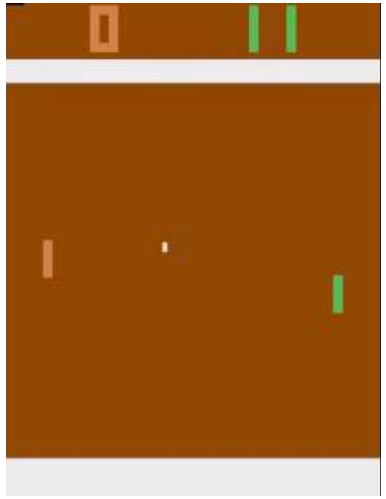
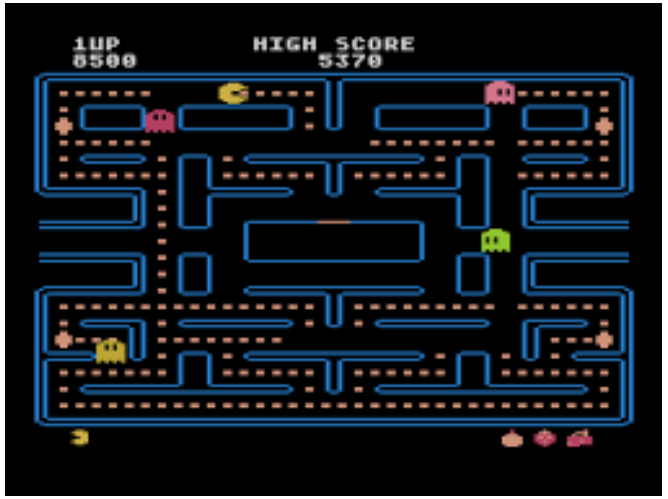
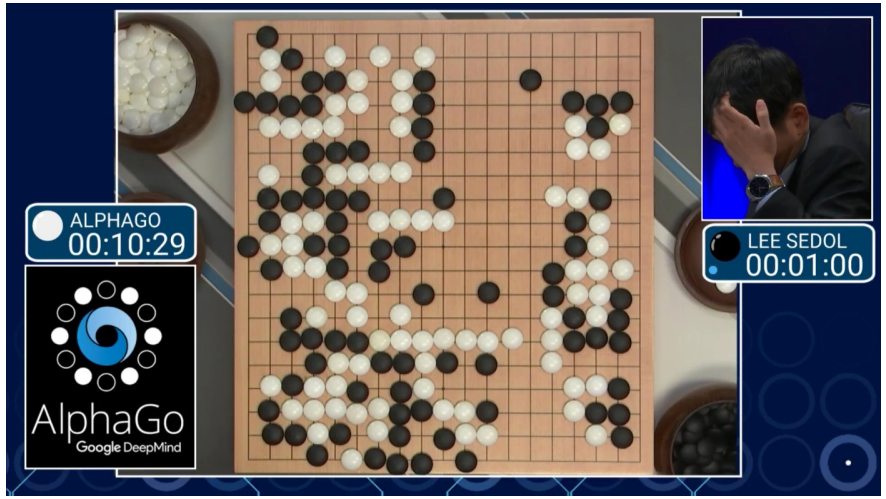
Verifiable Reinforcement Learning via Policy Extraction

Osbert Bastani, Yewen Pu, Armando Solar-Lezama

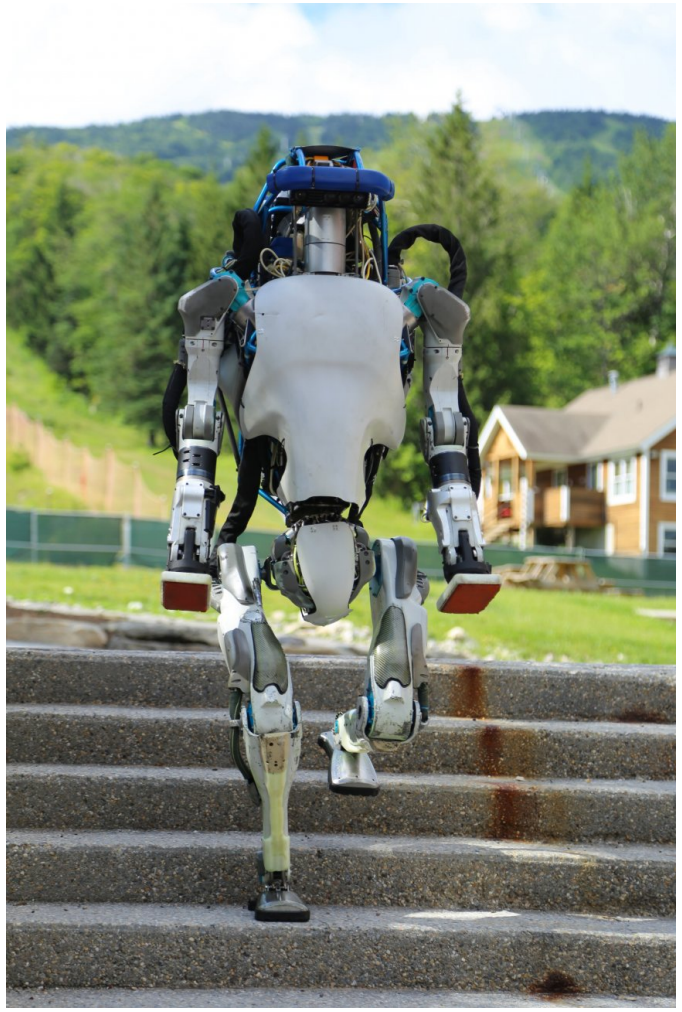
Deep Reinforcement Learning Successes

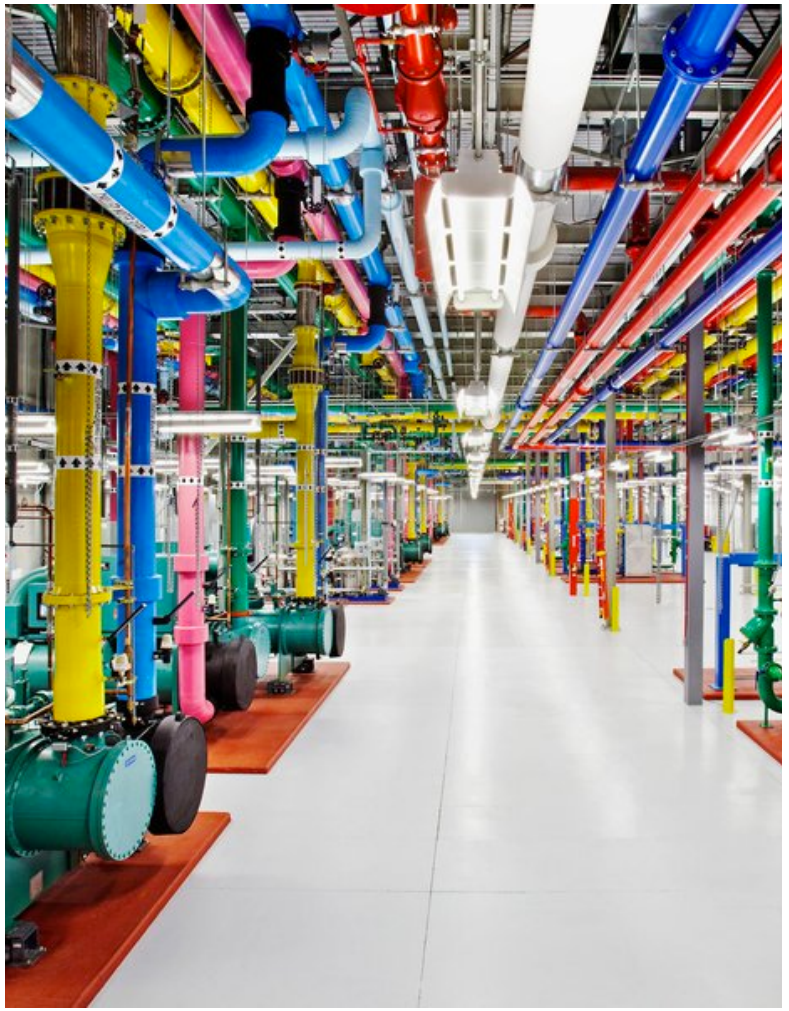




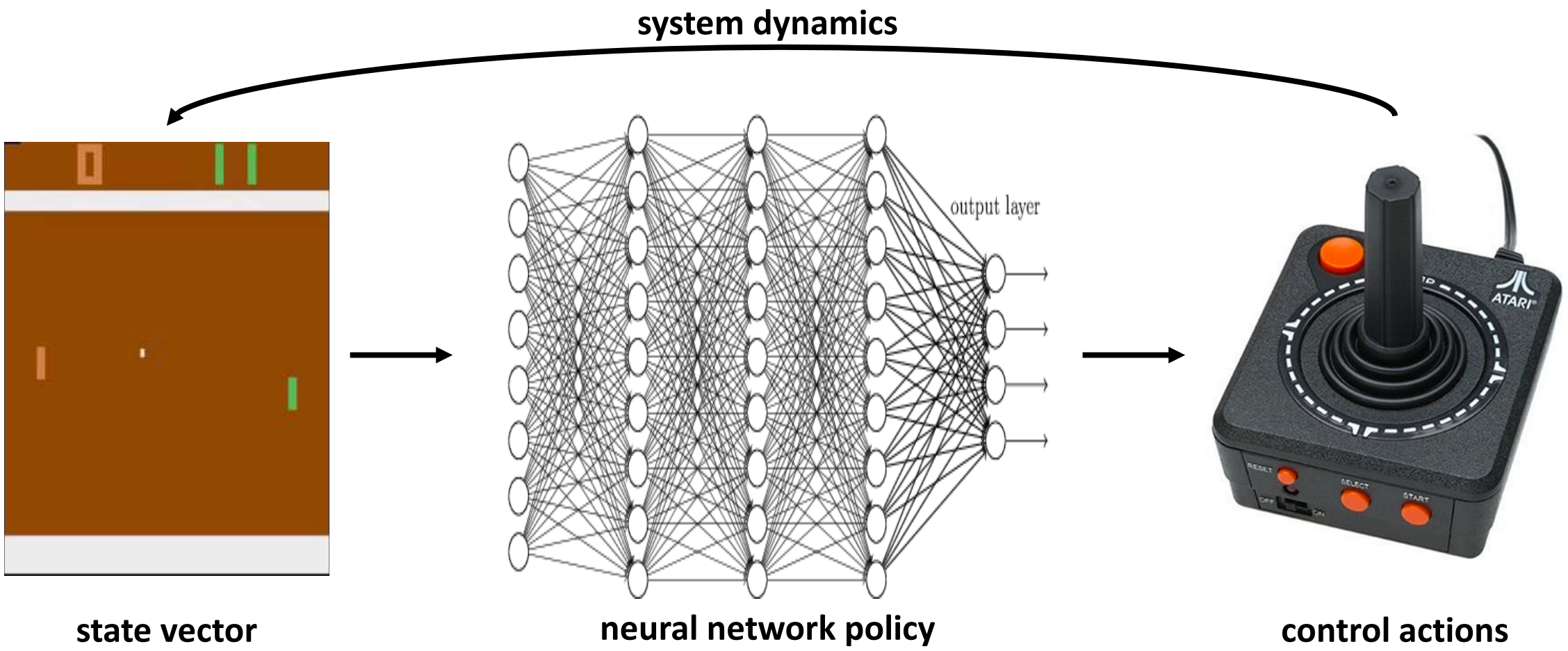


Safety Critical Applications

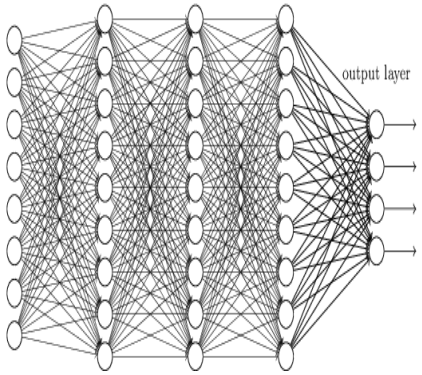


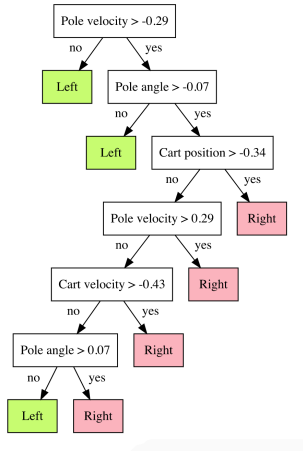


Deep Reinforcement Learning



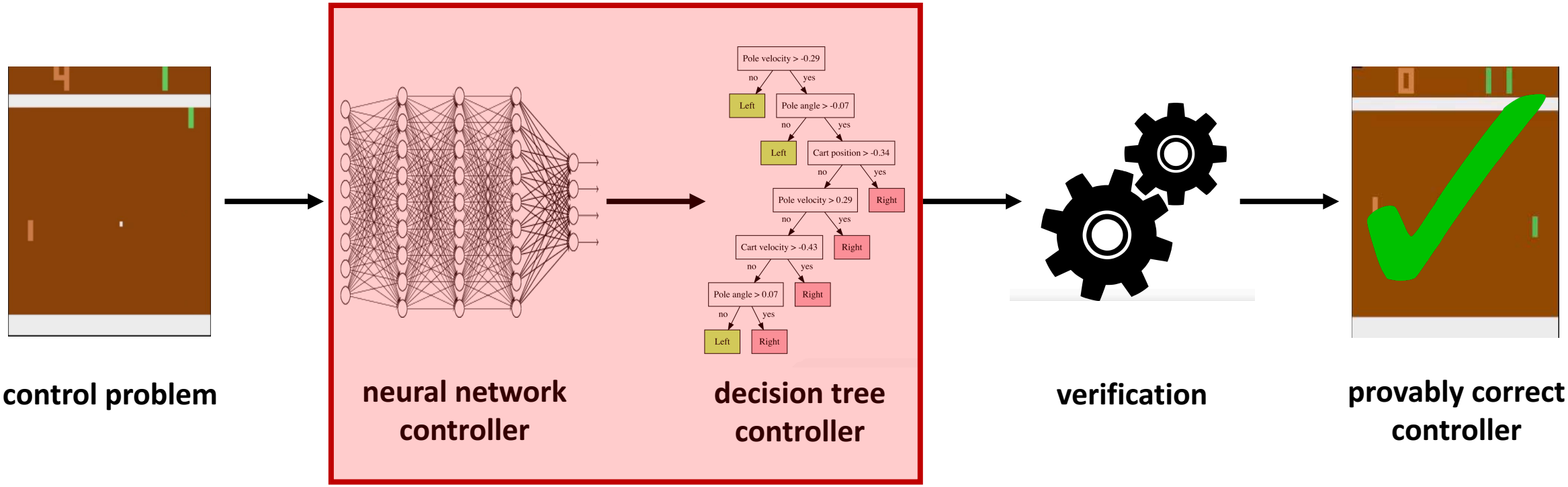
Learnability vs. Verifiability





Complex policies	✓	✓
Easy to verify	✗	✓
Easy to train	✓	✗

High-Level Approach



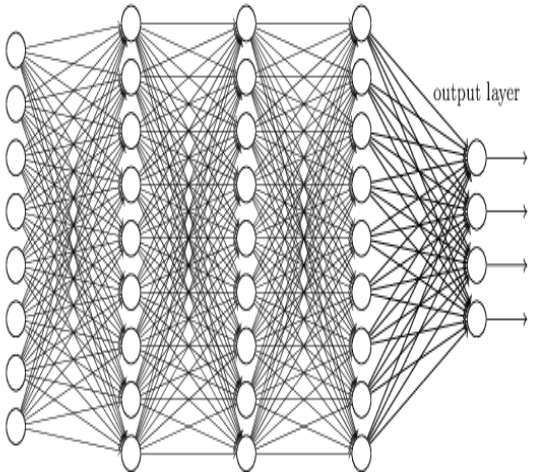
Imitation Learning

Examples from Human Expert

Autonomous Controller

Abbeel & Ng 2004

Imitation Learning

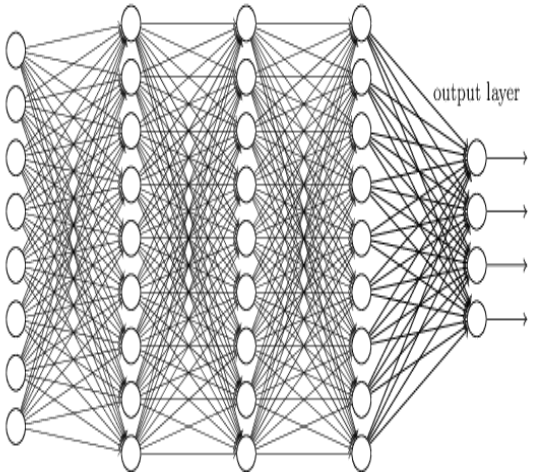


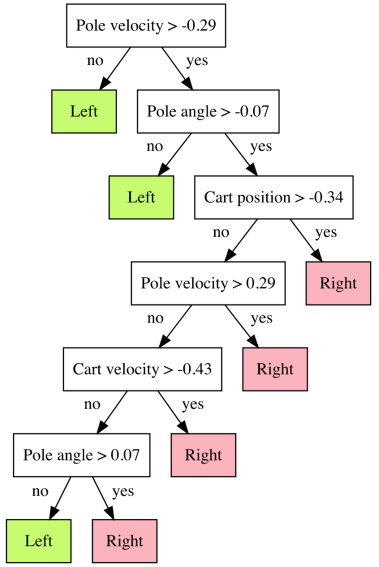
Examples from Neural Network

Autonomous Controller

Abbeel & Ng 2004

Imitation Learning



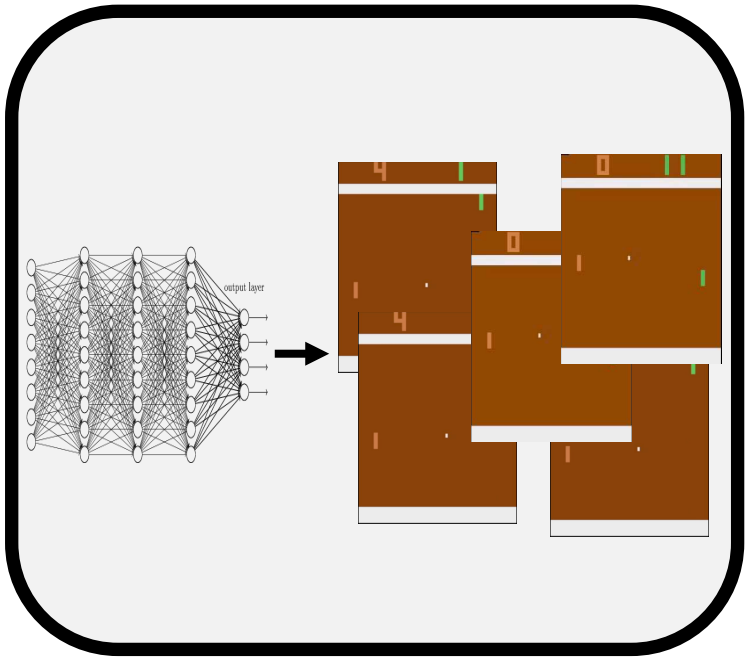


Examples from Neural Network

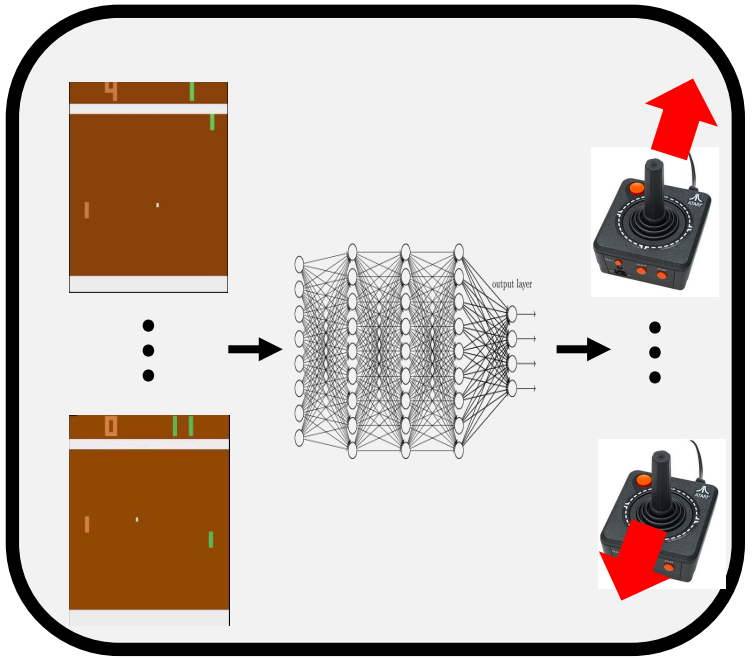
Decision Tree Controller

Abbeel & Ng 2004

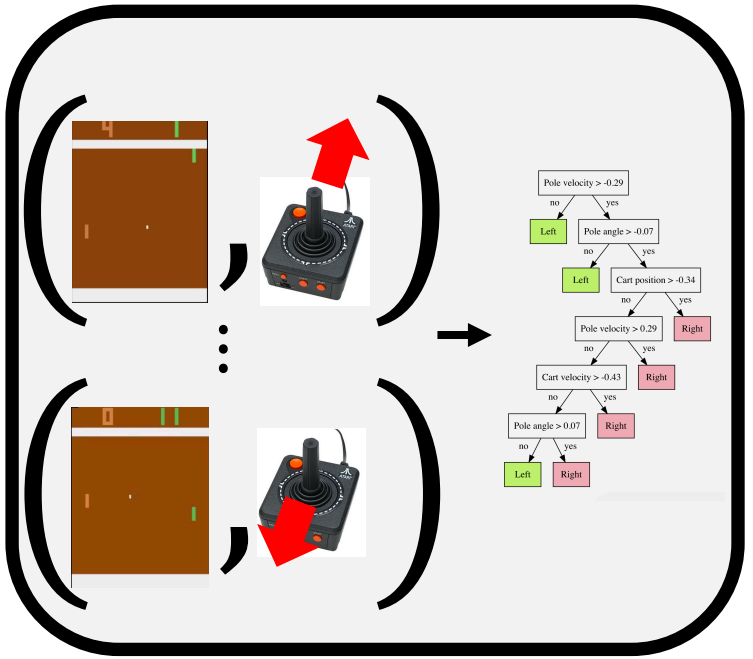
Imitation Learning (Supervised)



Step 1: Use NN to generate states



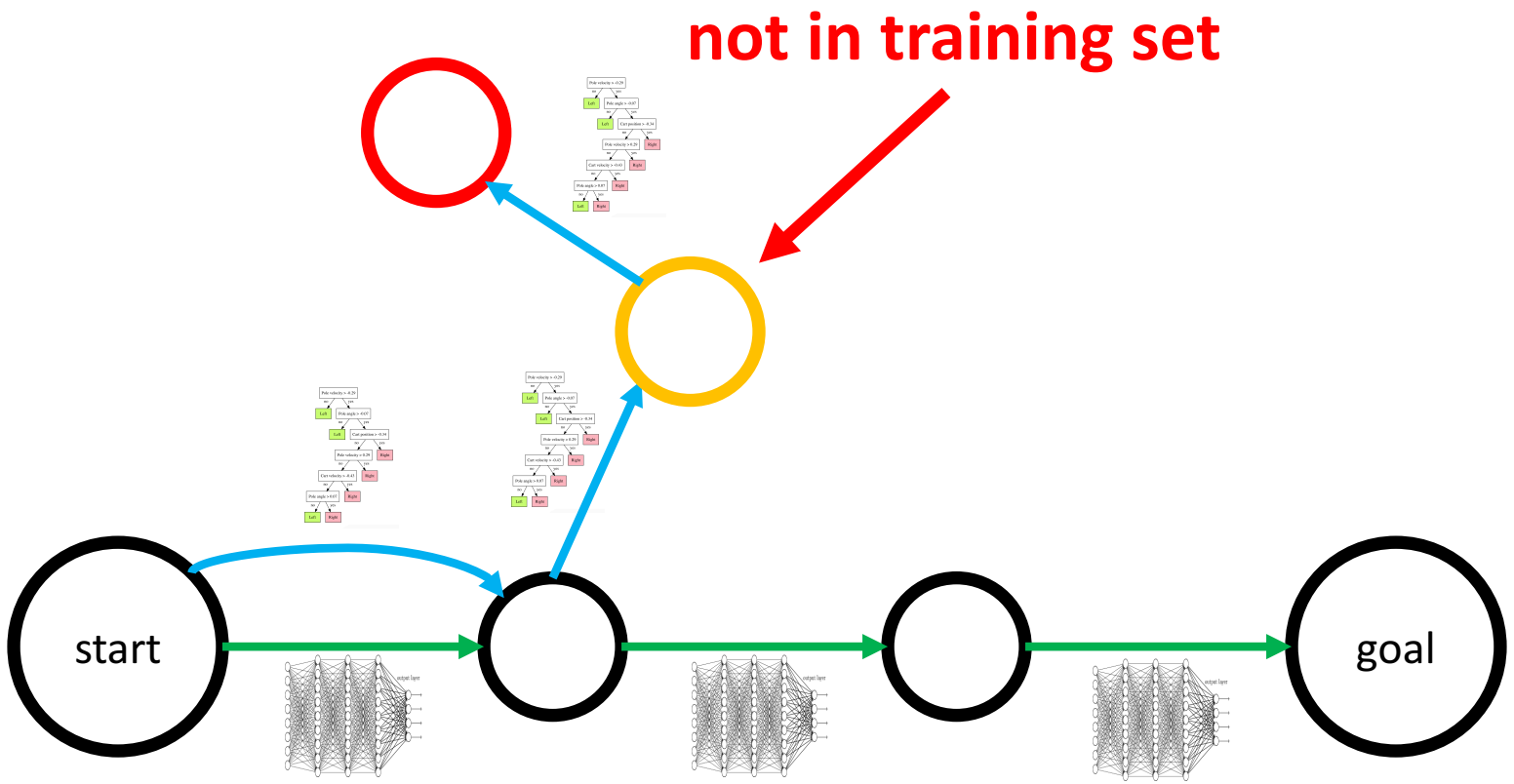
Step 2: Use NN to obtain actions



Step 3: Use supervised learning to train a decision tree

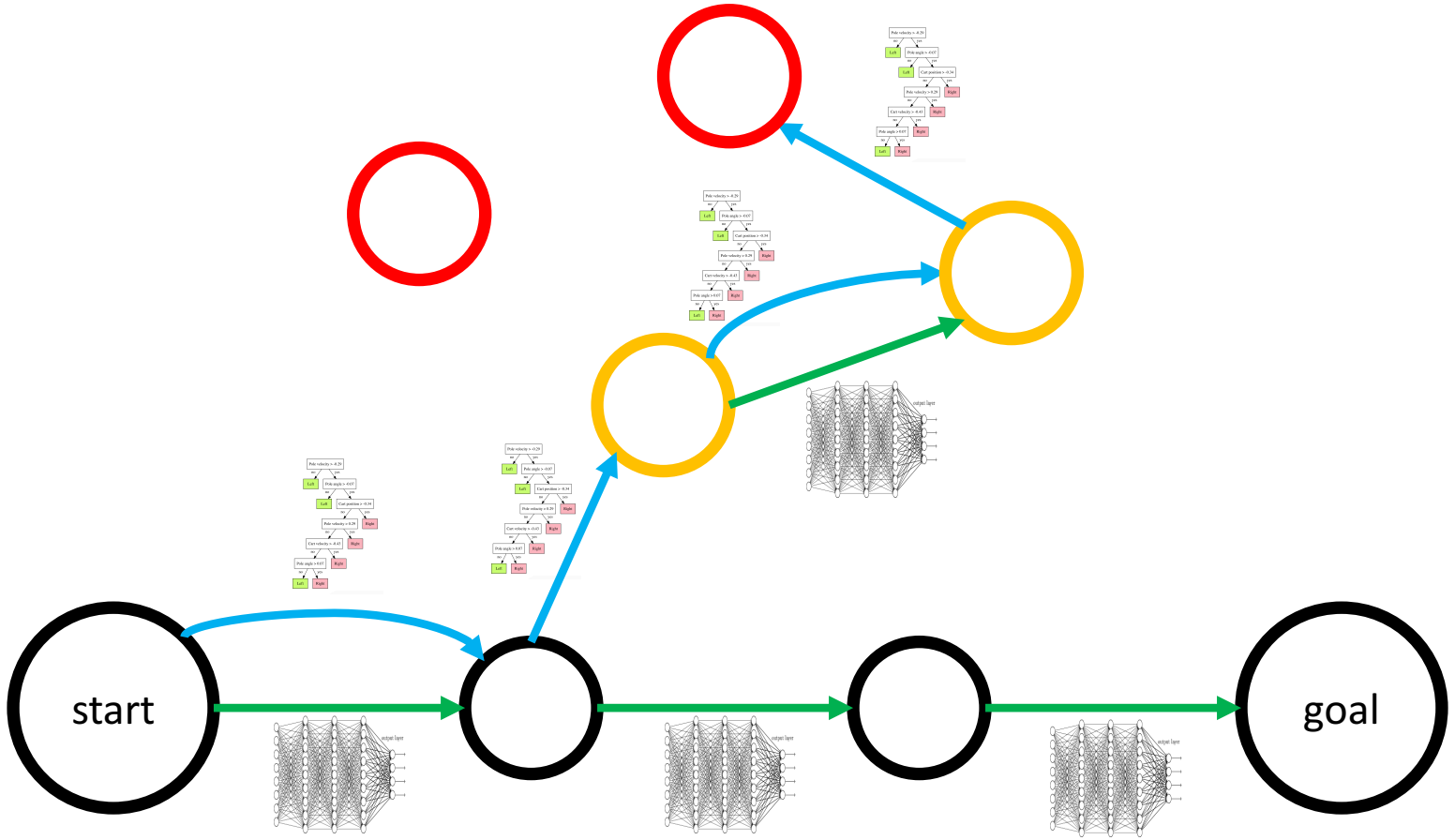
Ross & Bagnell 2011

Imitation Learning (Supervised)



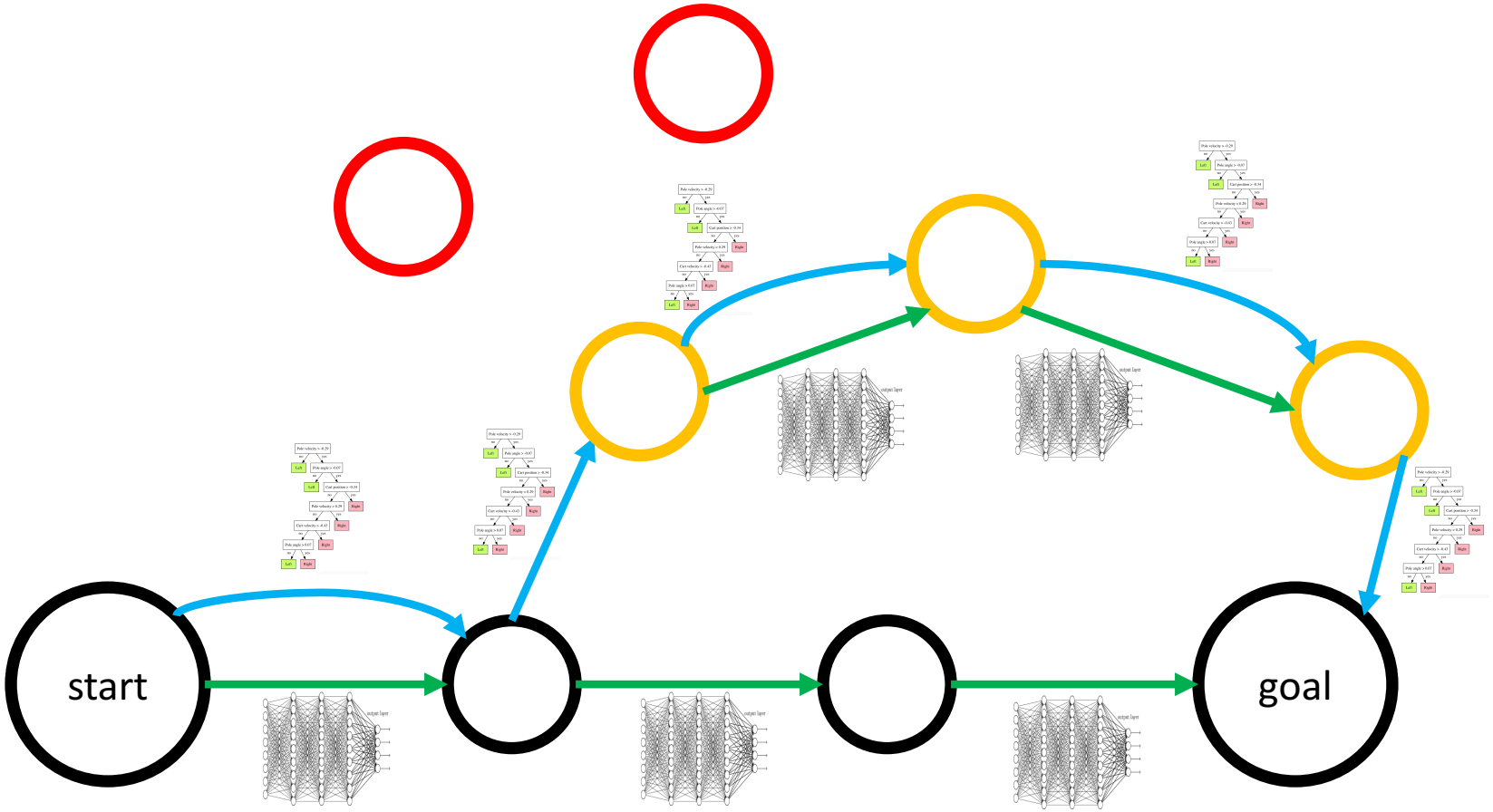
Ross & Bagnell 2011

Imitation Learning (Dataset Aggregation)



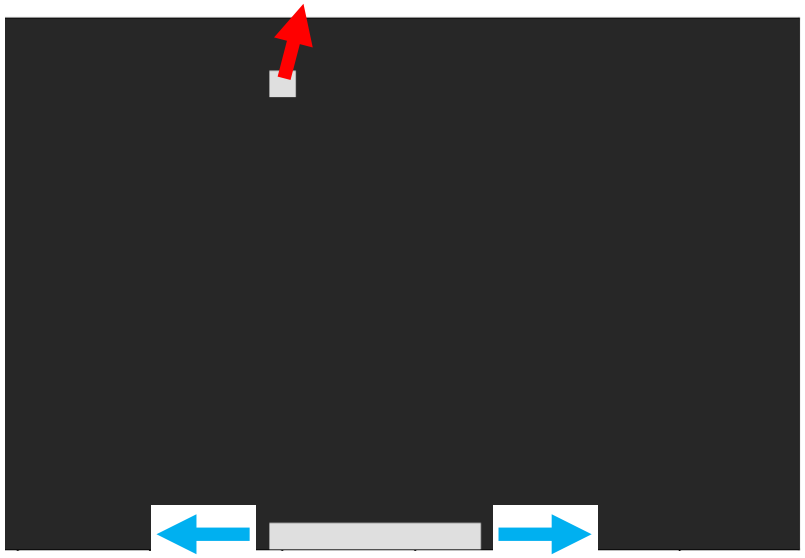
Ross & Bagnell 2011

Imitation Learning (Dataset Aggregation)

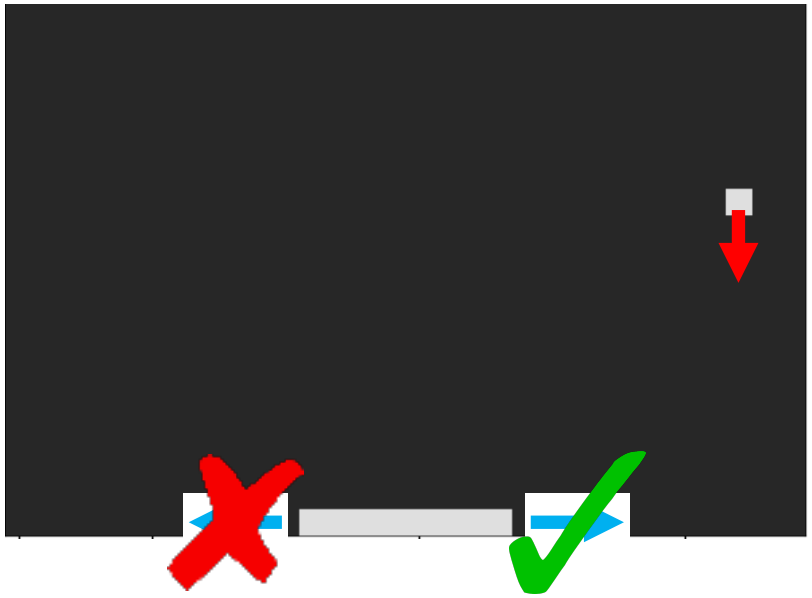


Ross & Bagnell 2011

Problem: Critical Actions



actions are similar



must move right!

Solution: Leverage $\boldsymbol{Q}$-Function

$Q(s, a) =$ “how good is action a in state s?”

Q-Dagger Algorithm

- Dataset Aggregation (Dagger) treats all state-action pairs equally:

$$\pi^* = \arg\min_{\pi} \sum_{(s,a)\in D} \mathbb{I}[\pi(s) = a]$$

- Q-Dagger weights state-action pairs by the Q-function

$$\pi^* = \arg\min_{\pi} \sum_{(s,a)\in D} Q(s,a) - Q\big(s,\pi(s)\big)$$

Theoretical Guarantees

Theorem. *For any $\delta > 0$, there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ such that*

$$J(\hat{\pi}) \leq J(\pi^*) + T\epsilon_N + \tilde{O}(1)$$

with probability at least $1 - \delta$, as long as $N = \tilde{\Theta}(\ell_{max}^2 T^2 \log(1/\delta))$.

Viper

- Extends Q-Dagger to extract decision tree controllers

Algorithm 1 Decision tree policy extraction.

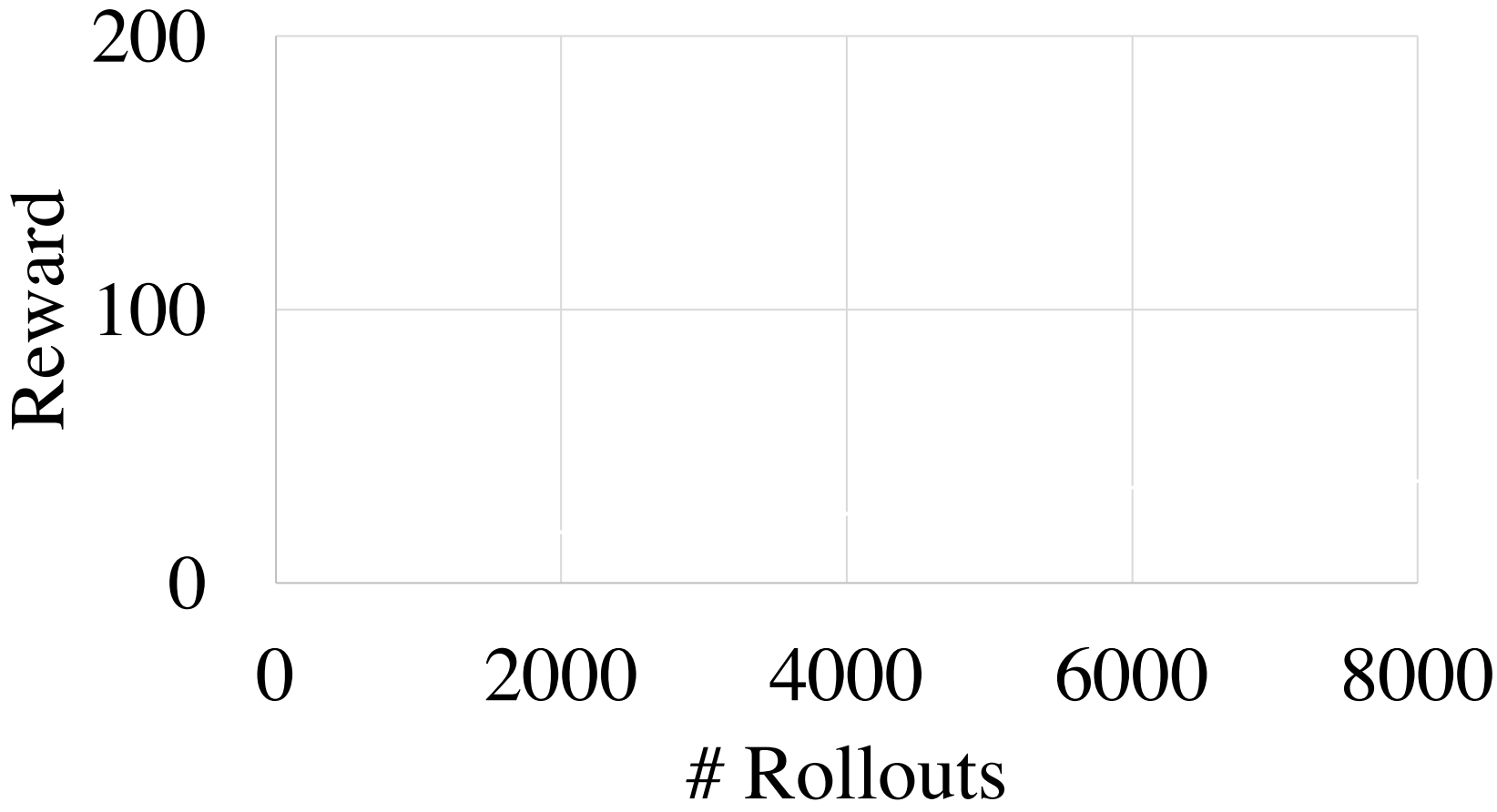
procedure VIPER($(S, A, P, R), \pi^*, Q^*, M, N$)
  Initialize dataset $\mathcal{D} \leftarrow \varnothing$
  Initialize policy $\hat{\pi}_0 \leftarrow \pi^*$
  for $i = 1$ **to** N **do**
    Sample M trajectories $\mathcal{D}_i \leftarrow \{(s, \pi^*(s)) \sim d^{(\hat{\pi}_{i-1})}\}$
    Aggregate dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$
    Resample dataset $\mathcal{D}' \leftarrow \{(s, a) \sim p((s, a)) \propto \tilde{\ell}(s)\mathbb{I}[(s, a) \in \mathcal{D}]\}$
    Train decision tree $\hat{\pi}_i \leftarrow \text{TrainDecisionTree}(\mathcal{D}')$
  end for
  return Best policy $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ on cross validation
end procedure

Evaluation

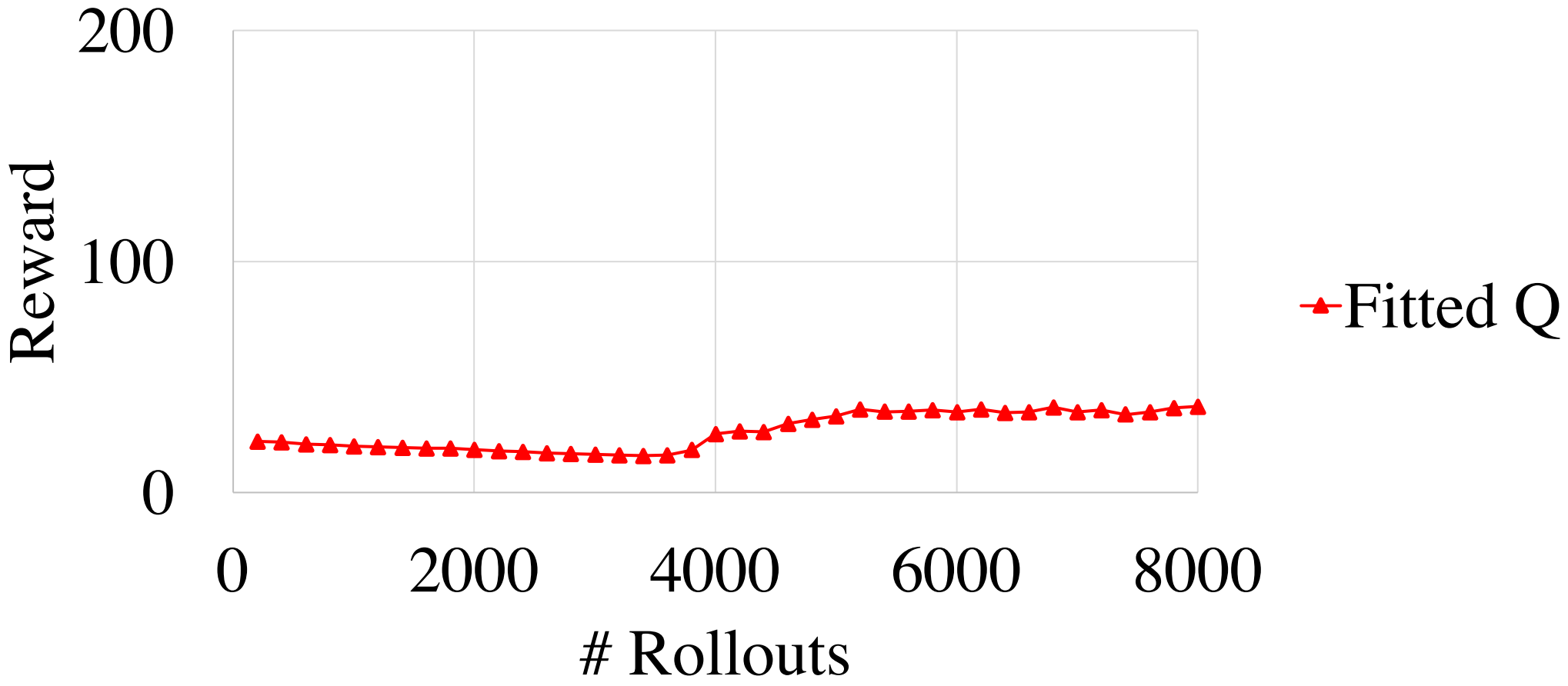
- Comparison to baselines
- Verified desirable properties

Comparison to Fitted $\boldsymbol{Q}$-Iteration (on Cart-pole)

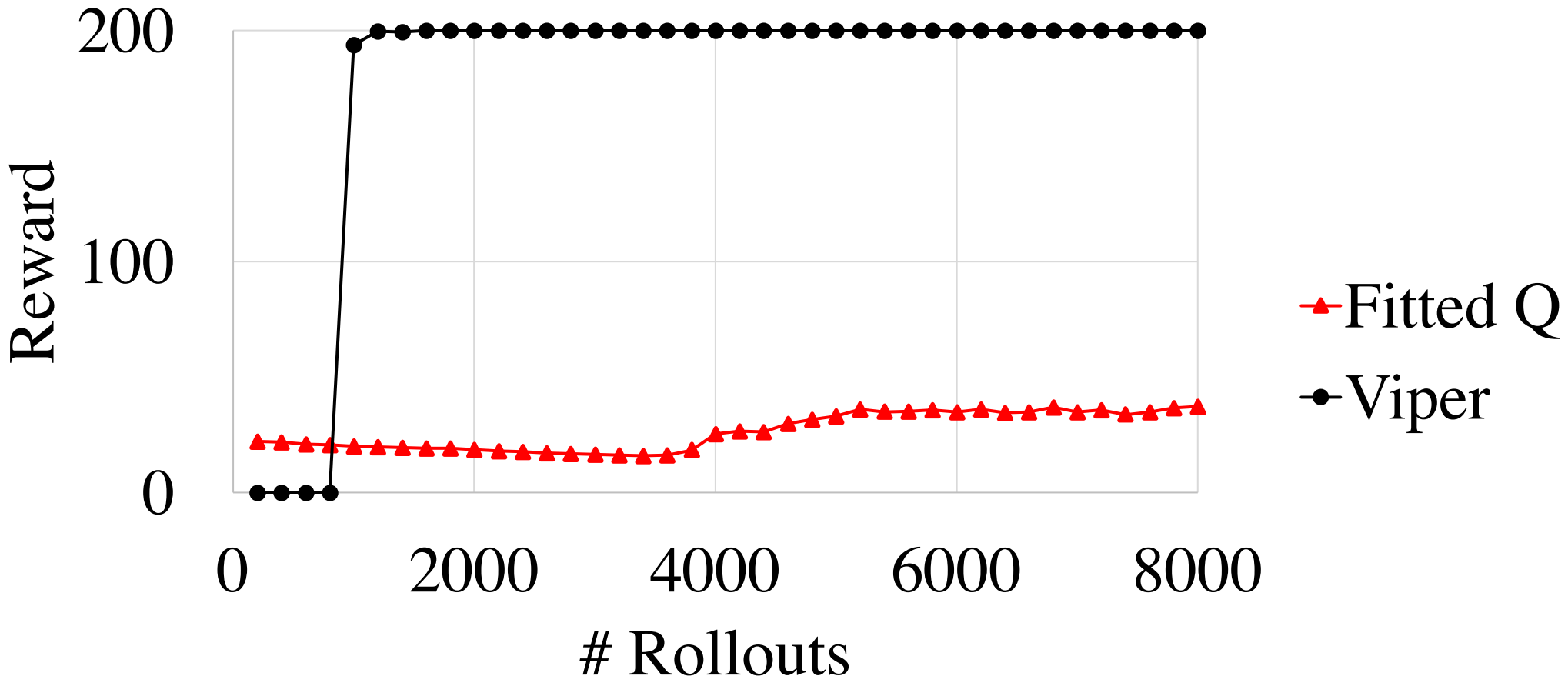
Comparison to Fitted $\boldsymbol{Q}$-Iteration (on Cart-pole)



Comparison to Fitted $\boldsymbol{Q}$-Iteration (on Cart-pole)

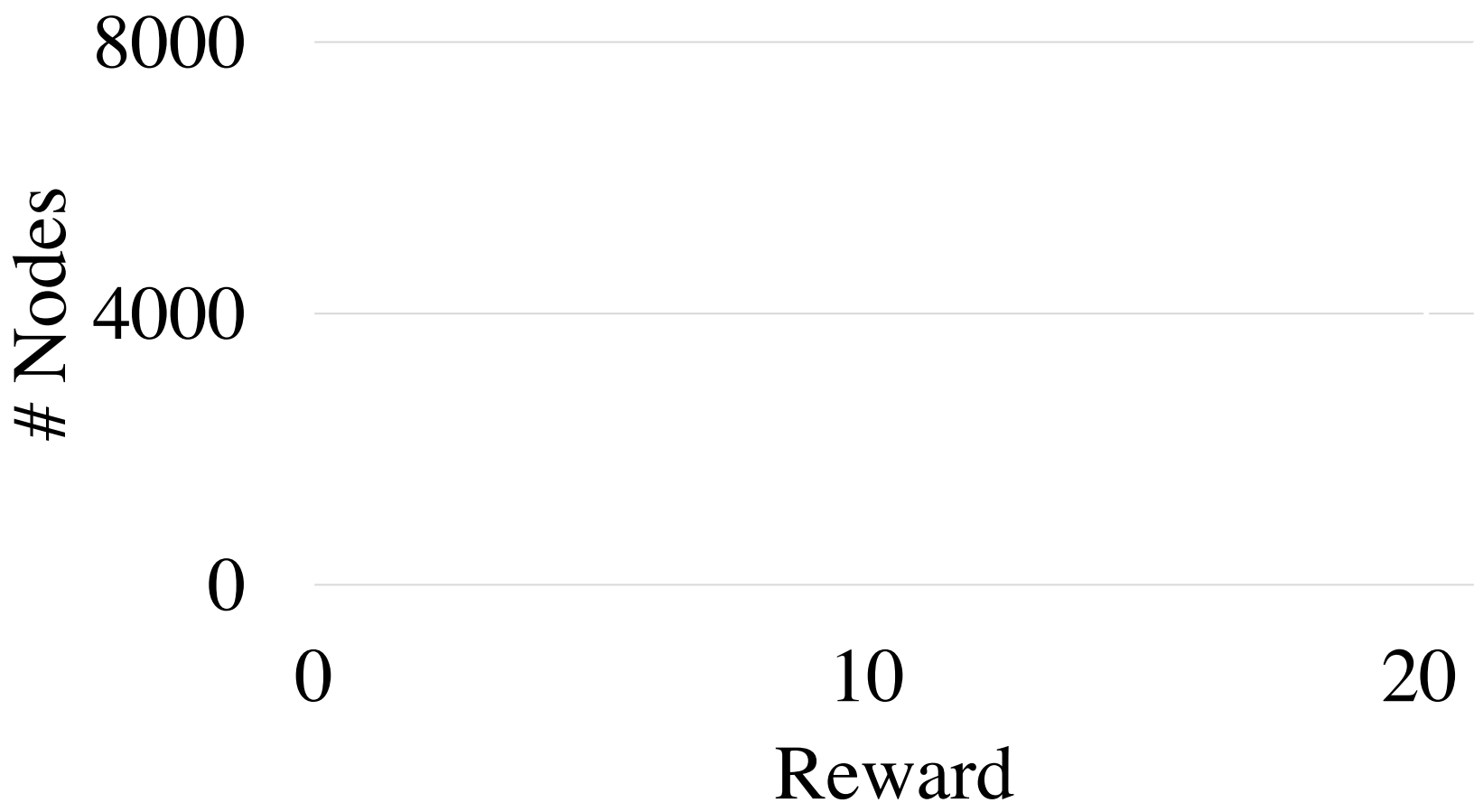


Comparison to Fitted $\boldsymbol{Q}$-Iteration (on Cart-pole)



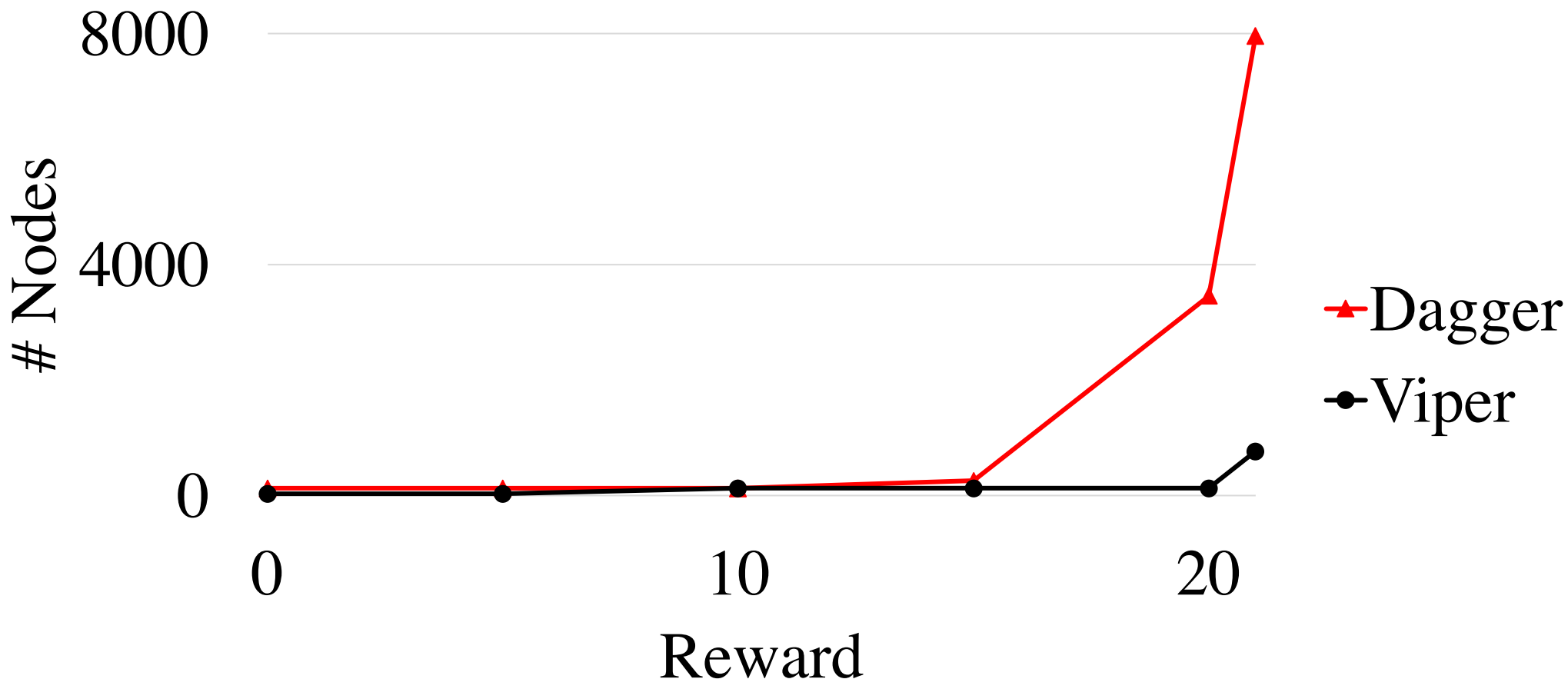
Comparison to Dagger (on Atari Pong)

Comparison to Dagger (on Atari Pong)

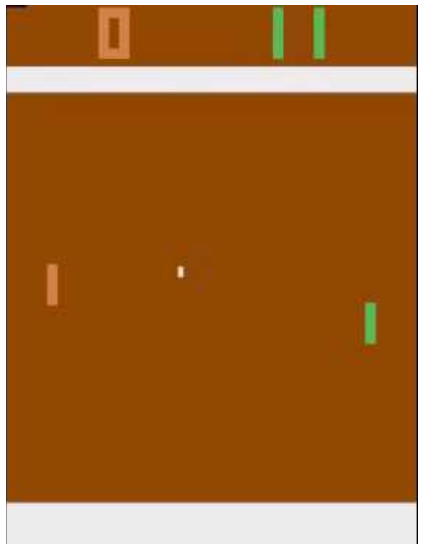


Comparison to Dagger (on Atari Pong)

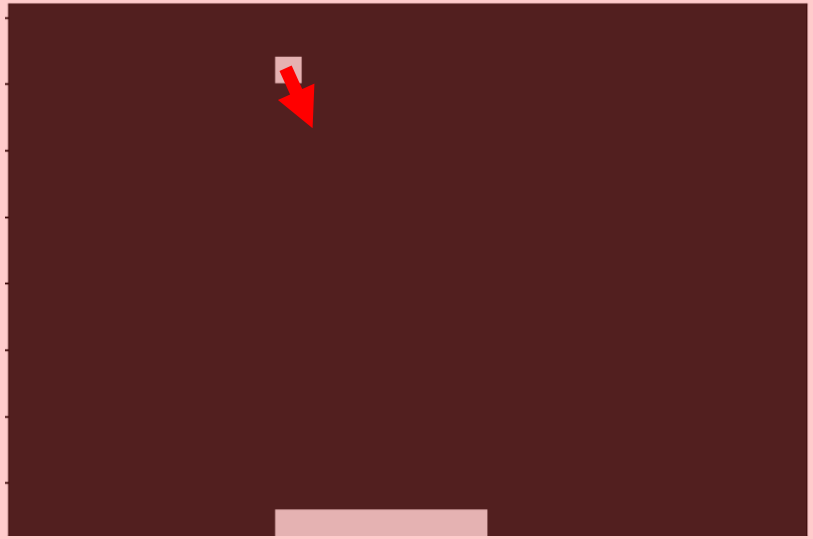
Comparison to Dagger (on Atari Pong)



Verification



robustness

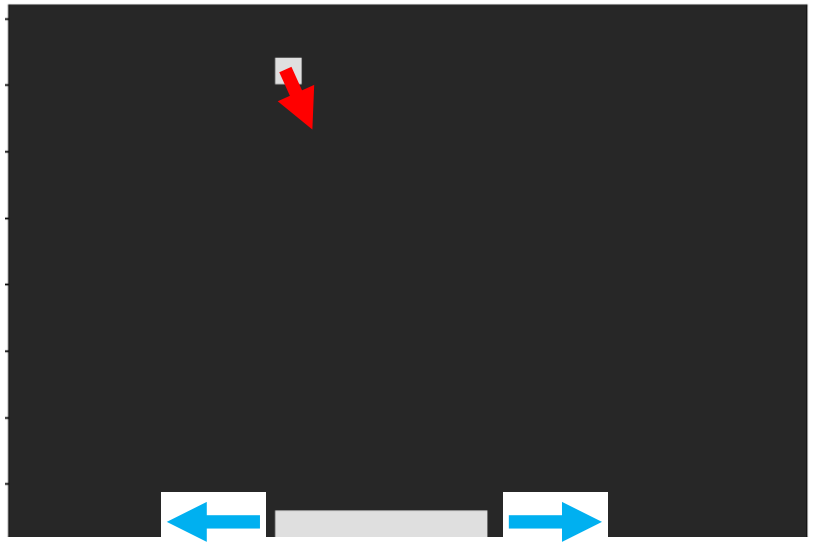


correctness
(never loses)

stable

Verifying Correctness of a Toy Pong Controller

- **Toy Pong**
  - states $= \mathbb{R}^5$
  - actions $= \{\text{left}, \text{right}, \text{stay}\}$
- **Neural network:**
  - trained using policy gradients
  - 600 neurons
- **Decision tree:**
  - extracted using Viper
  - 31 nodes



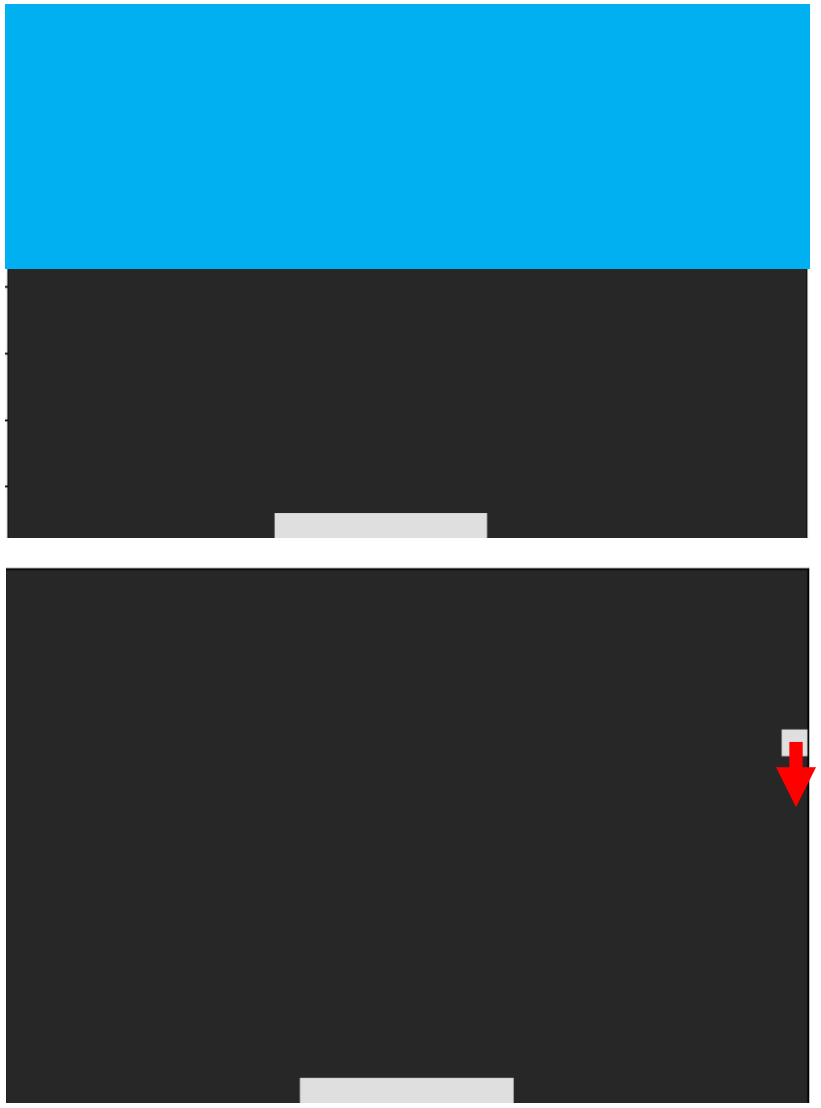
Verifying Correctness of a Toy Pong Controller

- **Inductive invariant:**

$$s(0) \in \text{blue} \Rightarrow s(t) \in \text{blue}$$

- **Verification algorithm**
  - dynamics are piecewise linear
  - SMT formula over linear arithmetic
  - solved by Z3 in < 5 seconds

- **Results:**
  - error when ball starts on the right
  - fixed when paddle is slightly longer!



Conclusion

Verifiability is critical to enabling application of deep reinforcement learning to safe-critical systems