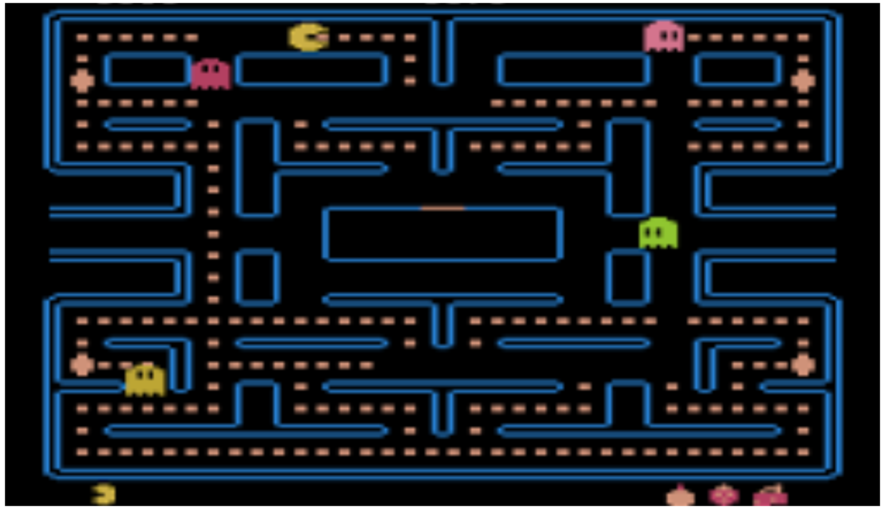
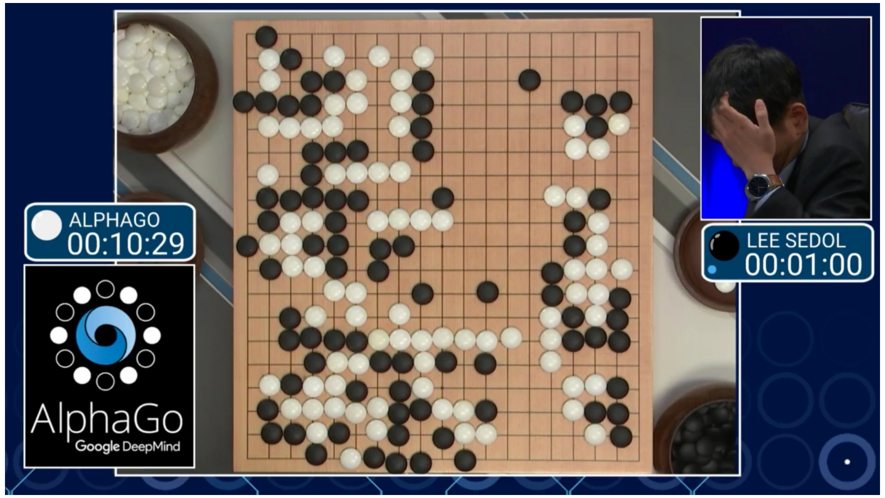
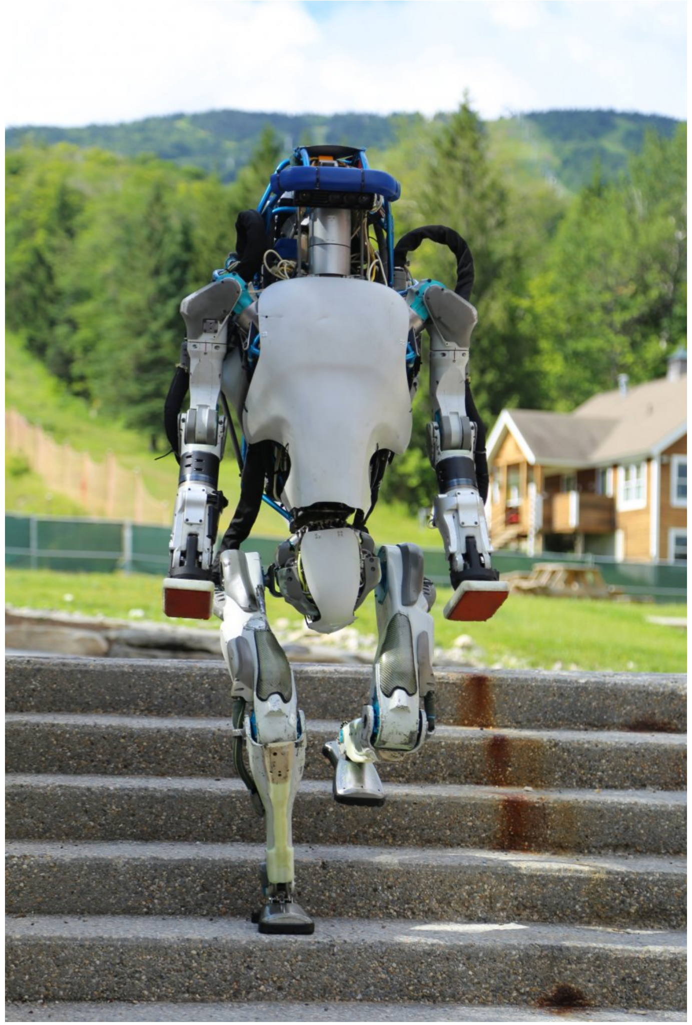
Verifiable Reinforcement Learning via Policy Extraction

Osbert Bastani, Yewen Pu, Armando Solar-Lezama

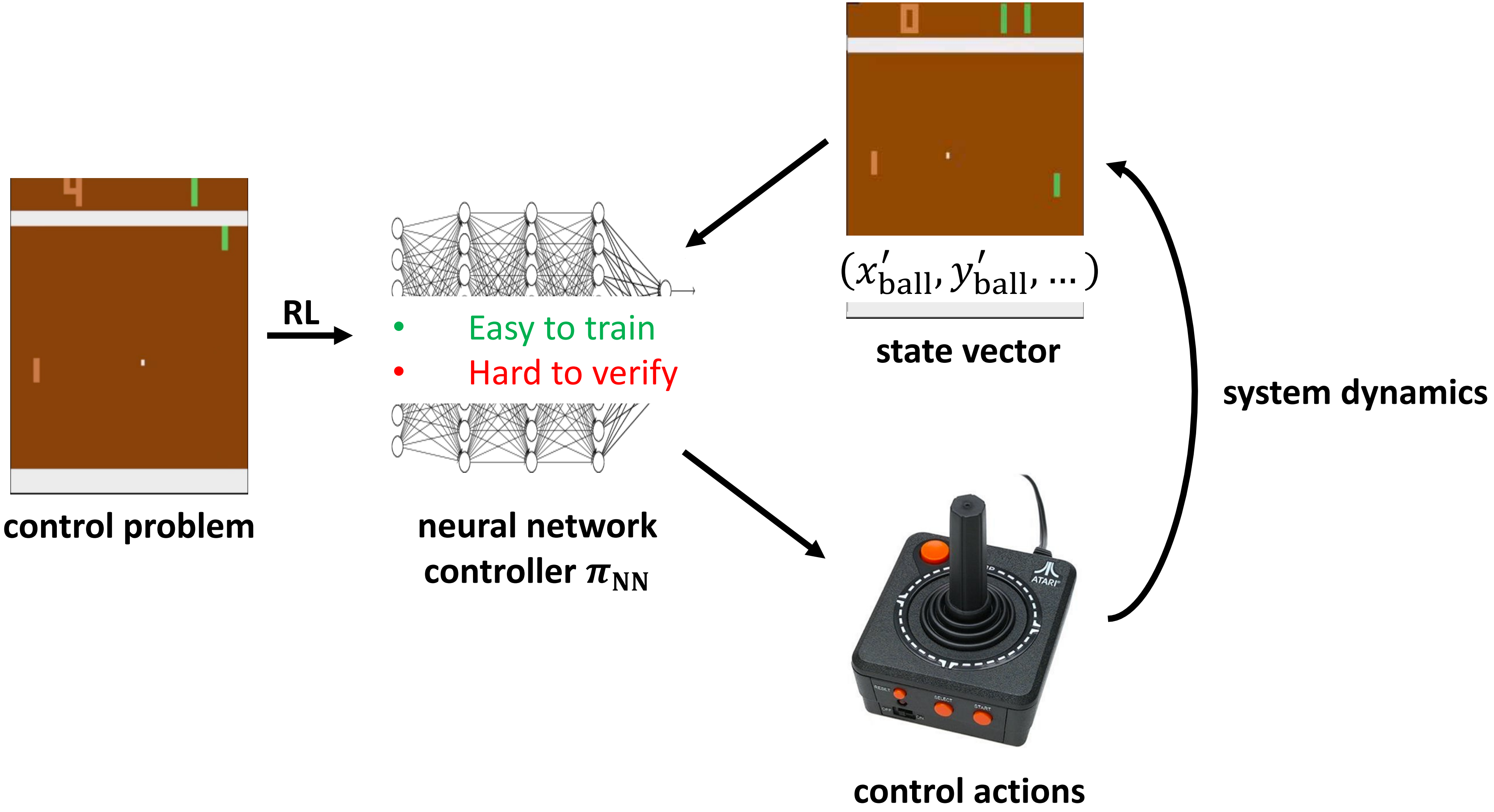
Deep Reinforcement Learning



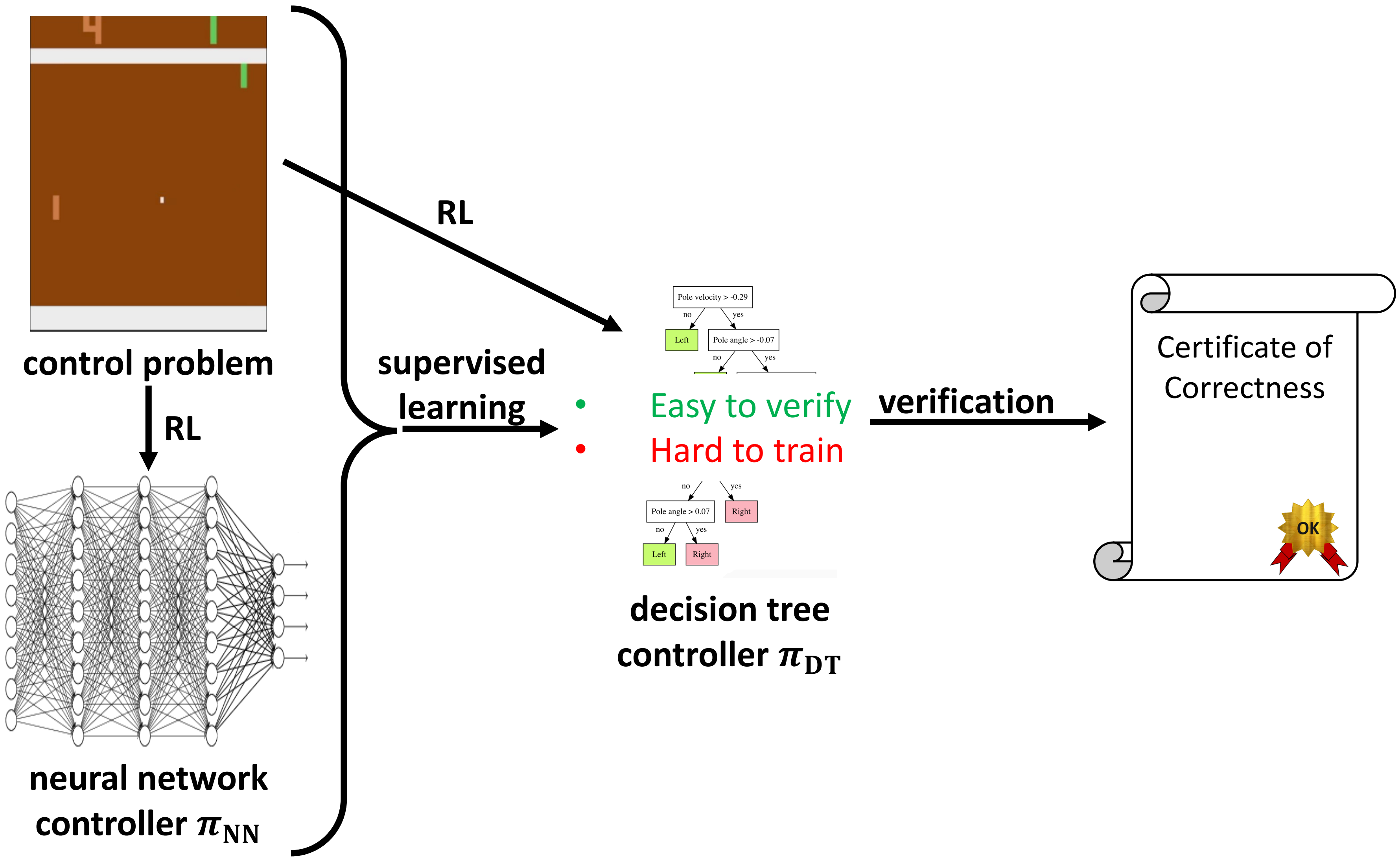




Deep Reinforcement Learning



Our Approach



Background

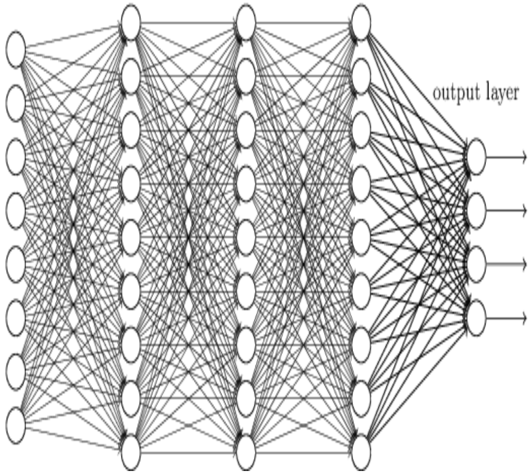
Imitation Learning

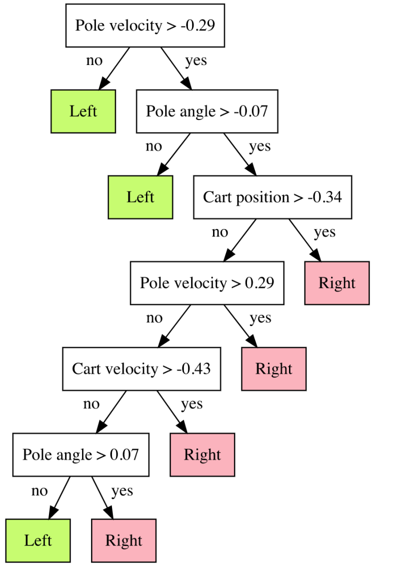
Demonstrations from Human Expert

Controller

Abbeel & Ng 2004

Imitation Learning



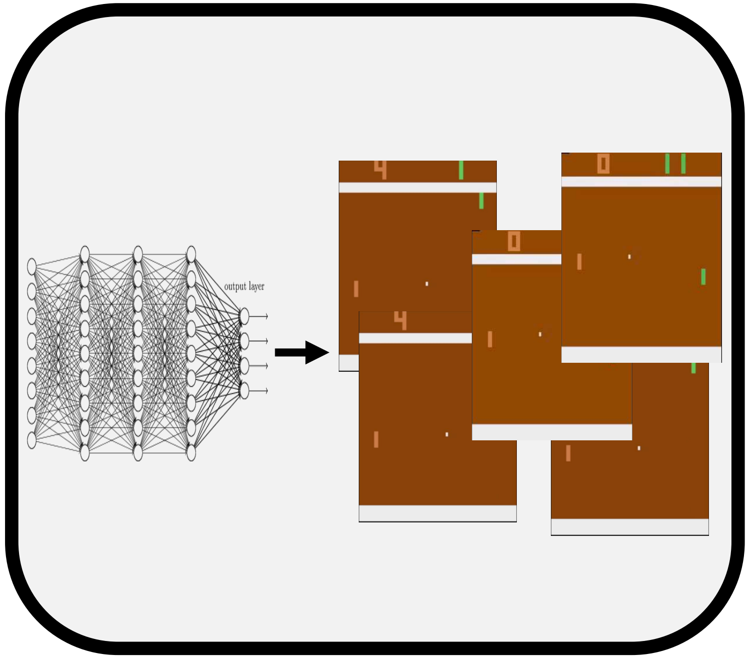


Demonstrations from Neural Network

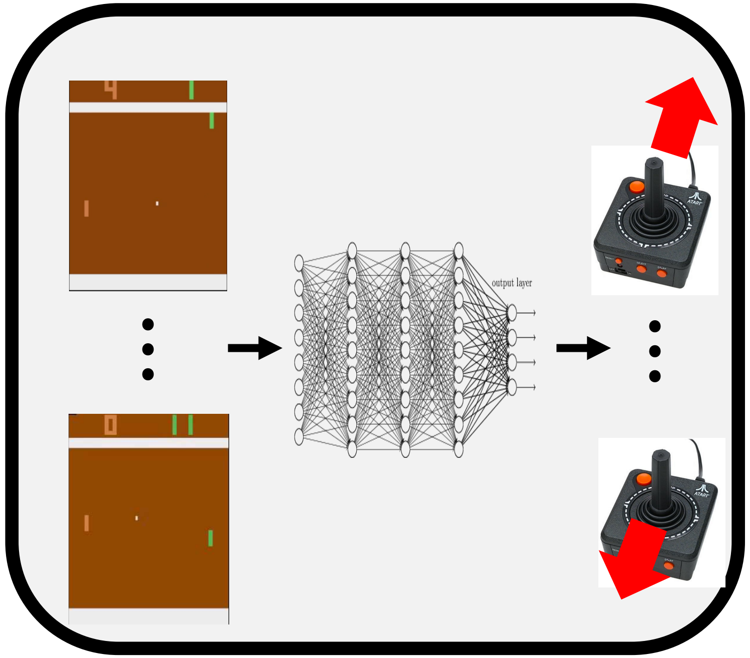
Decision Tree Controller

Abbeel & Ng 2004

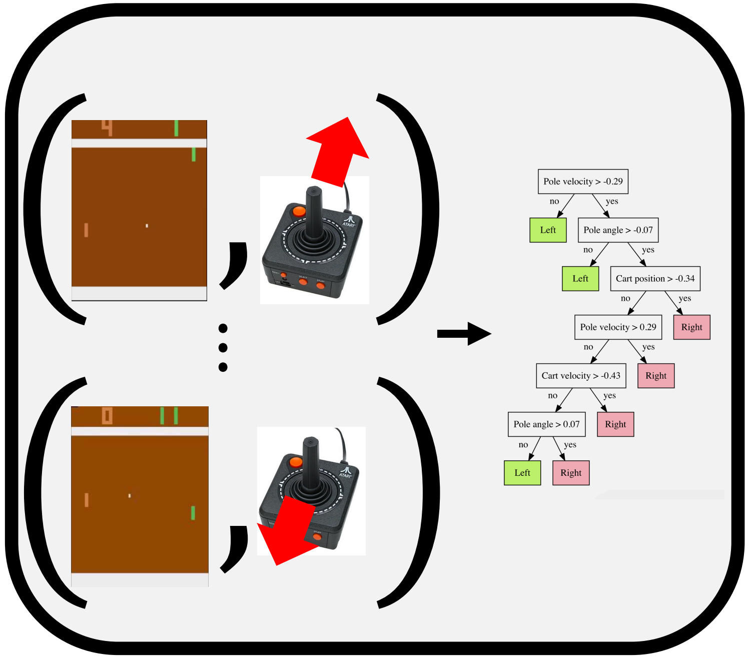
Imitation Learning



Step 1: Use NN to generate states



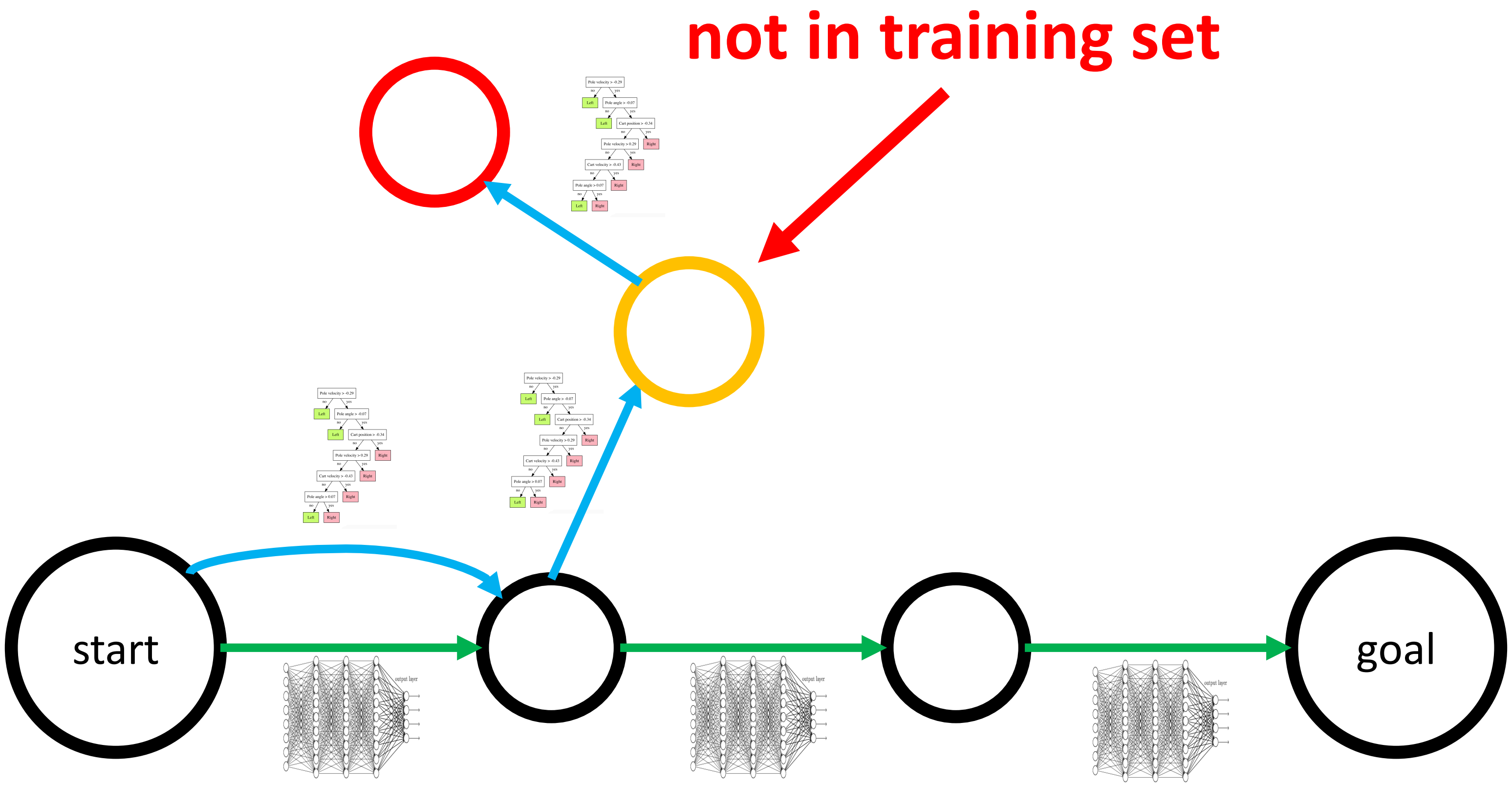
Step 2: Use NN to obtain actions



Step 3: Use supervised learning to train a decision tree

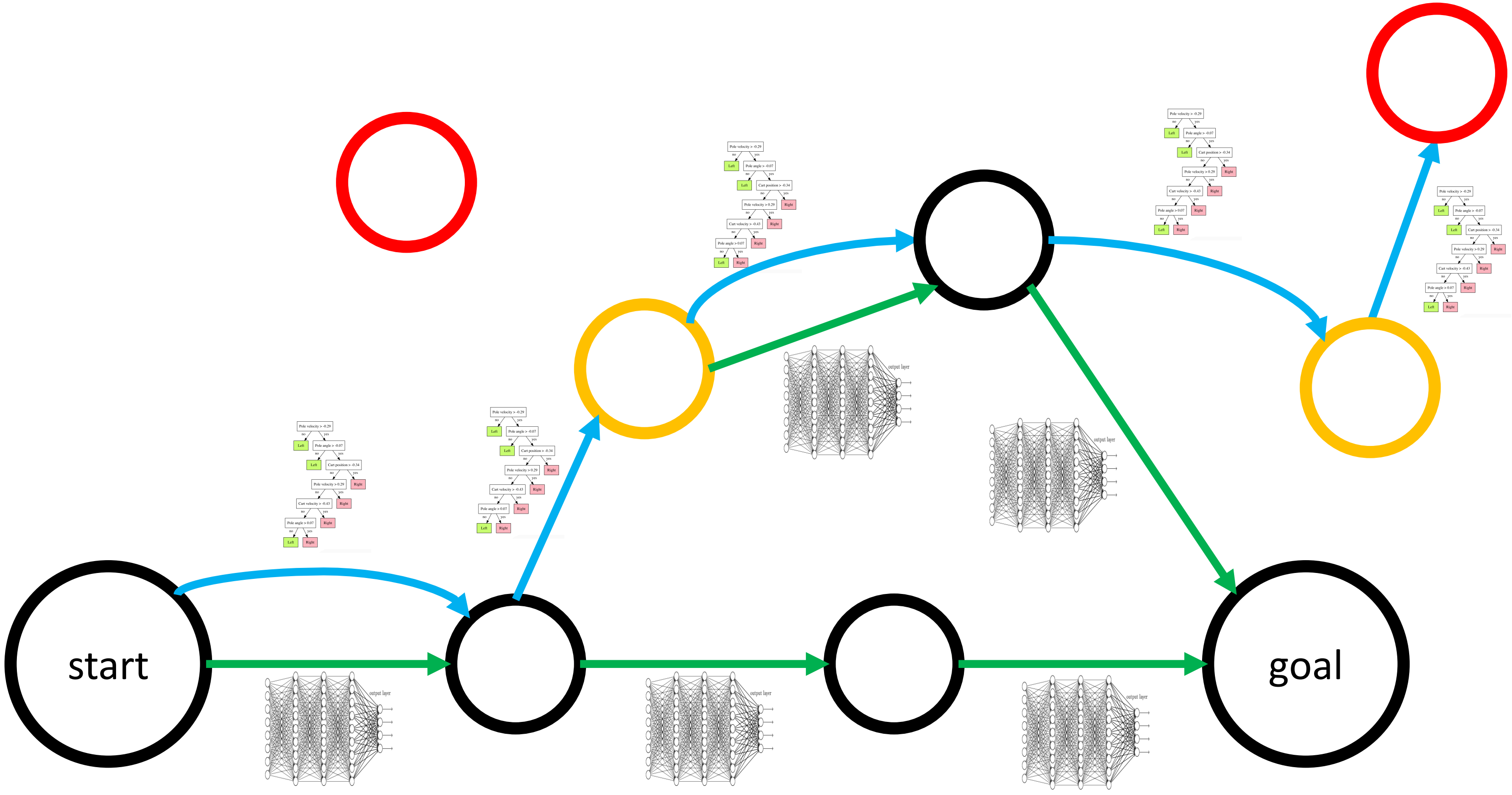
Ross & Bagnell 2011

Imitation Learning



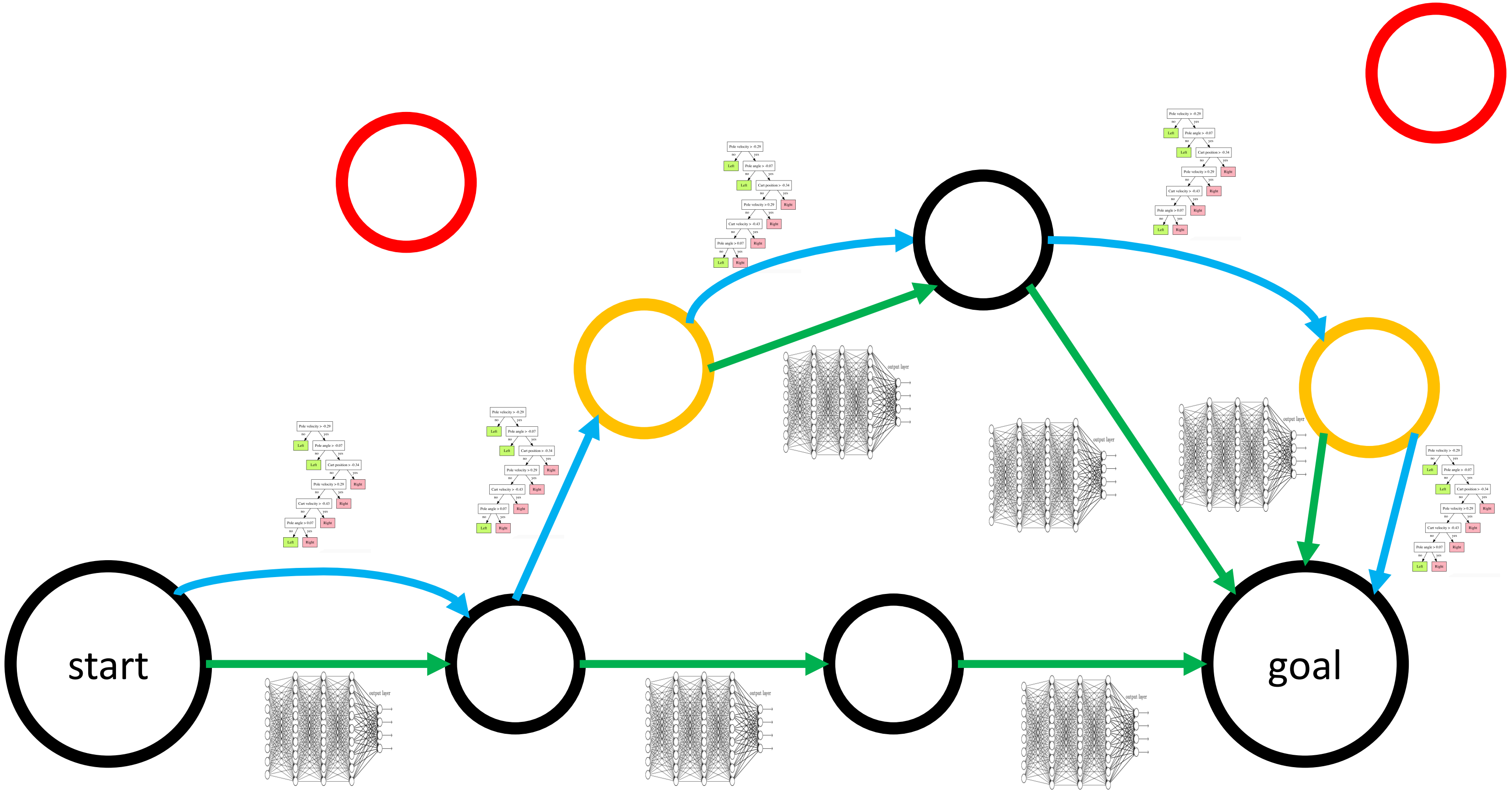
Ross & Bagnell 2011

Dataset Aggregation (DAgger)



Ross & Bagnell 2011

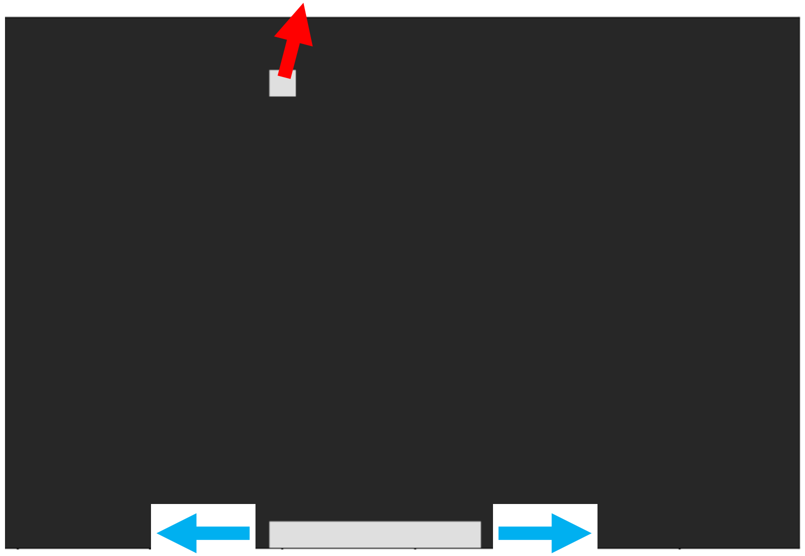
Dataset Aggregation (DAgger)



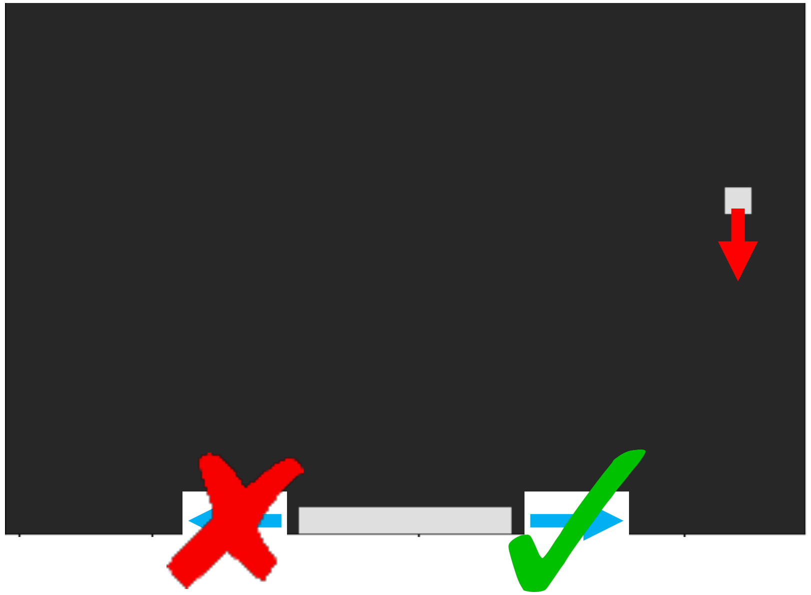
Ross & Bagnell 2011

Viper Algorithm

Insight: Critical States



actions are similar
(non-critical state)



must move right!
(critical state)

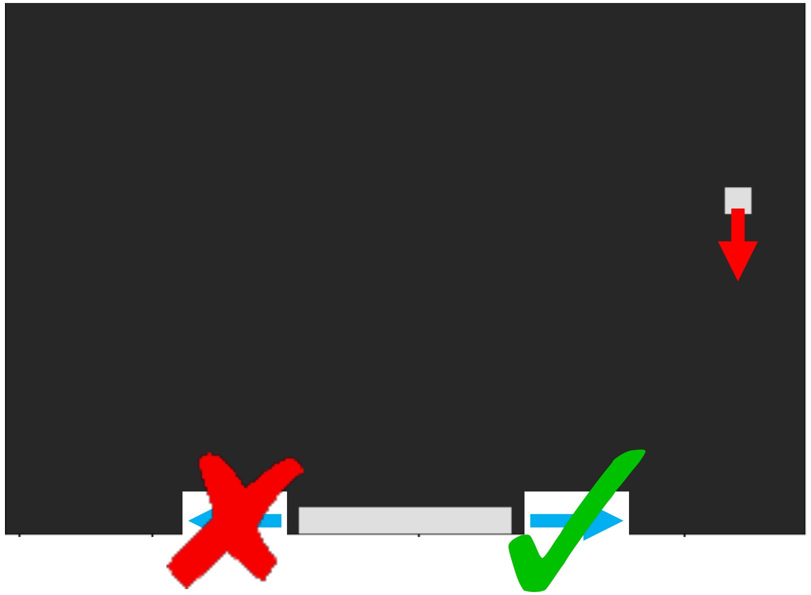
Our Approach: Leverage the $\boldsymbol{Q}$-Function

$$Q(s, a) = \text{“how good is action } a \text{ in state } s\text{?”} \in \mathbb{R}$$

Our Approach: Leverage the Q-Function

non-critical state (low priority)

$$\underbrace{Q\big(s, \pi_{\mathrm{NN}}(s)\big)}_{\text{optimal } Q \text{ value}} \approx \underbrace{\min_{a \in A} Q(s, a)}_{\text{worst-case } Q \text{ value}}$$



critical state (high priority)

$$\underbrace{Q\big(s, \pi_{\mathrm{NN}}(s)\big)}_{\text{optimal } Q \text{ value}} \gg \underbrace{\min_{a \in A} Q(s, a)}_{\text{worst-case } Q \text{ value}}$$

Viper Algorithm

- DAgger treats all state-action pairs equally:

$$\pi_{\mathrm{DT}} = \arg\min_{\pi} \sum_{s \in D} \mathbb{I}[\pi(s) = \pi_{\mathrm{NN}}(s)]$$

- Viper weights state-action pairs by the Q-function:

$$\pi_{\mathrm{DT}} = \arg\min_{\pi} \sum_{s \in D} \Big(\underbrace{Q(s, \pi_{\mathrm{NN}}(s))}_{\text{optimal } Q \text{ value}} - \underbrace{\min_{a' \in A} Q(s, a')}_{\text{worst-case } Q \text{ value}} \Big) \mathbb{I}[\pi(s) = \pi_{\mathrm{NN}}(s)]$$

Theoretical Guarantees

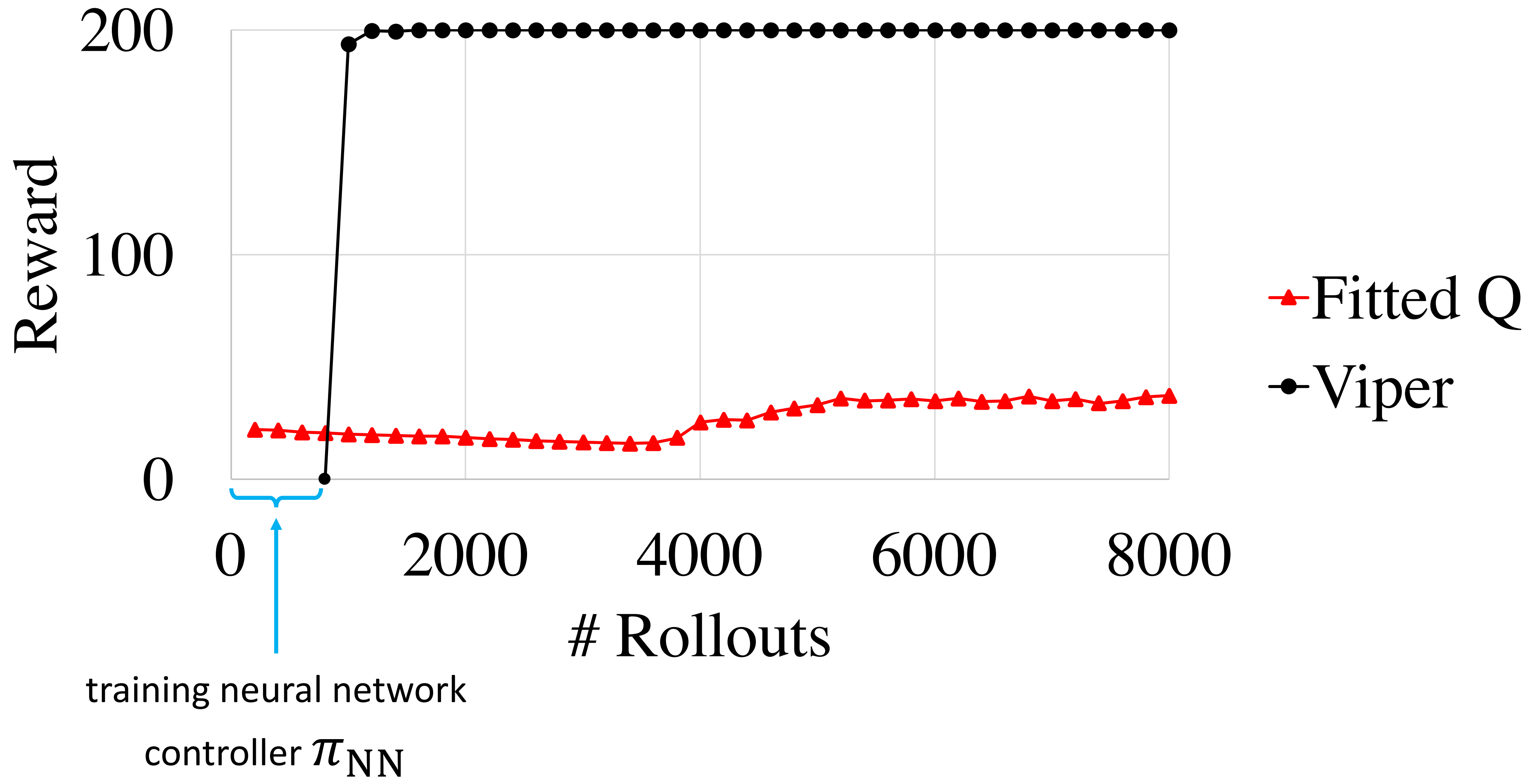
Theorem. *For any* $\delta > 0$*, there exists a policy* $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ *such that*

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

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

Evaluation

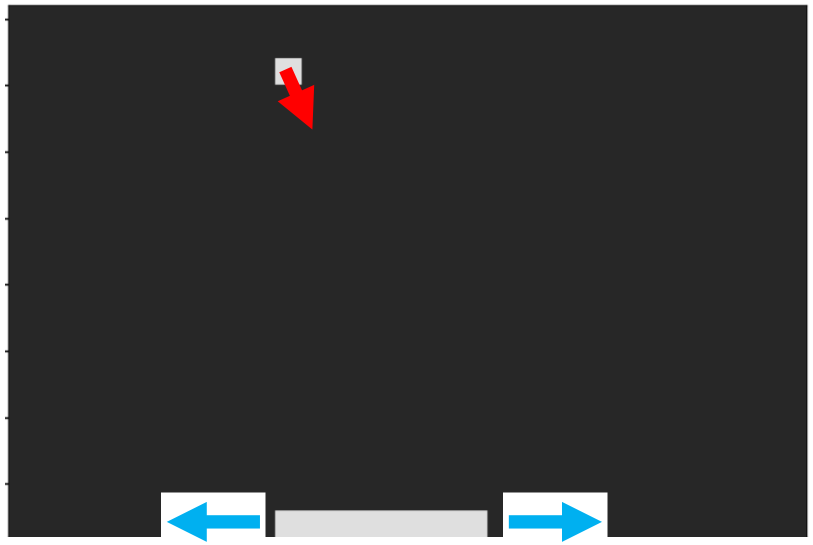
vs. Decision Trees via RL (on Cart-Pole)



vs. to DAgger (on Atari Pong)

Verifying Correctness of a Toy Pong Controller

- **Toy Pong**
  - $\text{states} = \mathbb{R}^5$
  - $\text{actions} = \{\text{left}, \text{right}, \text{stay}\}$
- **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