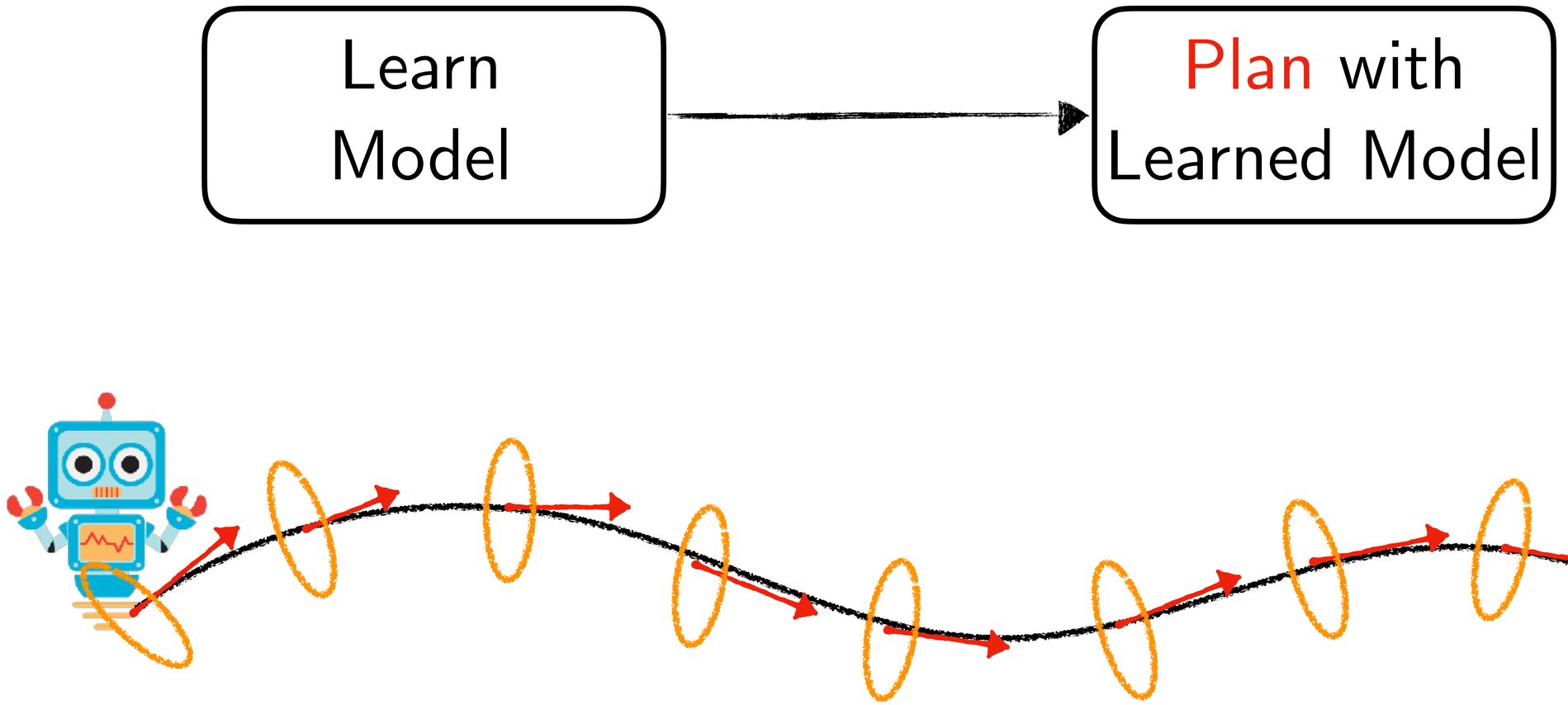
Review of Algorithms

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Model Based Reinforcement Learning



Strategy

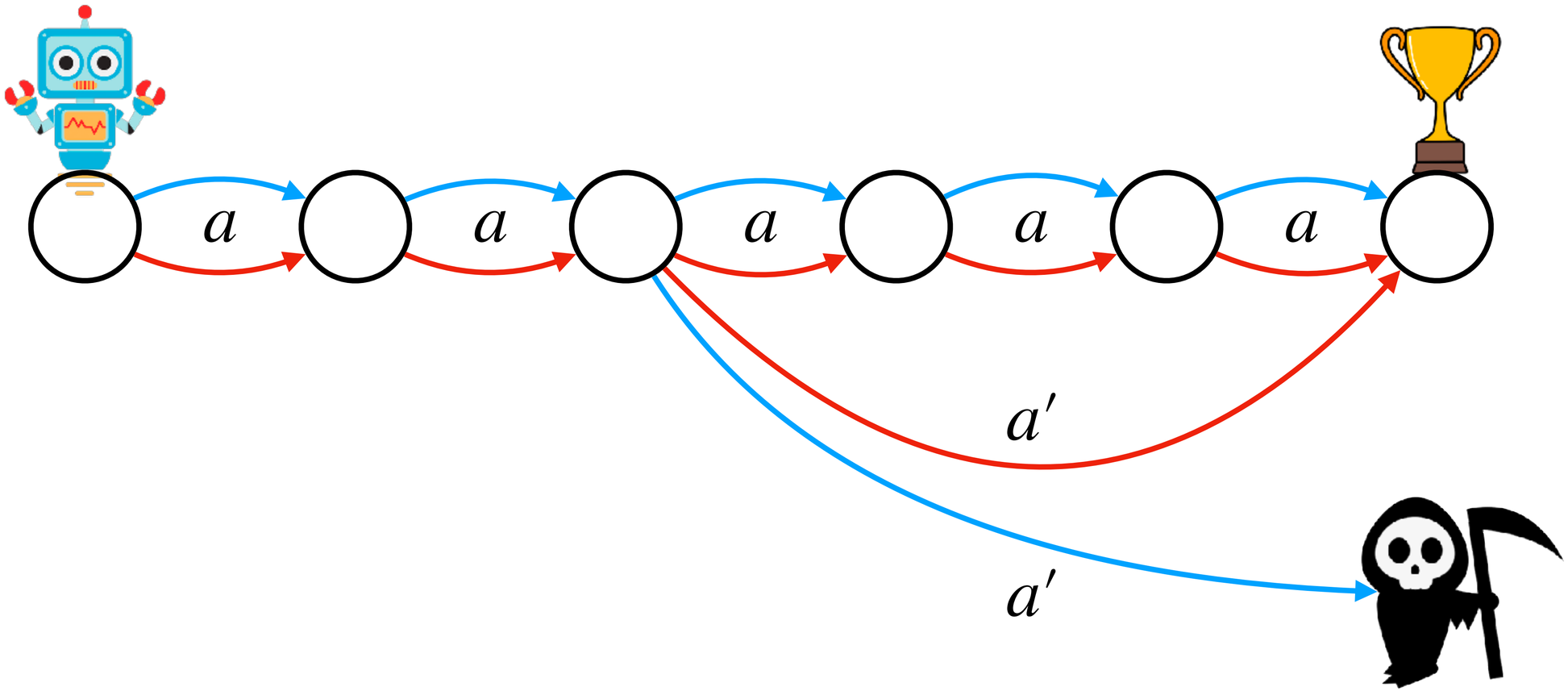
Train a model on state actions visited by the expert!

Model

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

World

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



In reality the shortcut ends in death ...

Strategy

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

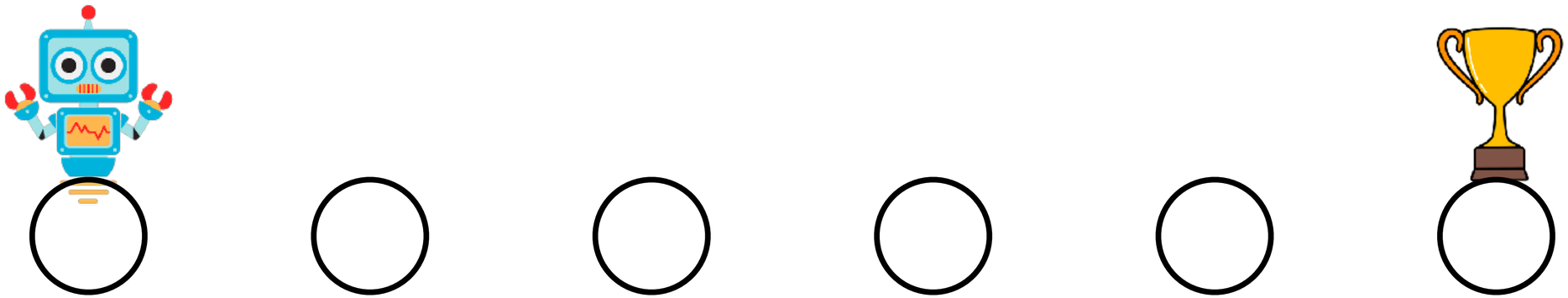
Train a model on state actions visited by the learner!

Model

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

World

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

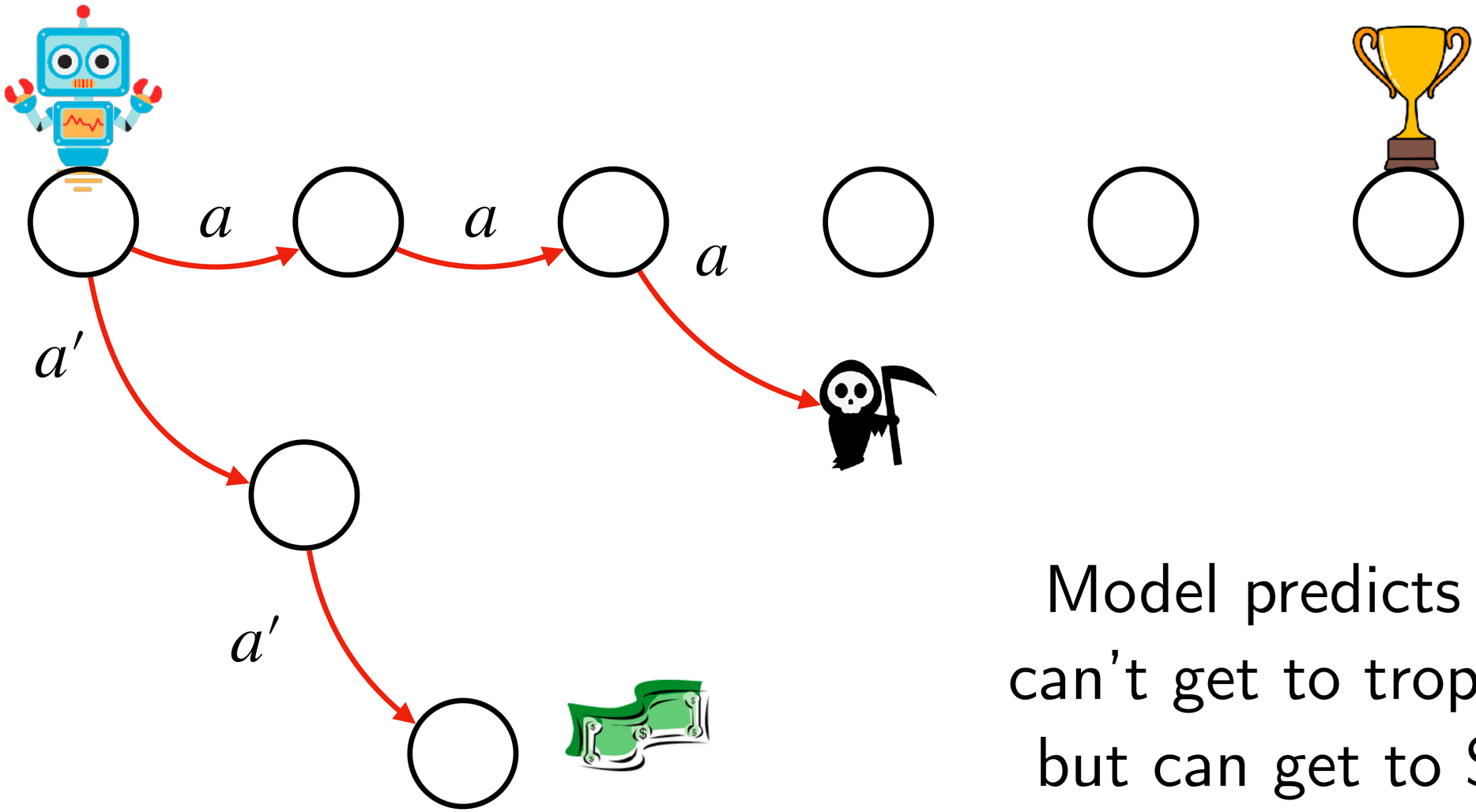


Model

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

World

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



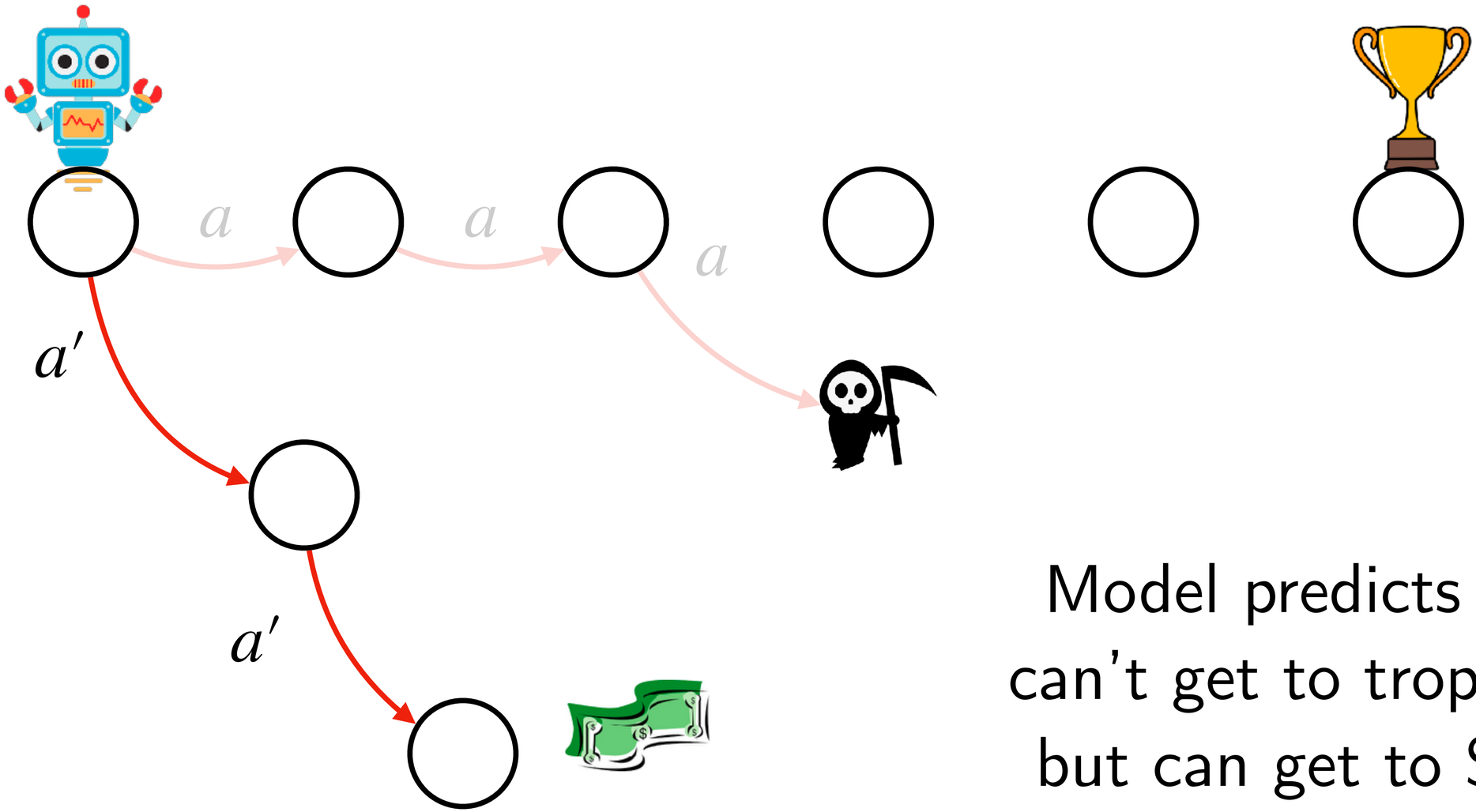
Model predicts it can't get to trophy, but can get to $1

Model

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

World

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



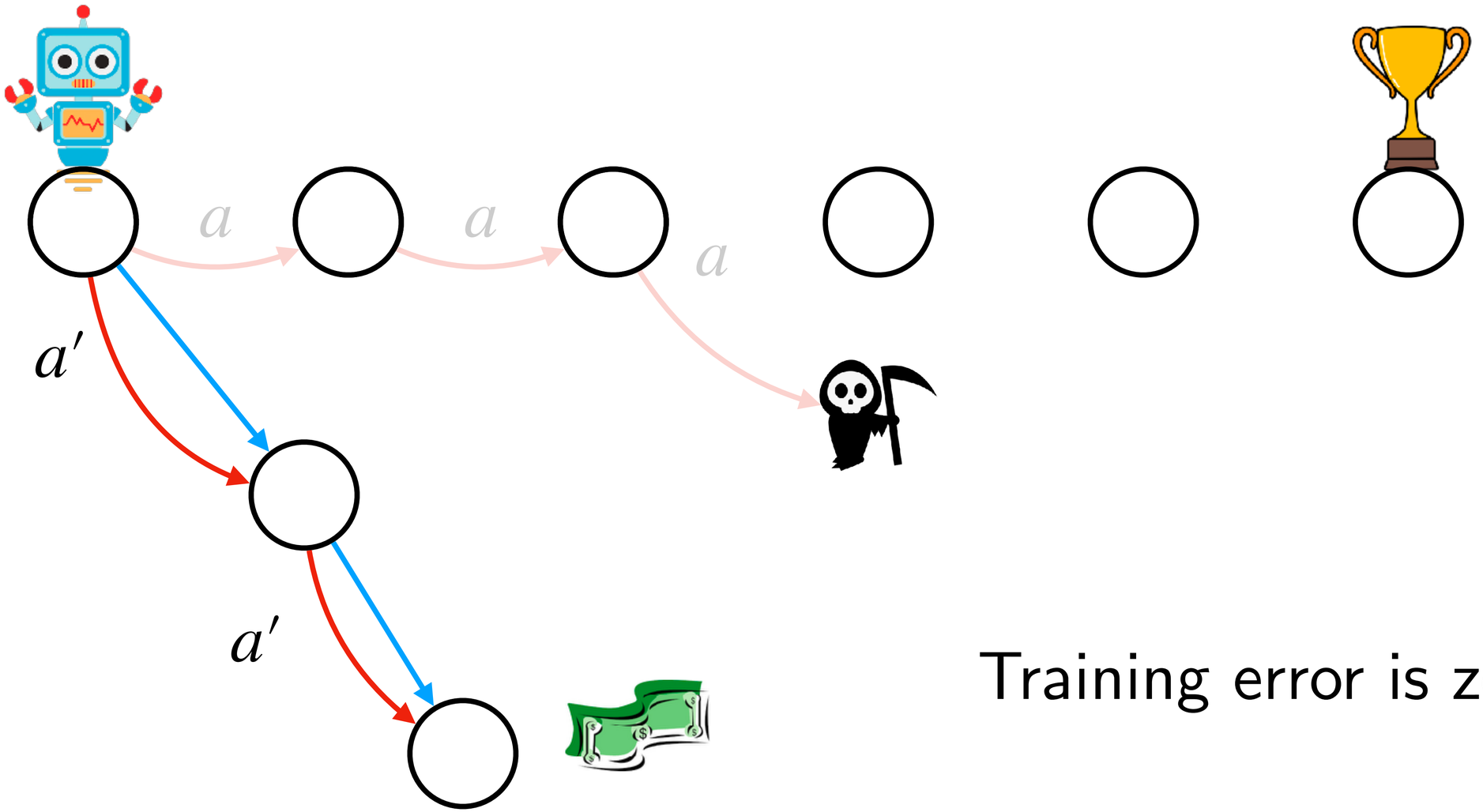
Model predicts it can't get to trophy, but can get to $1

Model

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

World

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



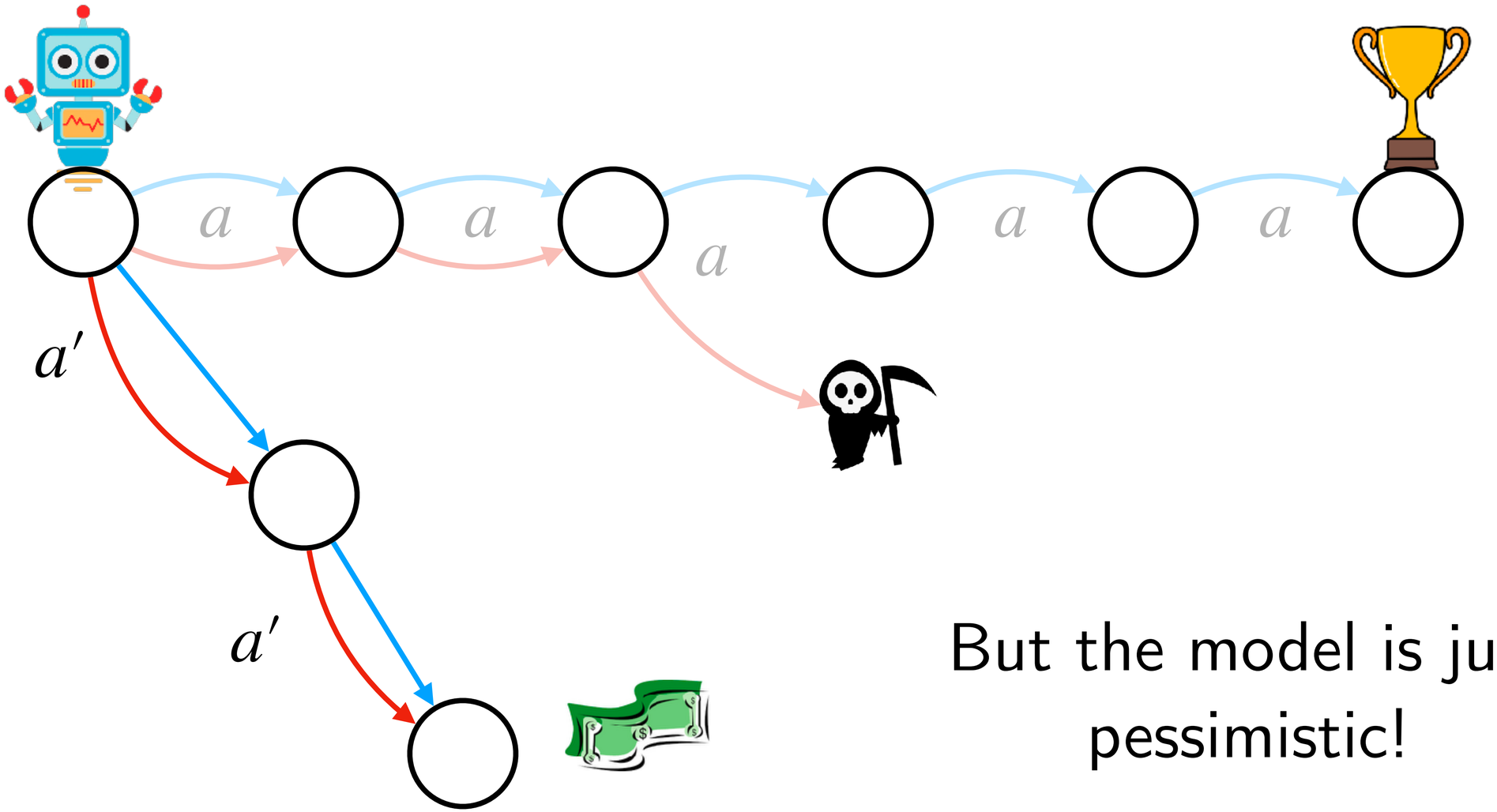
Training error is zero!

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!~~

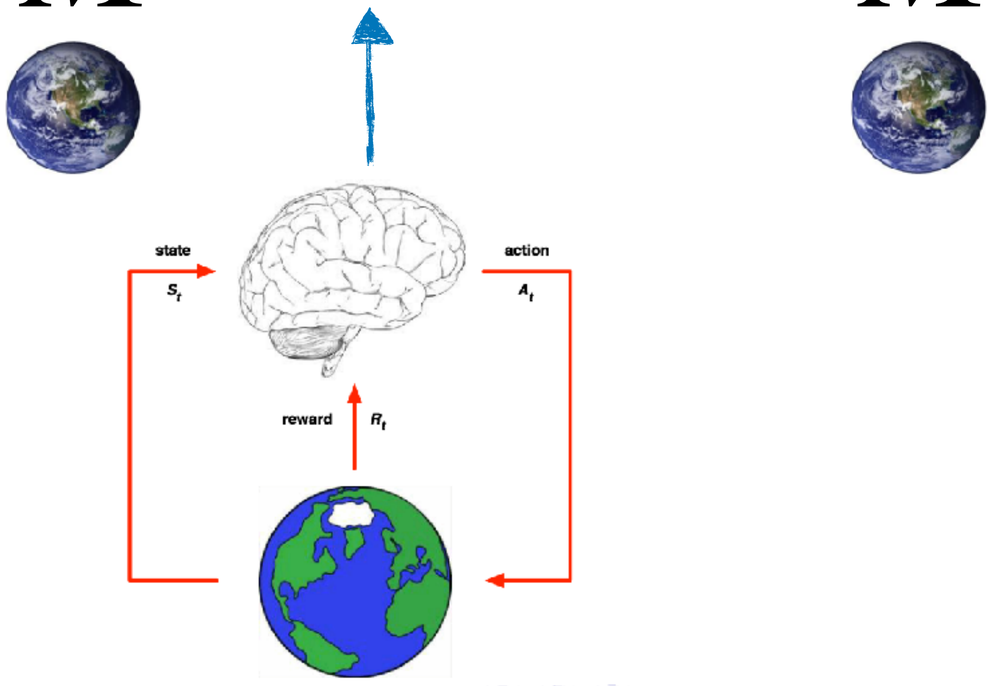
~~Train a model on state actions visited by the learner!~~

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

How do we derive this strategy?

What we care about is the performance difference?

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



Recap of Algorithms