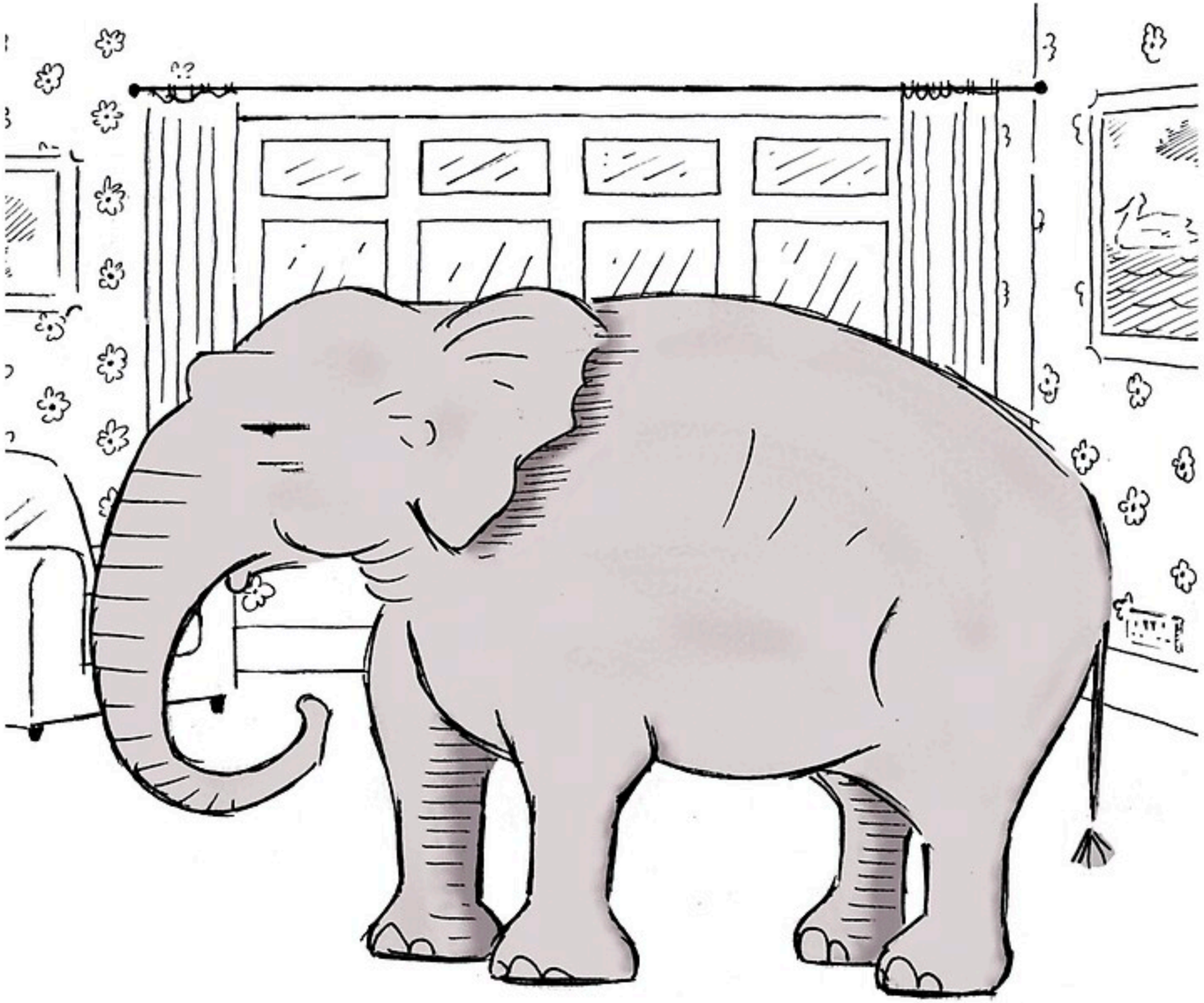
Planning with Inaccurate Models

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Elephant in the room: Why can't we just *learn a model*?

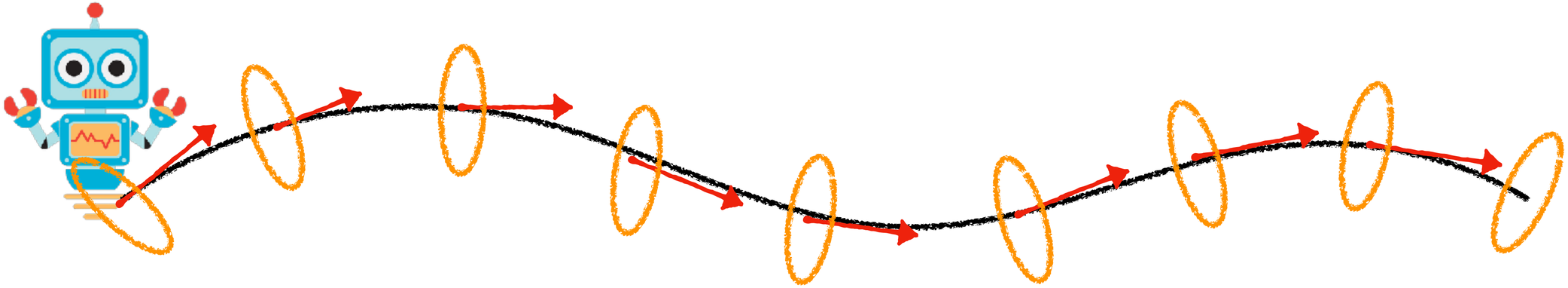


"Just pretend I'm not here..."

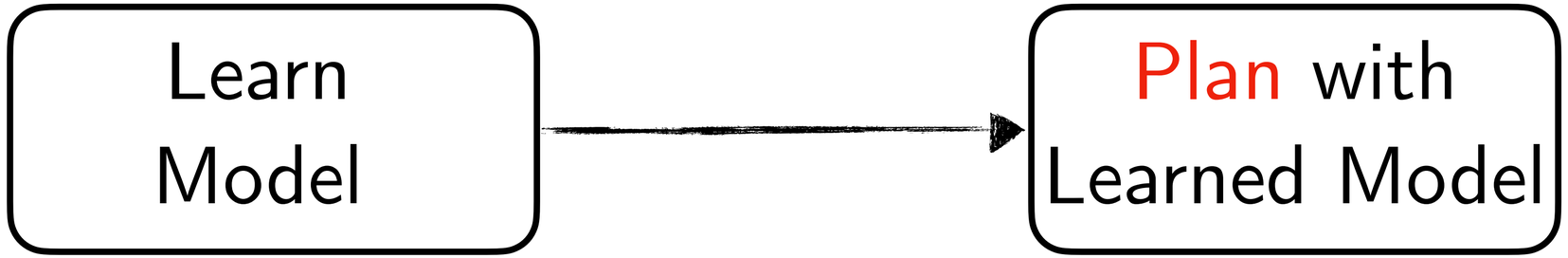
CTHamilton

Model Based Reinforcement Learning

Learn Model → Plan with Learned Model



Model Based Reinforcement Learning



Learning a model is EASY => Just supervised learning!!

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

Activity!

Think-Pair-Share

Think (30 sec): How will you model the dynamics of a helicopter? What planner would you use to control it?

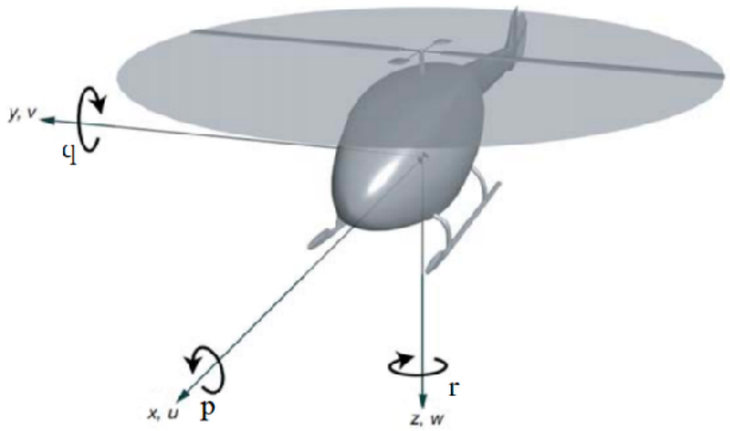
Pair: Find a partner

Learn Model → Plan with Learned Model

Share (45 sec): Partners exchange ideas

Part 1: System Identification

Learn Model → Plan with Learned Model

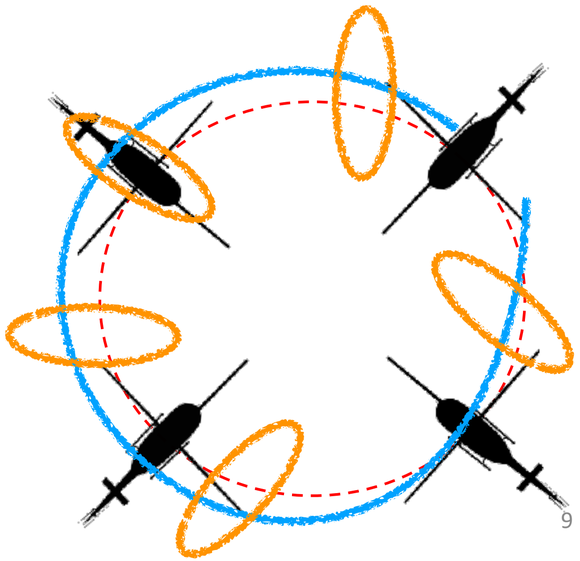


Predict velocities in "body" frame

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

Part 2: Planning

Learn Model → Plan with Learned Model



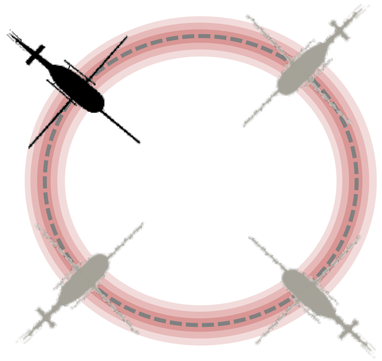
Use LQR with learnt models

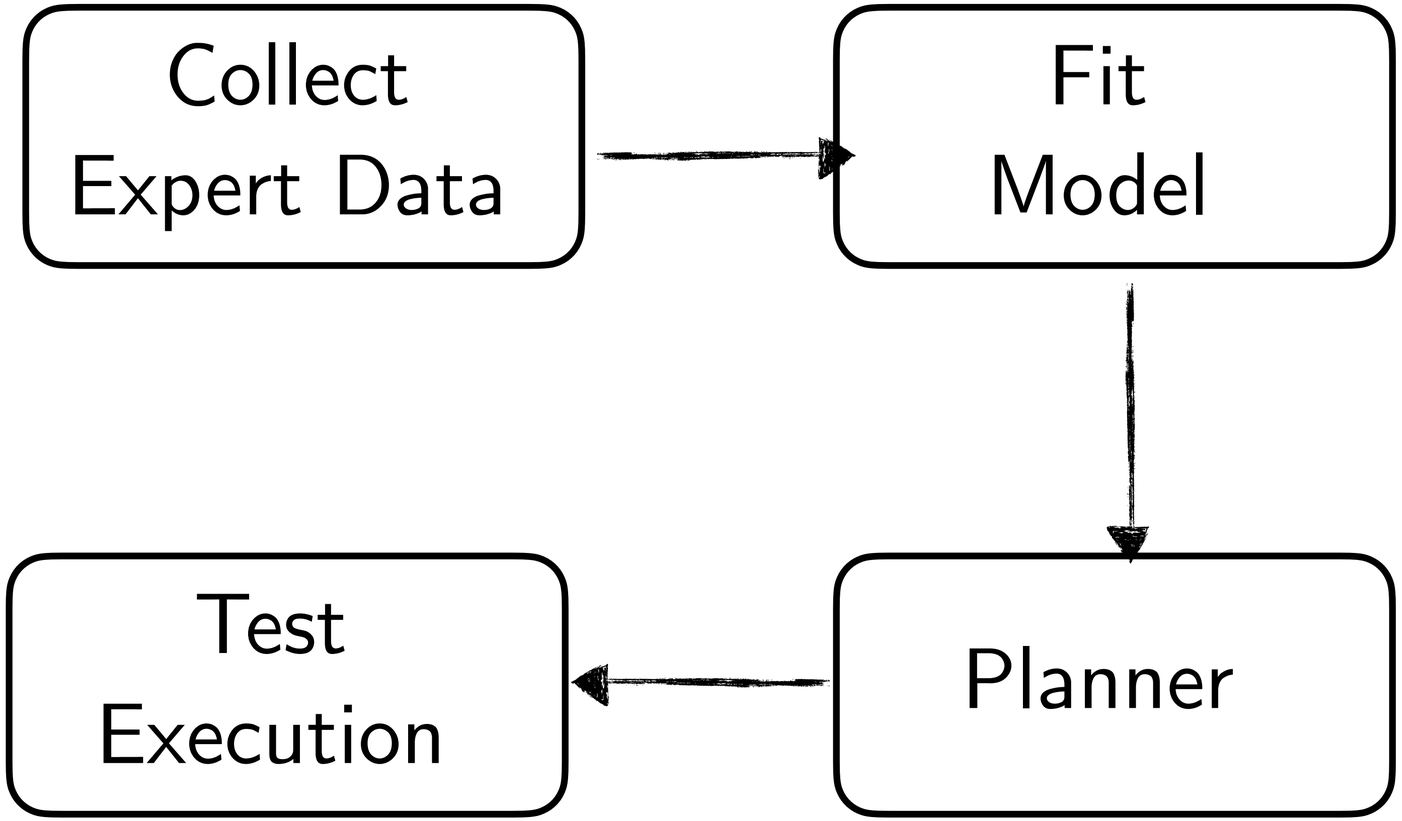
How do we collect data to train our model?

Strategy

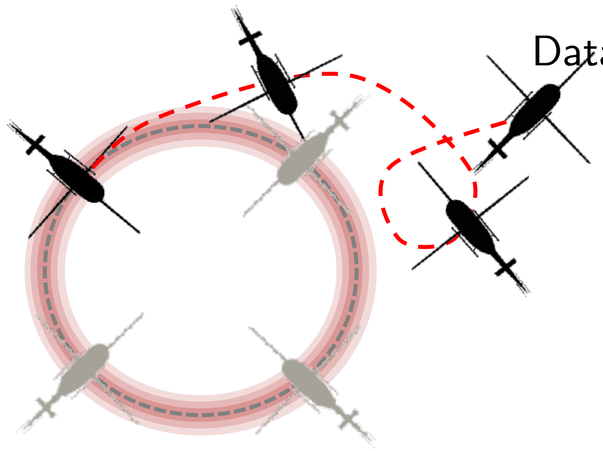
Train a model on state actions visited by the expert!

Vanilla Model Based RL





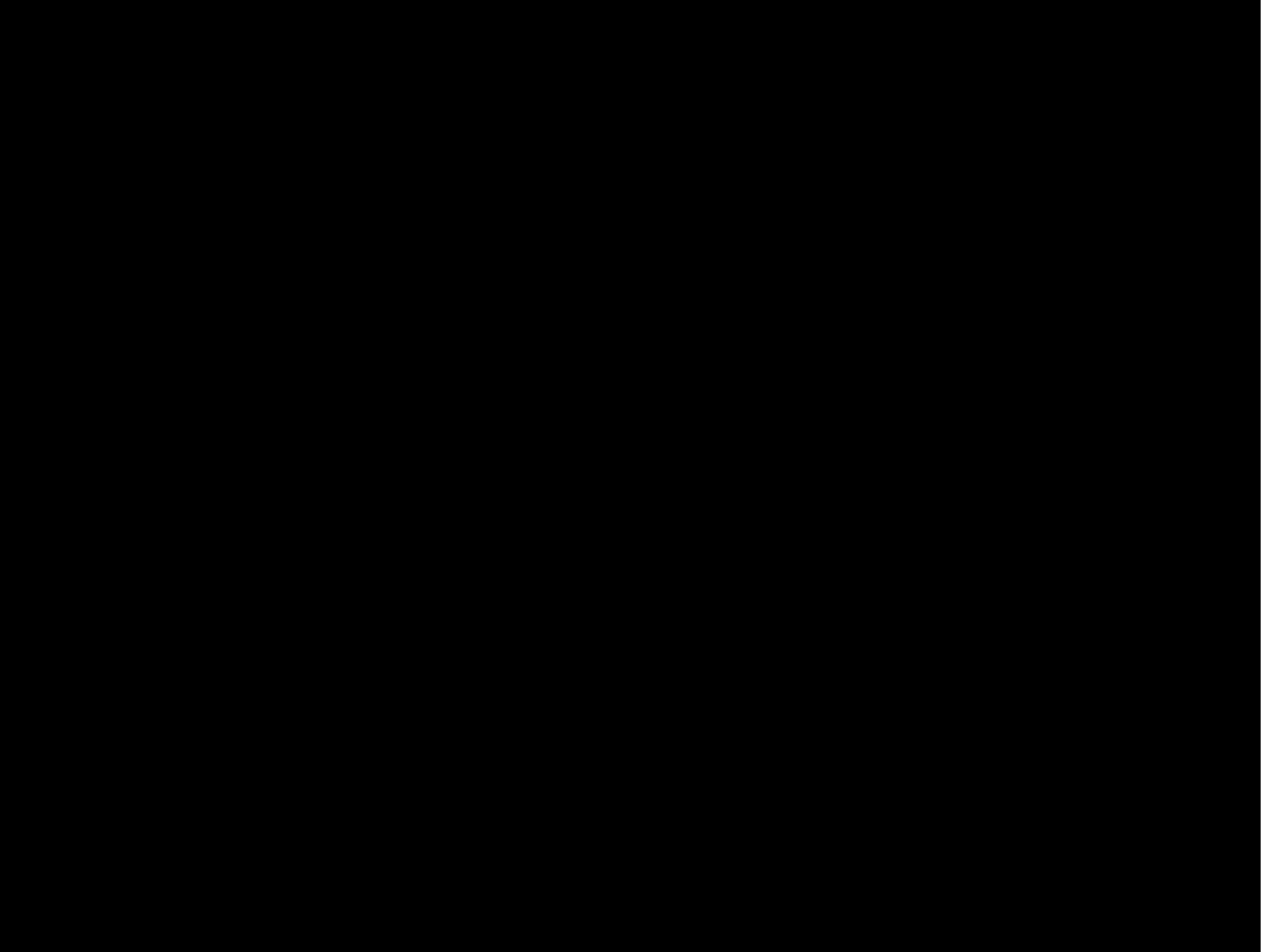
Problem: Train Test Mismatch!



Data gathering policy

Training Distribution $\neq$ Test Trajectory

Collect Expert Data → Fit Model → Planner → Test Execution

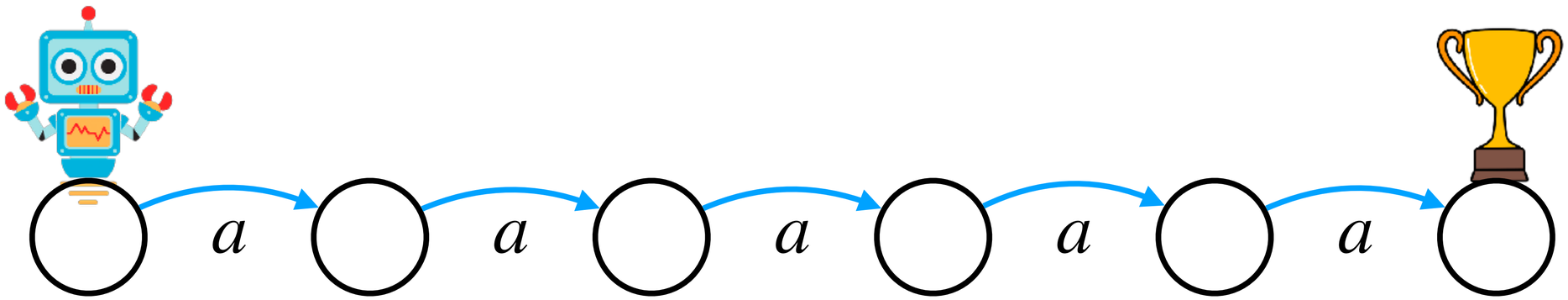


Super retro video of distribution mismatch!

Okay but what if my TRAIN error is ZERO!

World

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



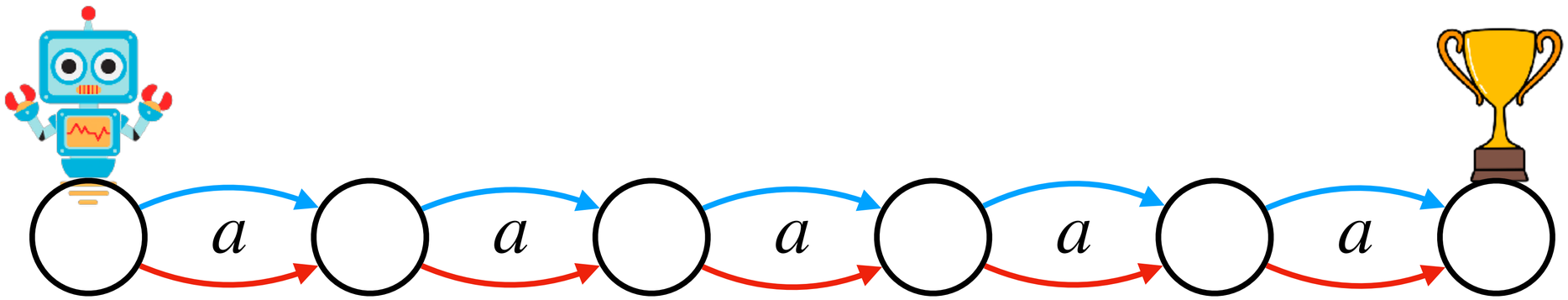
Experts picks action a to go to the goal

Model

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

World

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



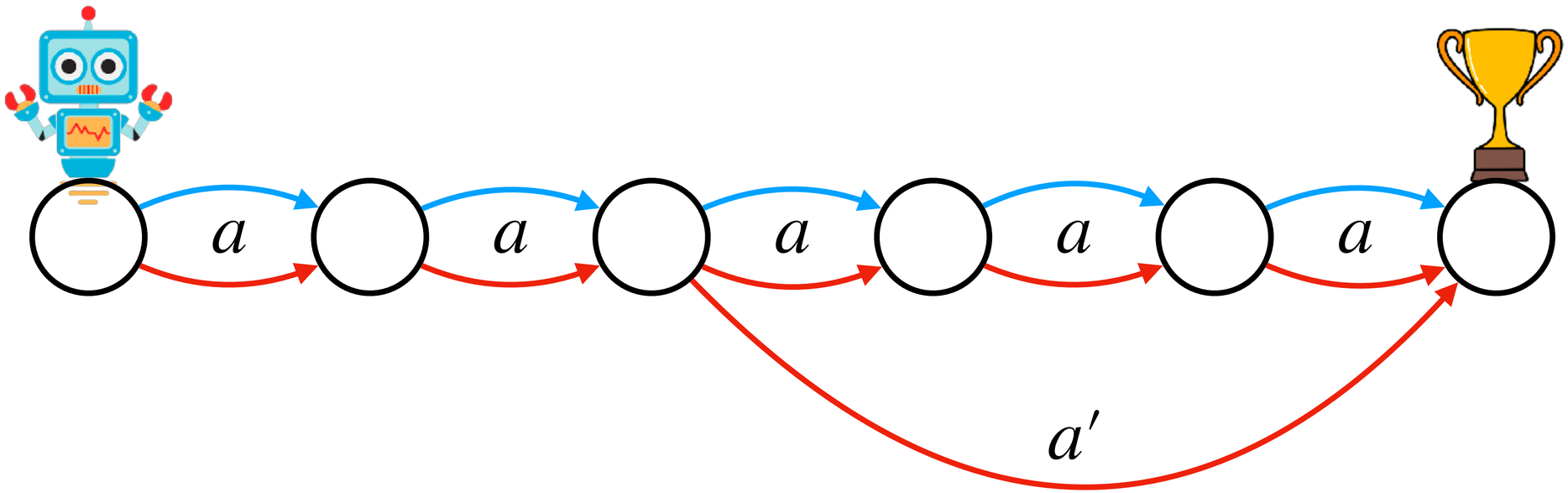
Model agrees with world, i.e. train error zero!

Model

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

World

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



What if the model is optimistic?

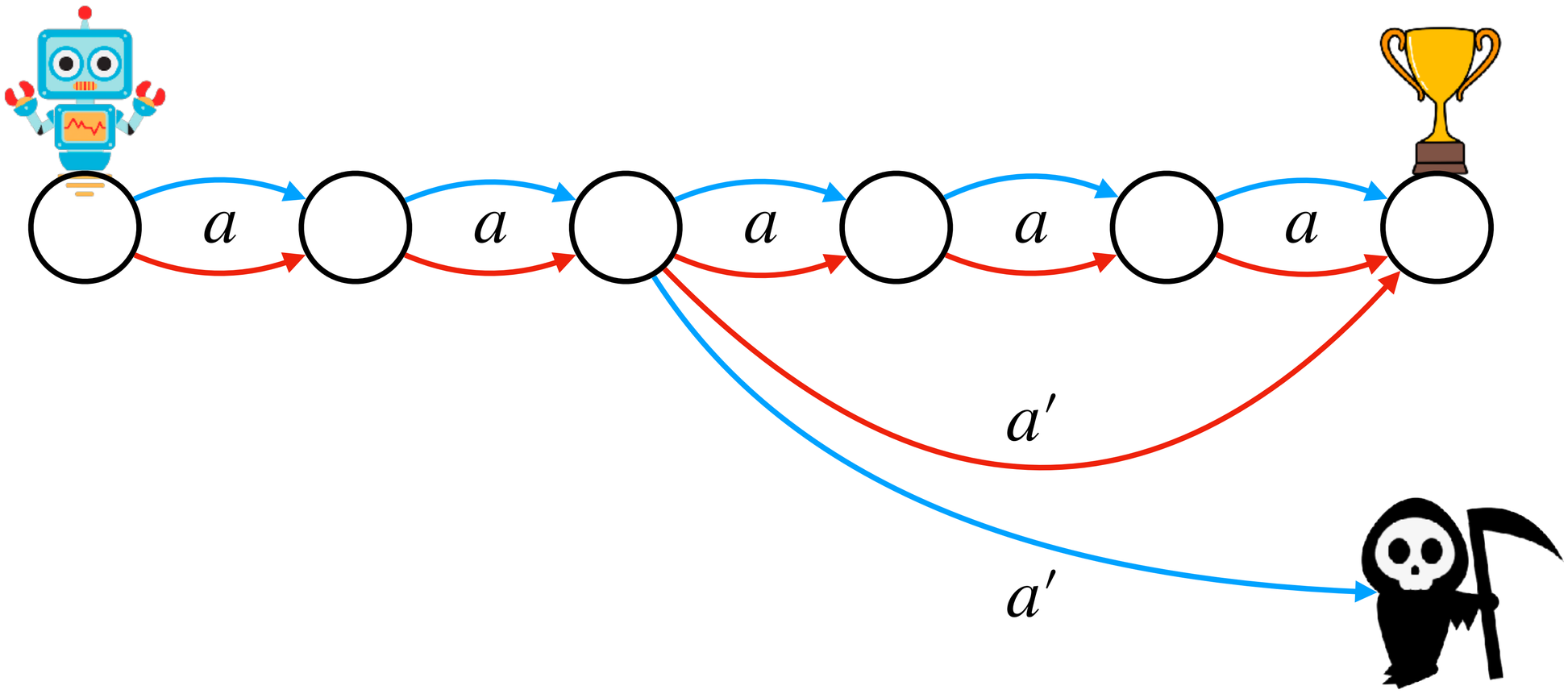
Predicts a short cut to the goal by taking action a'

Model

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

World

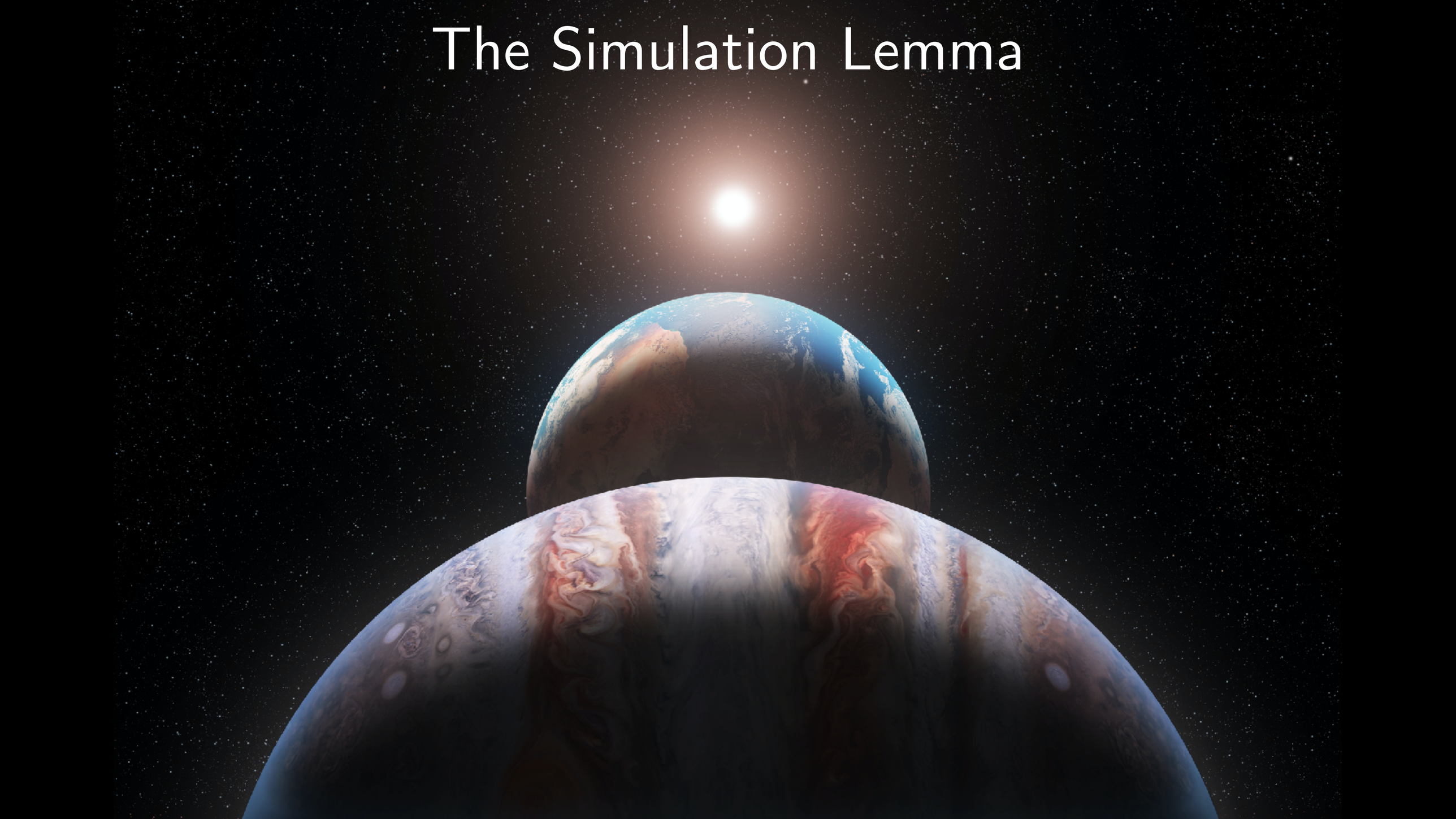
$s' = M^*(s, a)$



In reality the shortcut ends in death ...

Is there a theorem that tells me how error in my model affects error in my performance?

The Simulation Lemma

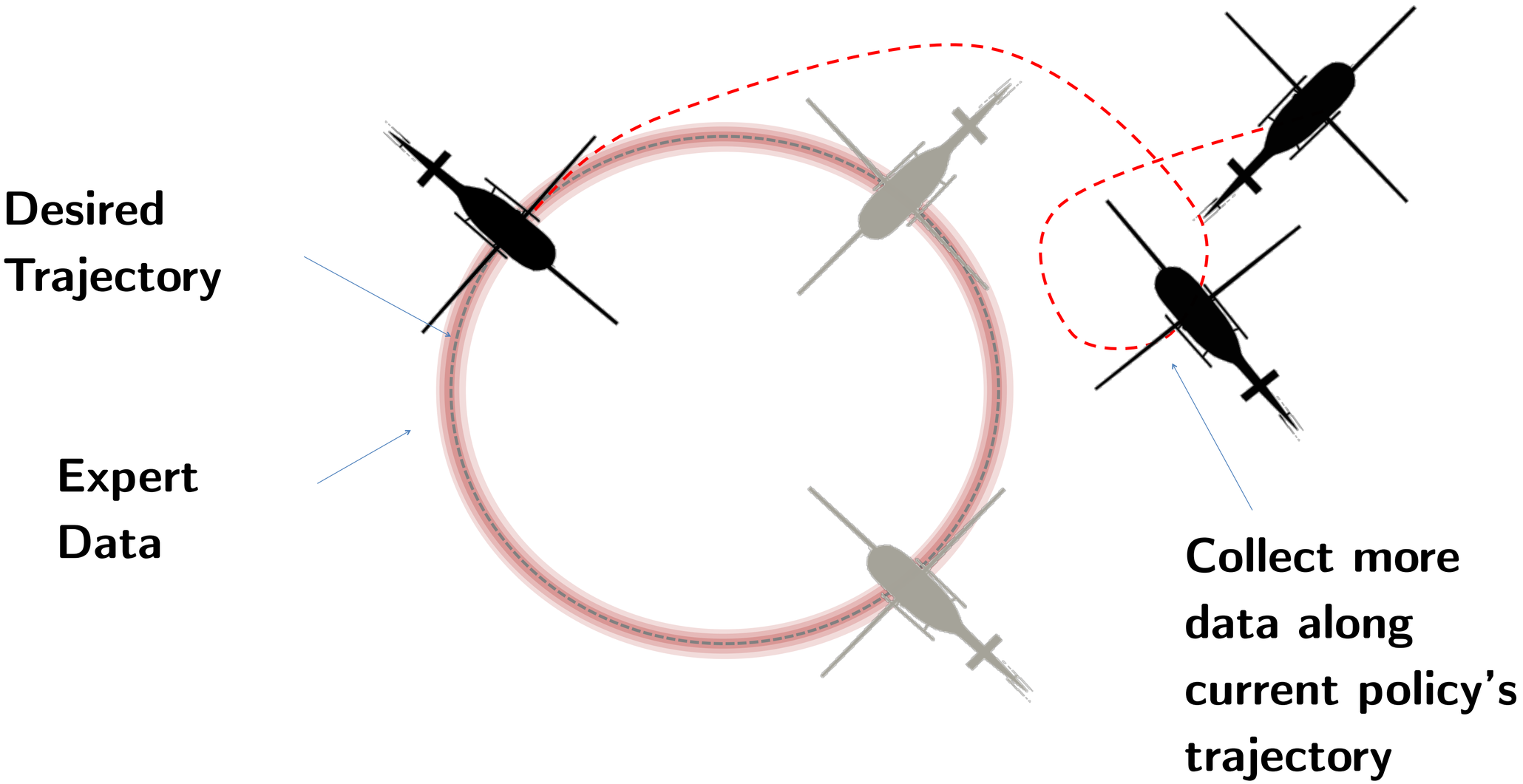


Strategy

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

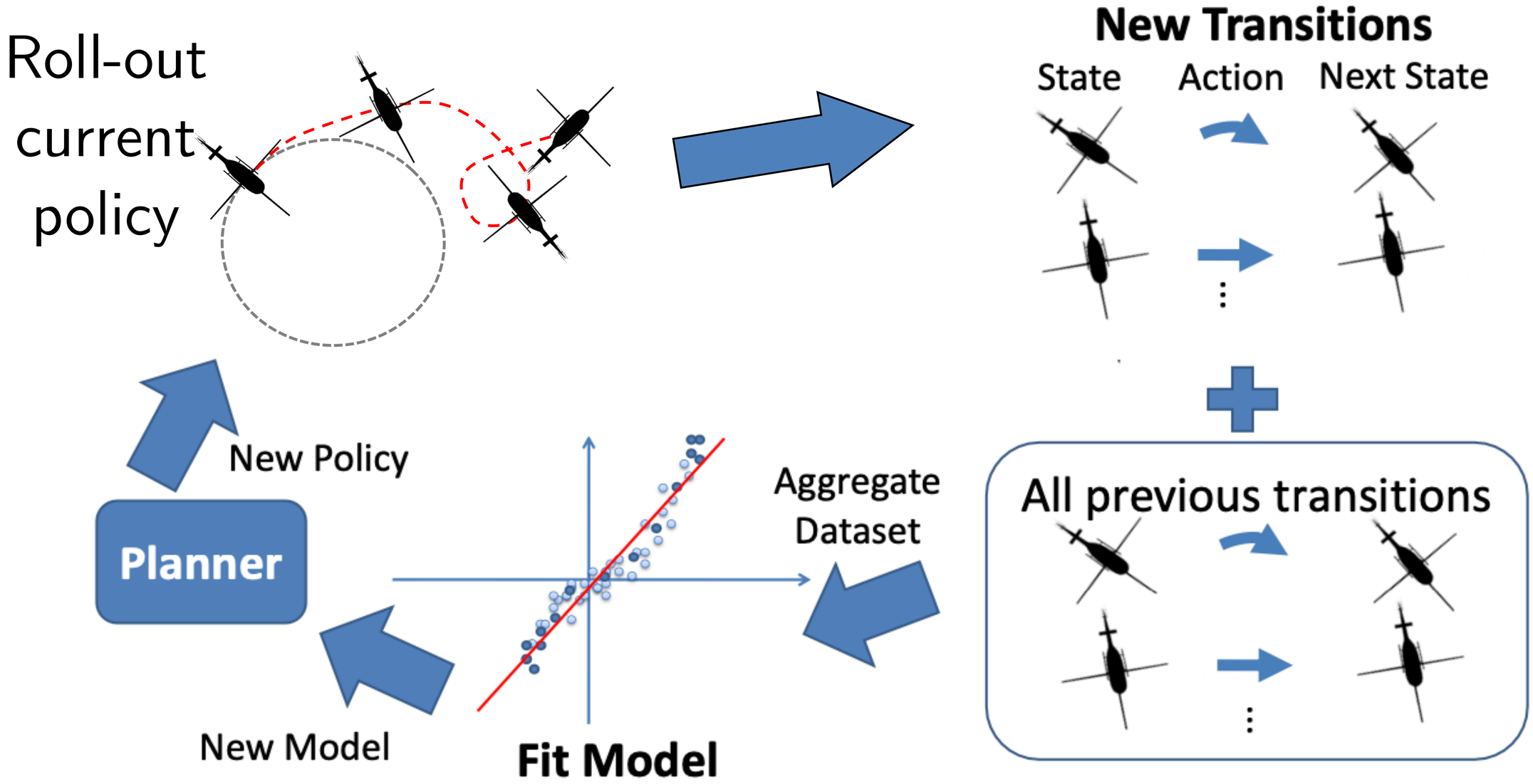
Train a model on state actions visited by the learner!

Improve model where policy goes



Don't we know an algorithm that does this?

DAGGER for Model-based RL!!



DAGGER
seems to work!

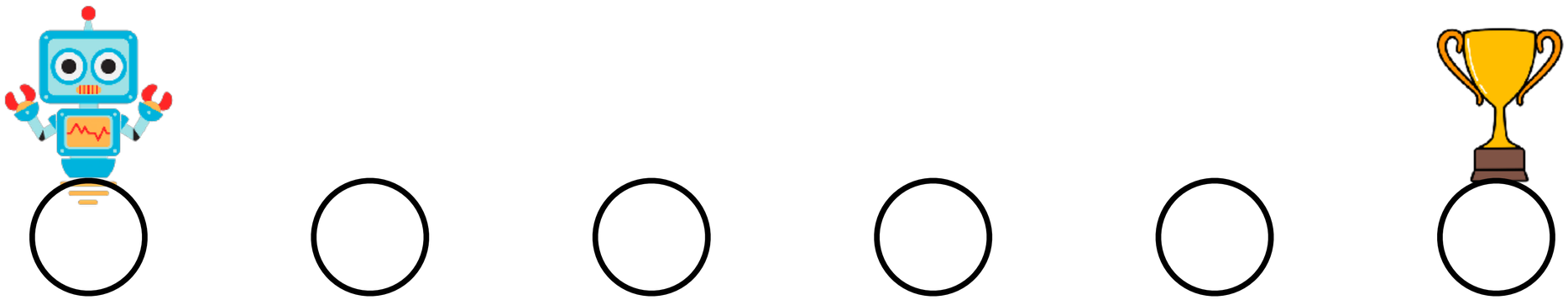
Are we done?

Model

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

World

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

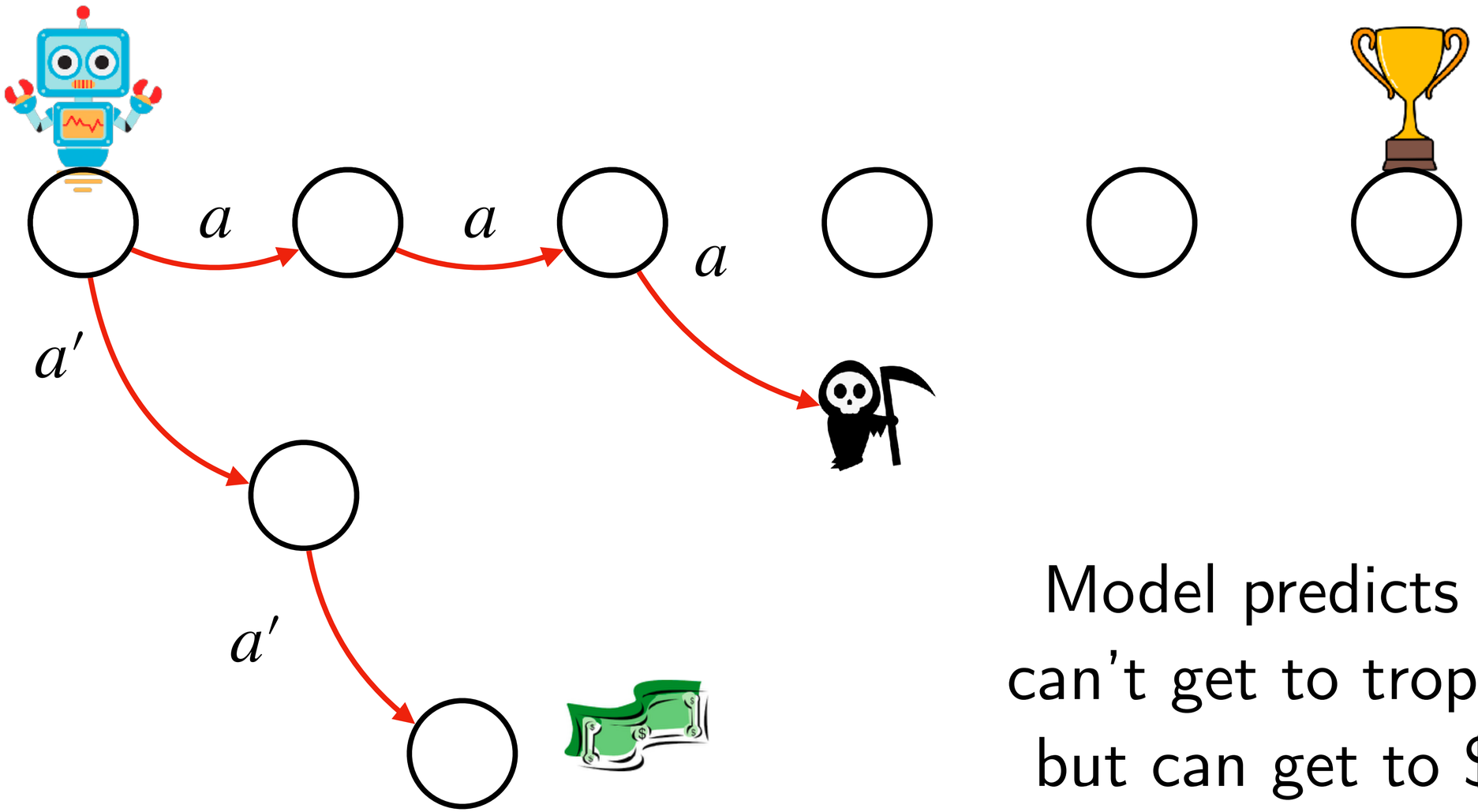


Model

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

World

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



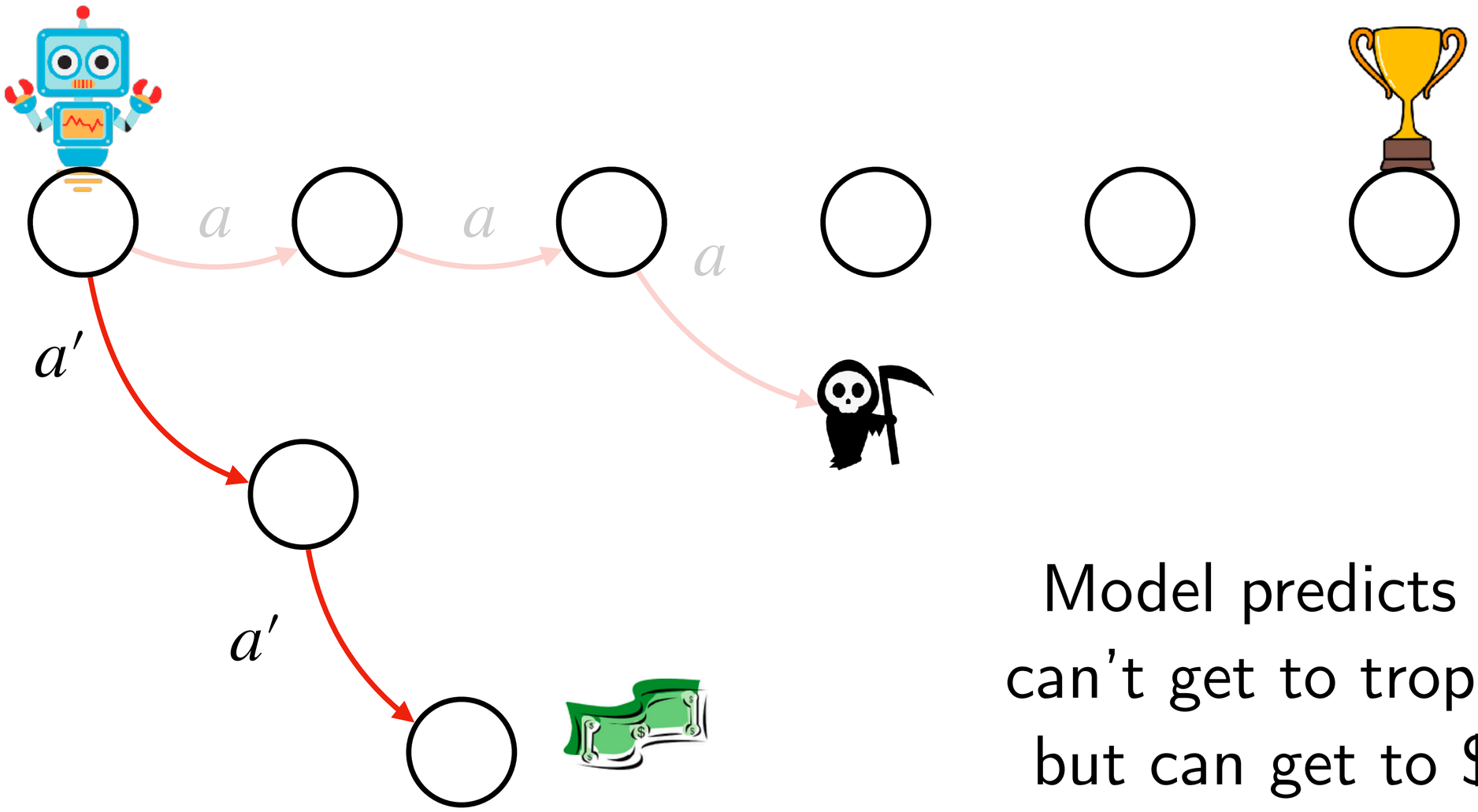
Model predicts it can't get to trophy, but can get to $1

Model

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

World

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



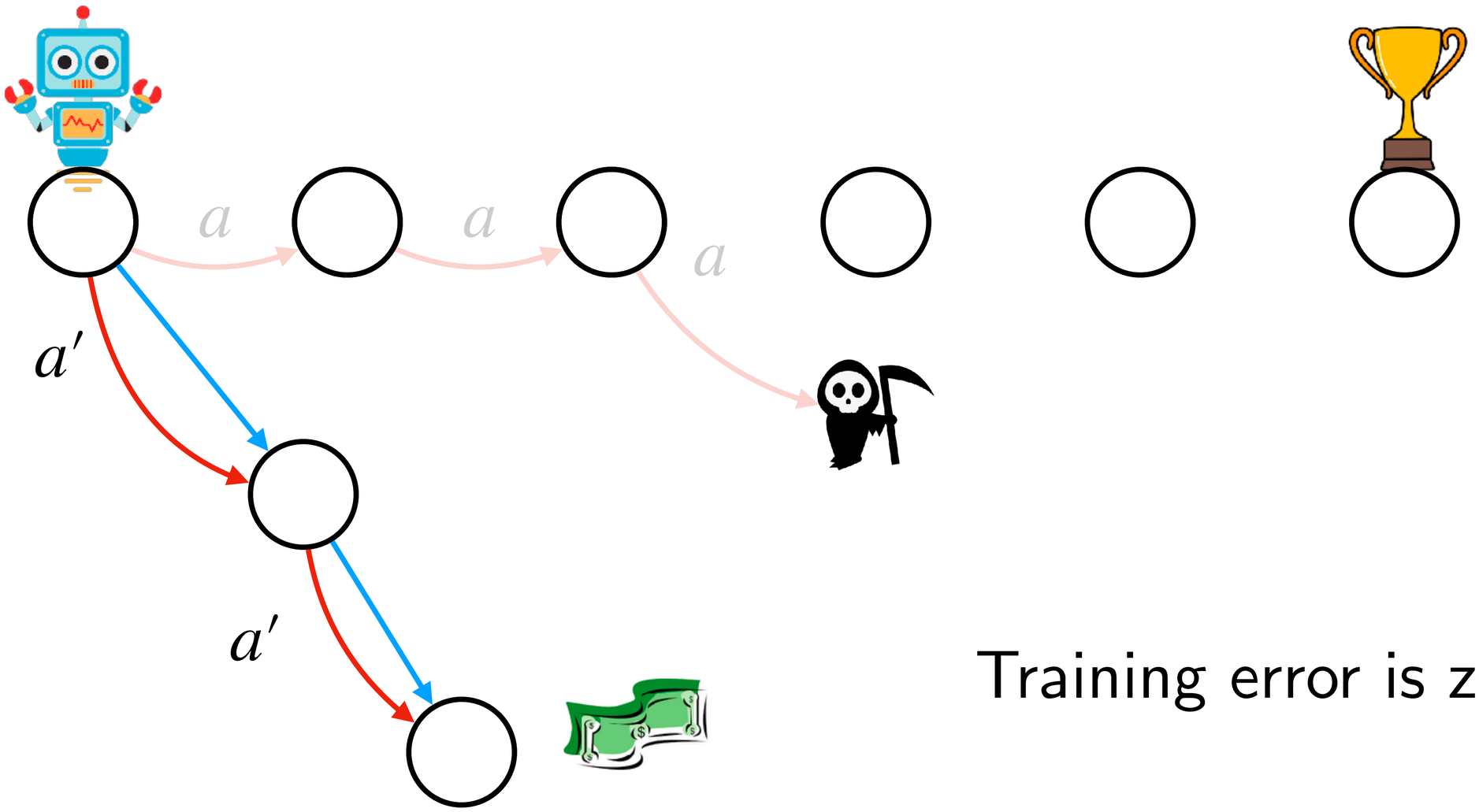
Model predicts it can't get to trophy, but can get to $1

Model

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

World

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



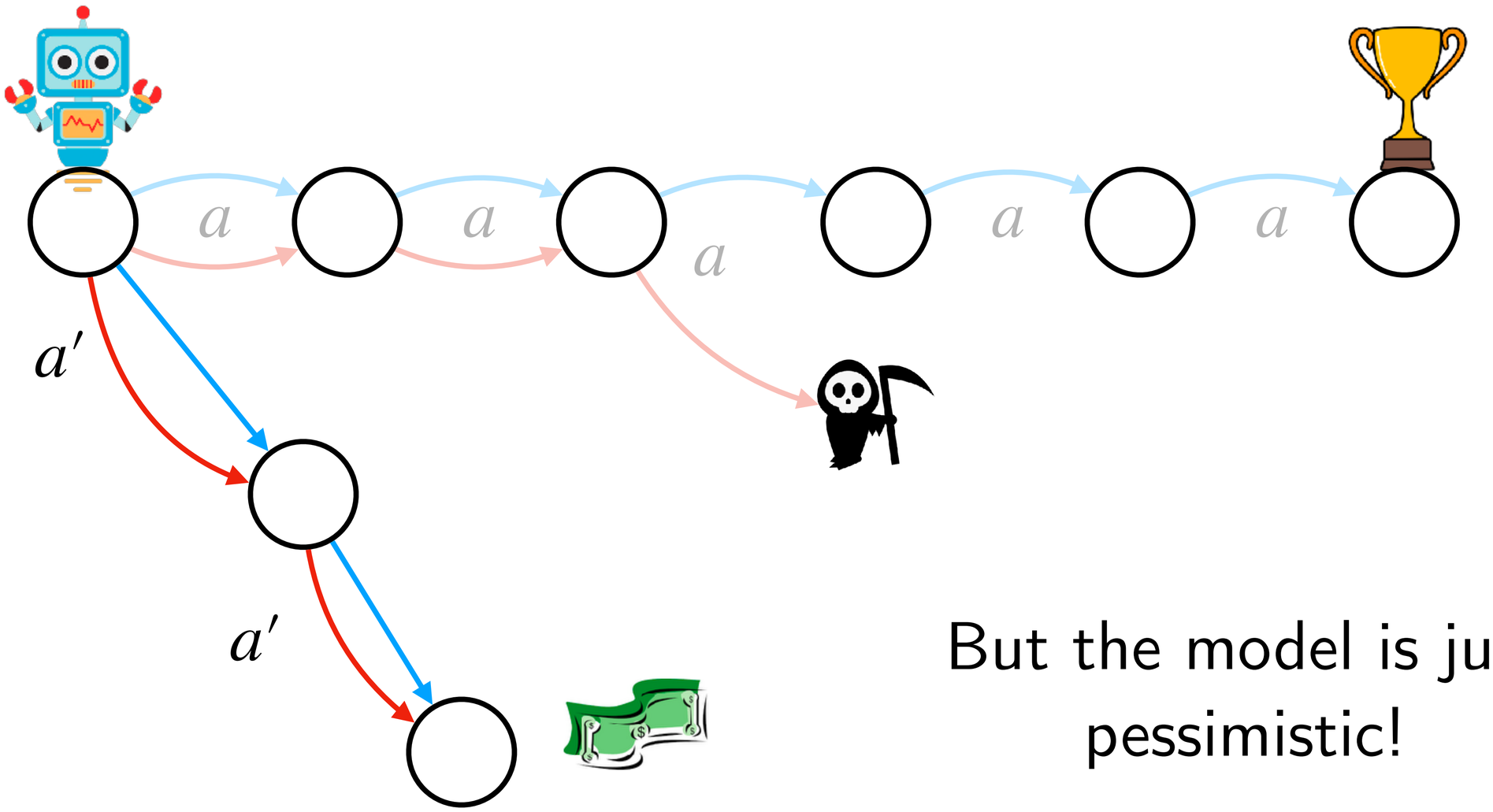
Training error is zero!

Model

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

World

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



But the model is just pessimistic!

Strategy

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

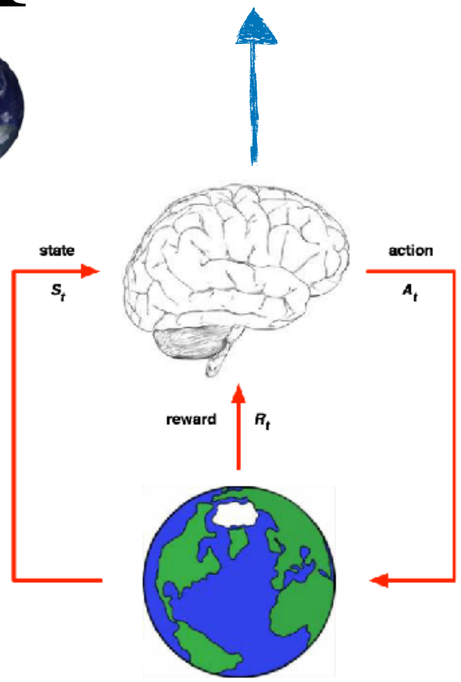
~~Train a model on state actions visited by the learner!~~

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

How do we derive this strategy?

What makes a model good?

$$J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^*)$$



The *Double* Simulation Lemma

Insight

- Make a model good only where the expert visits
- Make a model good where the learner visits
- Make a model good in both places