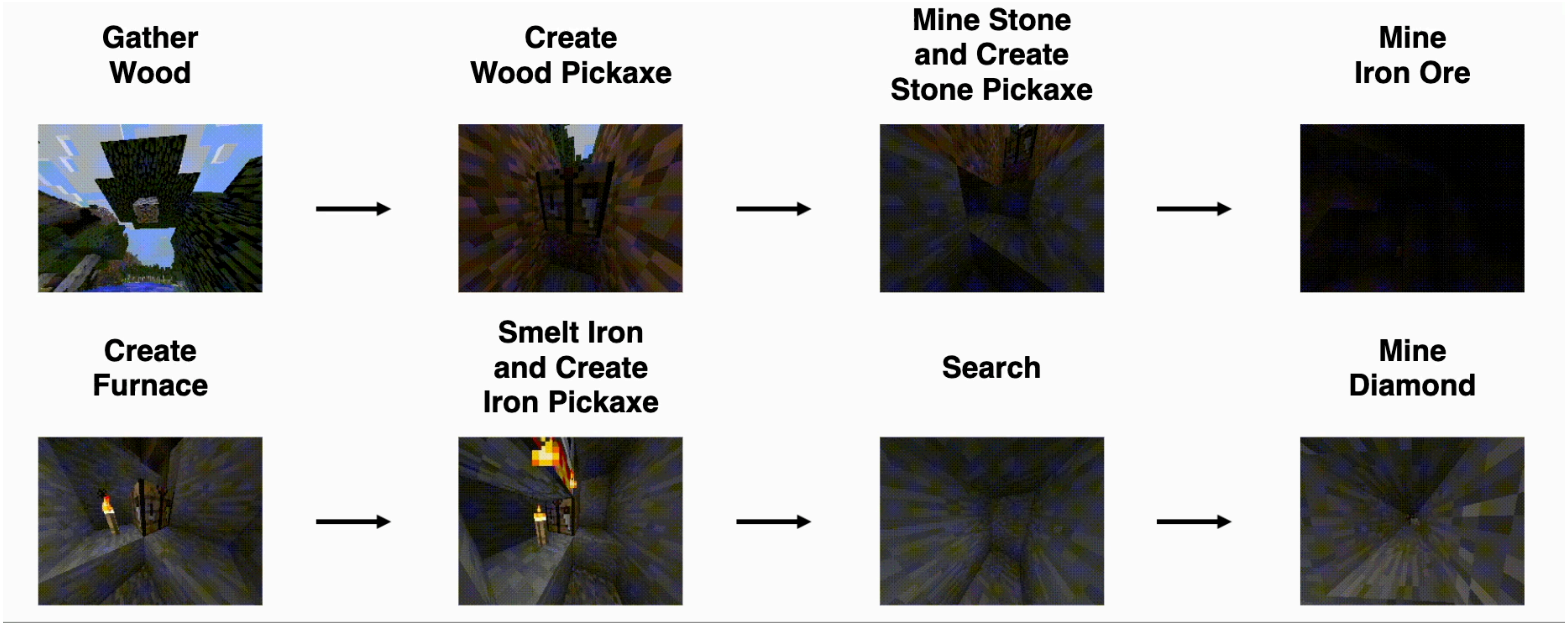
Danijar Hafner,[1,2] Jurgis Pasukonis,[1] Jimmy Ba,[2] Timothy Lillicrap[1]

[1]DeepMind [2]University of Toronto

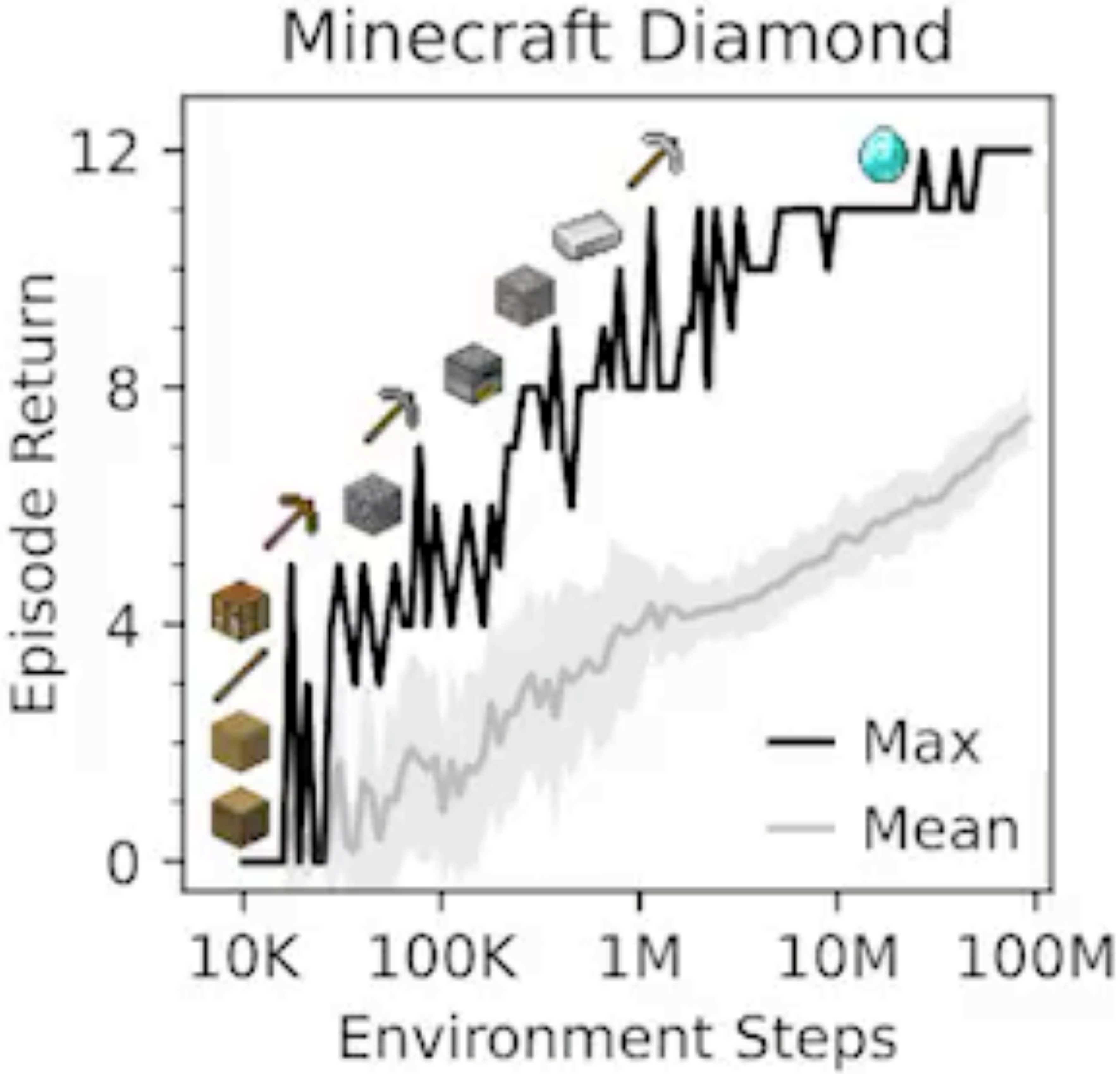
MineRL Diamond Challenge

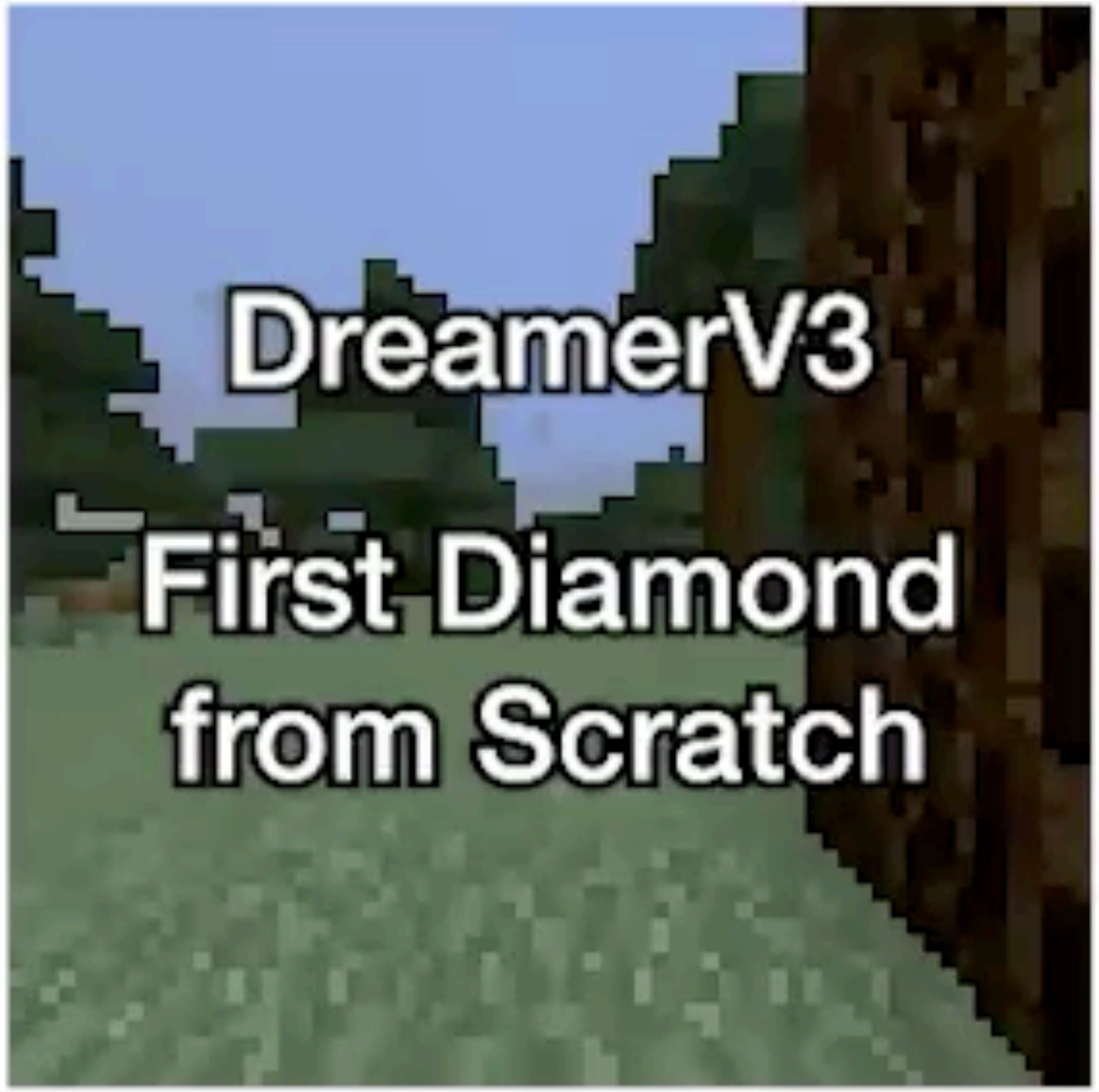


MineRL Diamond Challenge



DreamerV3 solved this task!





The DREAMER Algorithm

Dream to Control: Learning Behaviors by Latent Imagination

Danijar Hafner *
University of Toronto
Google Brain

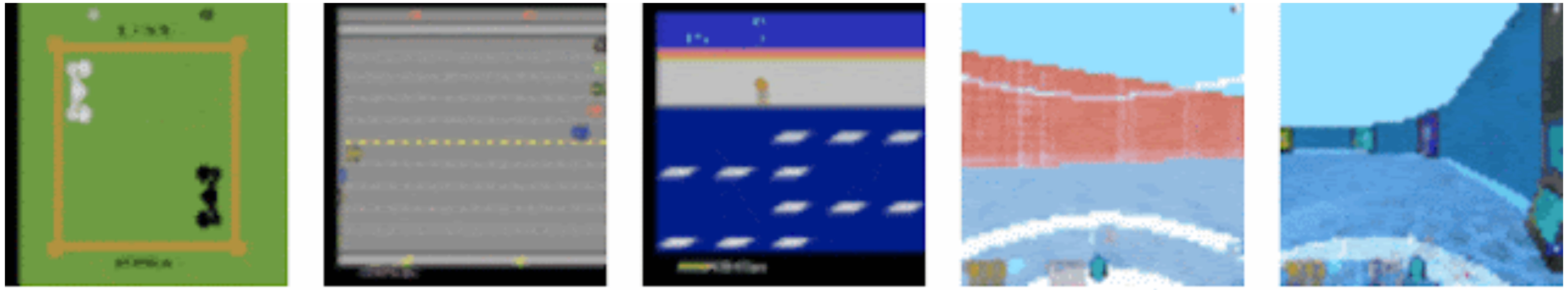
Timothy Lillicrap
DeepMind

Jimmy Ba
University of Toronto

Mohammad Norouzi
Google Brain

2020

Look at the videos below

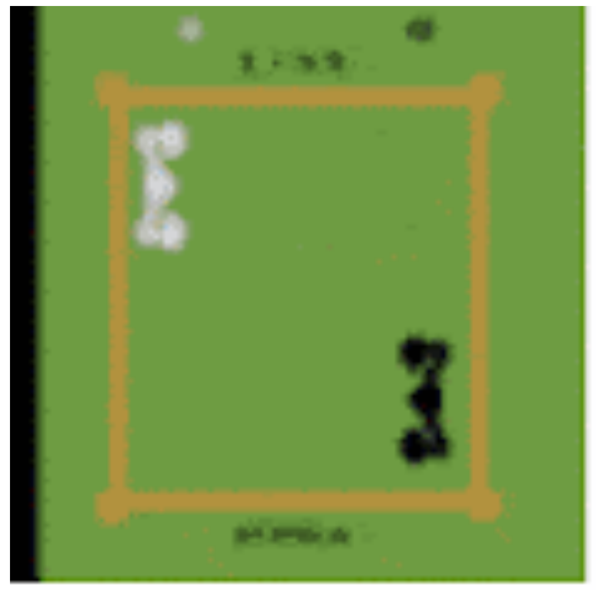


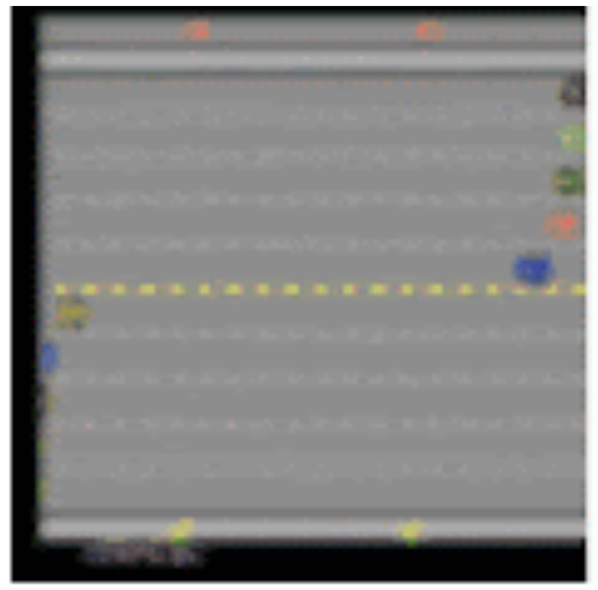
Boxing Freeway Frostbite Collect Objects Watermaze

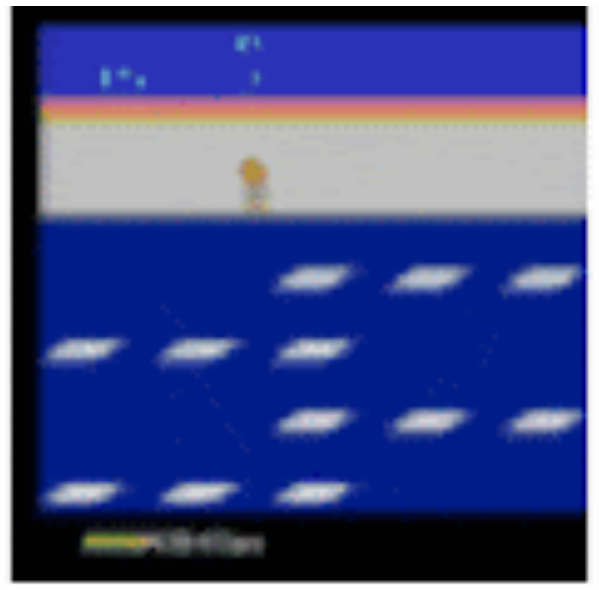
Sparse Cartpole Acrobot Swingup Hopper Hop Walker Run Quadruped Run

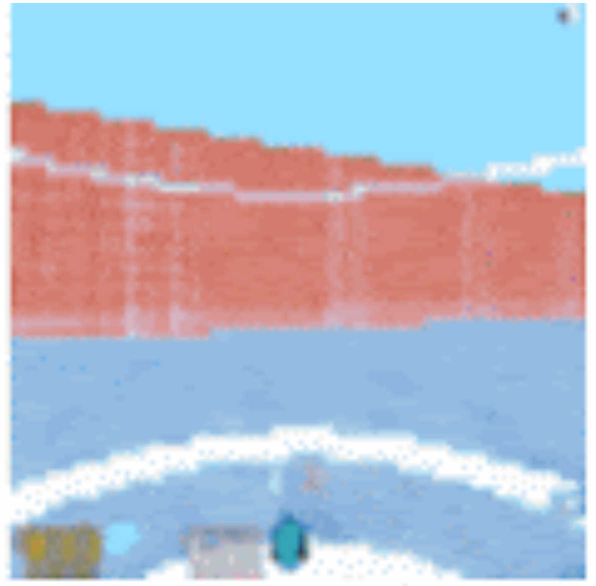
Is this from the actual simulator or predictions made by a model?

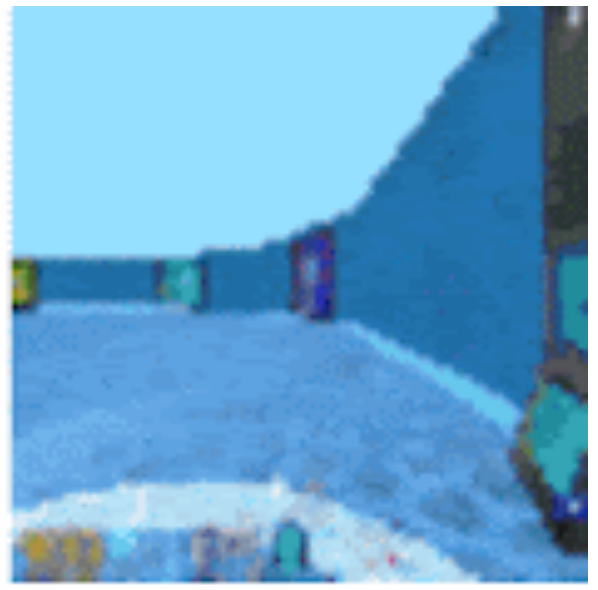
Look at the videos below


Boxing


Freeway


Frostbite


Collect Objects


Watermaze


Sparse Cartpole


Acrobot Swingup


Hopper Hop

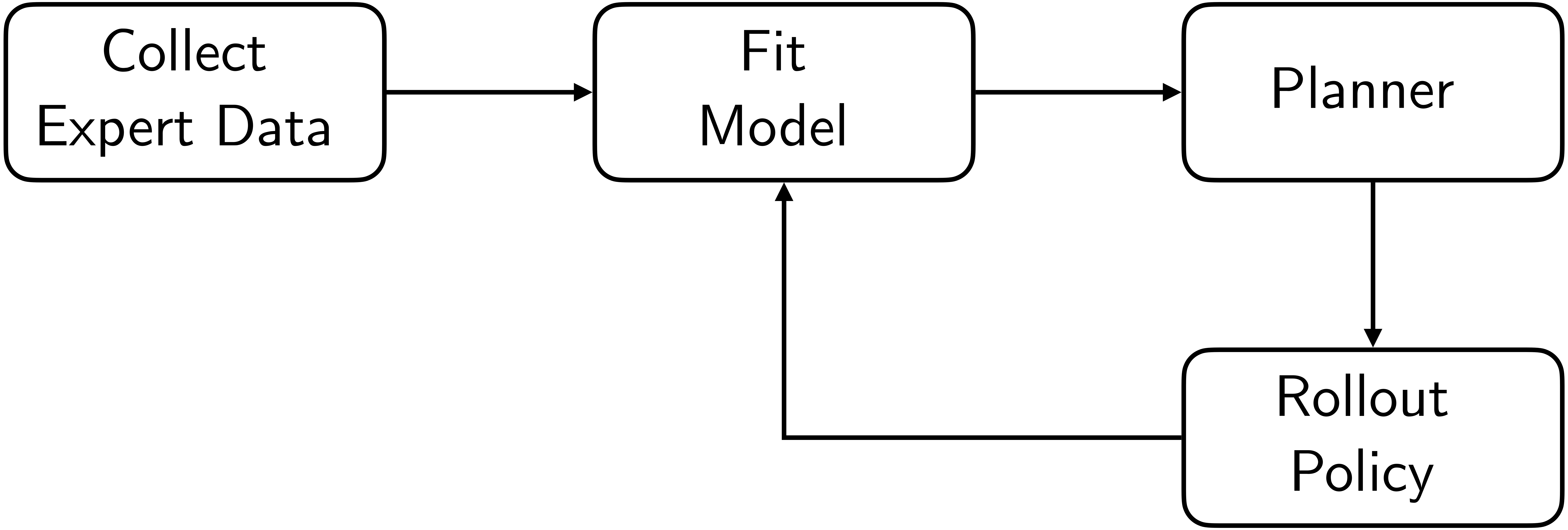

Walker Run


Quadruped Run

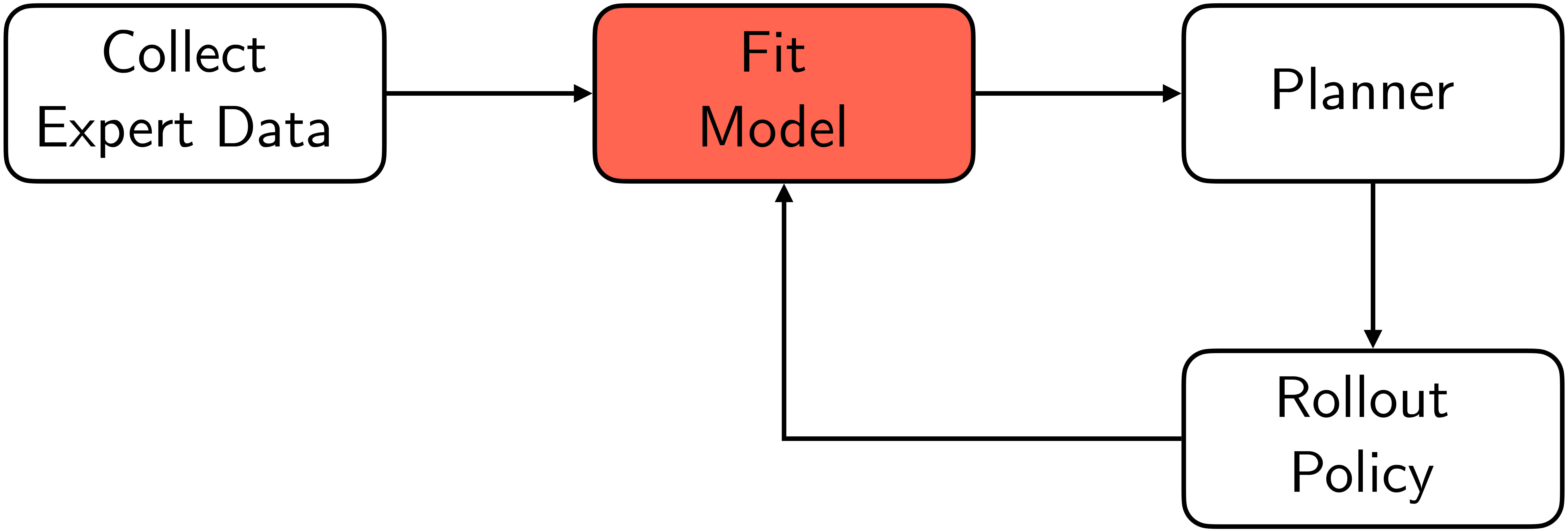
Predictions by a model!

Recap: Model-based RL

(Ross & Bagnell, 2012)



How does DREAMER fit a model?



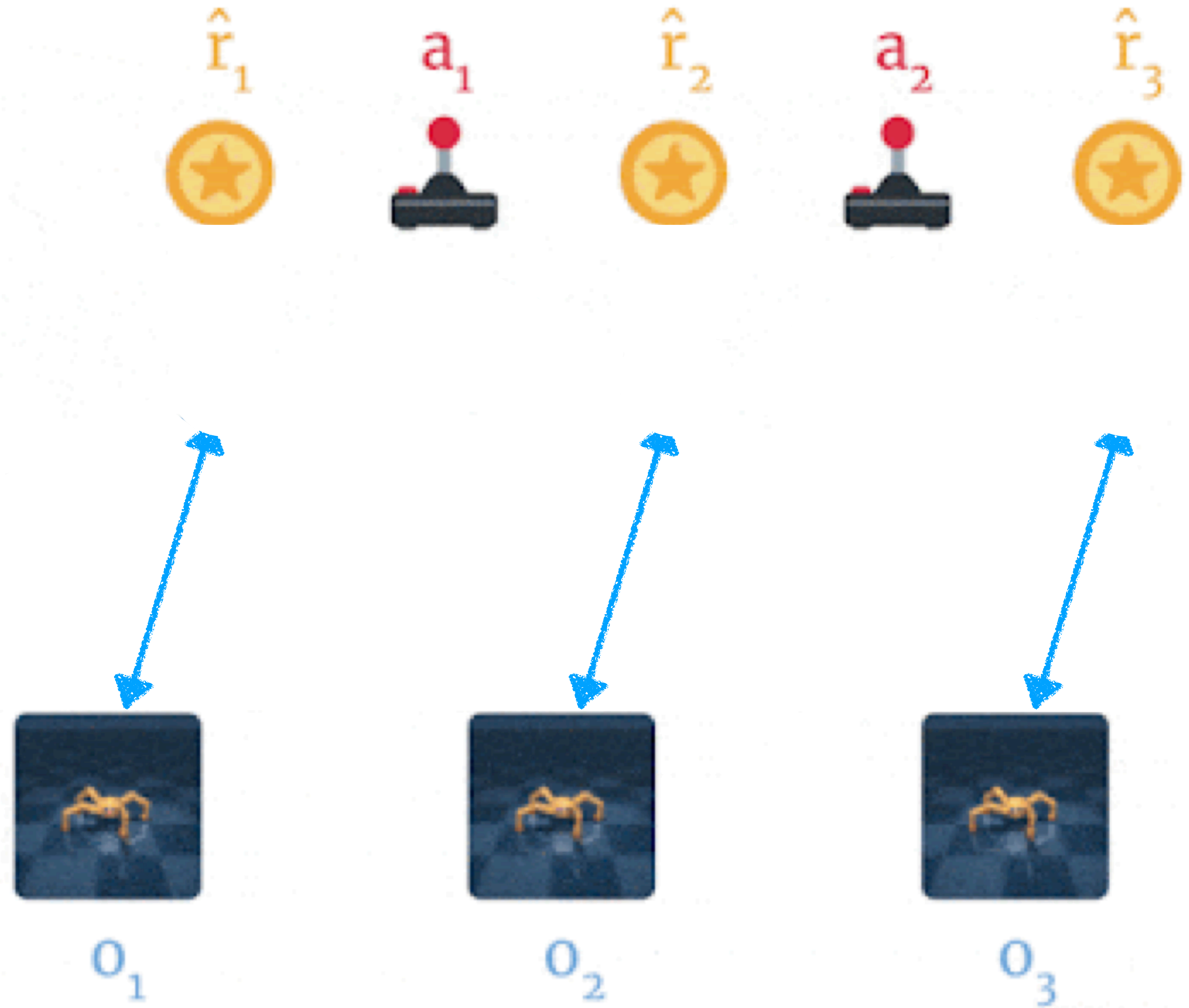
Goal: Fit a Model given data

Given Data:
Observations, rewards,
actions

Goal: Fit a Model given data

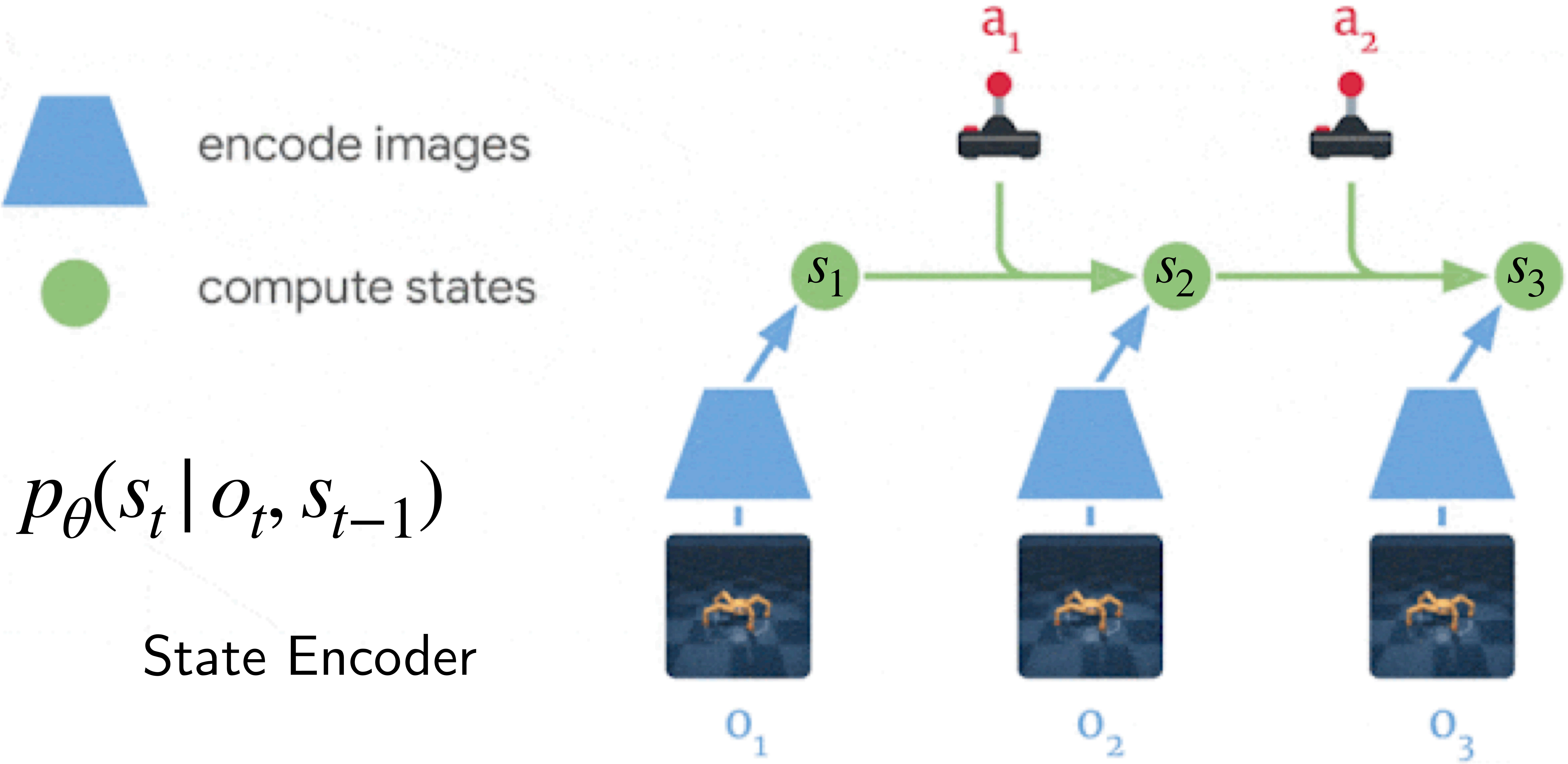
Given Data:
Observations, rewards,
actions

Predict:
States,
Dynamics Function,
Reward Function



Actions

Observations



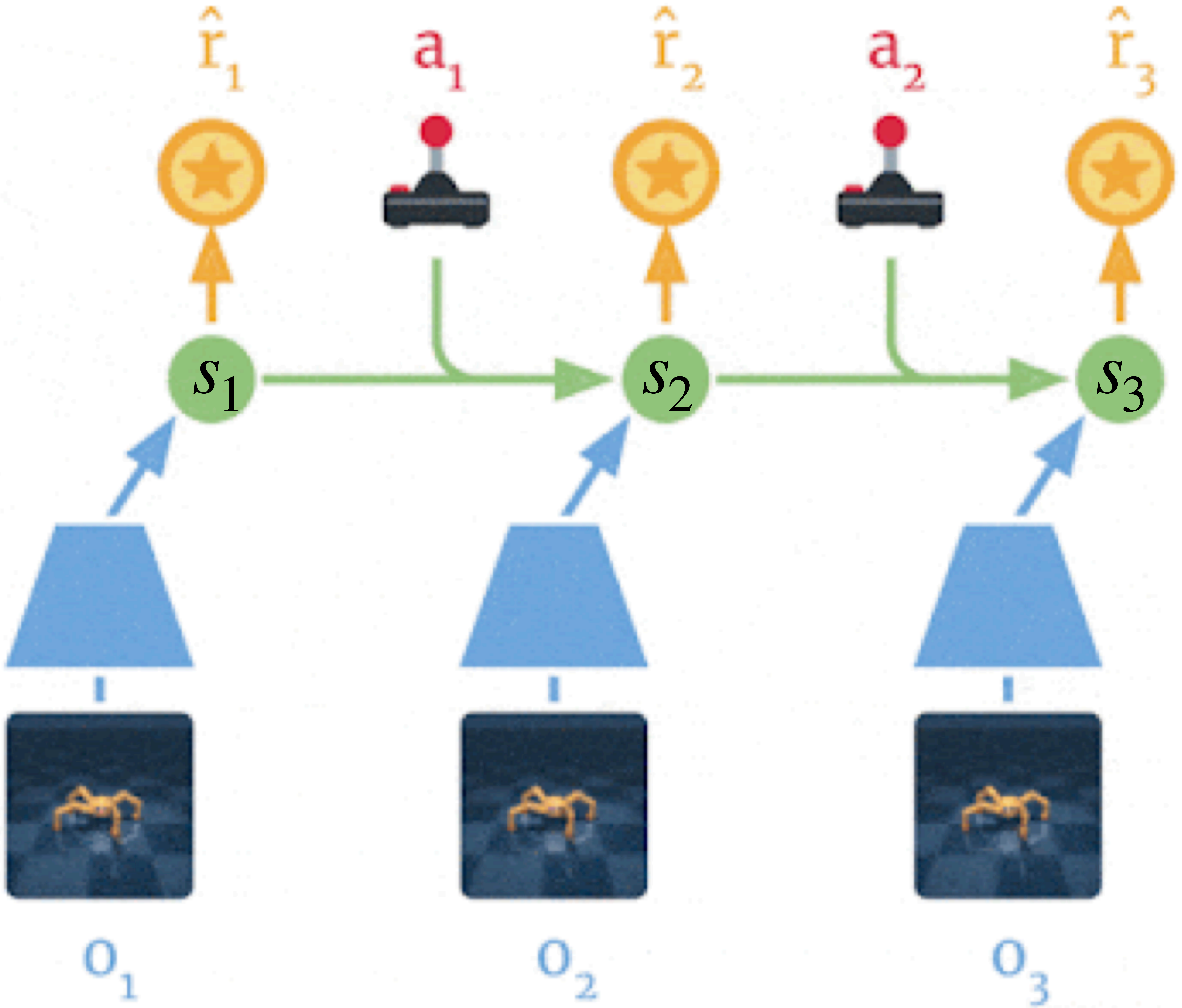
$$p_\theta(s_t \mid o_t, s_{t-1})$$

State Encoder

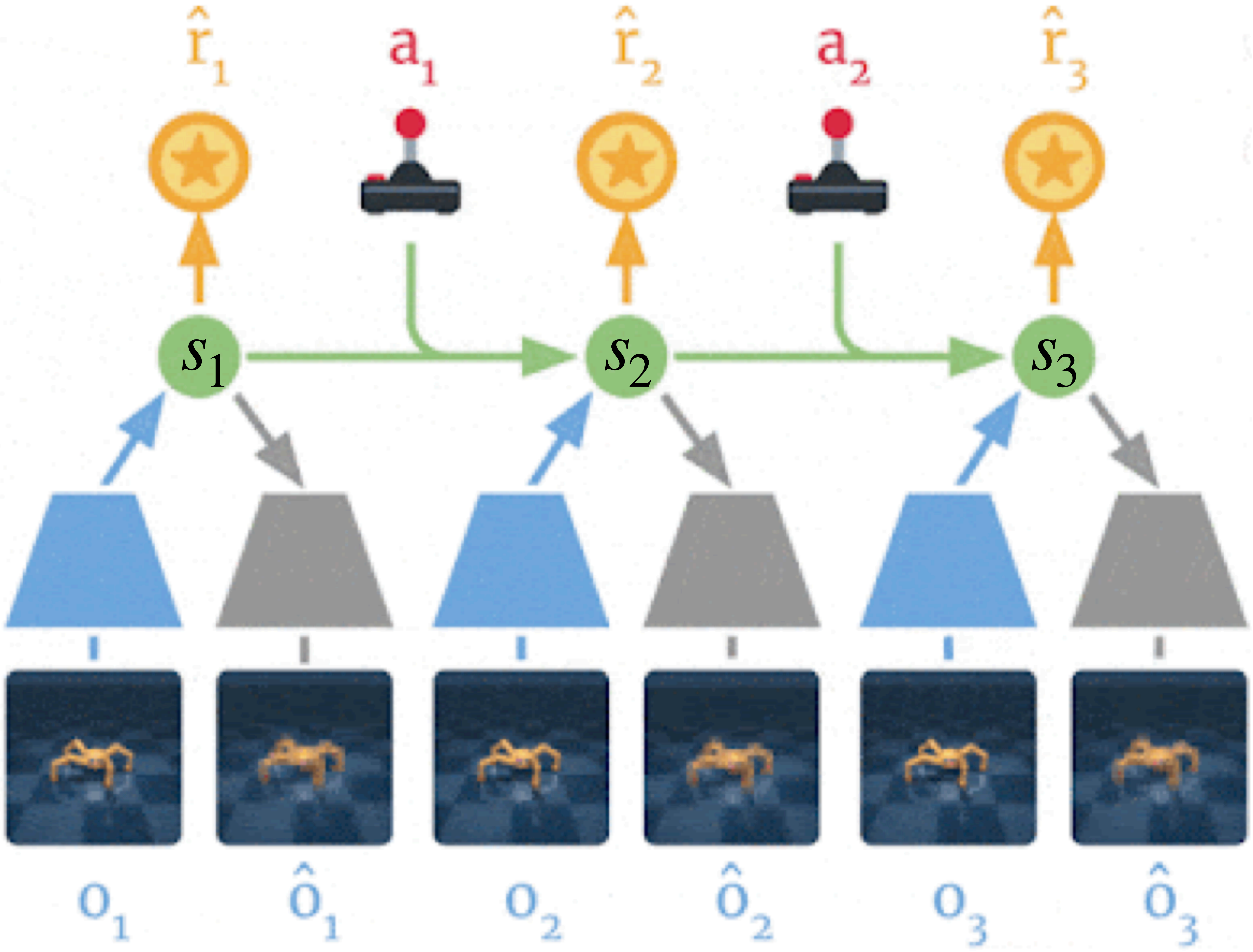
$$\ell = (r_t - \hat{r}_t)^2$$

$$q_\theta(r_t \mid s_t)$$

Reward Decoder



$$\ell = (o_t - \hat{o}_t)^2$$

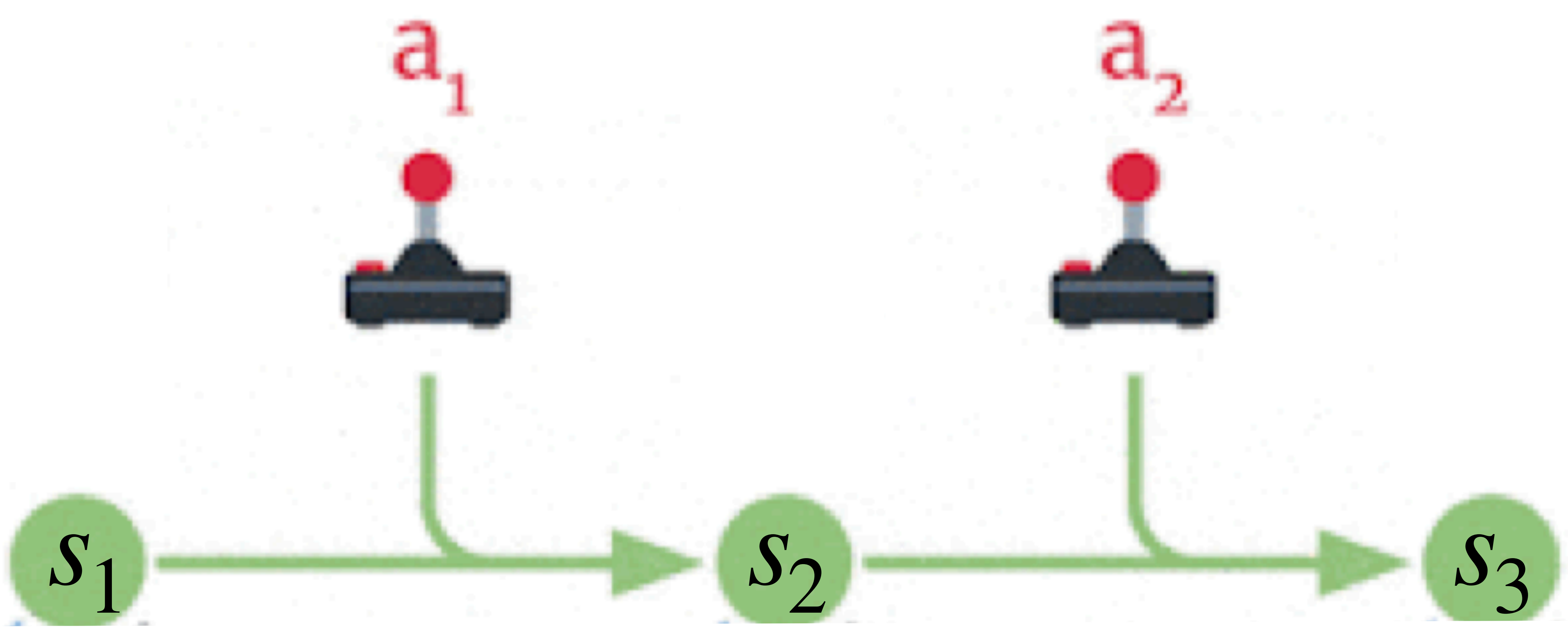


$$q_\theta(o_t \mid s_t)$$

Observation Decoder

$$q_\theta(s_{t+1} \mid s_t, a_t)$$

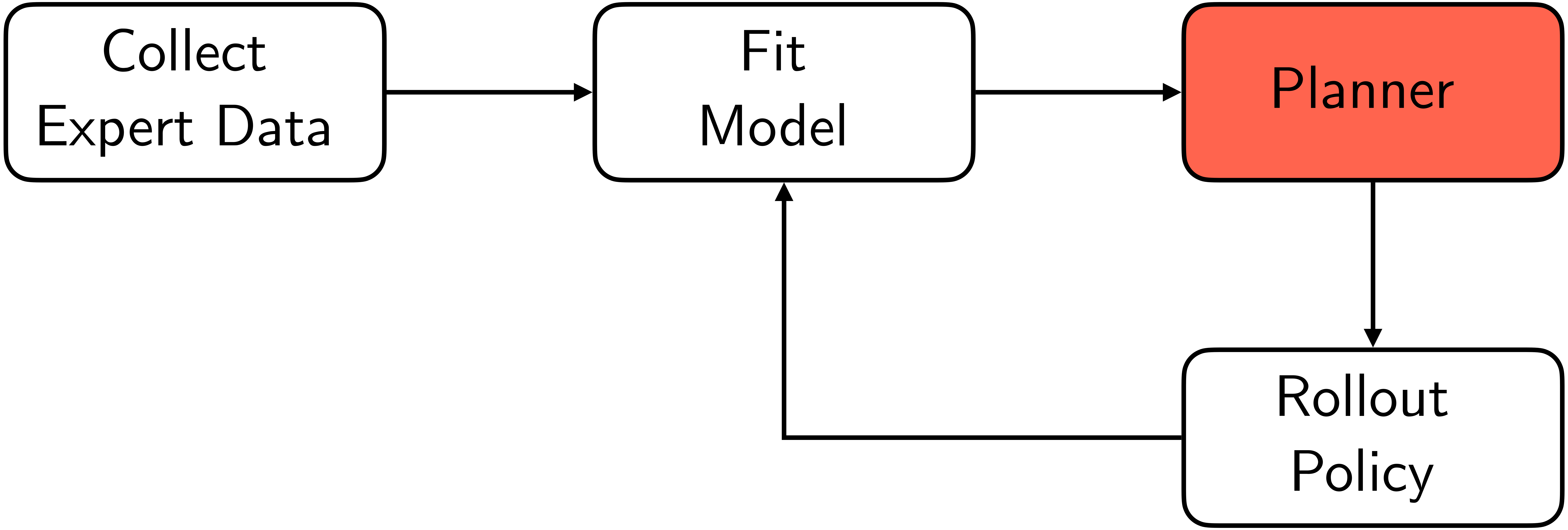
Dynamics
Function



Results: Learning World Model

Results: Learning World Model

How does DREAMER do planning?



Goal: Learn a Policy using Actor-Critic

$\pi_\phi(a_t \mid s_t)$

Actor

$V_\psi(s_t)$

Critic

From rollouts in the model

$q_\theta(s_t \mid s_{t-1}, a_{t-1})$

Recall: Actor-Critic

Start with an arbitrary initial policy $\pi_\phi(a \mid s)$

while *not converged* **do**

Roll-out $\pi_\phi(a \mid s)$ in the model $q_\theta(s' \mid s, a)$ to collect trajectories $D = \{s^i, a^i, r^i, s^i_+\}_{i=1}^N$

Fit value function $V_\psi(s^i)$ using TD, i.e. minimize $(r^i + \gamma V_\psi(s^i_+) - V_\psi(s^i))^2$

Compute advantage $\hat{A}(s^i, a^i) = r(s^i, a^i) + \gamma V_\psi(s^i_+) - V_\psi(s^i)$

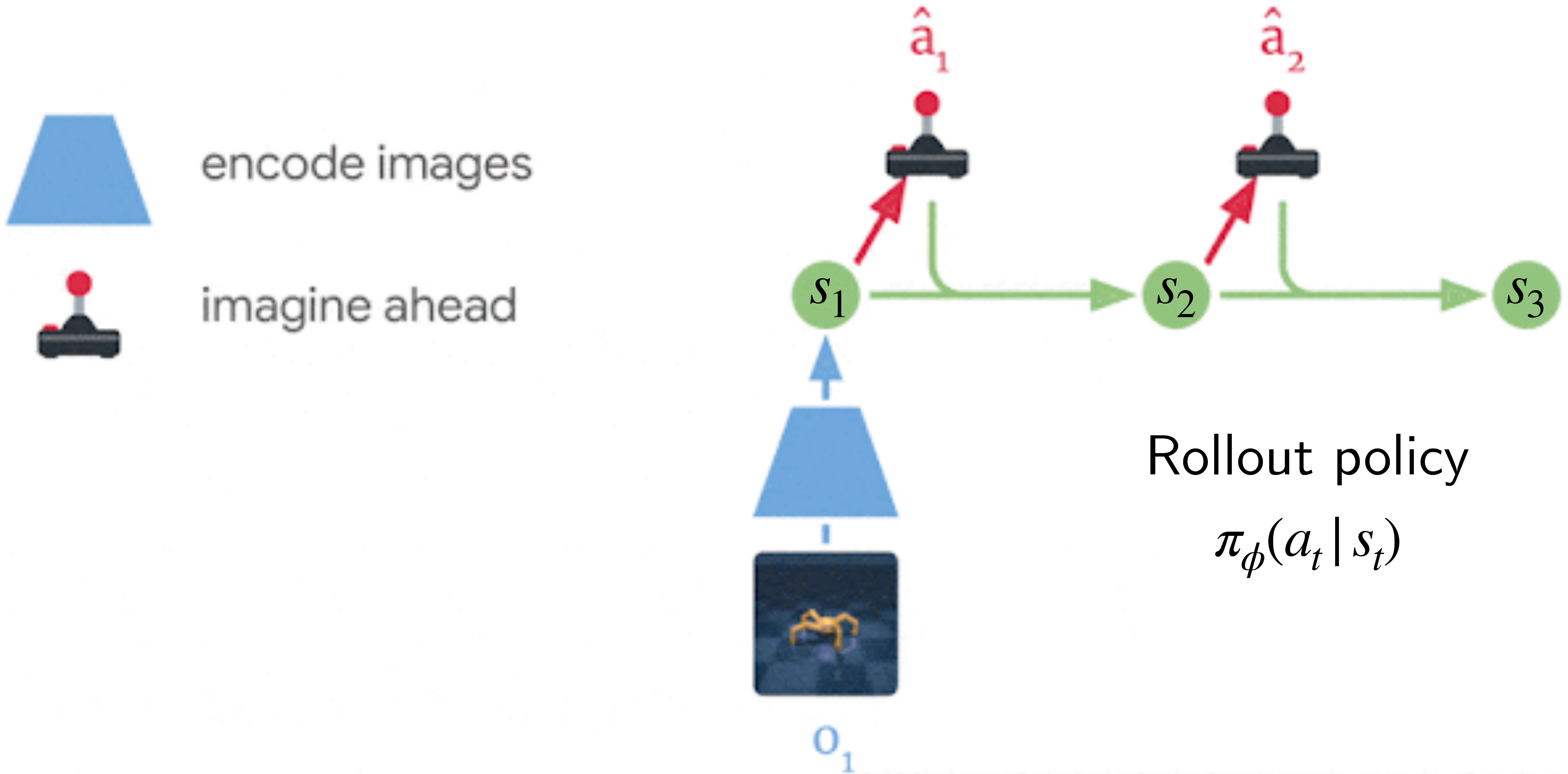
Compute gradient

$$\nabla_\phi J(\phi) = \frac{1}{N}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\phi(a^i_t \mid s^i_t)\, \hat{A}(s^i, a^i)\right]$$

Update parameters $\quad \phi \leftarrow \phi + \alpha \nabla_\phi J(\phi)$

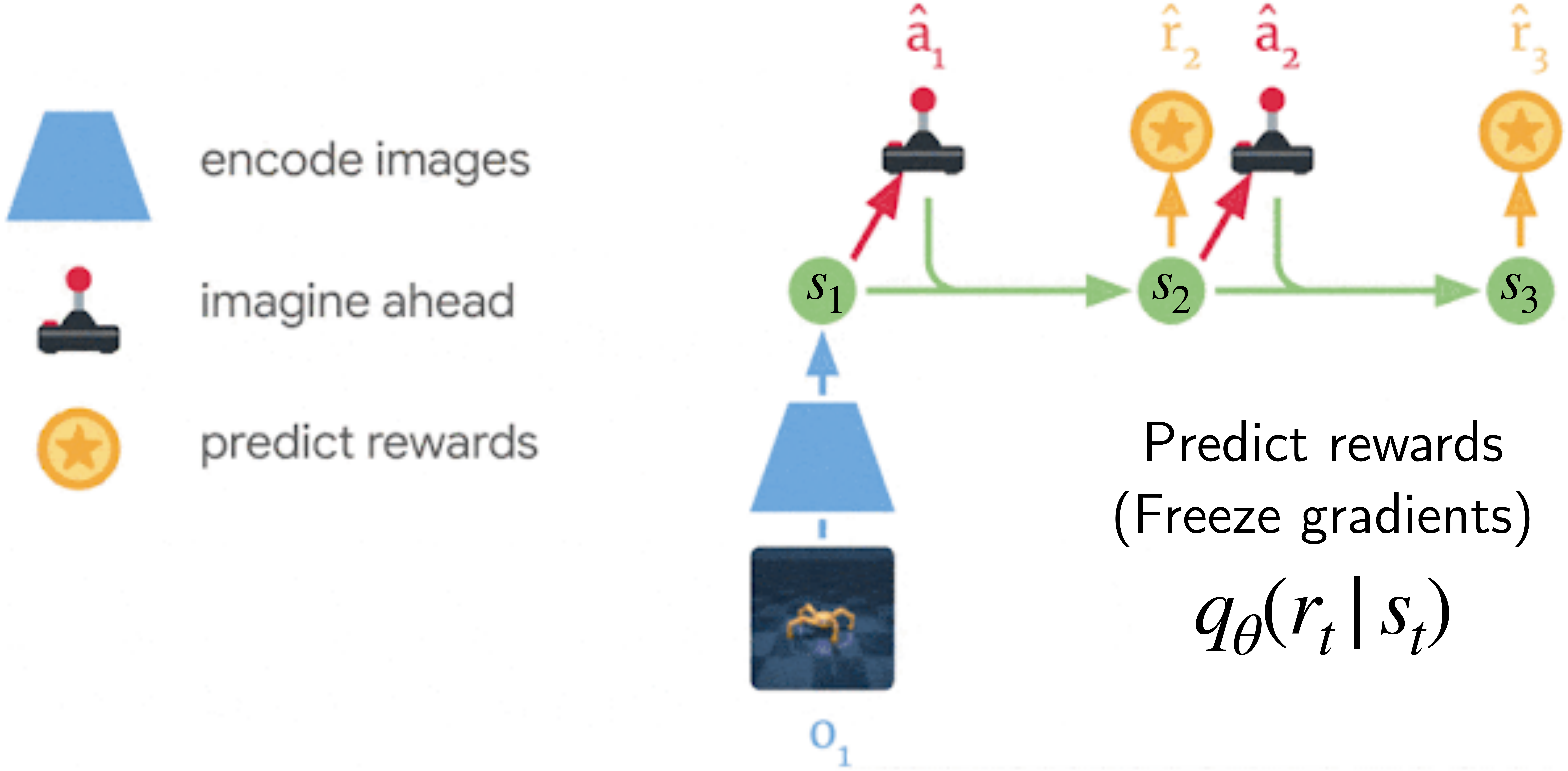
o_1

encode images



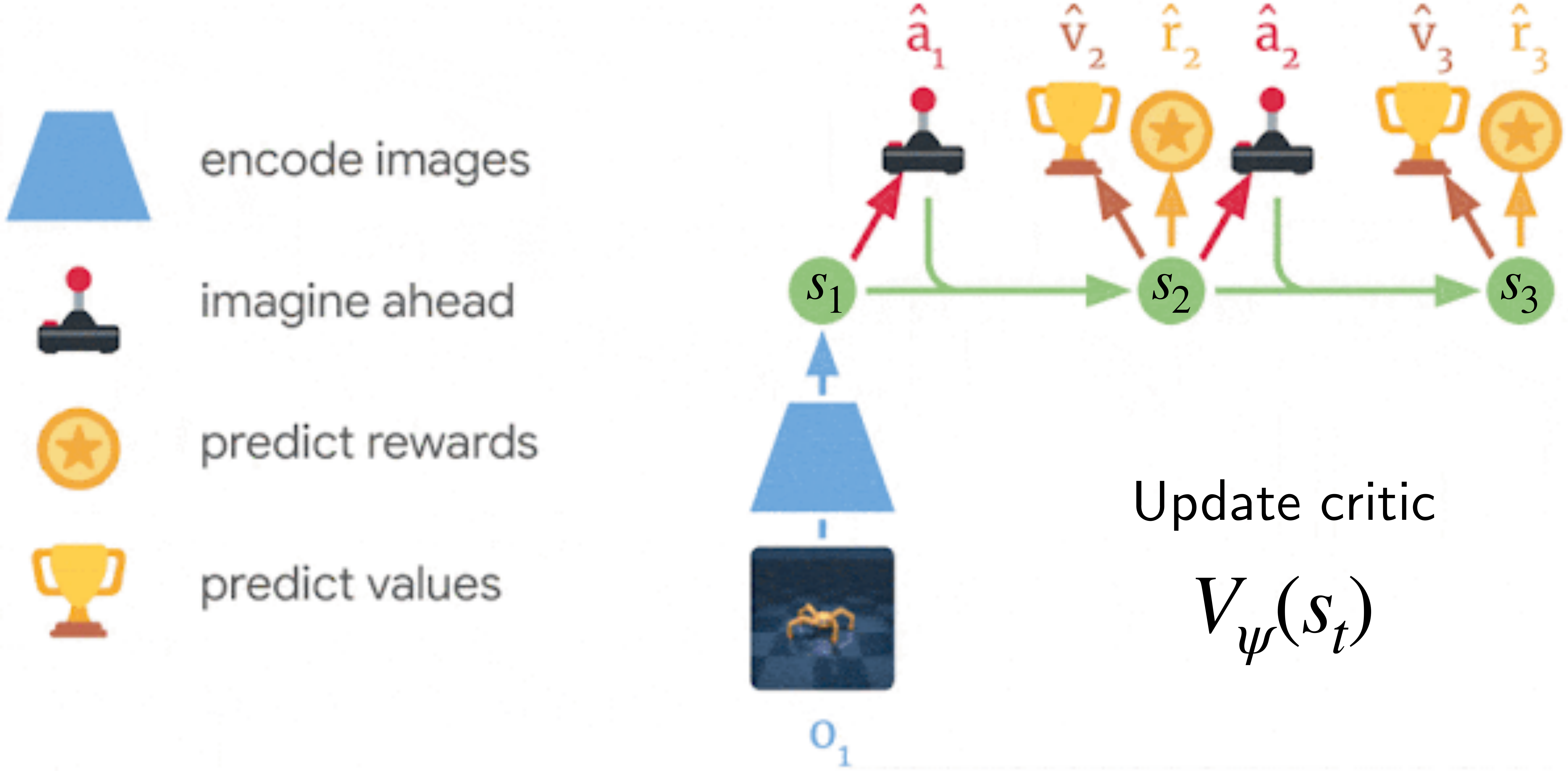
Rollout policy

$$\pi_\phi(a_t \mid s_t)$$



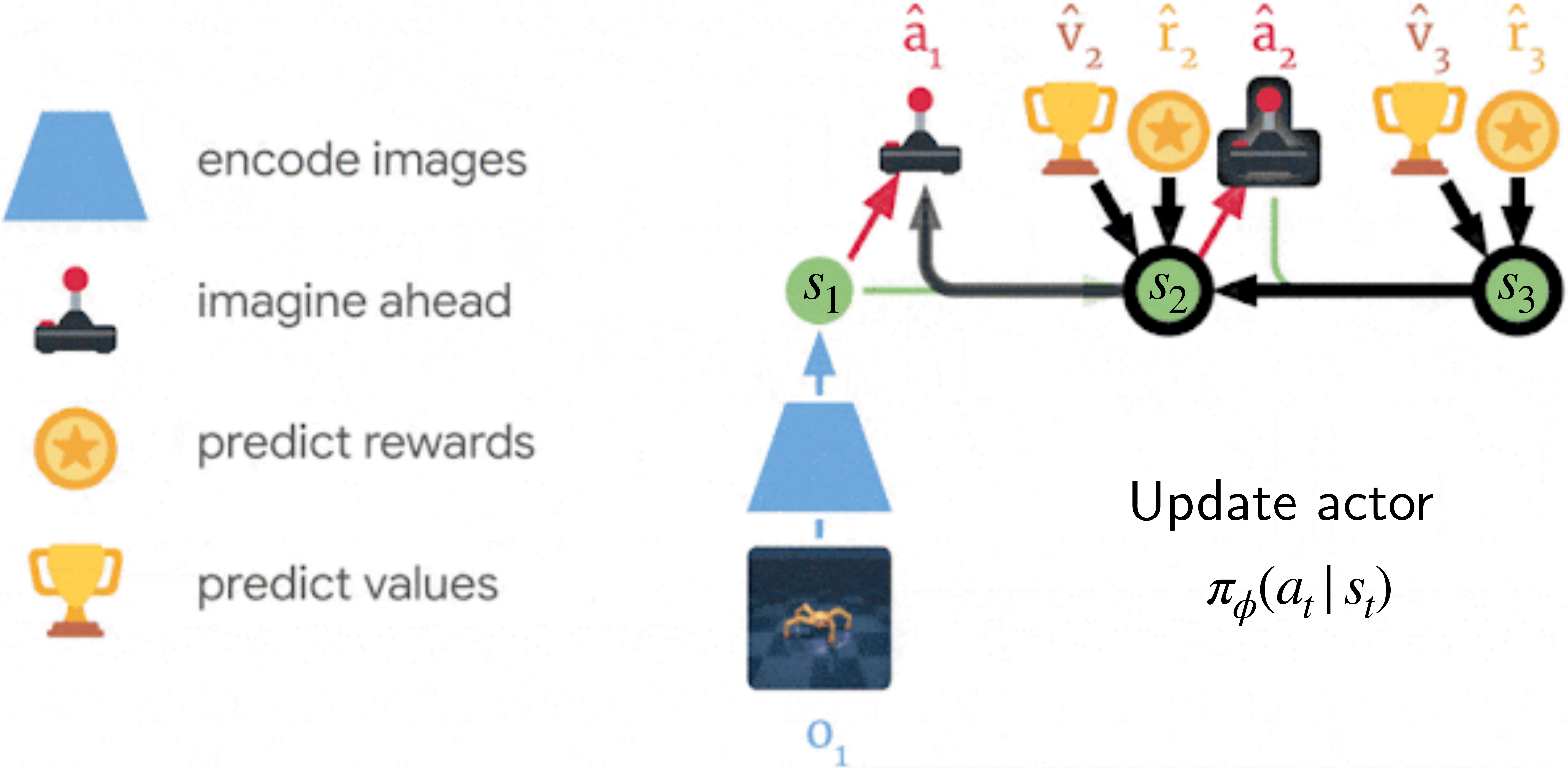
Predict rewards
(Freeze gradients)

$$q_\theta(r_t \mid s_t)$$



Update critic

$$V_\psi(s_t)$$



Update actor

$$\pi_\phi(a_t \mid s_t)$$

DREAMER: Results

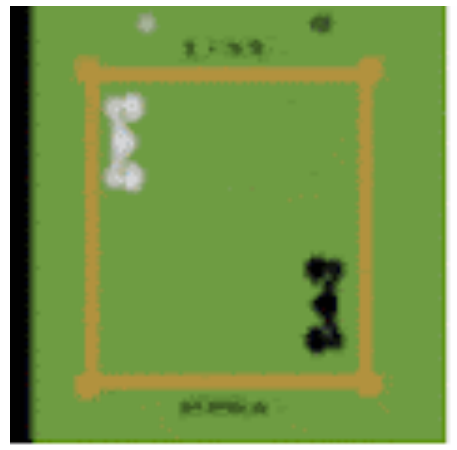

Sparse Cartpole

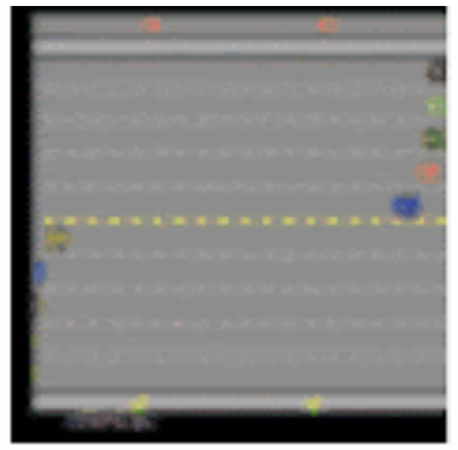

Acrobot Swingup

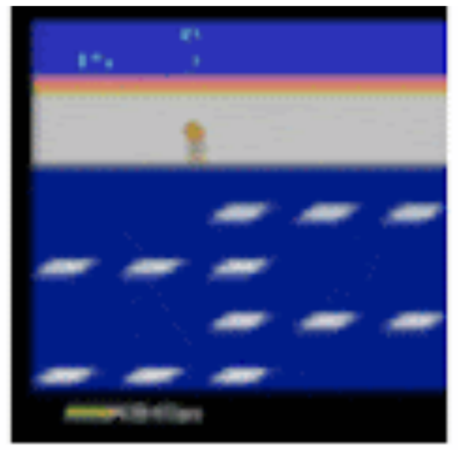

Hopper Hop


Walker Run


Quadruped Run

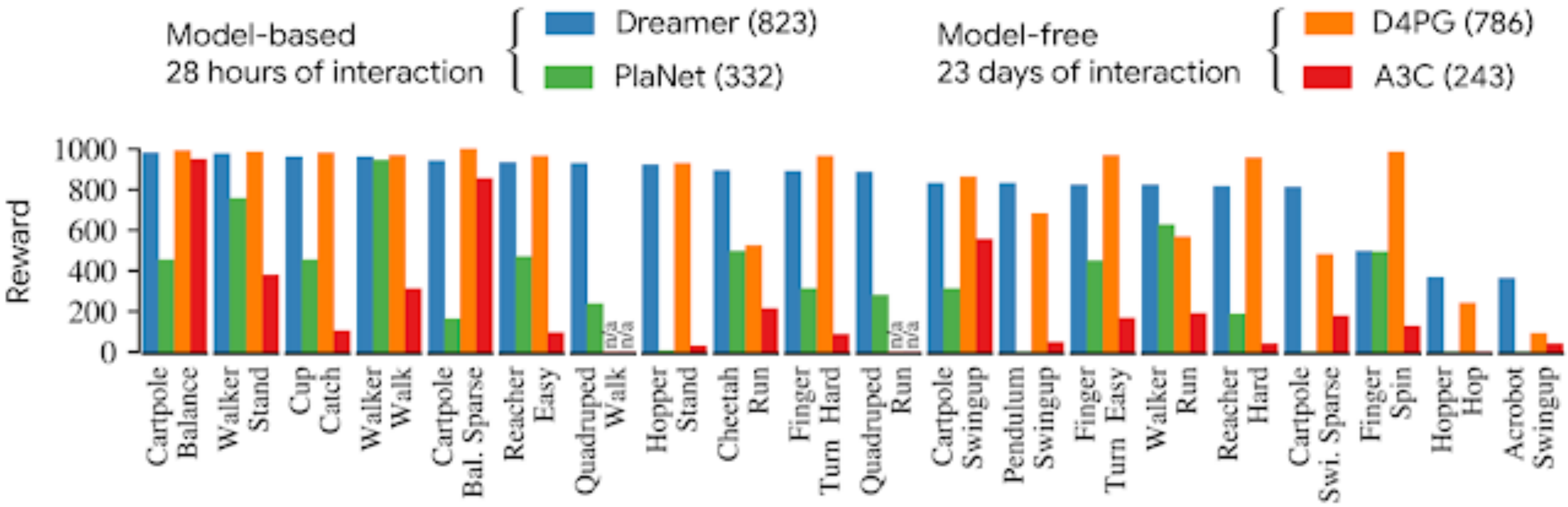

Boxing


Freeway


Frostbite


Collect Objects


Watermaze



DREAMER is a template for Model-based RL

But there are many challenges as we scale to harder real-world applications

DREAMER V2:

Tackling the world of Atari Games

MASTERING ATARI WITH DISCRETE WORLD MODELS

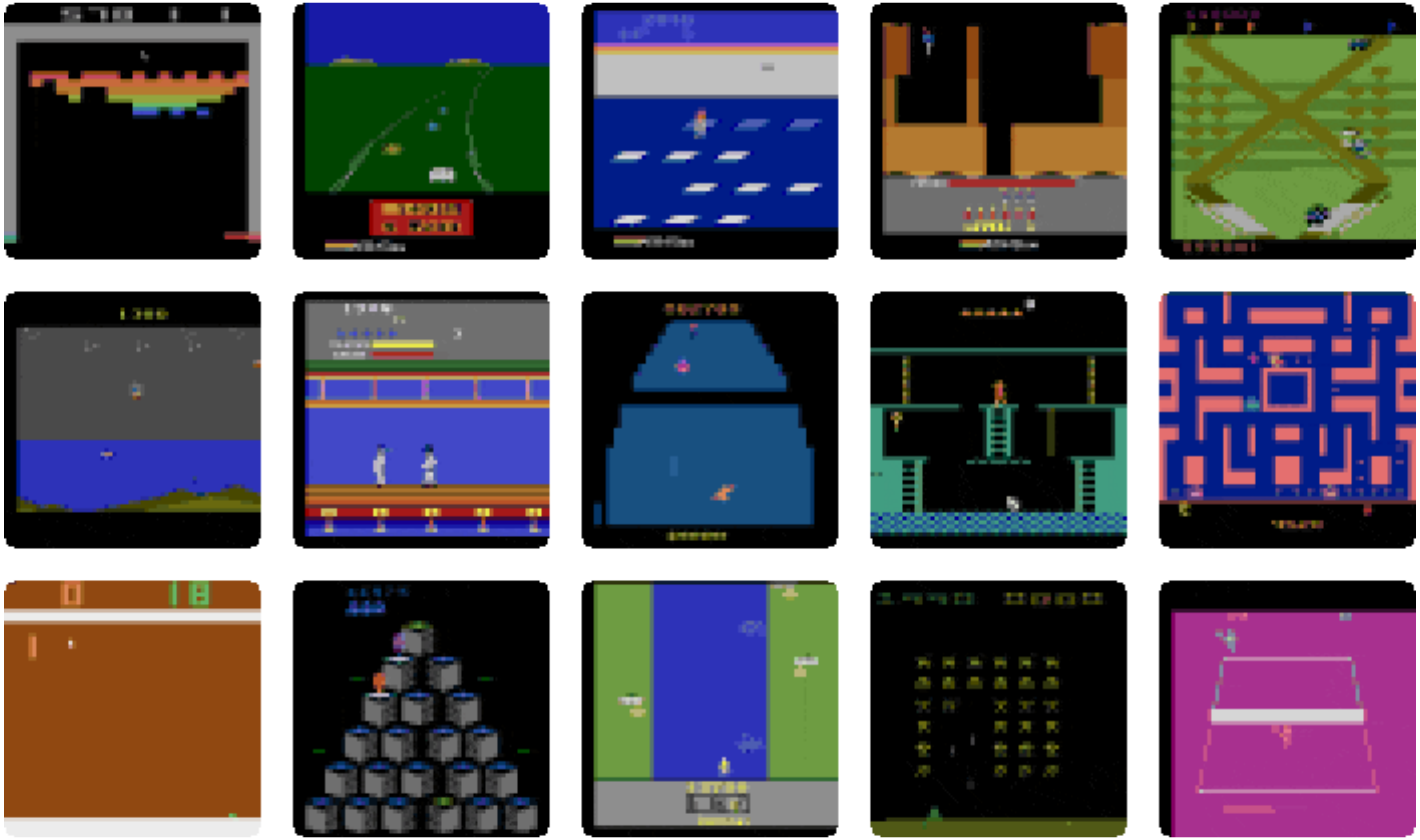
2021

Danijar Hafner *
Google Research

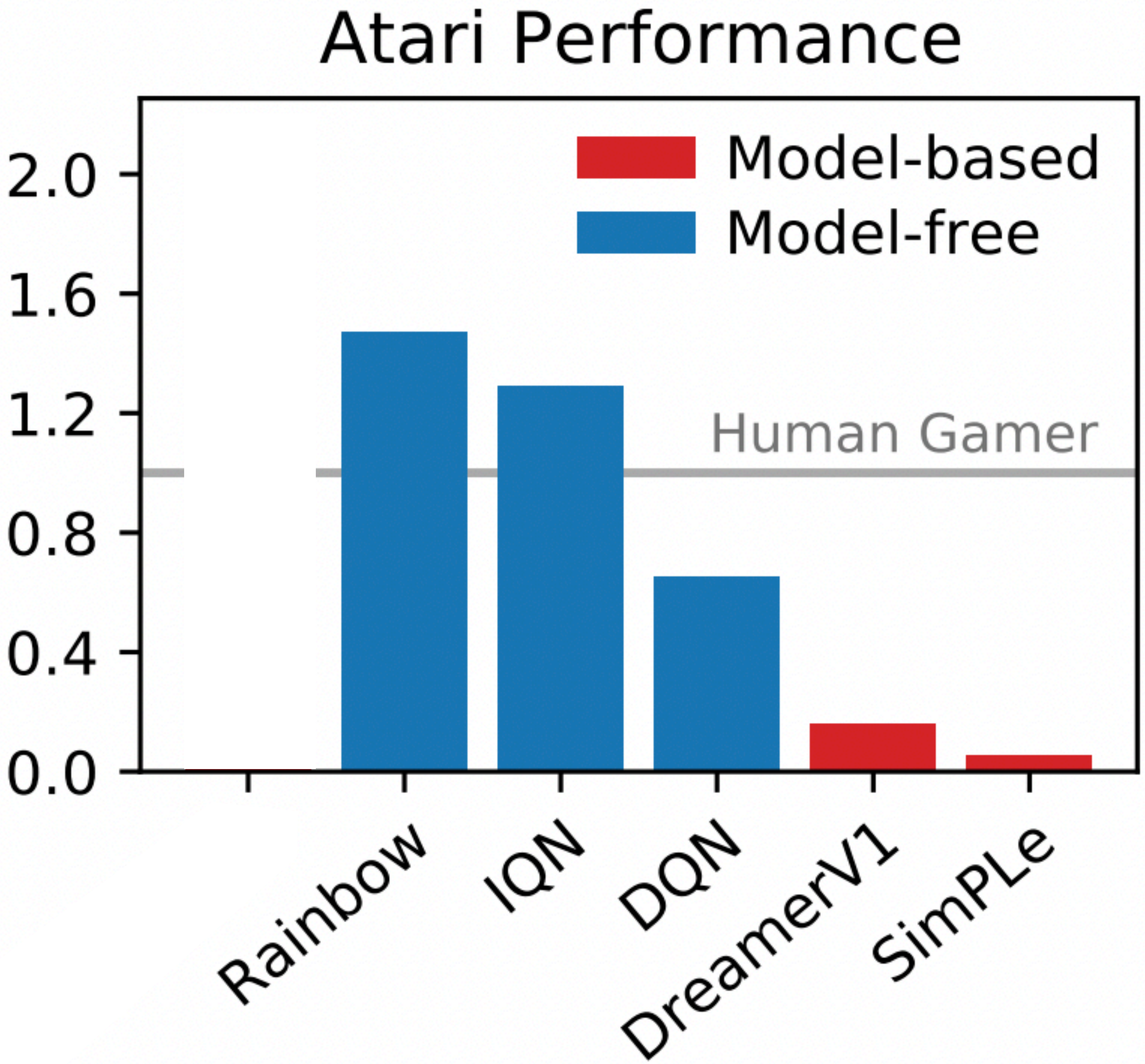
Timothy Lillicrap
DeepMind

Mohammad Norouzi
Google Research

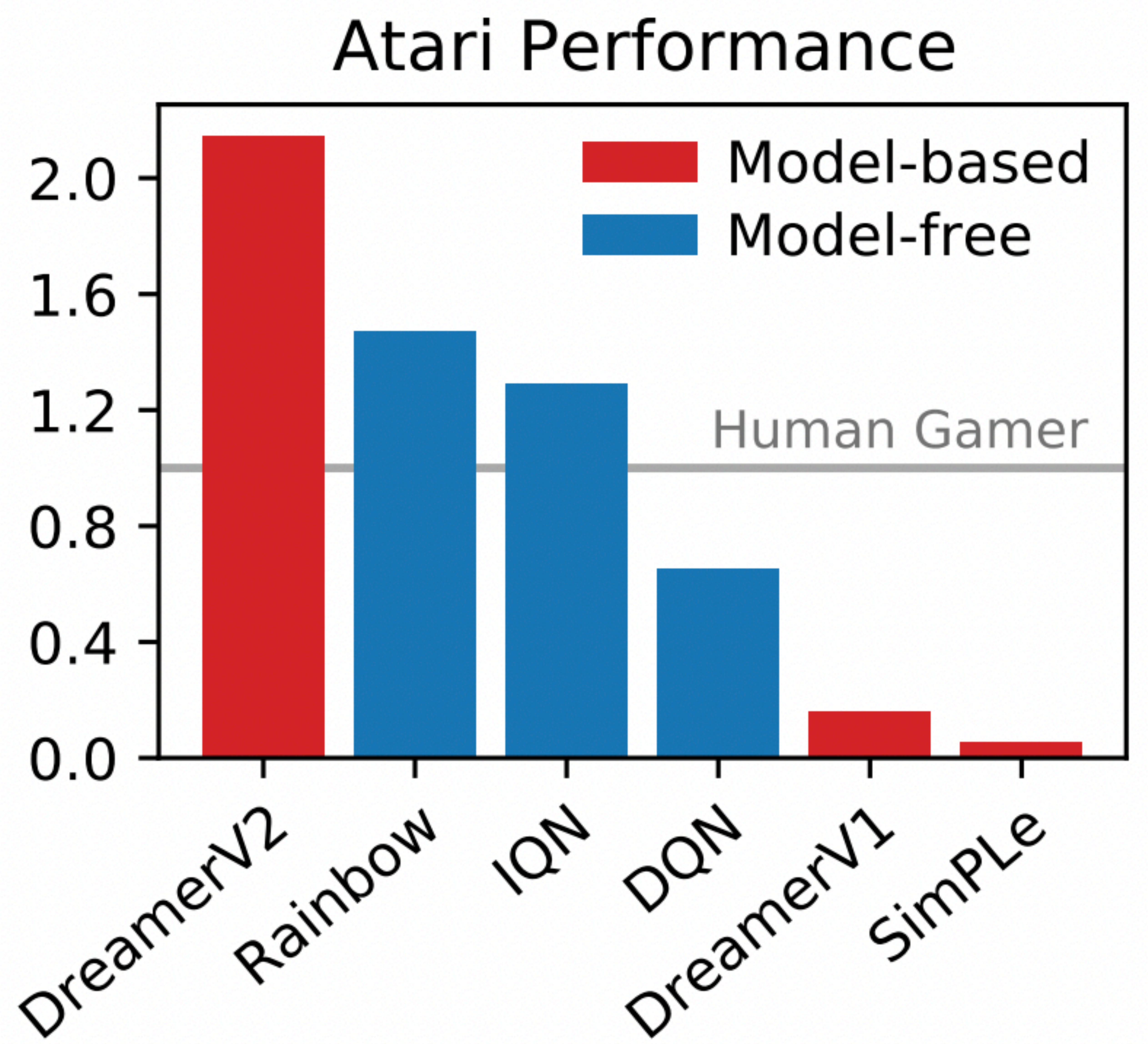
Jimmy Ba
University of Toronto

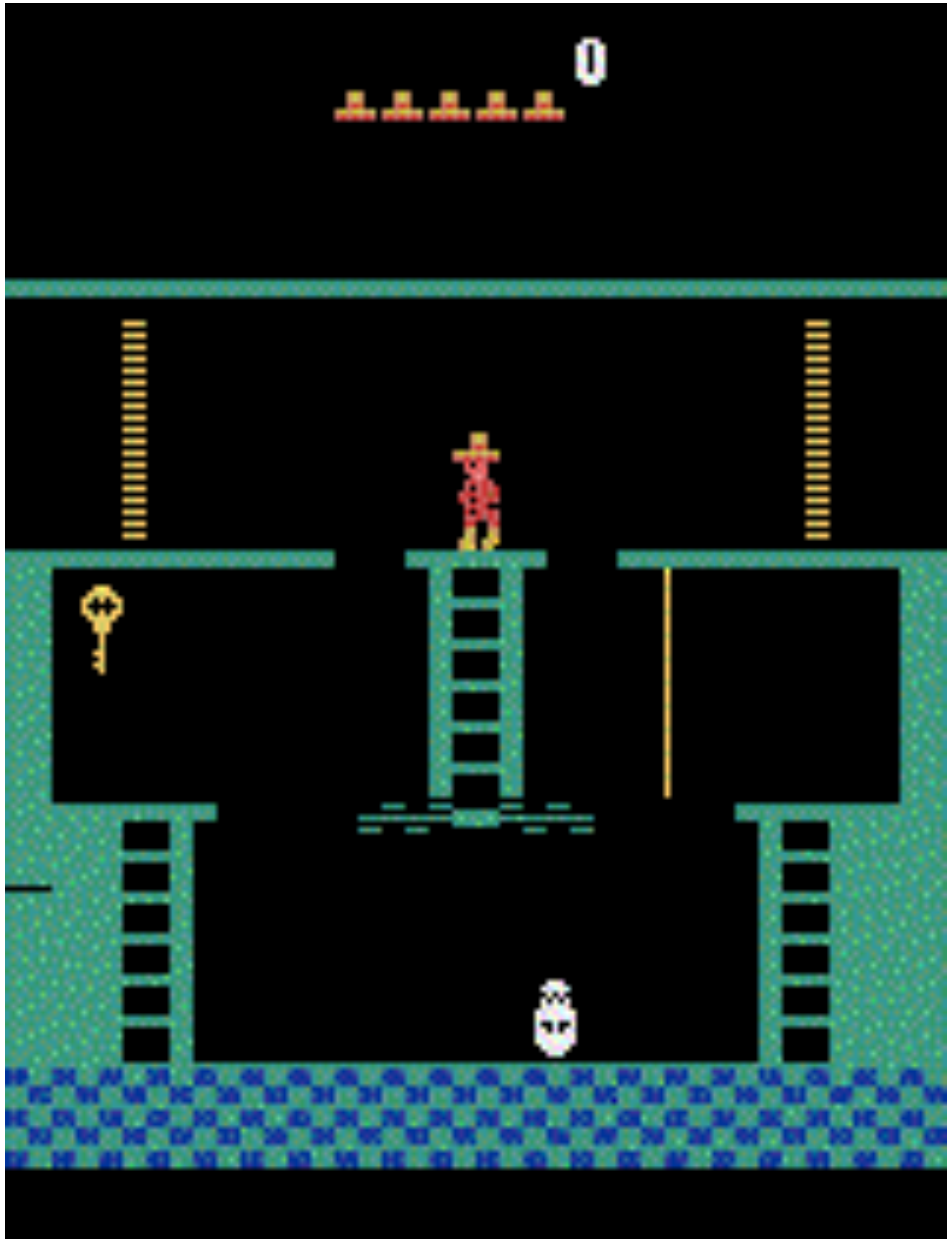


Atari was hard for Model Based RL



DreamerV2 beats all model free!



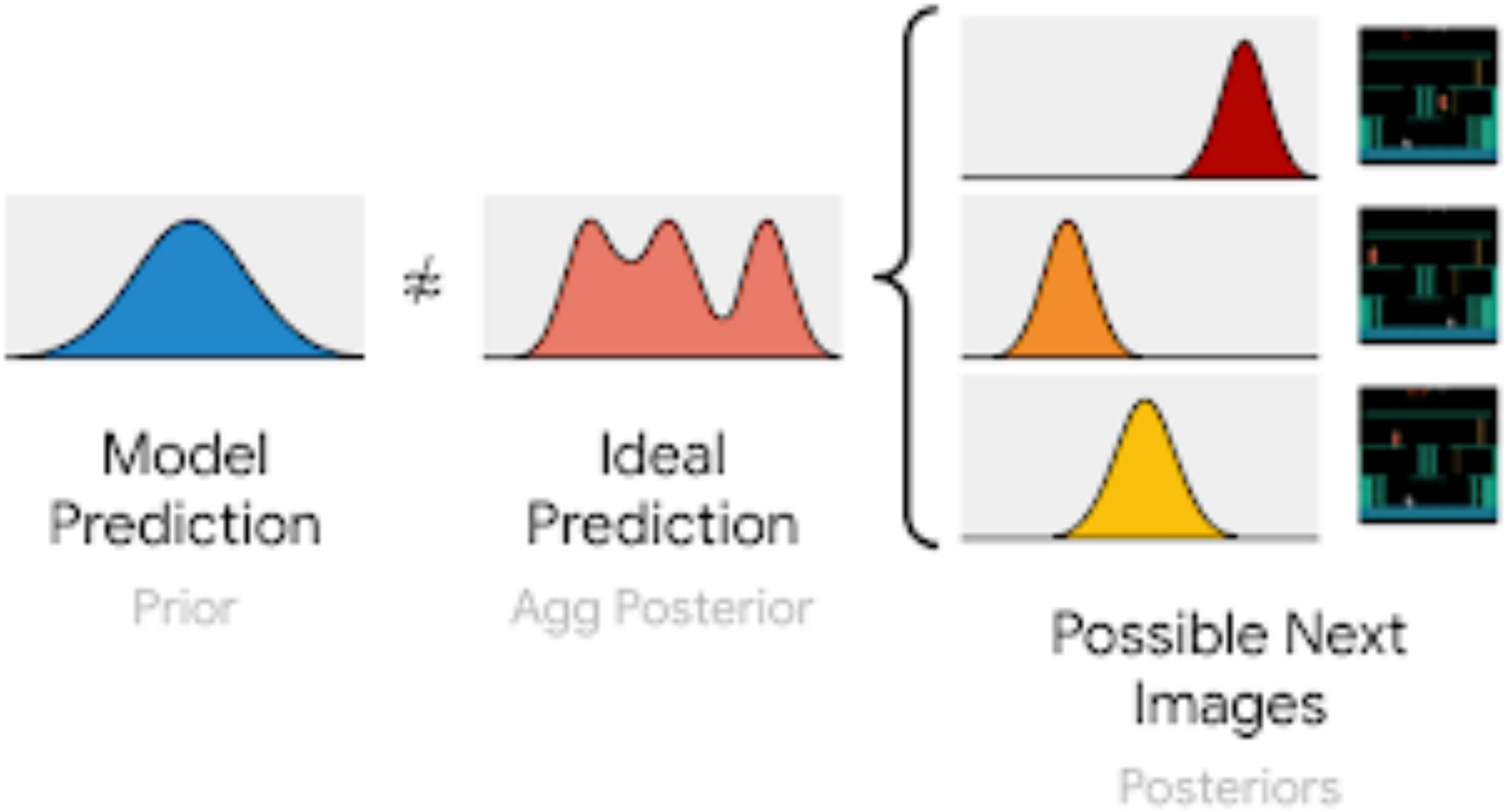


Montezuma's Revenge:

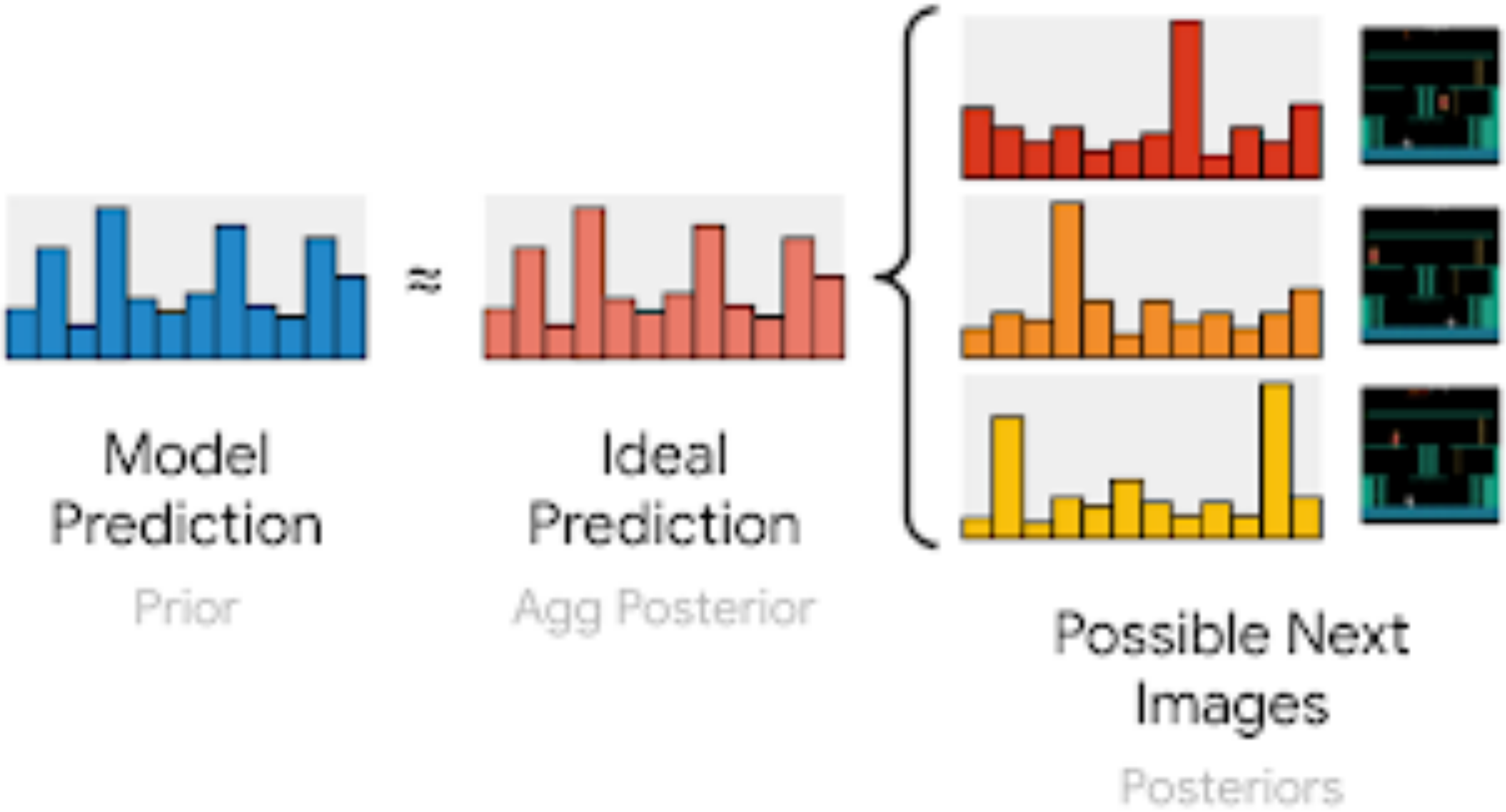
A really challenging
Atari Game!

Challenge: Dreamer V1 predicts a single mode of dynamics

Dreamer V1 predicts single mode dynamics

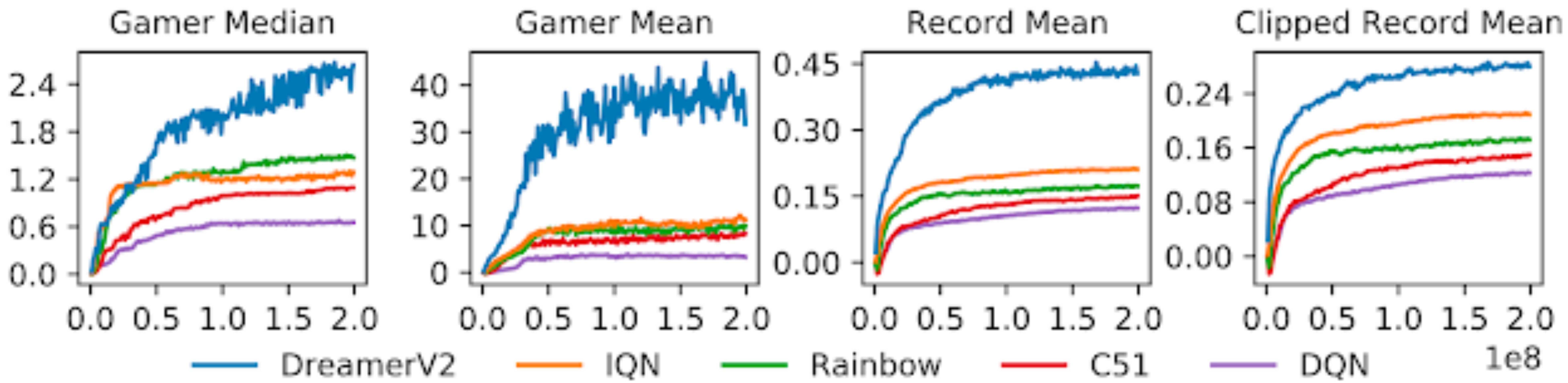


Idea: Predict multiple discrete modes!



$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
a_1
a_2
h_1
h_2
h_3
$\hat{z}_1$
z_1
$\hat{z}_2$
z_2
$\hat{z}_3$
z_3
min KL
min KL
min KL
x_1
x_2
x_3
Sparse Representation
32 categoricals
32 classes each


True
Model


Gamer Median
Gamer Mean
Record Mean
Clipped Record Mean
DreamerV2
IQN
Rainbow
C51
DQN
1e8