Assessing the Robustness of Deep RL Algorithms

Michael L. Littman

Brown University

Department of Computer Science

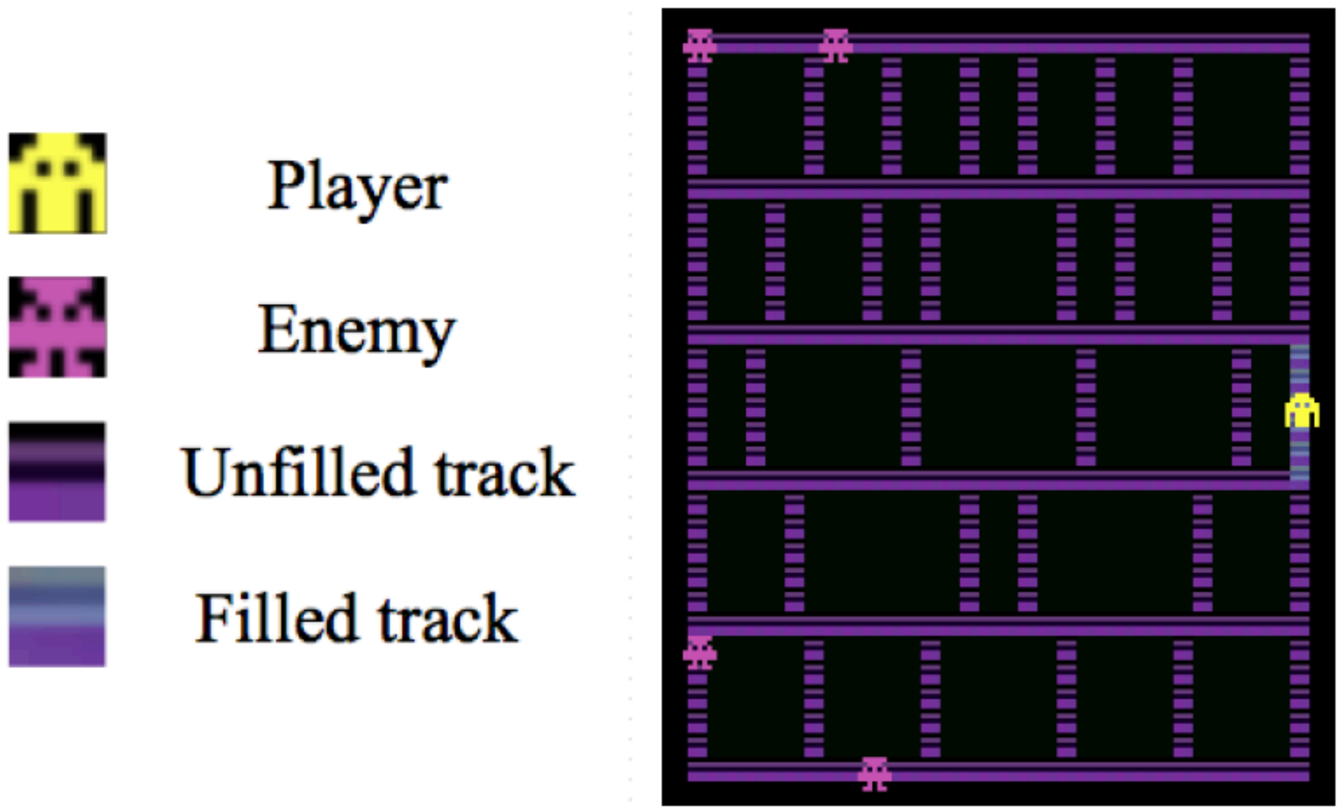
Background

- Started off interested in explainable RL: Why does DQN choose the moves it does Atari?
- Ended up wondering if any explanation at all is possible...

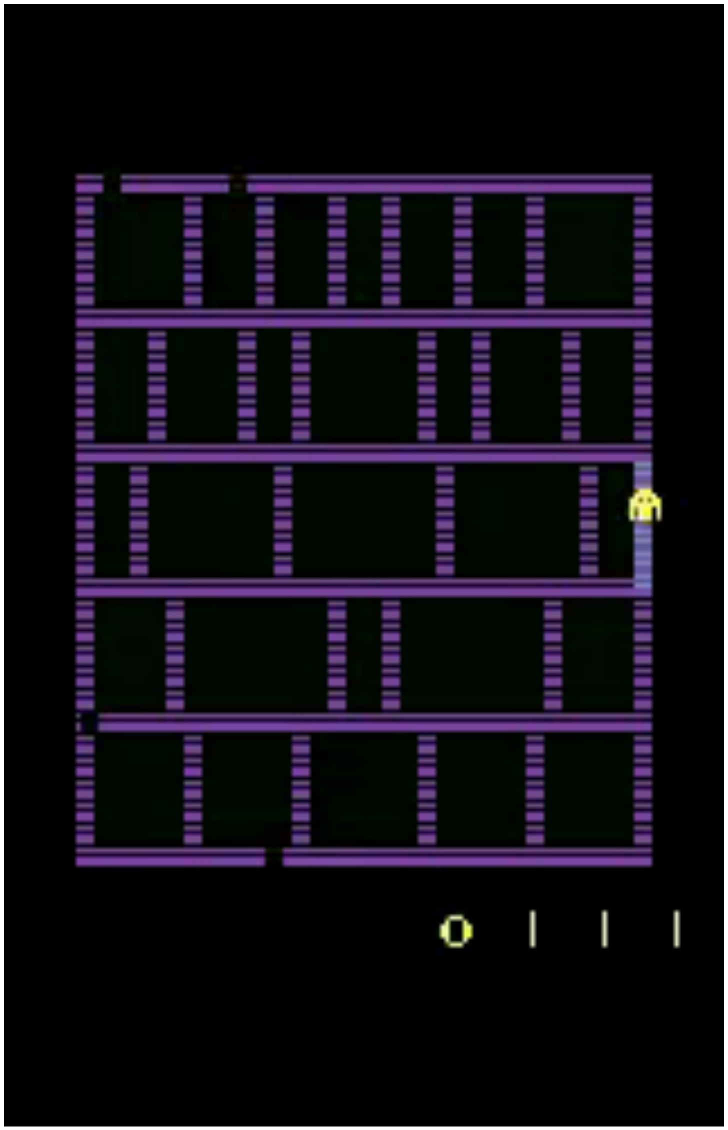
- Punchline: Generalizing Q values is hard.

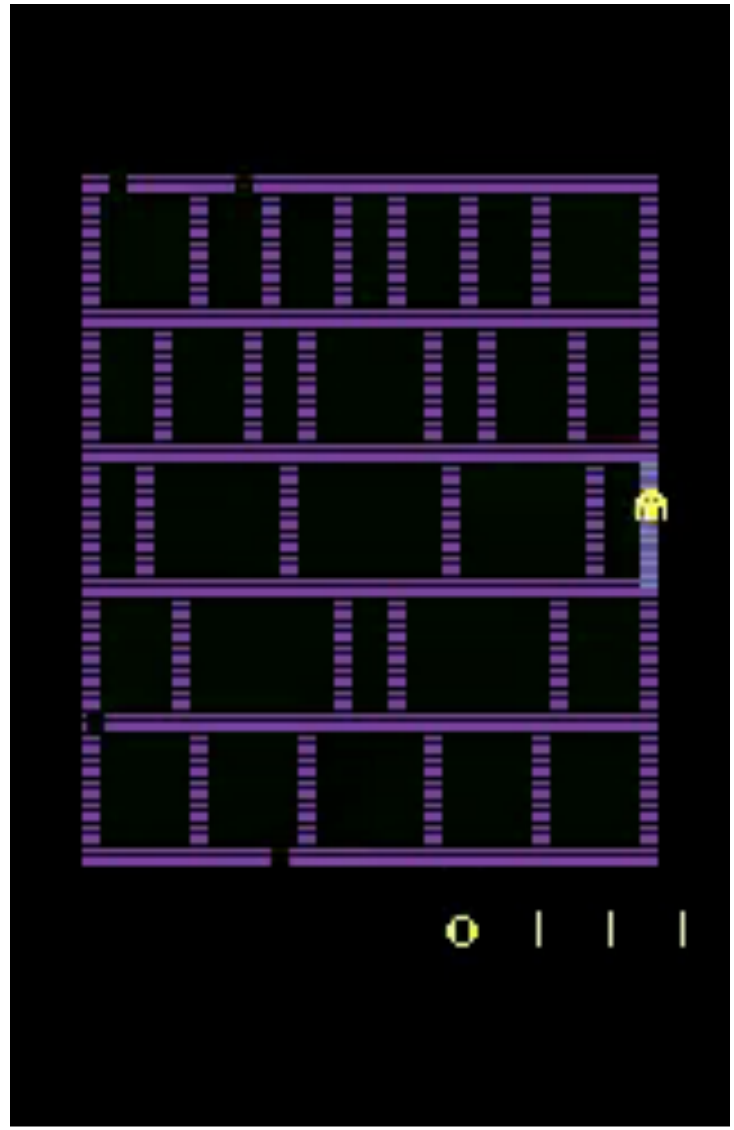
Case Study 1: Amidar

(Witty, Lee, Tosch, Atrey, Littman, Jensen 18)



Fancy Footwork





How Does It Do It? Explanation

Explain why explain matters.

- Provides assurances. Can we trust it?
- Suggests improvements.

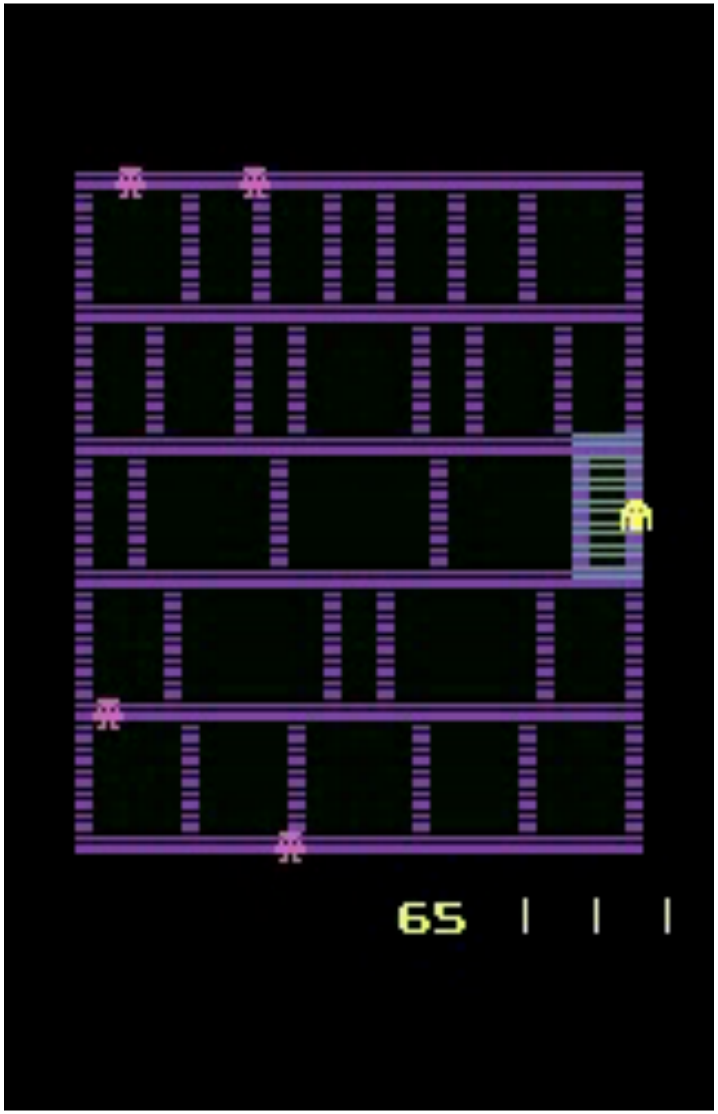
Expecting:

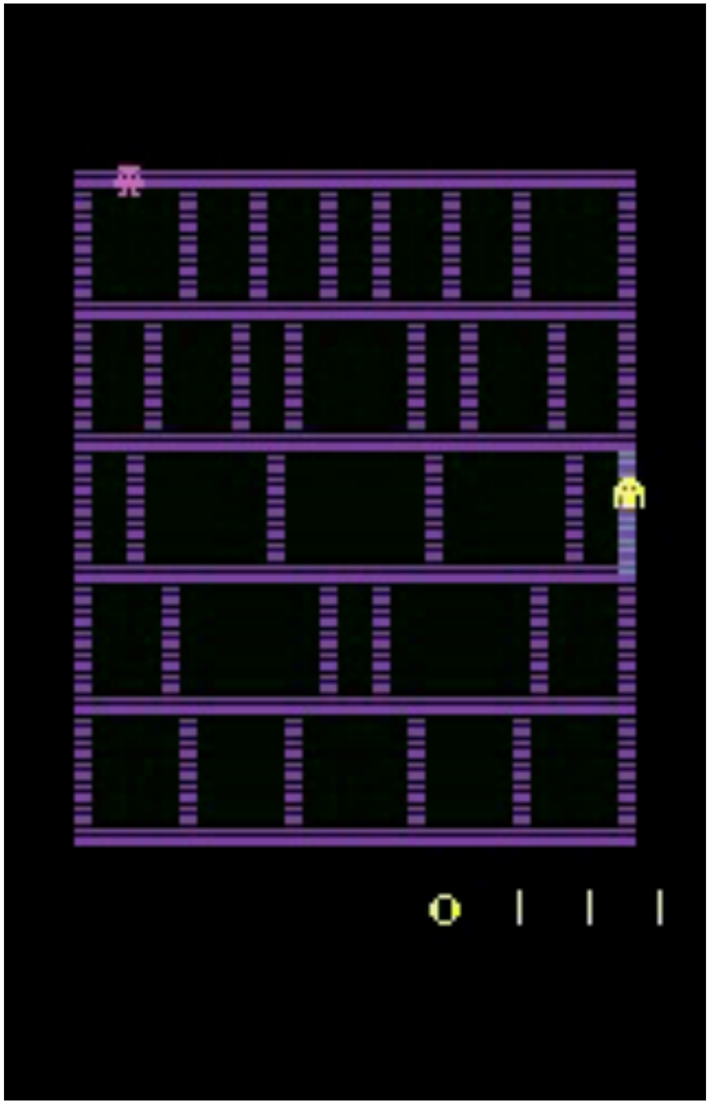
- Avoid enemies, seek out unfilled lines.
- We know it didn't learn about the corners.
- Evasive patterns? Priorities for filling board?

Methodology:

- Intervene and observe result.

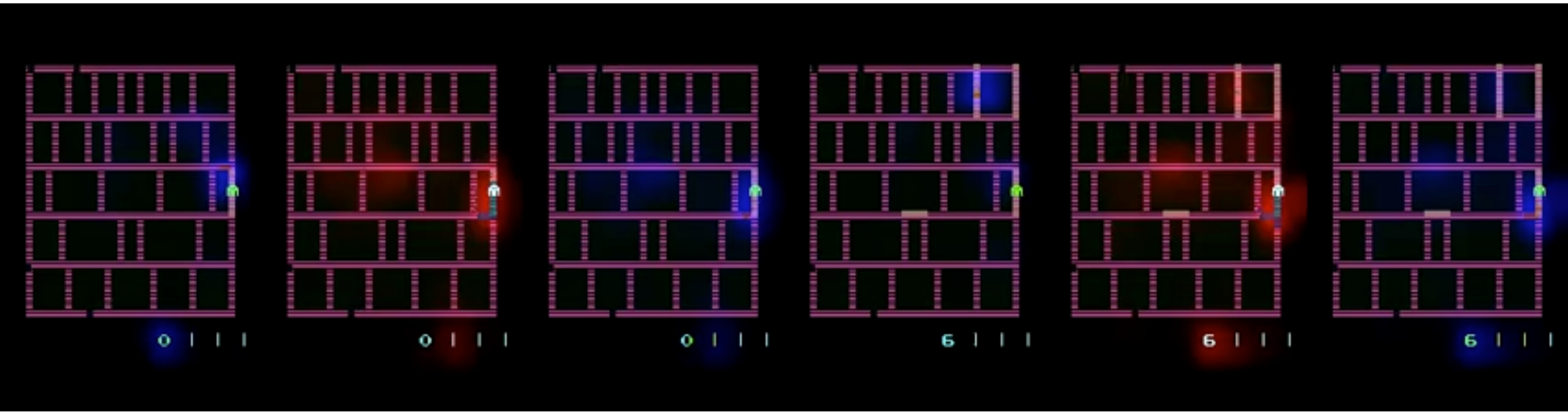
Examples





Saliency Plot

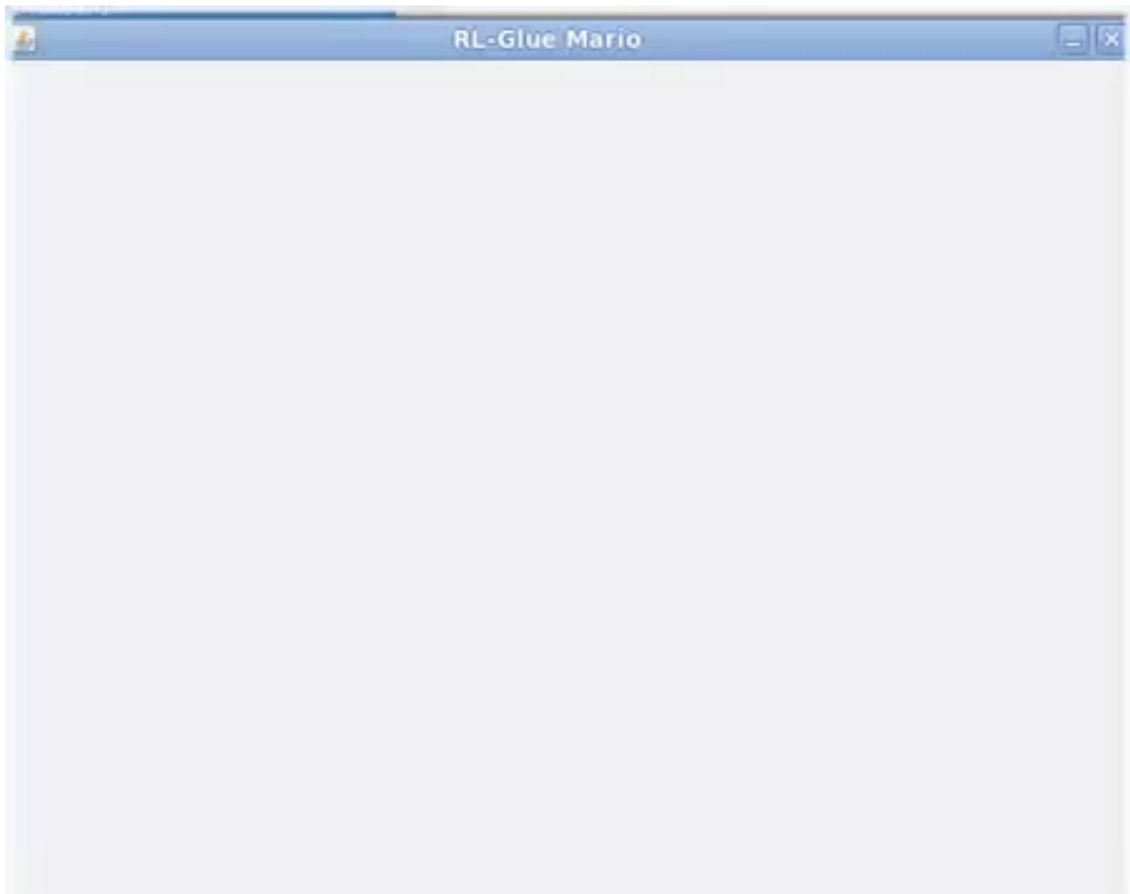
(Greydanus, Koul, Dodge, Fern 17)



- What makes big changes in action choice or value prediction if blurred out? What does the learned network pay attention to?
- Player and score.

Memorized movement

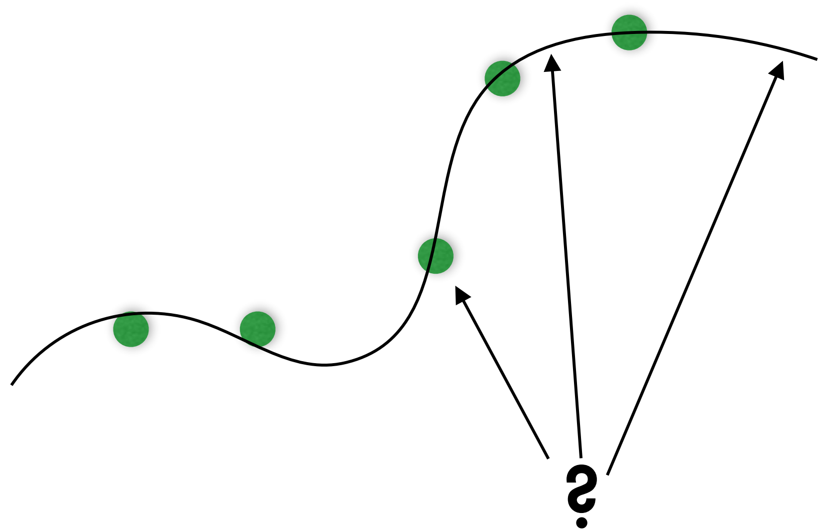
- Instead of learning principles, learned a path.



(Goschin et al. 12)

Step Back: Assessing Learning

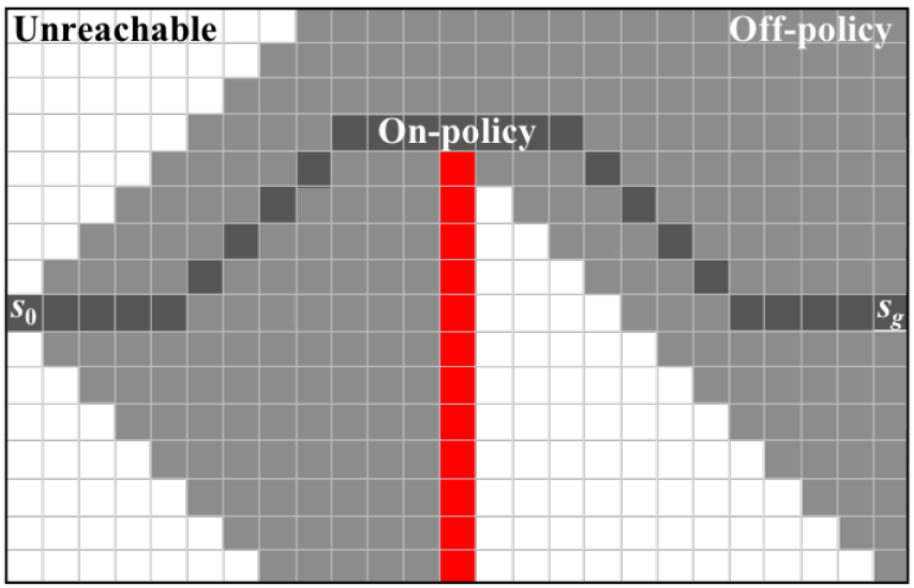
- Supervised learning:
  - Training examples
  - <u>Interpolation</u>: Examples from same distribution
  - Extrapolation: Out of sample.



Weakest to strongest measures of generalization.

Step Back: Assessing Learning

- Reinforcement learning:
  - Training examples ⇢ <u>On-policy states</u>
  - Interpolation ⇢ Off-policy states
  - Extrapolation ⇢ Unreachable states



Weakest to strongest measures of generalization.

Generating Testing States

Off-policy

- Stochasticity. k off-policy actions (k-OPA) in sequence.
- Human agents. What situations do people encounter? (Starts? Swaps in the middle.)
- Synthetic agents. Separately trained/built agents used to produce states.

Unreachable (via intervening on latent state):

- Existential: Enemies, line fill
- Parameterized: Position of player, enemies

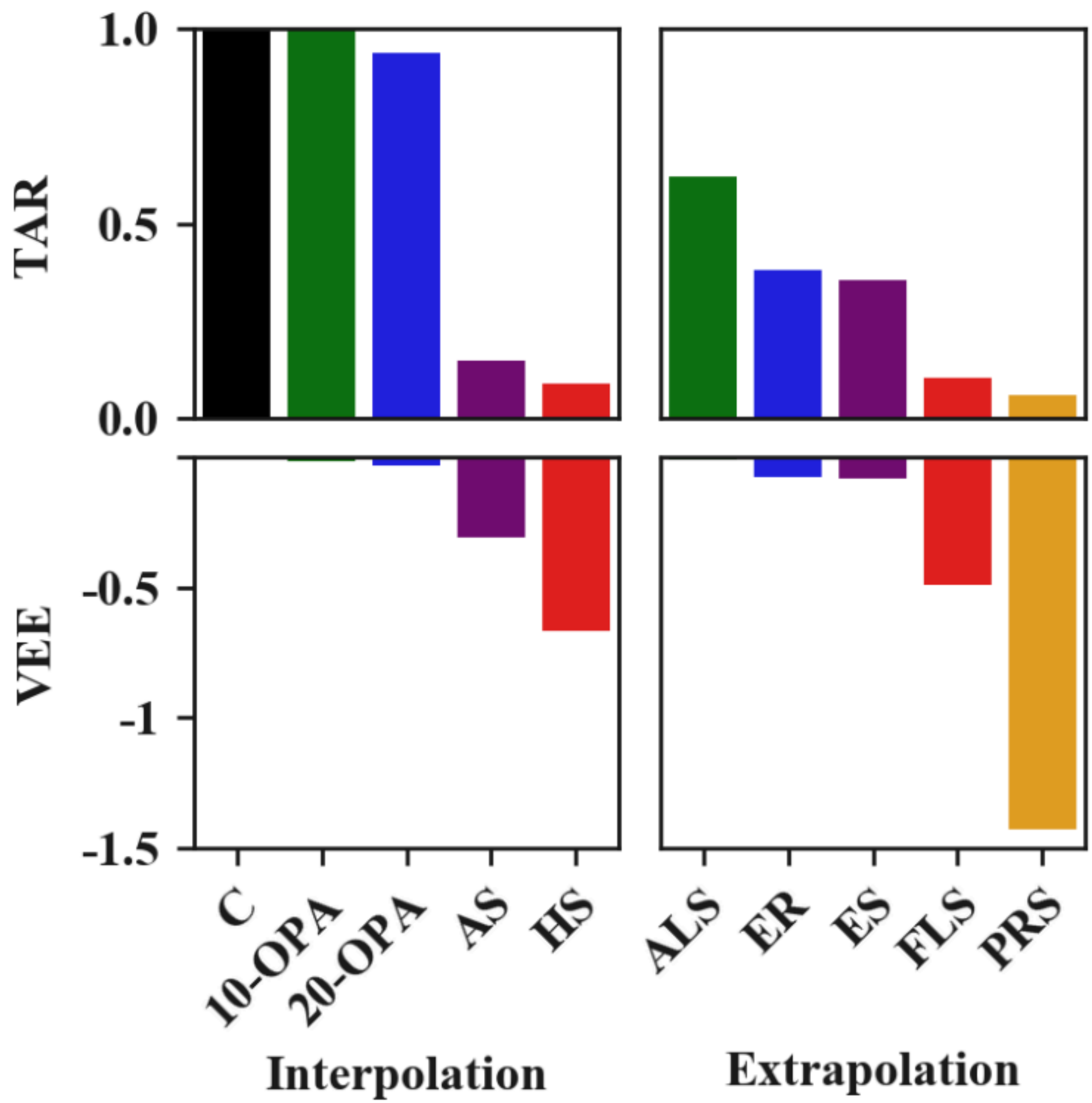
Evaluation Metrics

- VEE: Value estimation error
  - Internally, network predicts future reward. Compare to actual reward obtained.
- TAR: Total accumulated reward

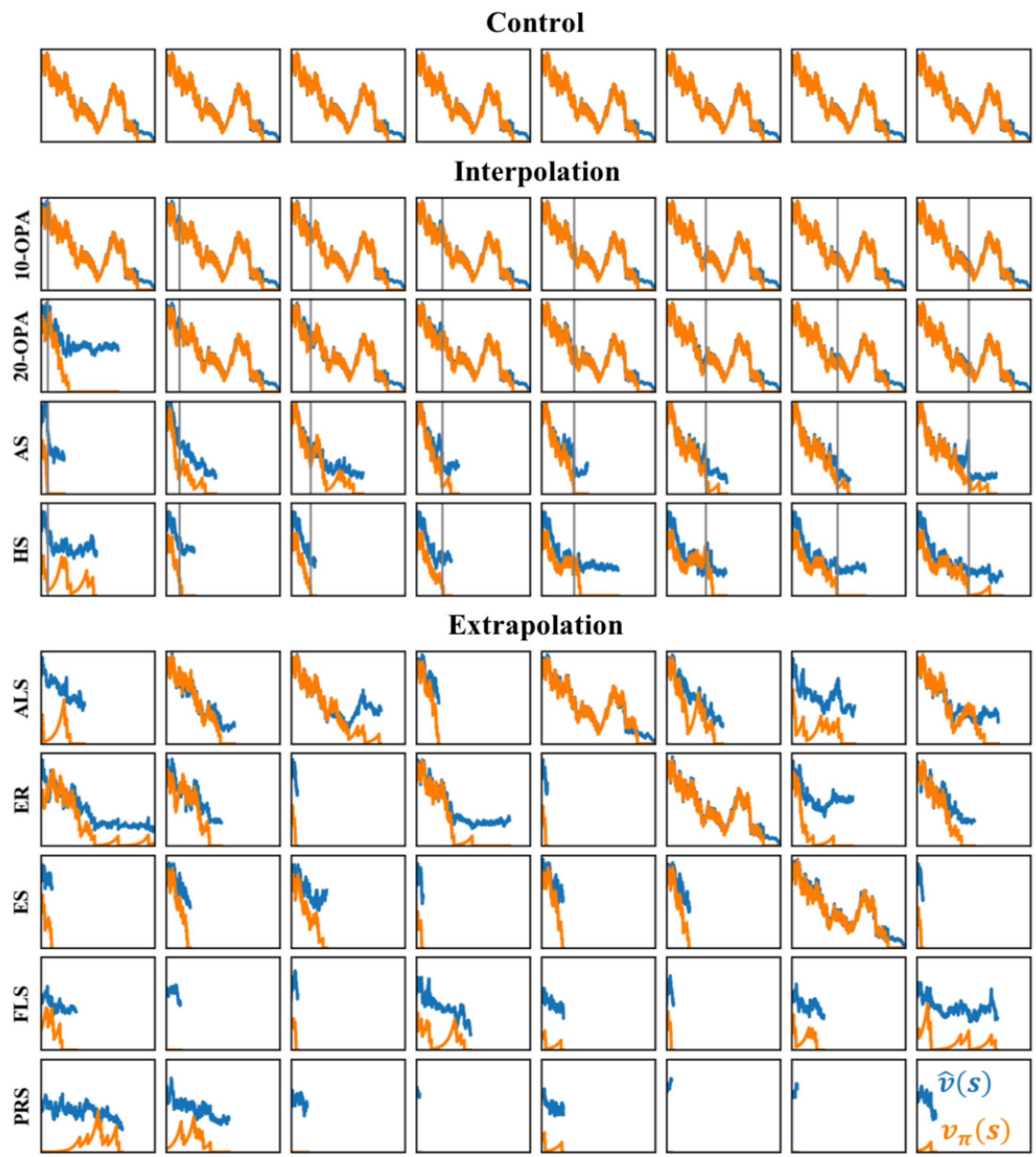
- Not enough to just do well (high TAR) if it's for the wrong reason (high VEE).
- Not enough to know what you will do (low VEE) if it's bad (low TAR).

Generalization Results

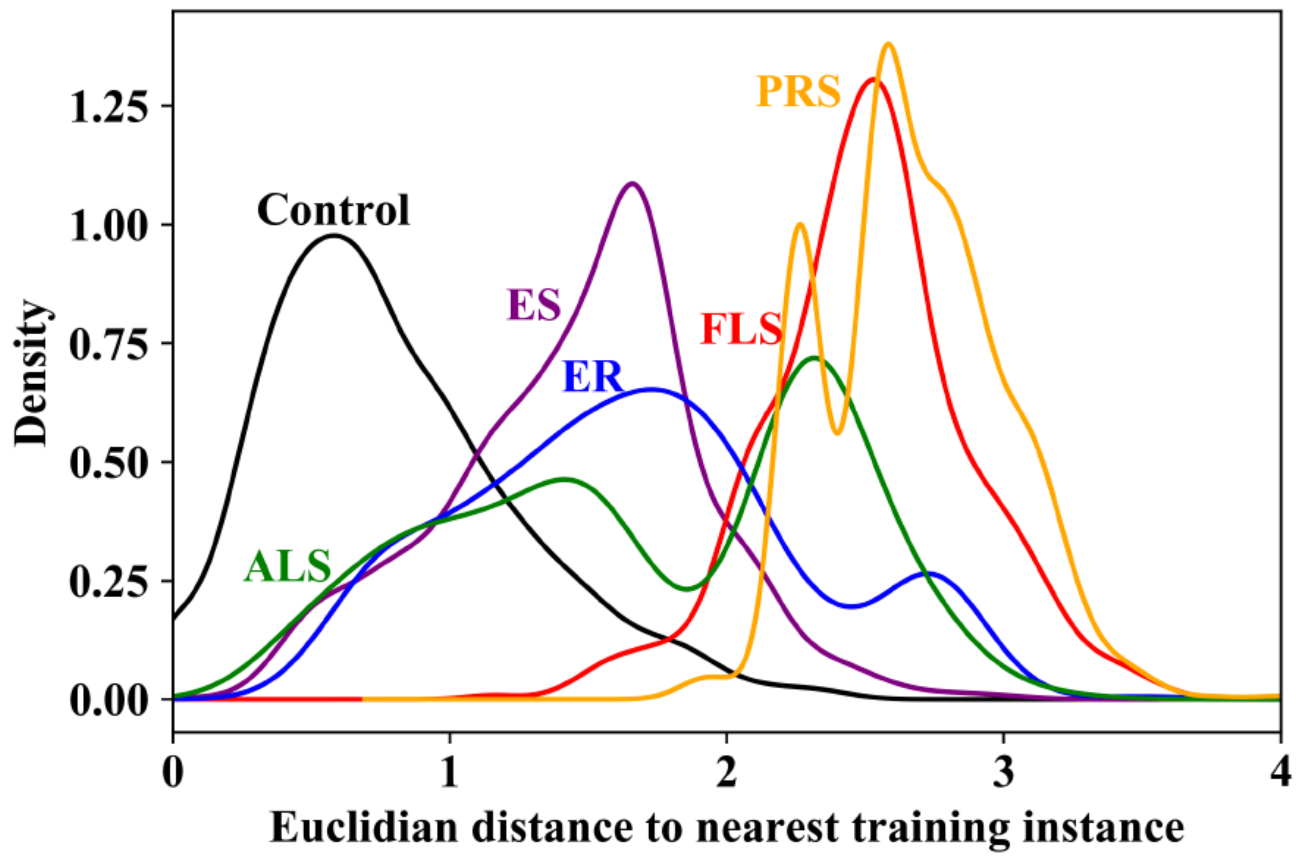
- C: control
- n-OPA: off policy actions
- AS: agent starts
- HS: human starts
- ALS: add line segments
- ER: enemy removal
- ES: enemy shift
- FLS: filled line segments
- PRS: player random start



VEE and TAR Correlate



Novel States Not "Recognized"

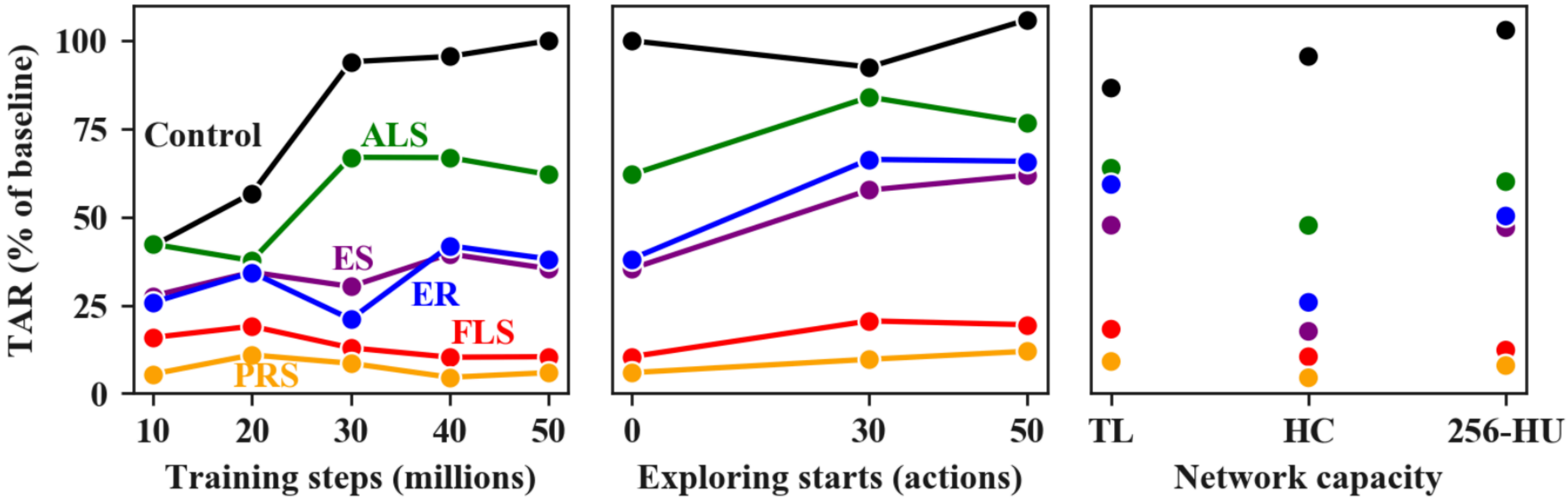


- The learned representation does not find the novel states to be like those seen in training.

Improving Generalization

- Supervised learning:
  - More data.
  - Simpler model / regularization.
- Reinforcement learning:
  - Increasing data via increasing training time.
  - Diversifying training data via random starts.
  - Reducing model capacity.

Modifying Training



(1) more training overfits

(2) diversifying training experience helps a bit

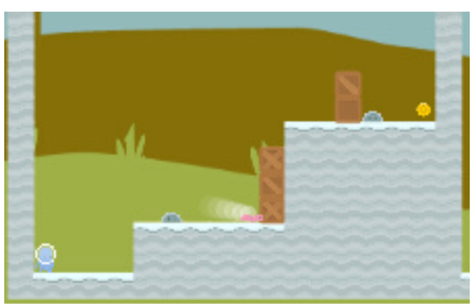
(3) reductions to model capacity are mixed

Case Study 2: CoinRun

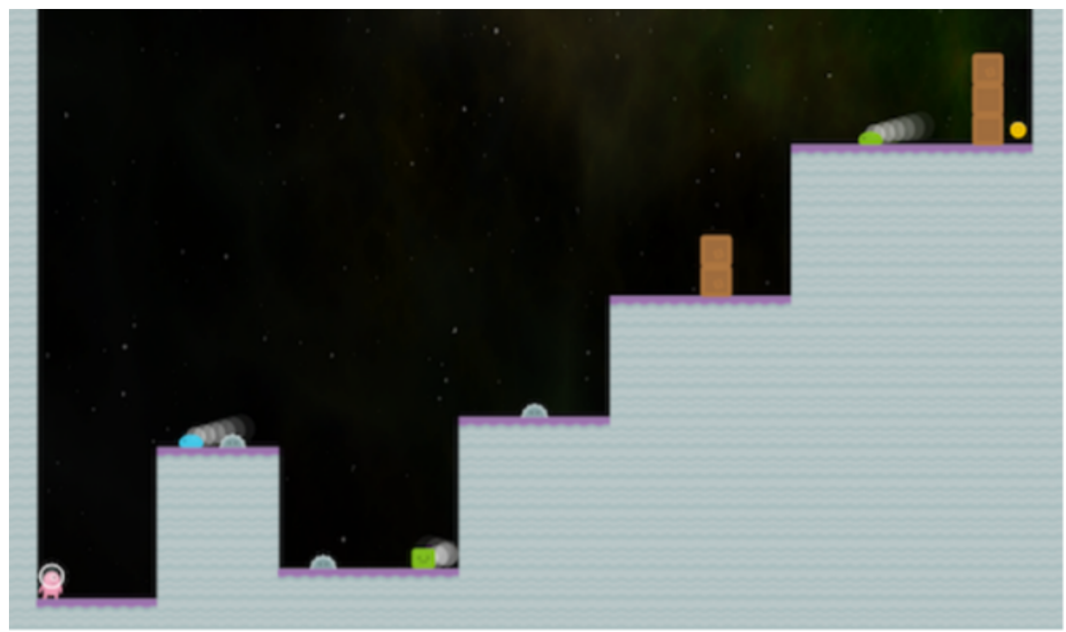
(Zhang and Littman, last week)

- Methodology and platform (Cobbe et al., 19). Collect the single coin to end the level.
- Agent spawns far left, coin on far right.
- Obstacles, enemies. Level ended by death, coin, or 1000 steps. Difficulty from 1 to 3.

1



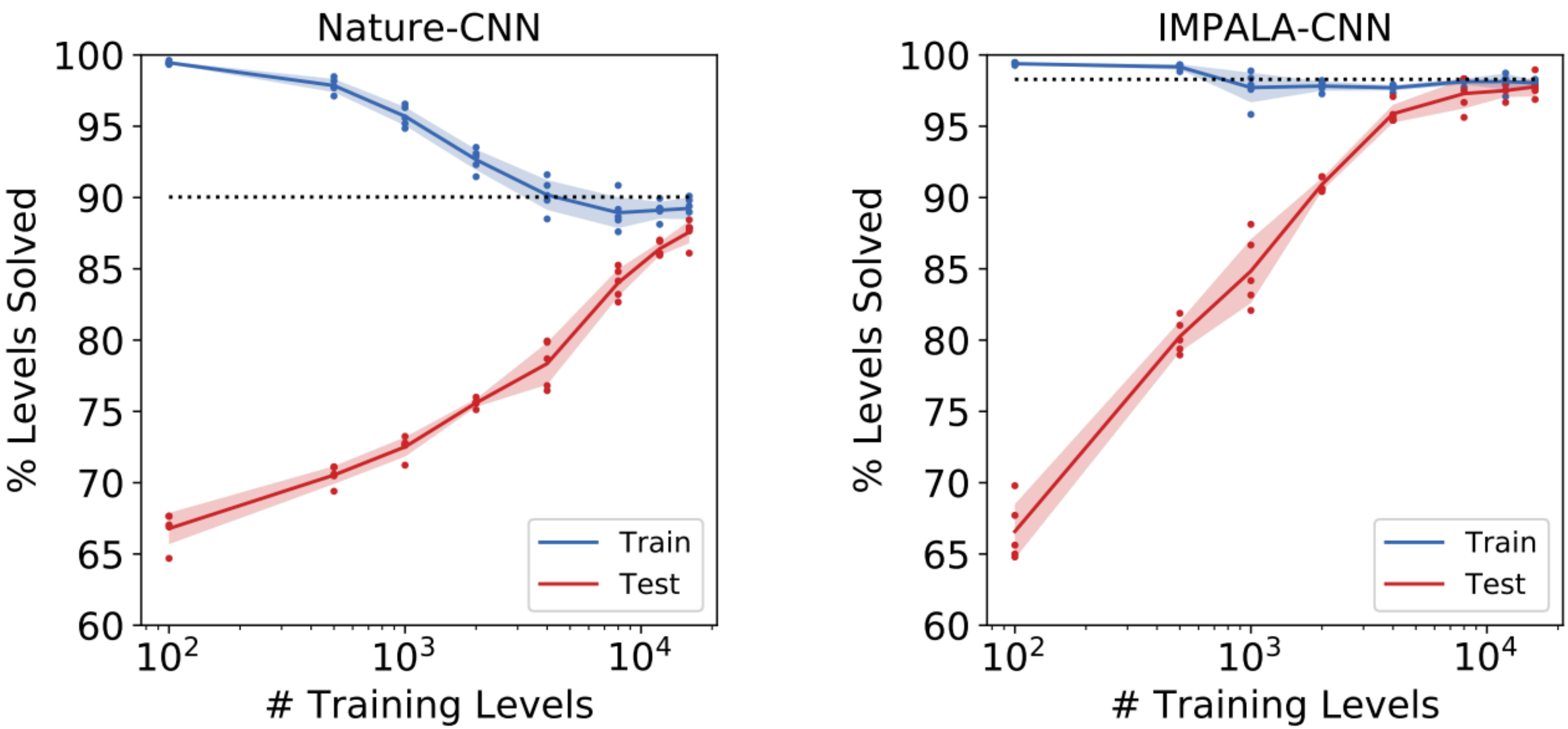
2



3

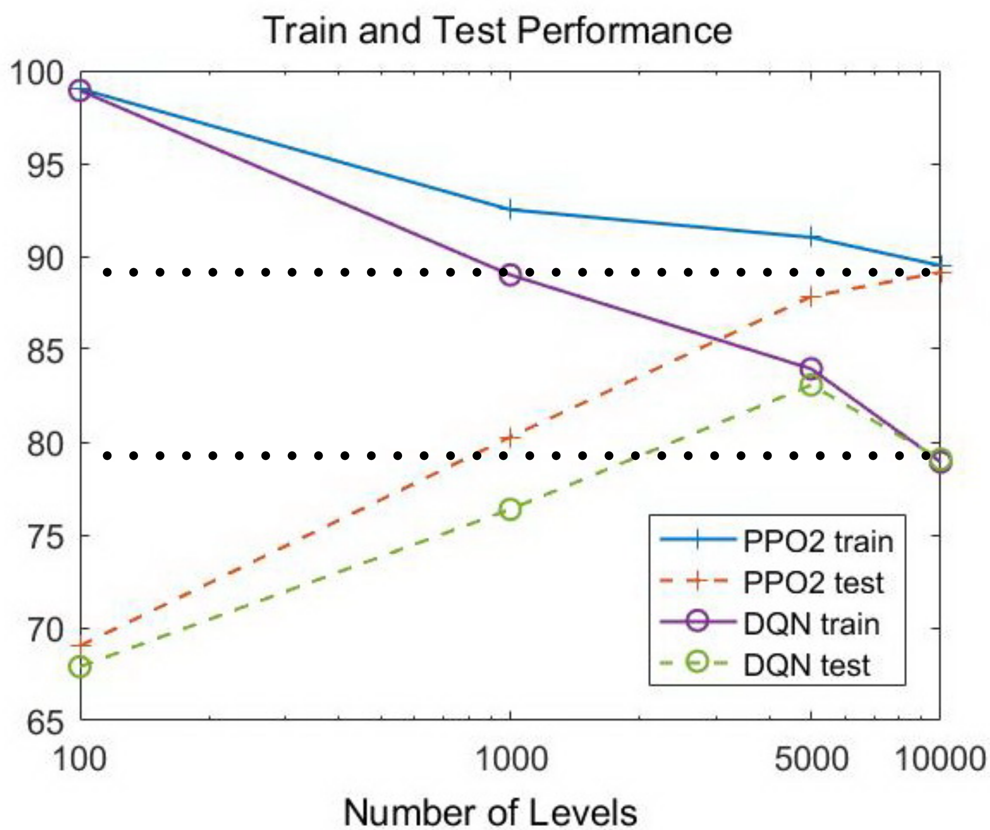
Results from CoinRun Paper

- Looked at two networks. Overfitting observed. Used PPO. DQN not reported.



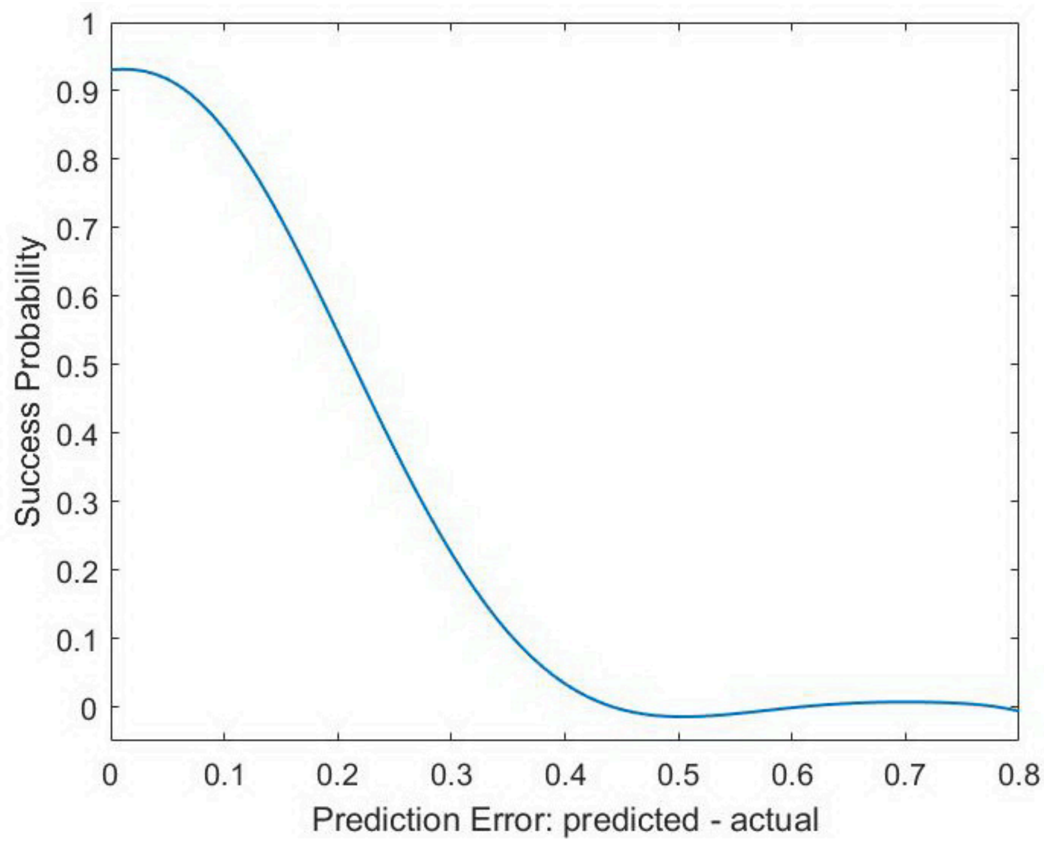
Compare Policy Search, DQN

- Switched to difficulty 2 only. Test on 10k.
- DQN: 20M steps. PPO2: 50M steps. Nature net.
- DQN generalized (but less well).



Prediction Errors

- High prediction error associated with failure.
- Prediction error lower in training than testing.
- Training = testing given enough data.



Summary

- Good RL performance seductive: look closer.
- Analogy between RL and supervised learning subtle.
- DQN non-generalization in Amidar, CoinRun, weak in CoinRun difficulty 2.
- Prediction error and internal representation distance good predictors of poor generalization.
- Adjusting training volume, model capacity, and exploration help (a bit).
- Future work:
  - Compare to model-based RL!

Prediction Errors