Model-based Reinforcement Learning

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Do not worry about prelims!

We will grade on a curve (and try to be lenient!)

It's only 20% of the grade

Instead focus on final project (20%) and in-class quiz (10%)
[This is totally in your control to get full marks!]

Final Project

Will be posting doc with full instructions + partner finding today

Deliverables: Project proposal (in 2 weeks), final project, video, peer review

Groups of upto 3

Expectations (next slide)

Final project expectations

We have the following expectations for the final project:
* You should apply a concept that you learned in class
* Your project should be well-scoped so it is achievable in the short time that you have
* Your project should have some degree of science. It can't just be "I tried X algorithm on Y environment and here are the results".
* Instead, clearly state a hypothesis that you are testing

E.g. We expect warm starting RL from BC in this environment would lead to faster convergence because it reduces the need to explore.

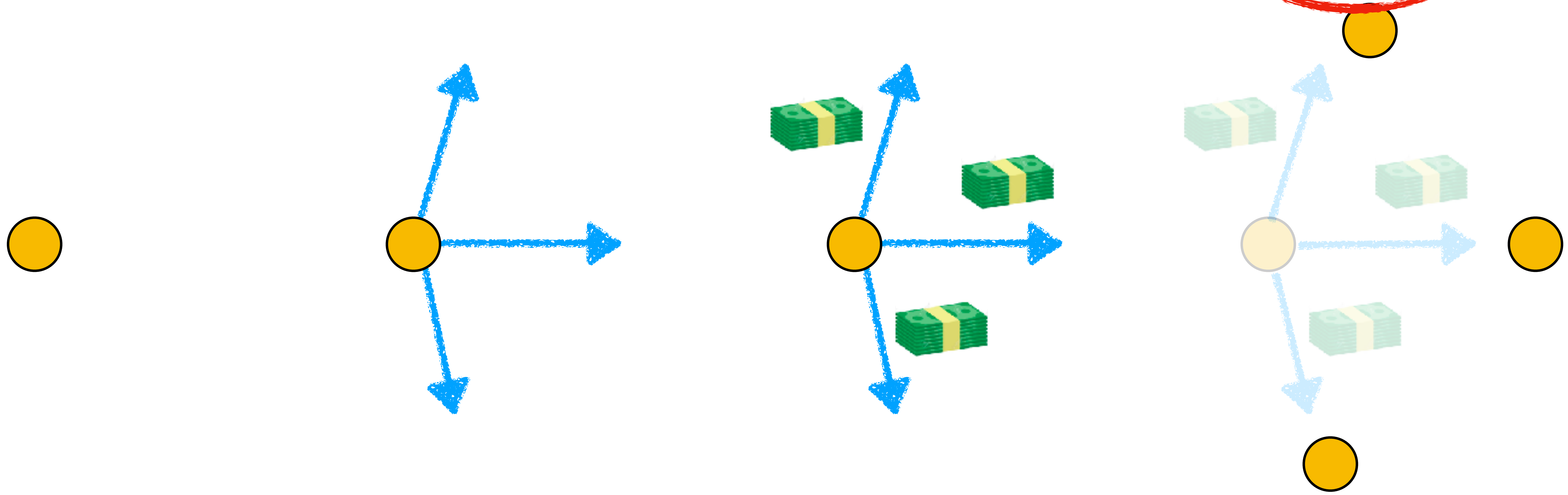
Today's class

- ☐ What is model-based RL?
- ☐ How NOT to learn a model?
- ☐ DAgger for model-based RL

How do we solve the MDP if the transition model is unknown?

Unknown Model

$$< S , A , C , \mathcal{T} >$$



All the RL you learned up until now is Model-Free RL

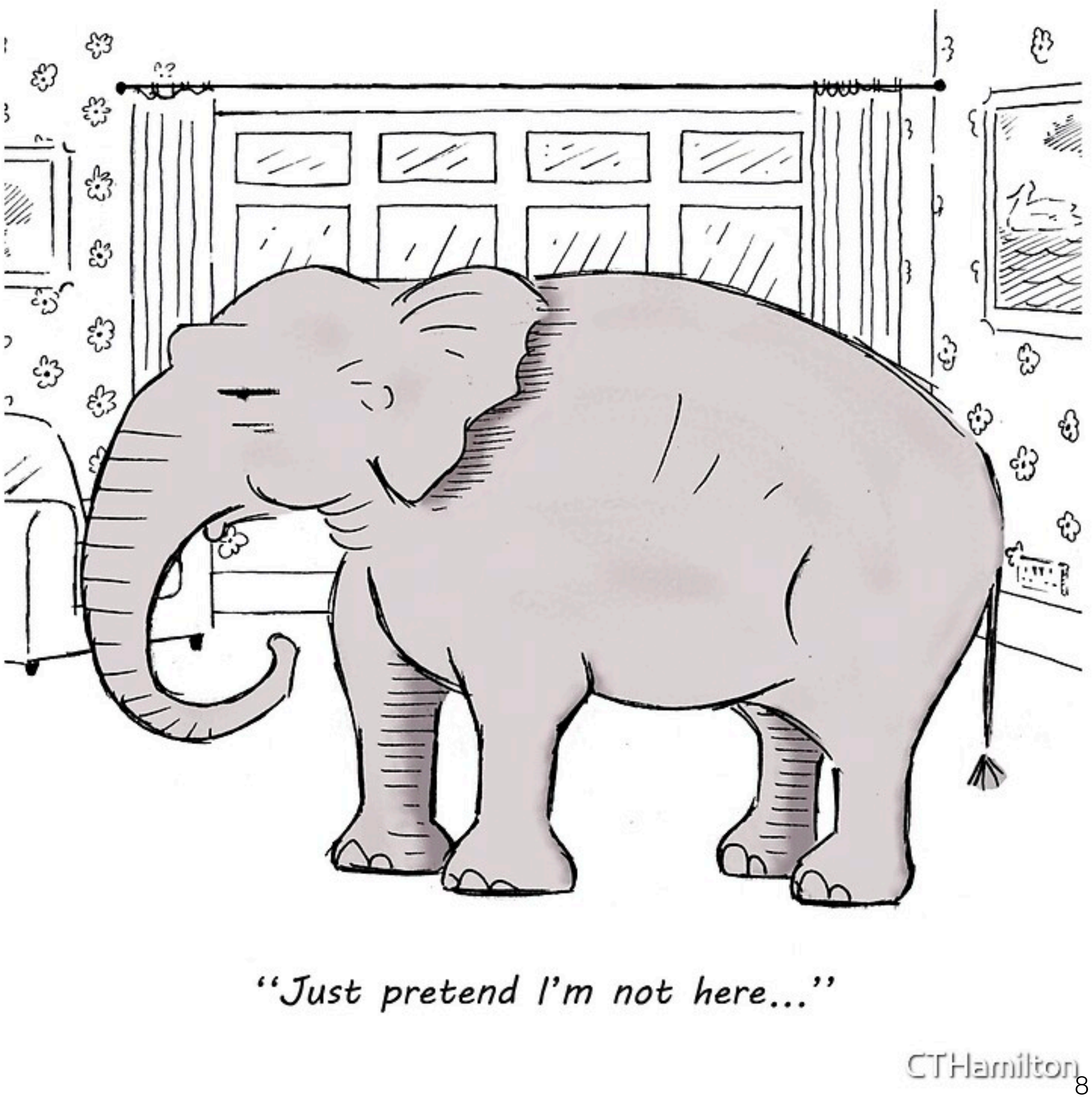
RL

=

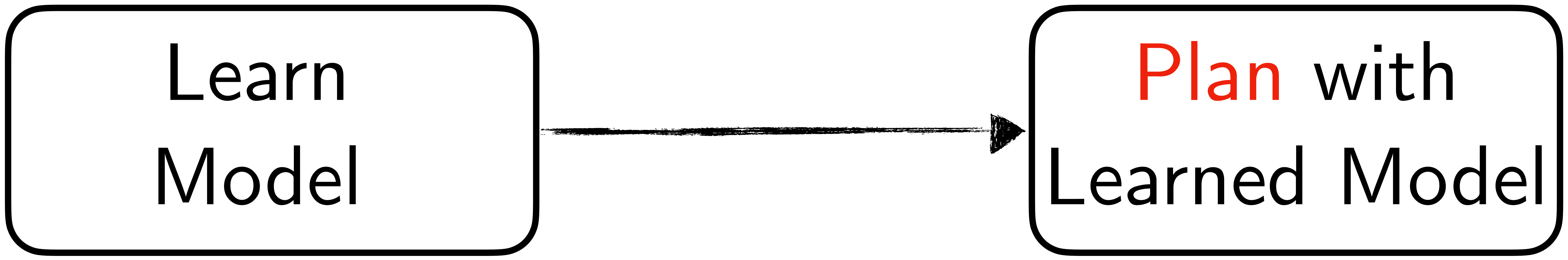
Learn model

+

Plan with model



Model Based Reinforcement Learning



Use *any* planning algorithm you like:

Value iteration / Policy iteration / LQR / A* search / RRTs ...

What is a model?

What is a model?

s_t , a_t

Model
$\hat{M}(s_t, a_t)$

What is a model?

$s_t \, , \, a_t$

$\downarrow$

Model

$\hat{M}(s_t, a_t)$

$\downarrow$

s_{t+1}

Models are simulators of the real world dynamics

Why model-based RL?

What happens if we run model-free RL in the real world?

All practical robotics rely on evaluating policies in a simulator before deploying

Models work very well in *theory*

Model-Based Reinforcement Learning with a Generative Model is Minimax Optimal

Alekh Agarwal
Microsoft
alekha@microsoft.com

Sham Kakade
University of Washington
sham@cs.washington.edu

Lin F. Yang
University of California, Los Angeles
linyang@ee.ucla.edu

April 7, 2020

Models work in *practice*

Hafner et al. 2023

Learning Models.

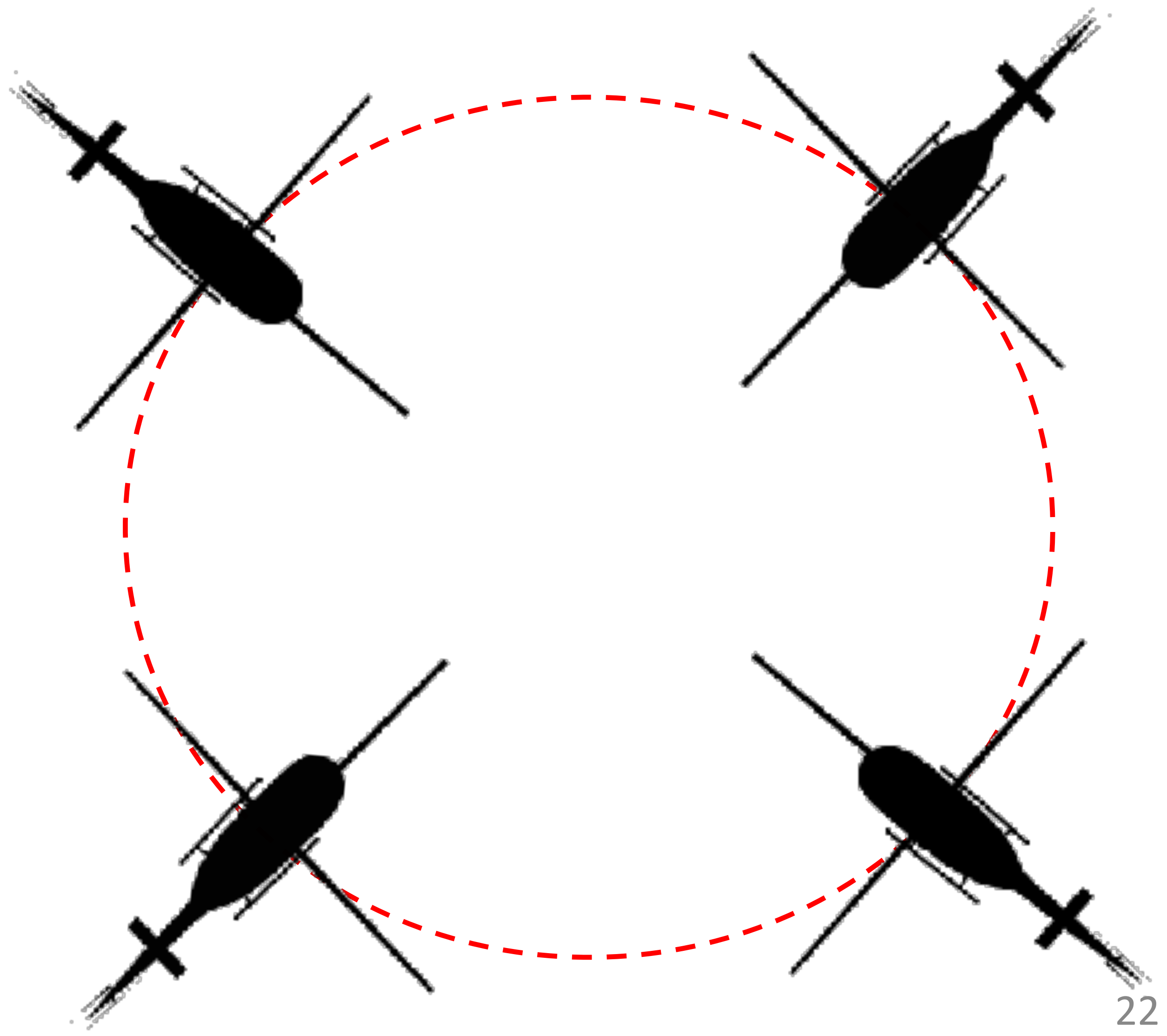
What is the machine learning setup for learning models?

Goal: We want to fit a model $\hat{M}(s, a)$
to the real world dynamics $M^*(s, a)$

What is the Input? Output? Loss? Model Class? Data?

Activity!

Example: Helicopter Aerobatics



(Super cool work by Pieter Abeel et al. https://people.eecs.berkeley.edu/~pabbeel/autonomous_helicopter.html)

Suppose I want to learn a helicopter model. What is input / output / loss?

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

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

Think-Pair-Share

Think (30 sec): What model will you use for learning?
What planner would you use to execute a maneuver?

Pair: Find a partner

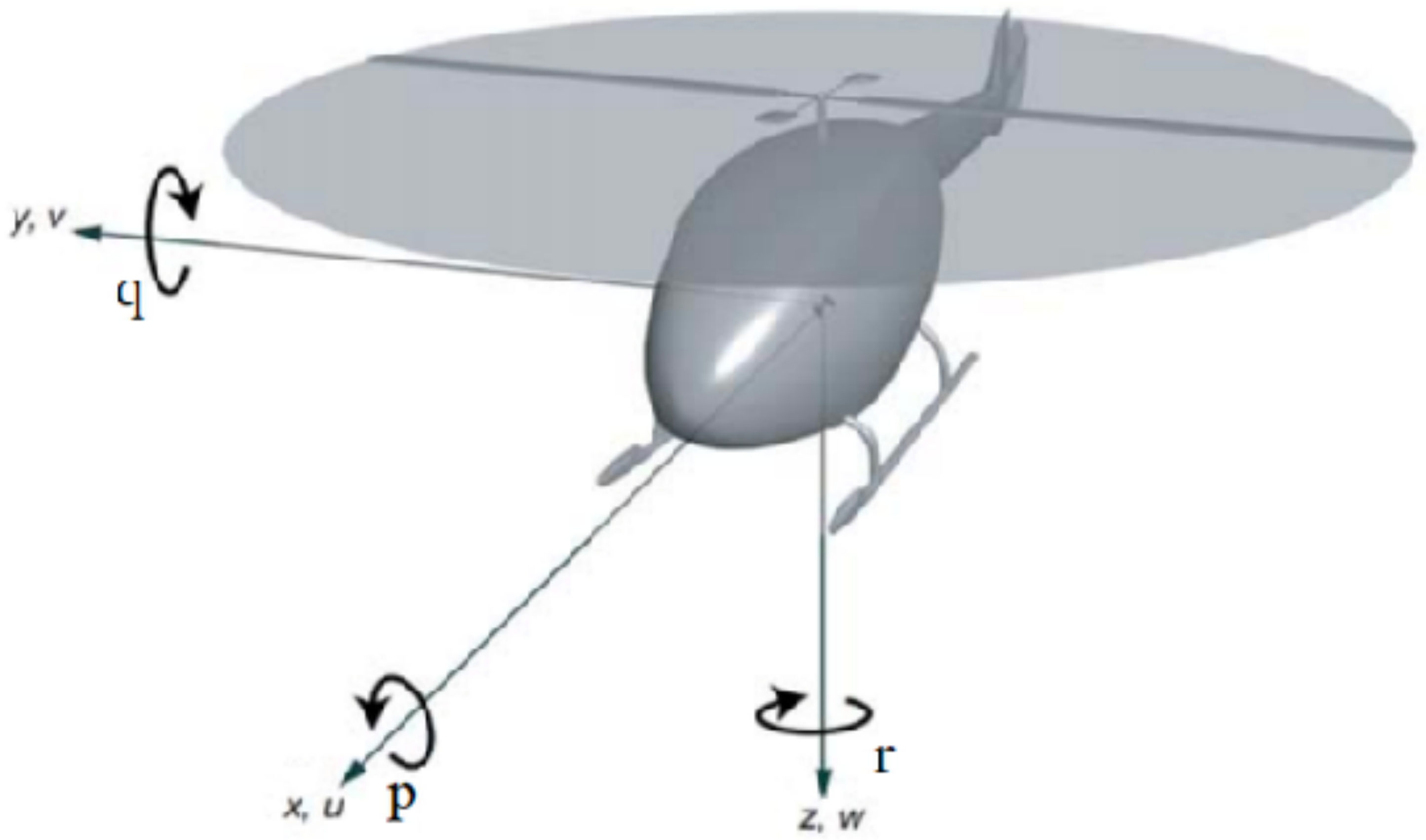
Learn Model → Plan with Learned Model

Share (45 sec): Partners exchange ideas

What did Abeel and Ng do?

Part 1: Learn a Model

Learn Model → Plan with Learned Model

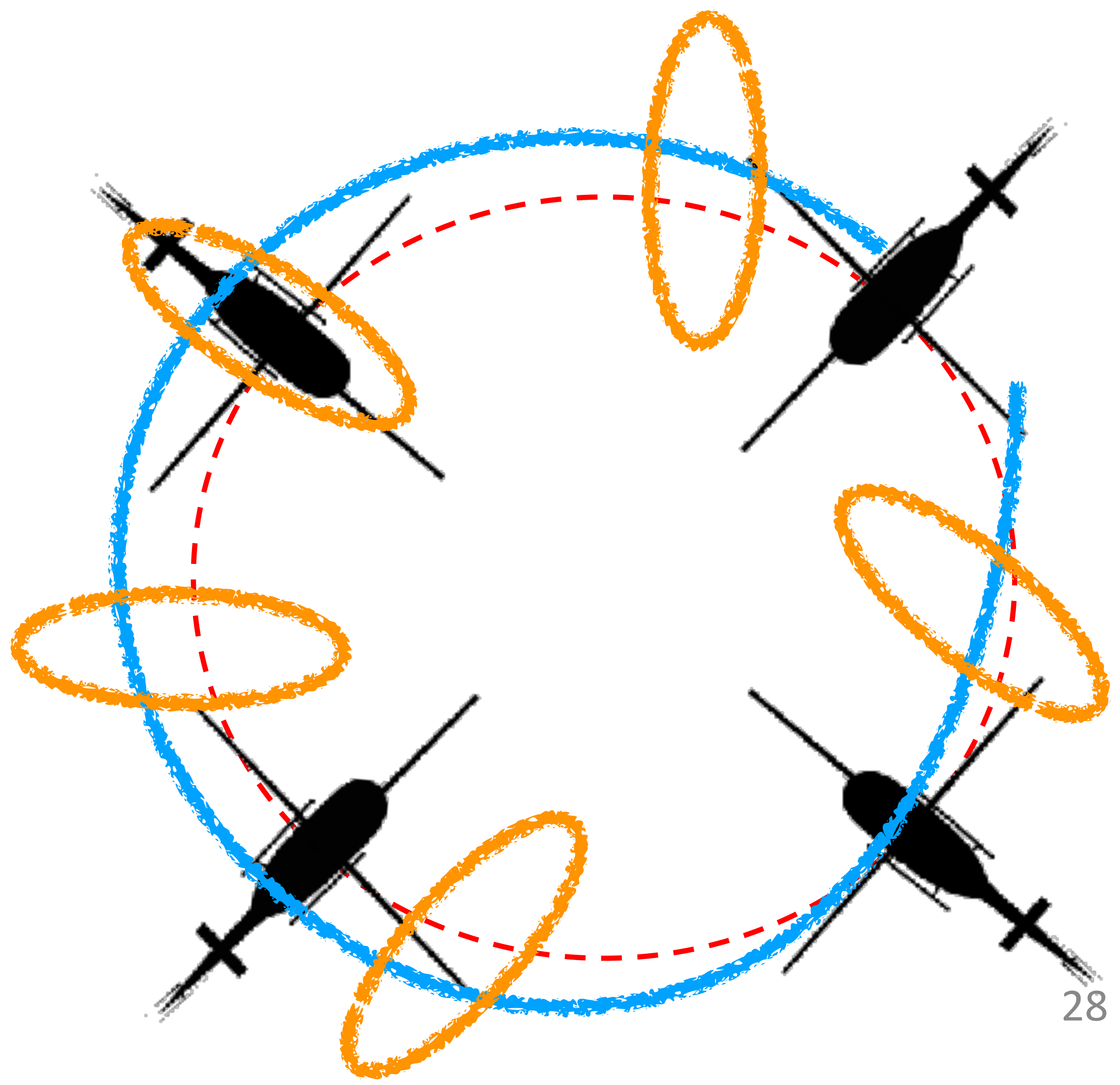


Write down a physics model and learn parameters

$$
\begin{aligned}
\dot{u} &= vr - wq + A_x u + g_x + w_u, \\
\dot{v} &= wp - ur + A_y v + g_y + D_0 + w_v, \\
\dot{w} &= uq - vp + A_z w + g_z + C_4 u_4 + D_4 + w_w, \\
\dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + B_x p + C_1 u_1 + D_1 + w_p, \\
\dot{q} &= pr(I_{zz} - I_{xx})/I_{yy} + B_y q + C_2 u_2 + D_2 + w_q, \\
\dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + B_z r + C_3 u_3 + D_3 + w_r.
\end{aligned}
$$

Part 2: Plan with Learned Model

Learn Model → Plan with Learned Model



Use LQR by *linearizing* the model

Today's class

- ☑ What is model-based RL?

  Learn a model, plan with learned model

- ☐ How NOT to learn a model?

- ☐ DAgger for model-based RL

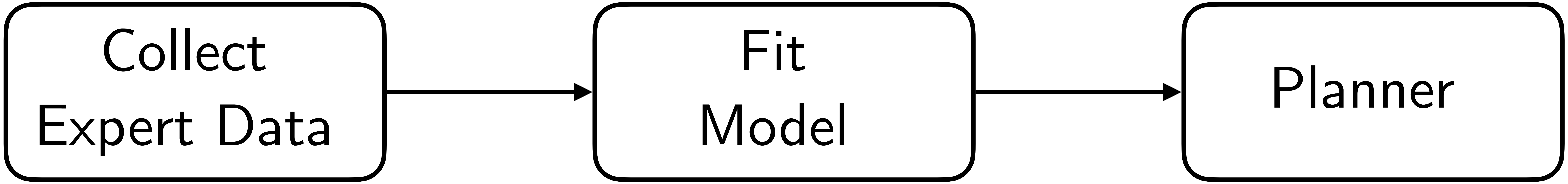
Question: How do you collect data for learning model?

(Super cool work by Pieter Abeel et al. https://people.eecs.berkeley.edu/~pabbeel/autonomous_helicopter.html)

Strategy

Train a model on state actions visited by the expert!

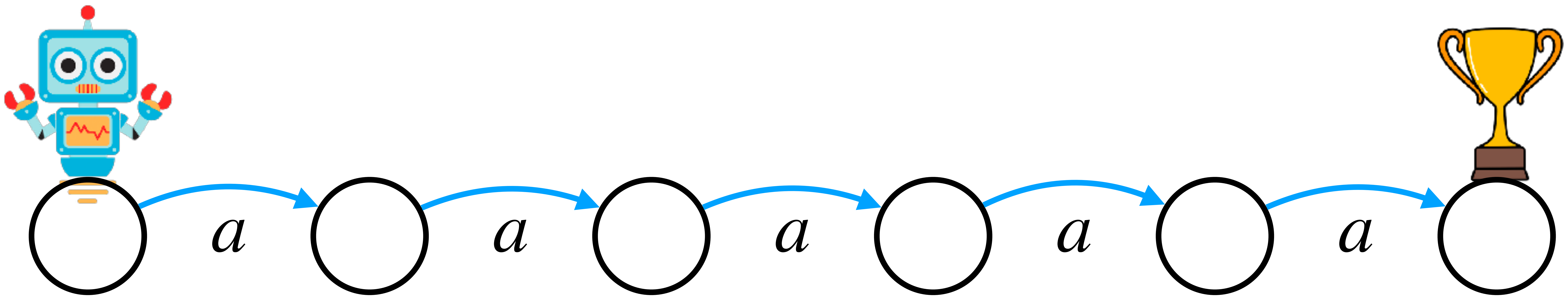
Model Based RL v1.0



*If I **perfectly** fit a model (i.e. training/validation error zero), this should work, right?*

World

s'$=M^*(s, a)$



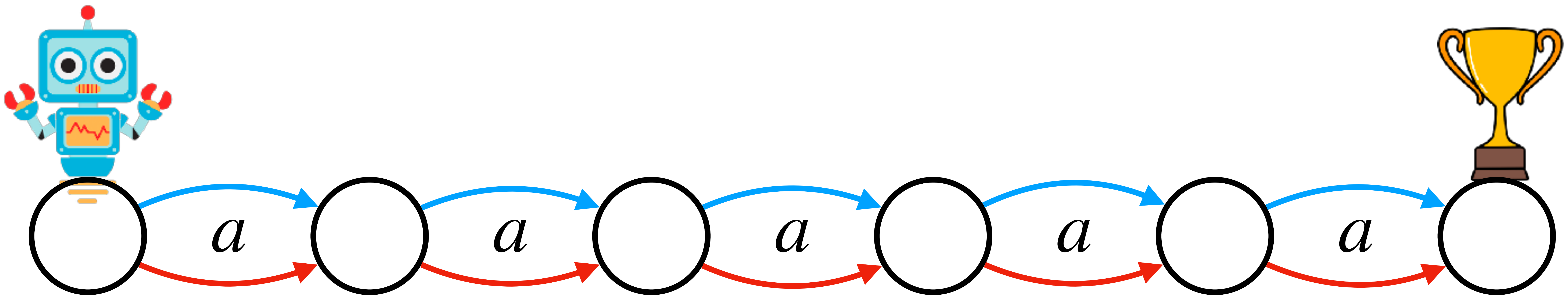
Experts picks action a to go to the goal

Model

$s'=\hat{M}(s,a)$

World

$s'=M^*(s,a)$



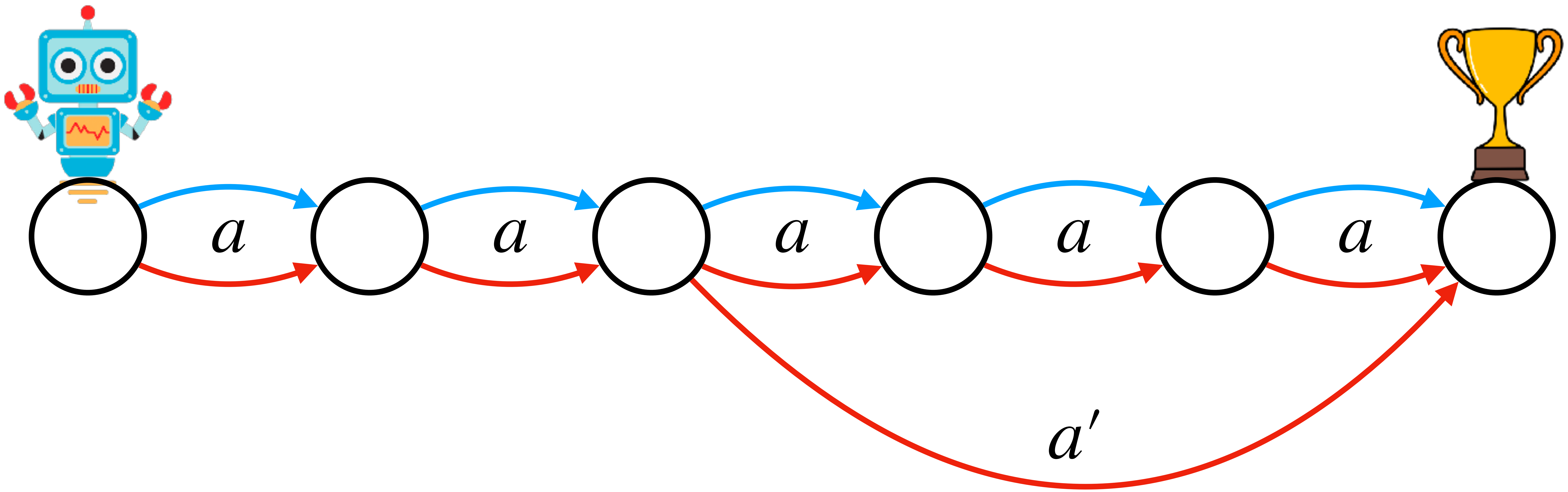
Model agrees with world, i.e. train error zero!

Model

s'=$\hat{M}(s, a)$

World

s'=$M^*(s, a)$

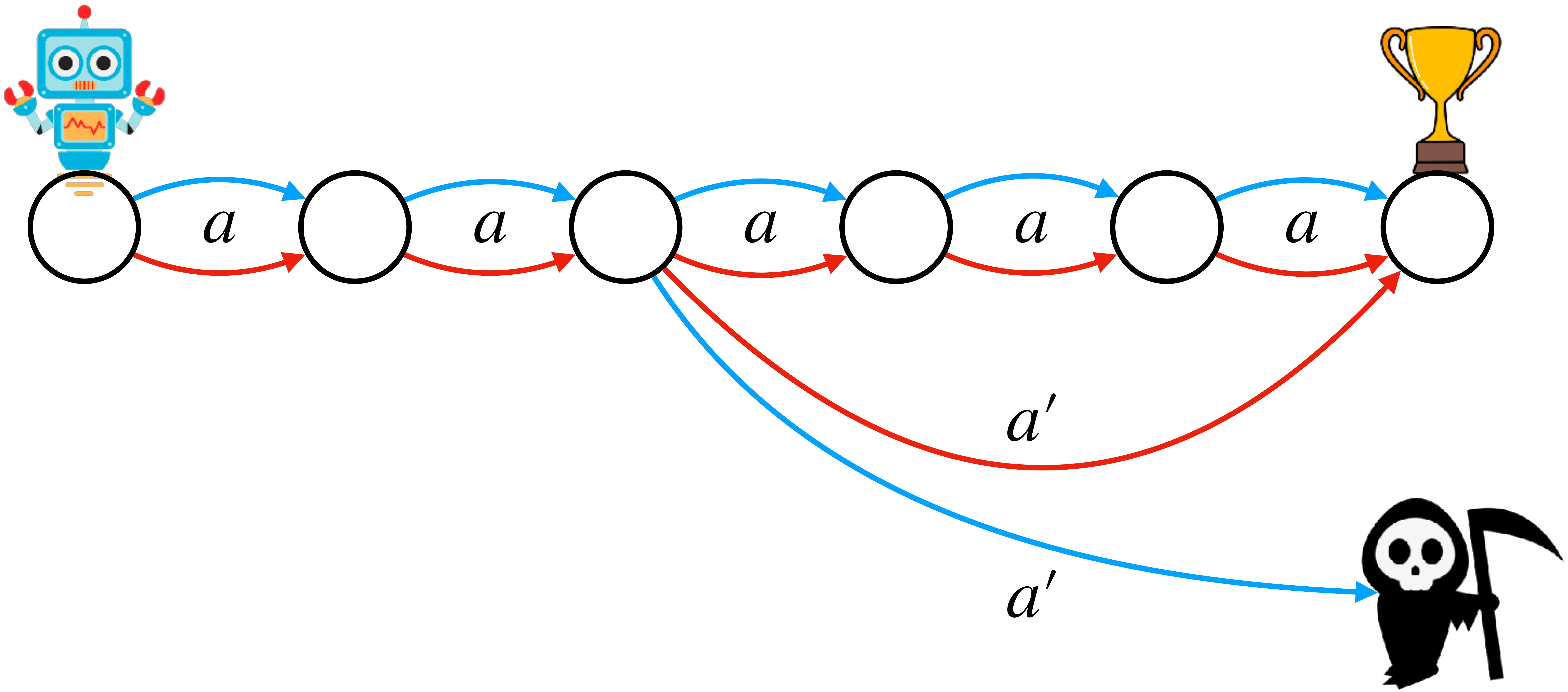


What if the model is optimistic?

Predicts a short cut to the goal by taking action a'

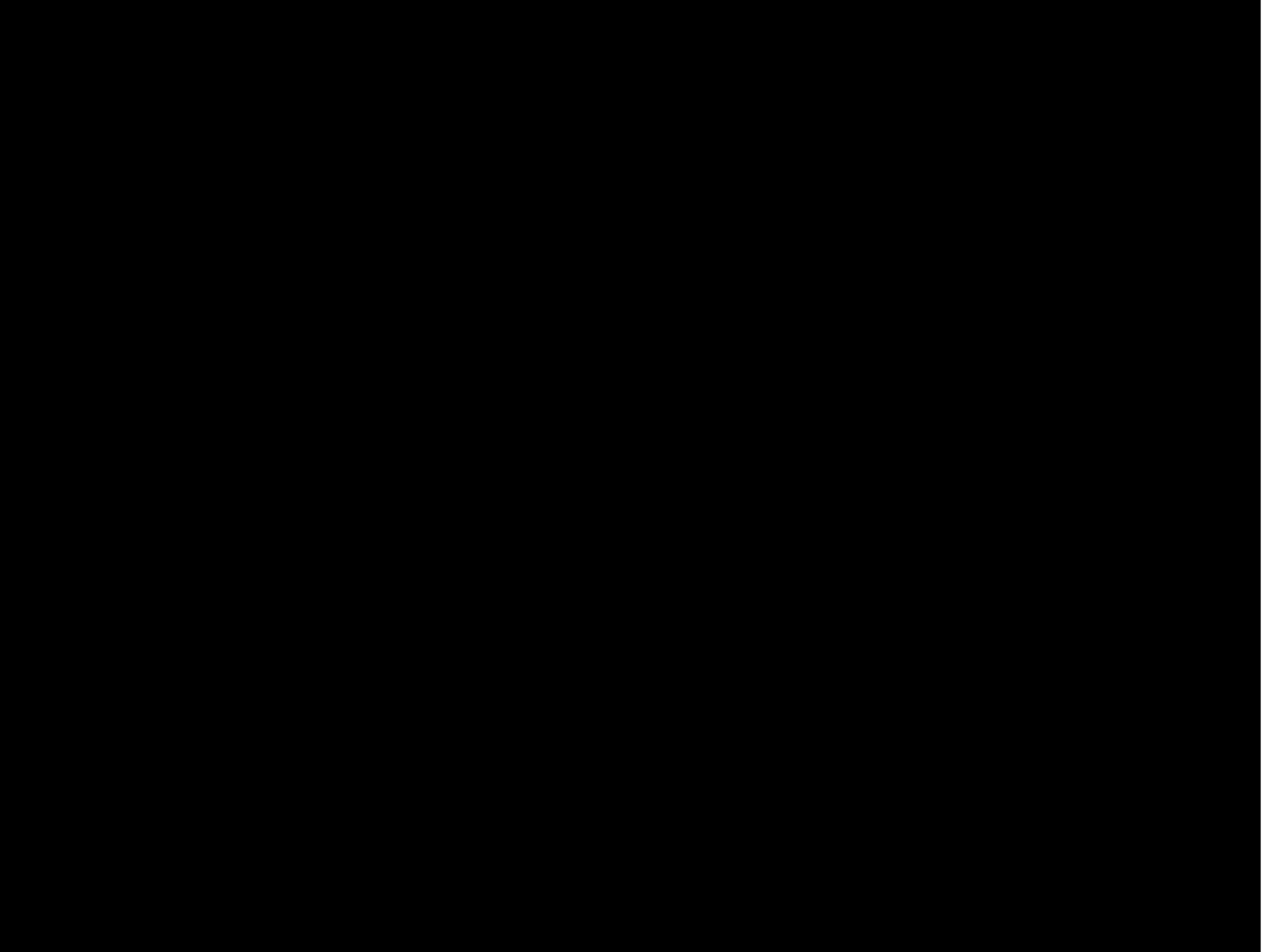
Model $s' = \hat{M}(s, a)$

World $s' = M^*(s, a)$



In reality the shortcut ends in death ...

Training just on expert data may result in *optimistic* models



Training on Expert Data

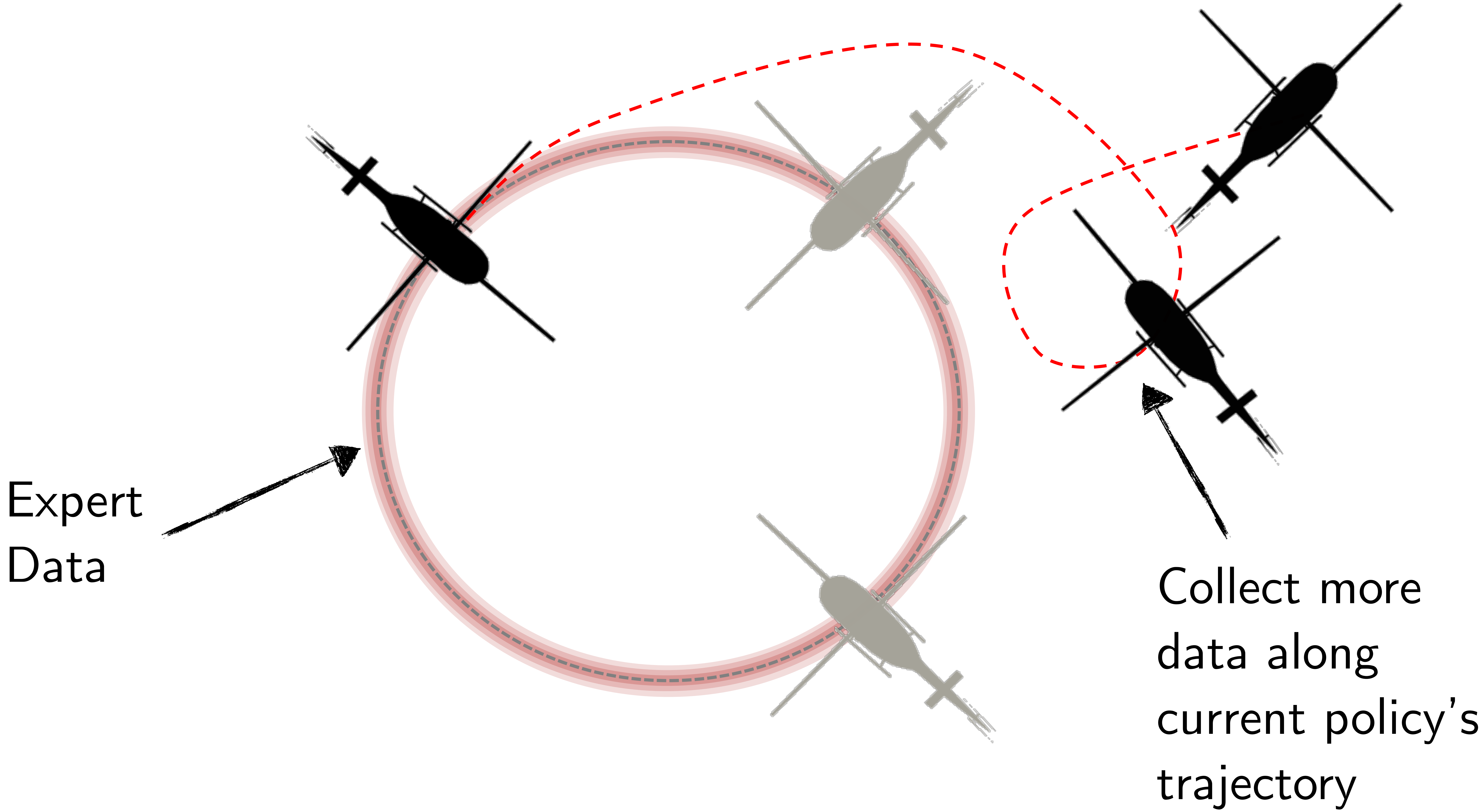
(From Ross and Bagnell, 2012)

Strategy

~~Train a model on state actions visited by the expert!~~

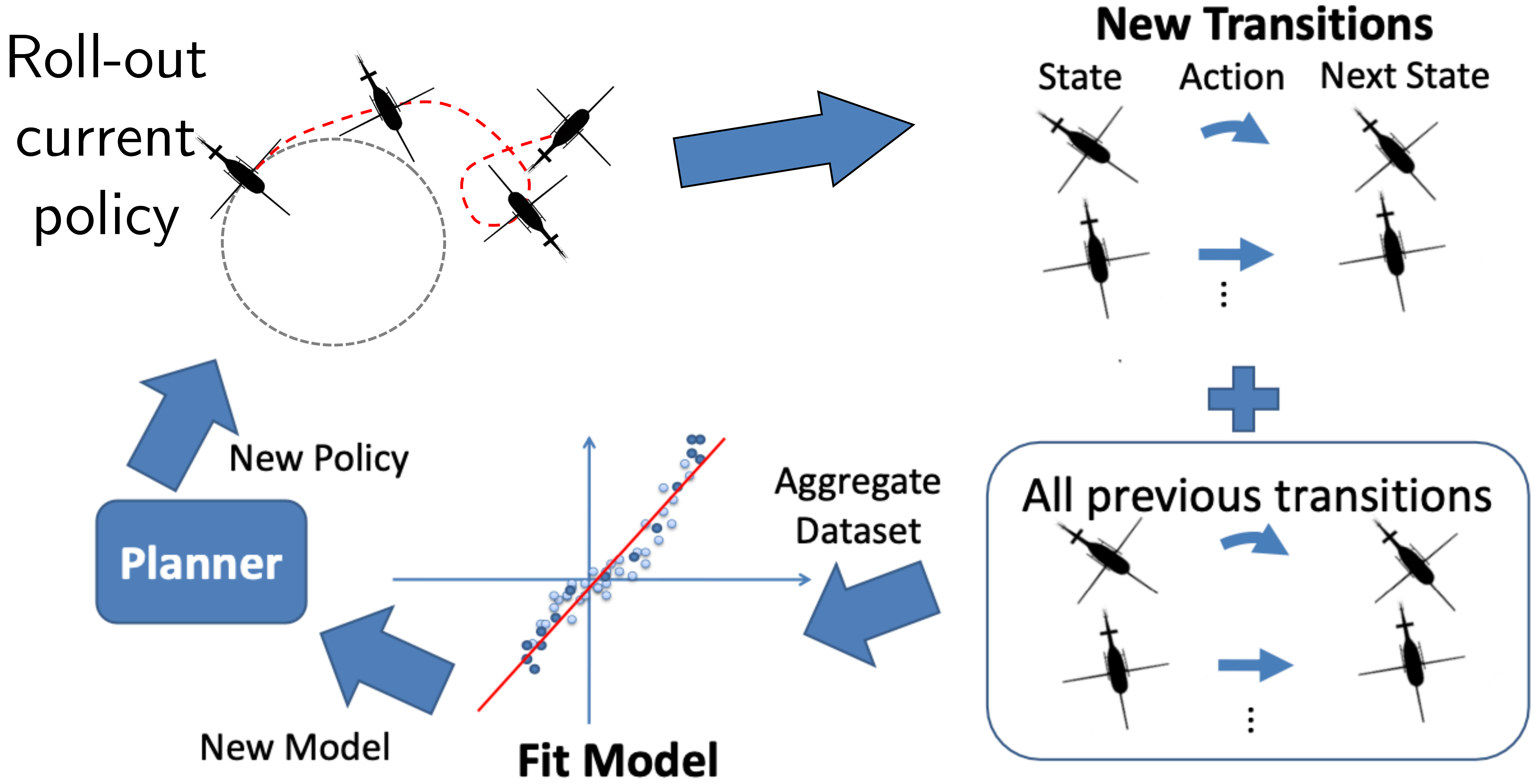
Train a model on state actions visited by the learner!

Improve model where policy goes

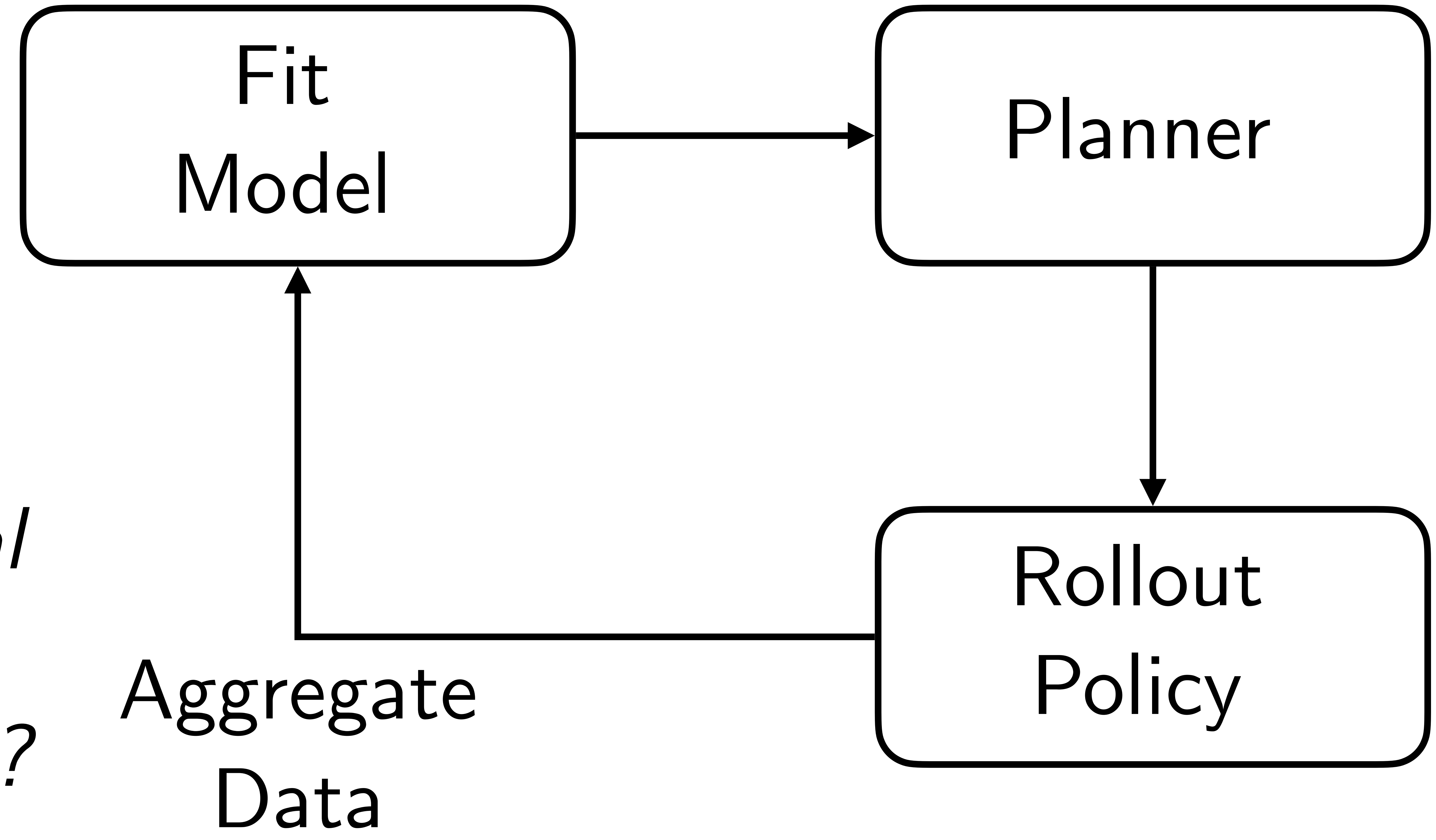


Don't we know an algorithm that does this?

DAGGER for Model-based RL!!



Model Based RL v2.0



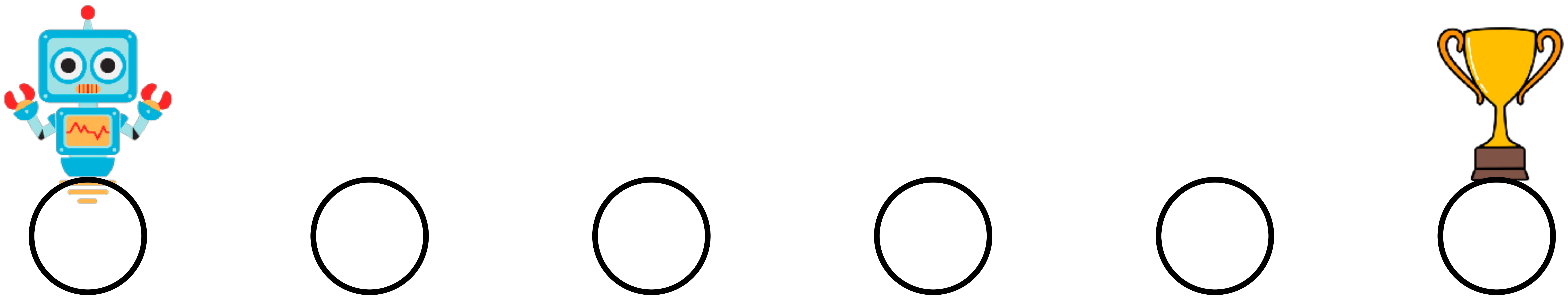
*If I **perfectly** fit a model (i.e. training/val error zero), this should work, right?*

Model

$s' = \hat{M}(s, a)$

World

$s' = M^*(s, a)$

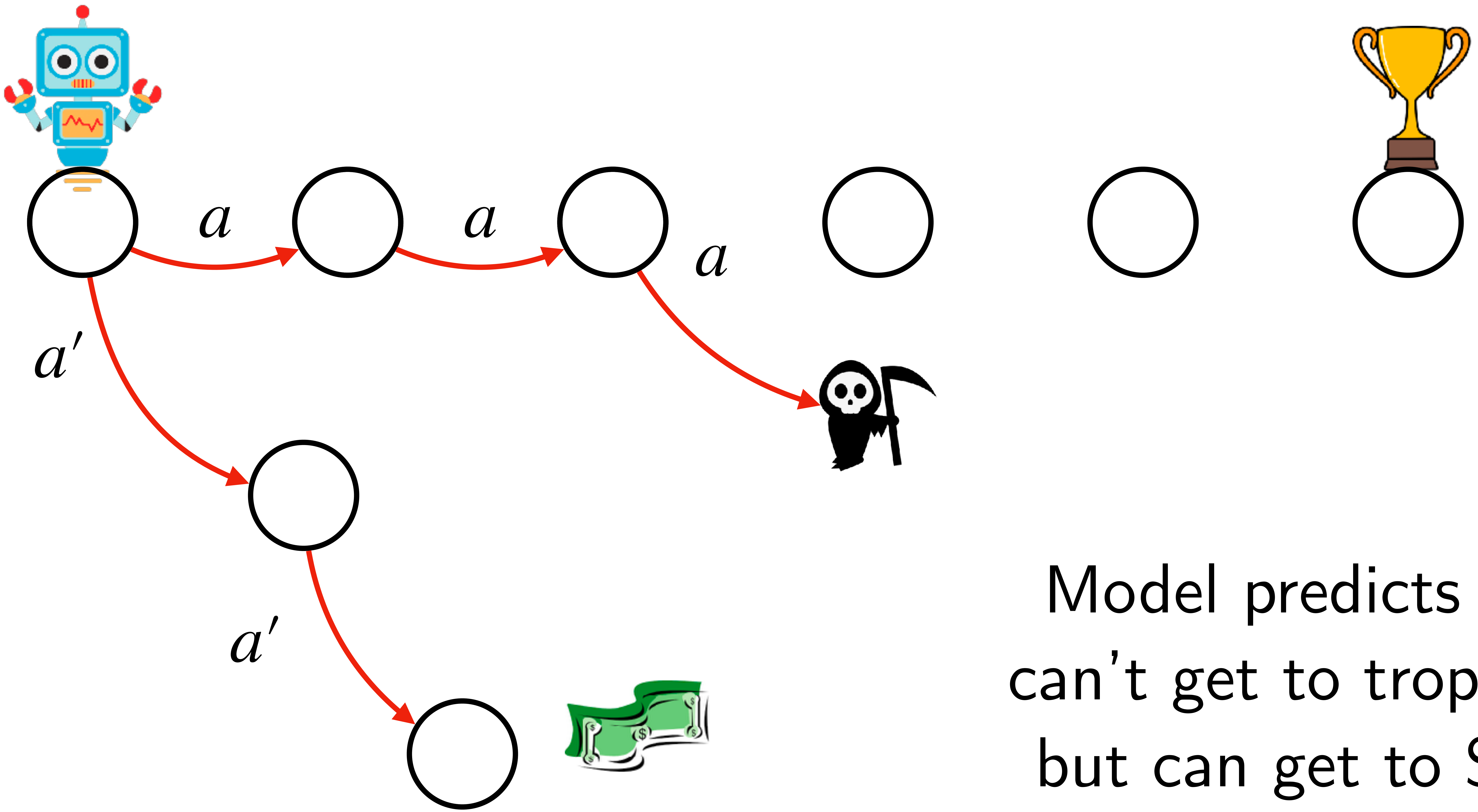


Model

$s' = \hat{M}(s, a)$

World

$s' = M^*(s, a)$



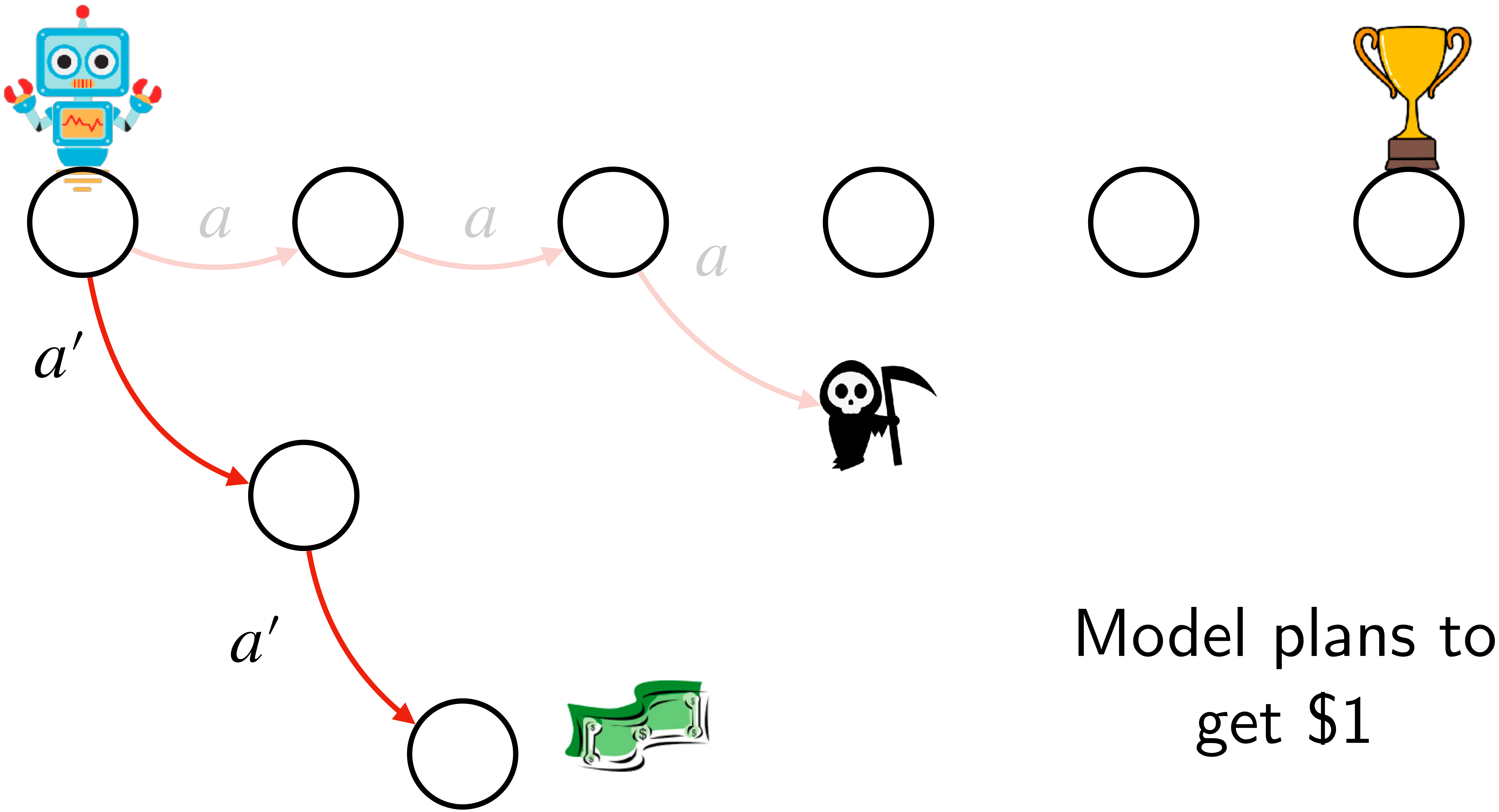
Model predicts it can't get to trophy, but can get to $1

Model

s'$=\hat{M}(s, a)$

World

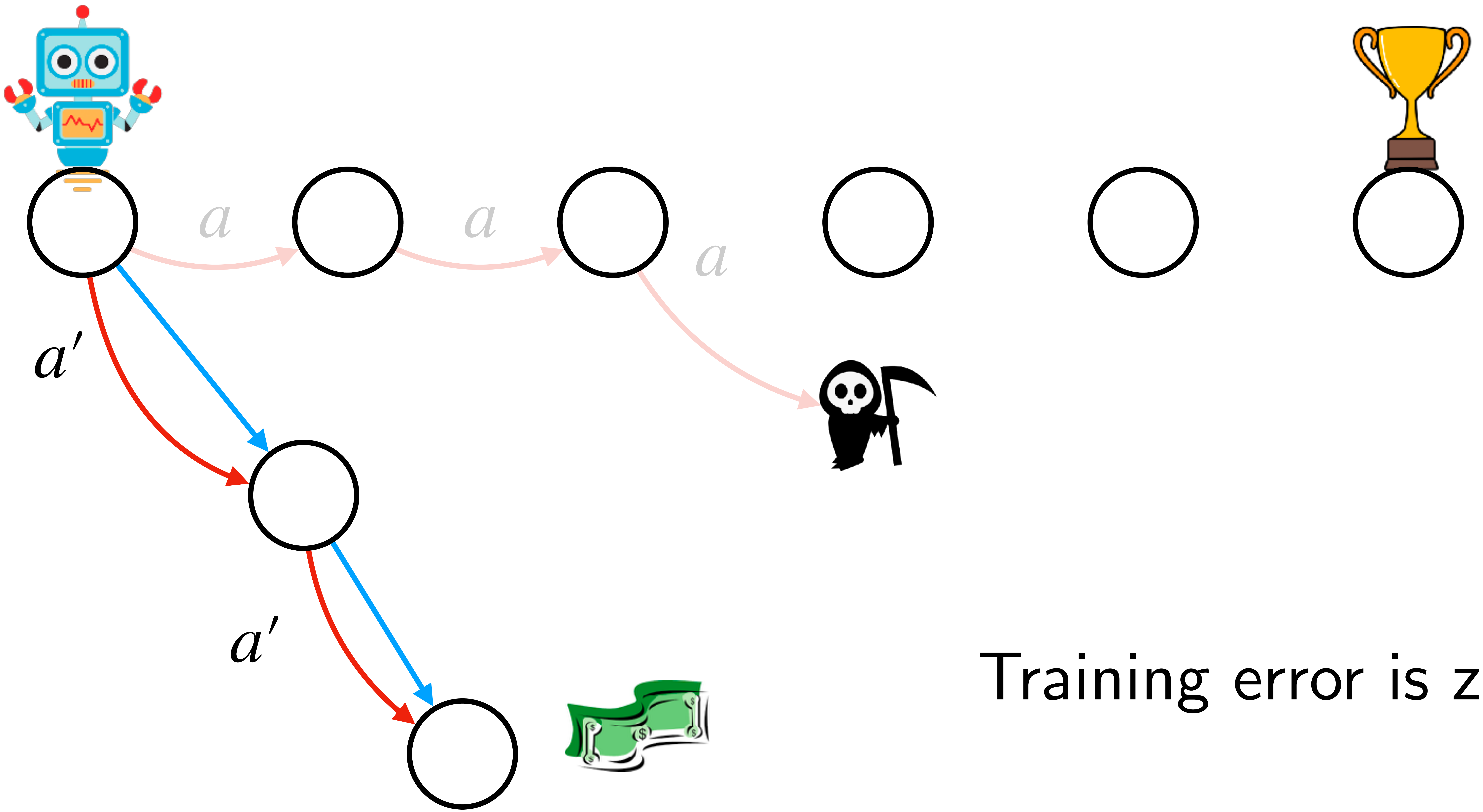
s'$=M^*(s, a)$



Model plans to get $1

Model $\quad$ World

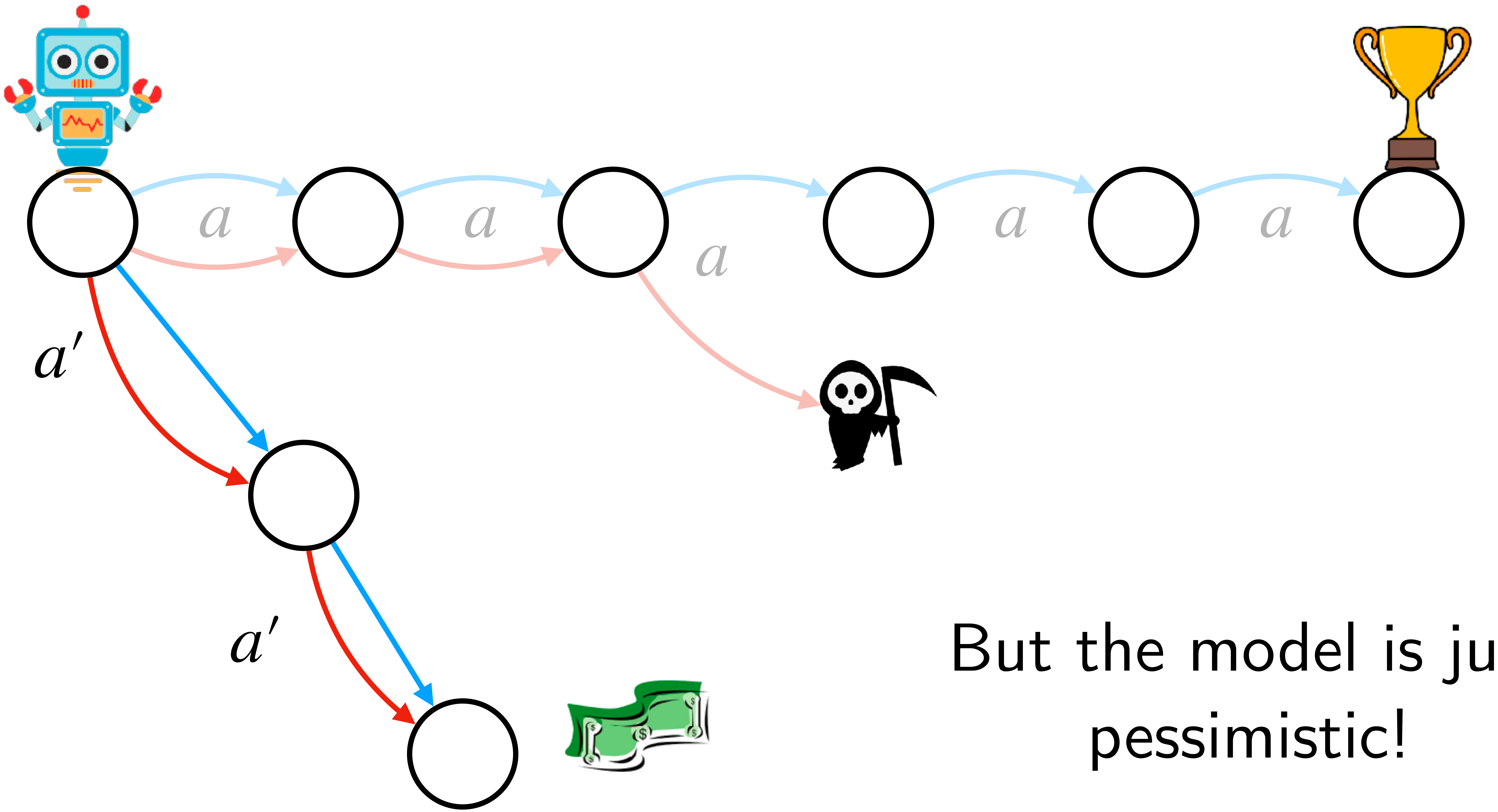
$s' = \hat{M}(s, a)$ $\qquad$ $s' = M^*(s, a)$



Training error is zero!

Model
$s' = \hat{M}(s, a)$

World
$s' = M^*(s, a)$



But the model is just pessimistic!

Training just on learner data may result in *pessimistic* models

Today's class

- ☑ What is model-based RL?

  Learn a model, plan with learned model

- ☑ How NOT to learn a model?

  Don't train on only **expert** data (too **optimistic**),
  Don't train on only **learner** data (too **pessimistic**),

- ☐ DAgger for model-based RL

Strategy

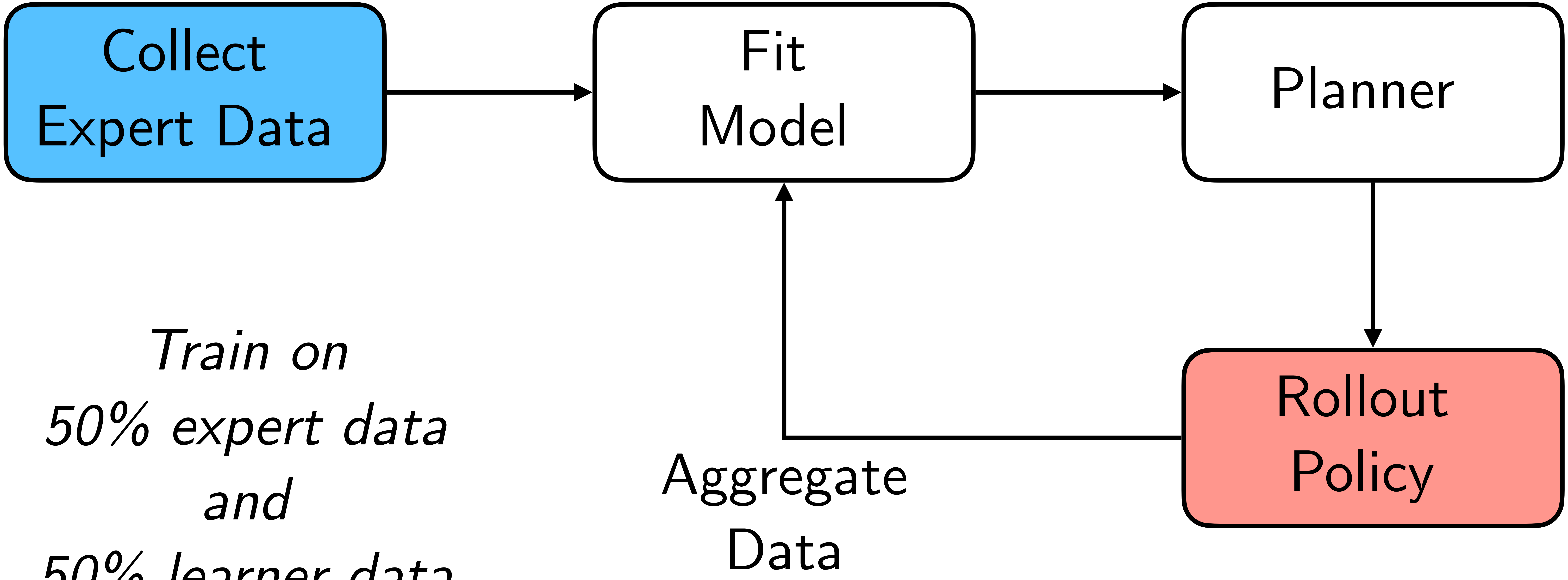
~~Train a model on state actions visited by the expert!~~

~~Train a model on state actions visited by the learner!~~

Train a model on state actions visited by
both the expert and the learner!

Model Learning with Planner in Loop

(Ross & Bagnell, 2012)



Train on
50% expert data
and
50% learner data

Training on 50% learner, 50% expert

(Not too optimistic or pessimistic)

Model Learning with Planner in Loop

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

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

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

For $i = 1, \ldots, N$

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

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

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

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

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

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

Select the best policy in $\pi_{1:N+1}$

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

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

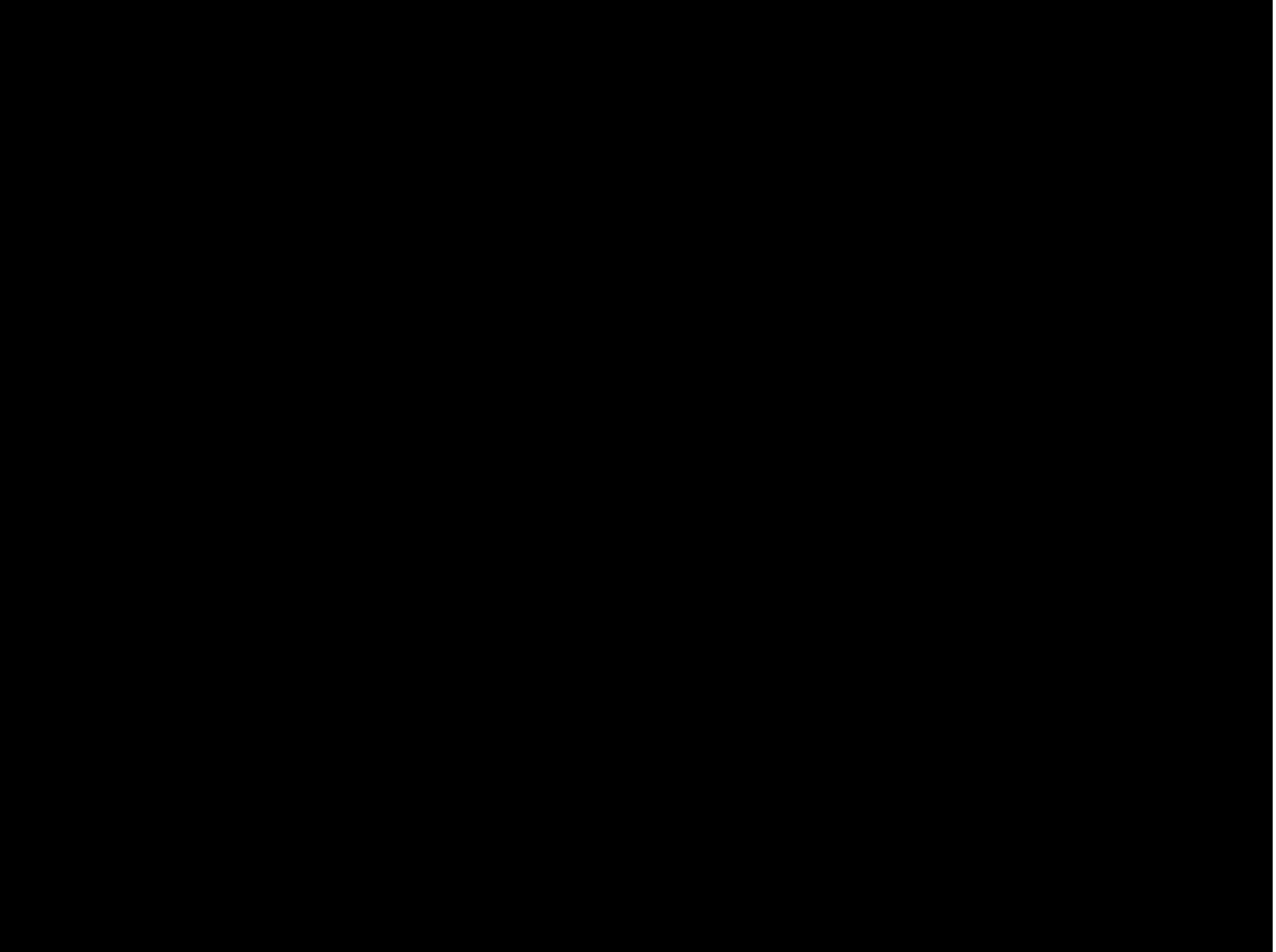
Aggregate data $\mathscr{D}_{\text{learner}} \leftarrow \mathscr{D}_{\text{learner}} \cup \mathscr{D}_i$

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 π_{i+1} in the model $\hat{M}_{i+1}$

Select the best policy in $\pi_{1:N+1}$



Model learning on both expert and learner data works!

(From Ross & Bagnell, 2012)

Today's class

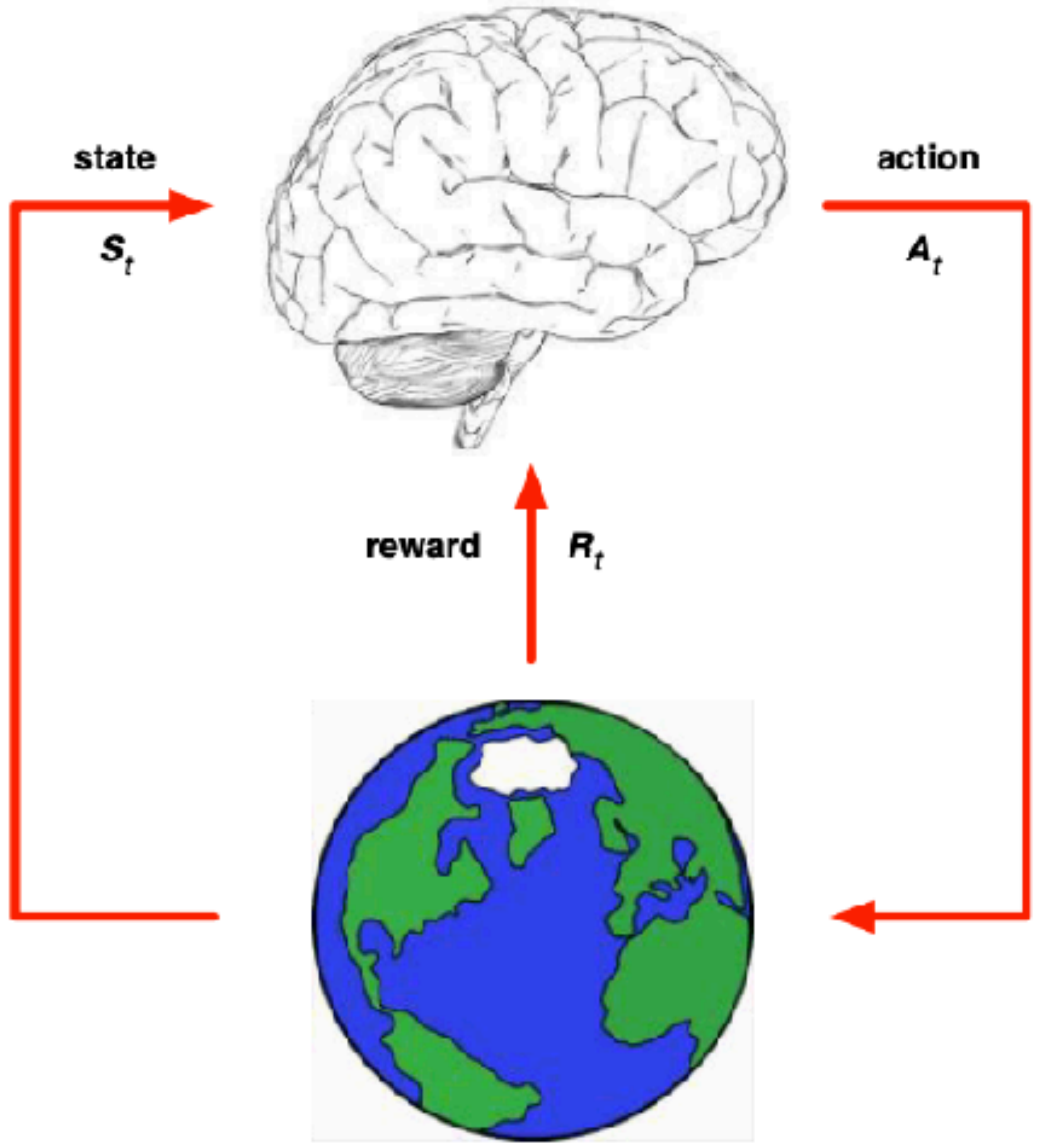
- ☑ What is model-based RL?
- ☑ How NOT to learn a model?
- ☑ DAgger for model-based RL

Why is 50-50 the right thing to do?

Performance Difference via Planning in Model Lemma

What makes for a good model?

Policy $\hat{\pi}$



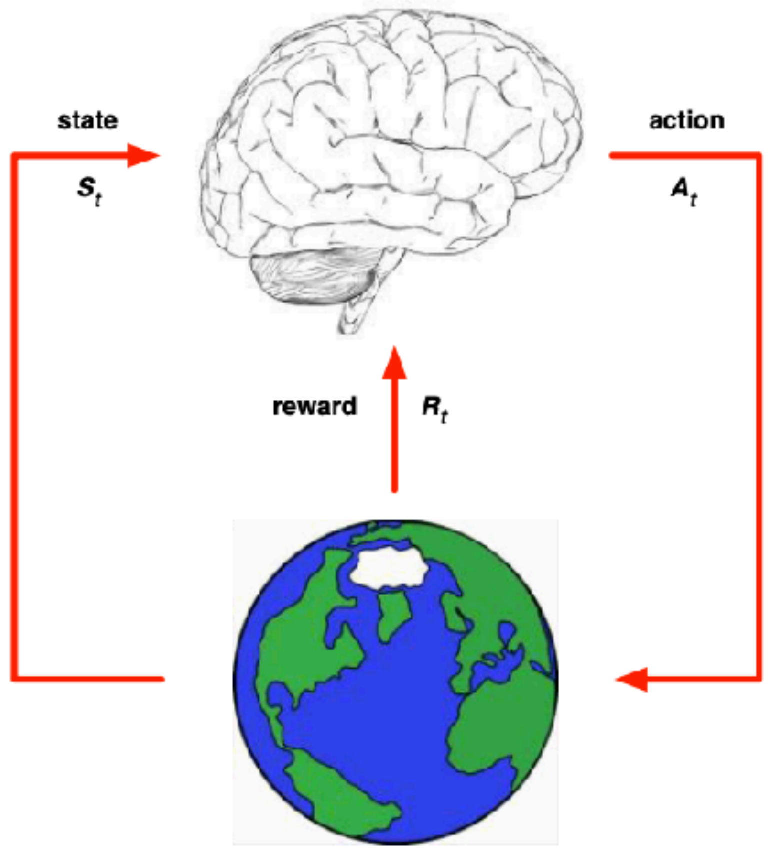
Model $\hat{M}$

A good model is one such that if

we plan with the model

we get a good policy

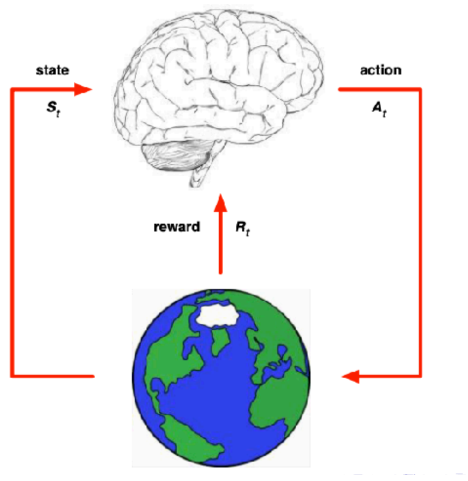
What makes for a good model?



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

A good model gives a good policy that has bounded performance difference

What makes for a good model?

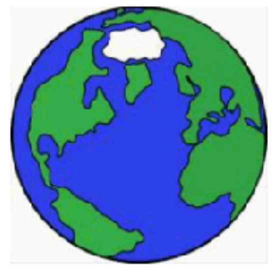
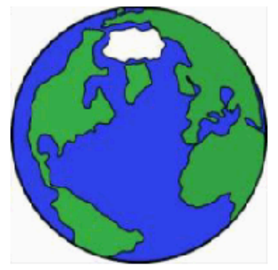


$$V^{\hat{\pi}}_{M^*}(s_0) - V^{\pi^\star}_{M^*}(s_0)$$

A good model gives a good policy that has bounded performance difference in the real world

What makes for a good model?

$$\left[V^{\hat{\pi}}_{M^*}(s_0) - V^{\hat{\pi}}_{\hat{M}}(s_0) \right] + \left[V^{\pi^\star}_{\hat{M}}(s_0) - V^{\pi^\star}_{M^\star}(s_0) \right]$$


Learner in real-world vs model

Expert in real-world vs model

$$+ \left[V^{\hat{\pi}}_{\hat{M}}(s_0) - V^{\pi^\star}_{\hat{M}}(s_0) \right]$$

Learner vs Expert in model


What makes for a good model?

$$\left[V^{\hat{\pi}}_{M^*}(s_0) - V^{\hat{\pi}}_{\hat{M}}(s_0)\right] + \left[V^{\pi^\star}_{\hat{M}}(s_0) - V^{\pi^\star}_{M^\star}(s_0)\right]$$

Learner in real-world vs model

Expert in real-world vs model

Fit model on learner data!

Fit model on expert data!

$$+ \left[V^{\hat{\pi}}_{\hat{M}}(s_0) - V^{\pi^\star}_{\hat{M}}(s_0)\right]$$

Learner vs Expert in model

Optimize policy in model!