Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations

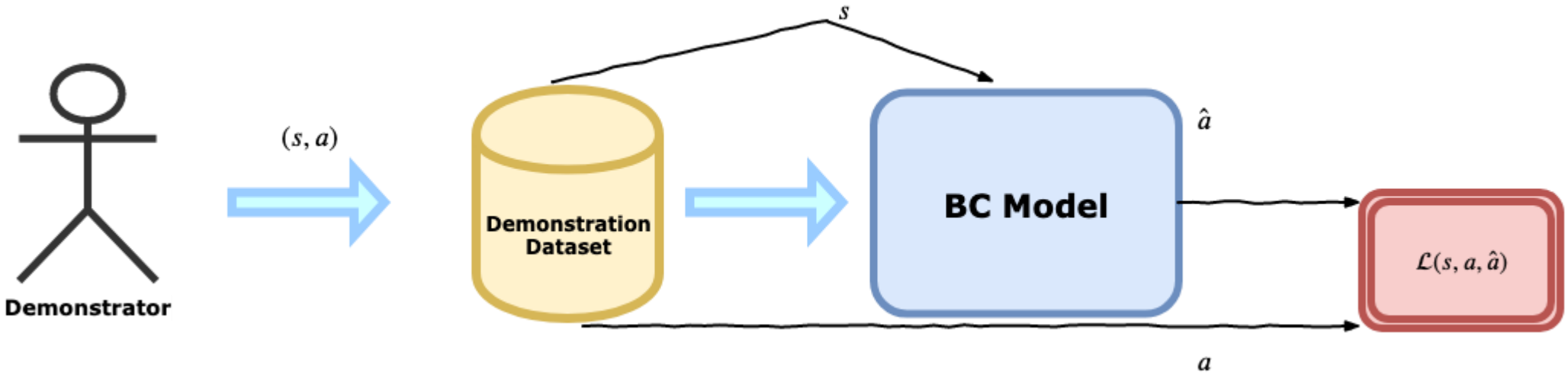
DANIEL S. BROWN*, WONJOON GOO*, **PRABHAT NAGARAJAN**, SCOTT NIEKUM

Motivation

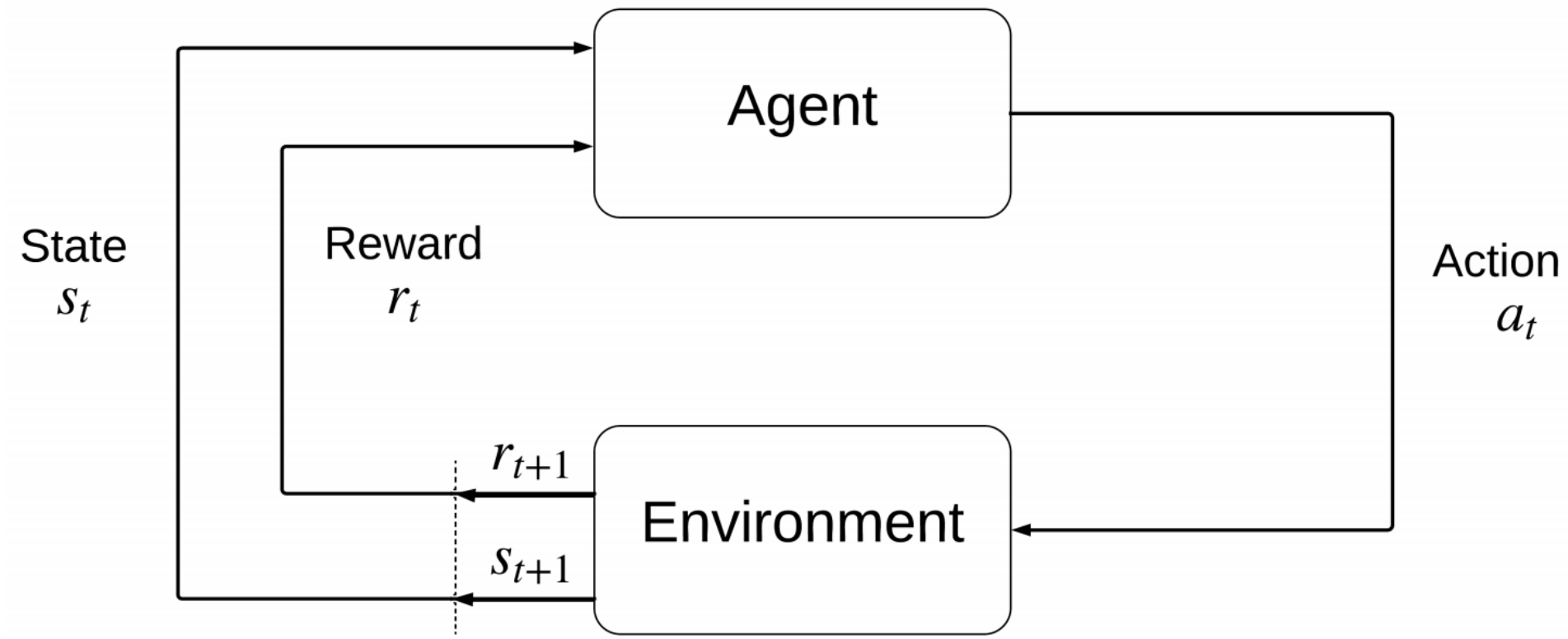
- **Imitation Learning** – (for this talk) Learning a policy (i.e. how to make decisions) for a task from demonstrations of the task
- Traditional imitation learning approaches fail to outperform the demonstrator (inverse RL, adversarial imitation, behavioral cloning)
  - Methods all based on some form of *matching* the demonstrator
  - Often assume demonstrator optimality
- **Issues: such "matching" methods can break task semantics.**

Background: Behavioral Cloning

- BC- *learn a direct mapping from states/contexts to trajectories/actions without recovering the* reward *function* **(Osa et. al, 2019)**
- Supervised learning

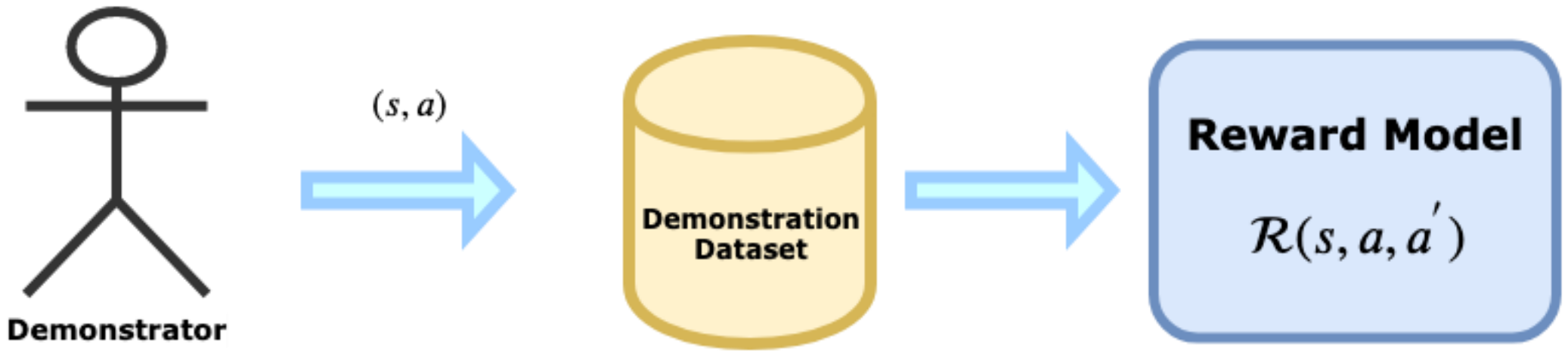


Reinforcement Learning



Modeled after (Sutton and Barto, 2018).

Inverse RL – RL + Learned Reward

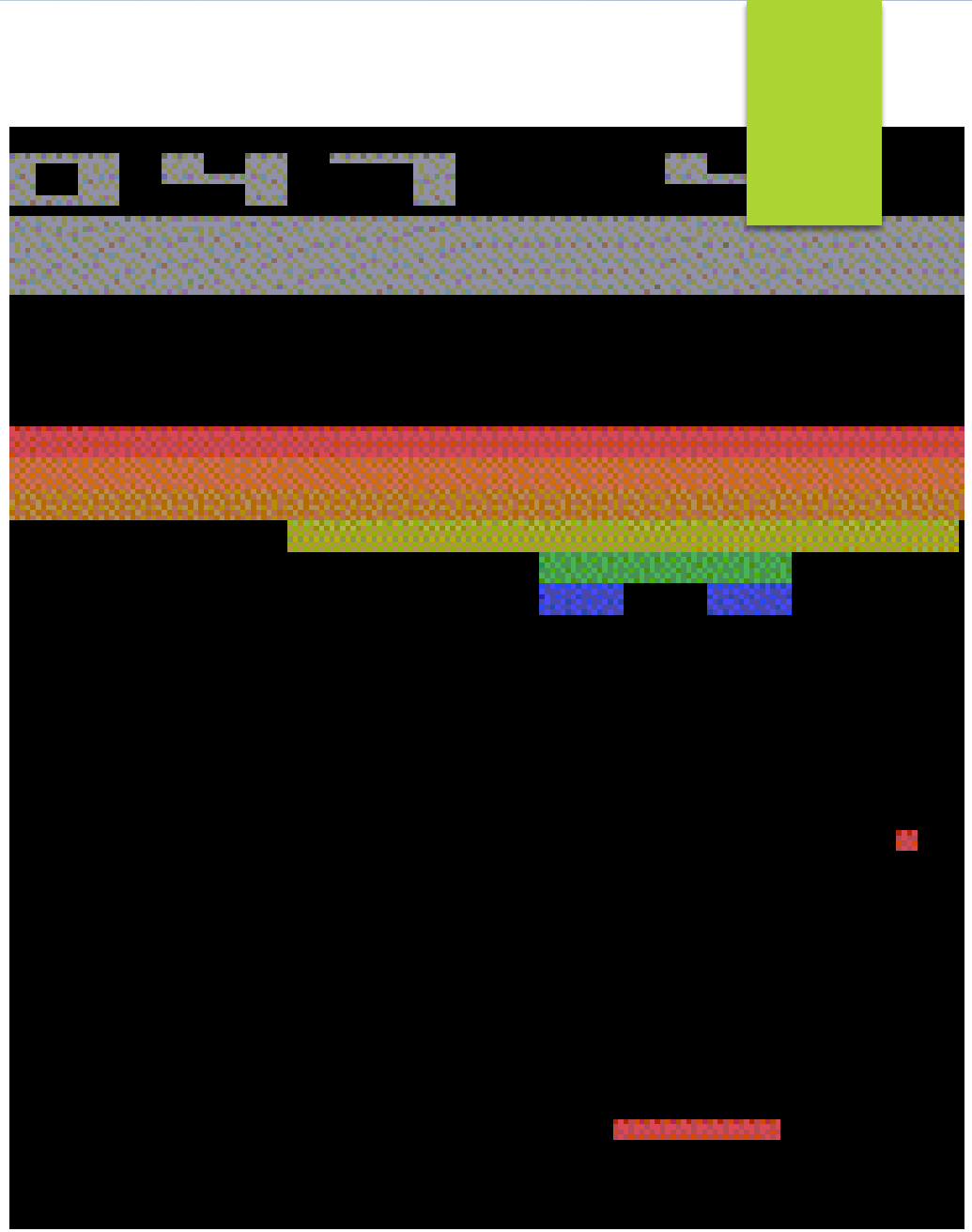


- Assume demonstrator is optimal
- Learn a reward function that makes demonstrator's actions appear optimal
- IRL vs. BC: IRL learns a reward function that makes demonstrator's decisions optimal. BC traditionally learns to mimic demonstrator's decisions

Scenario - Demonstration

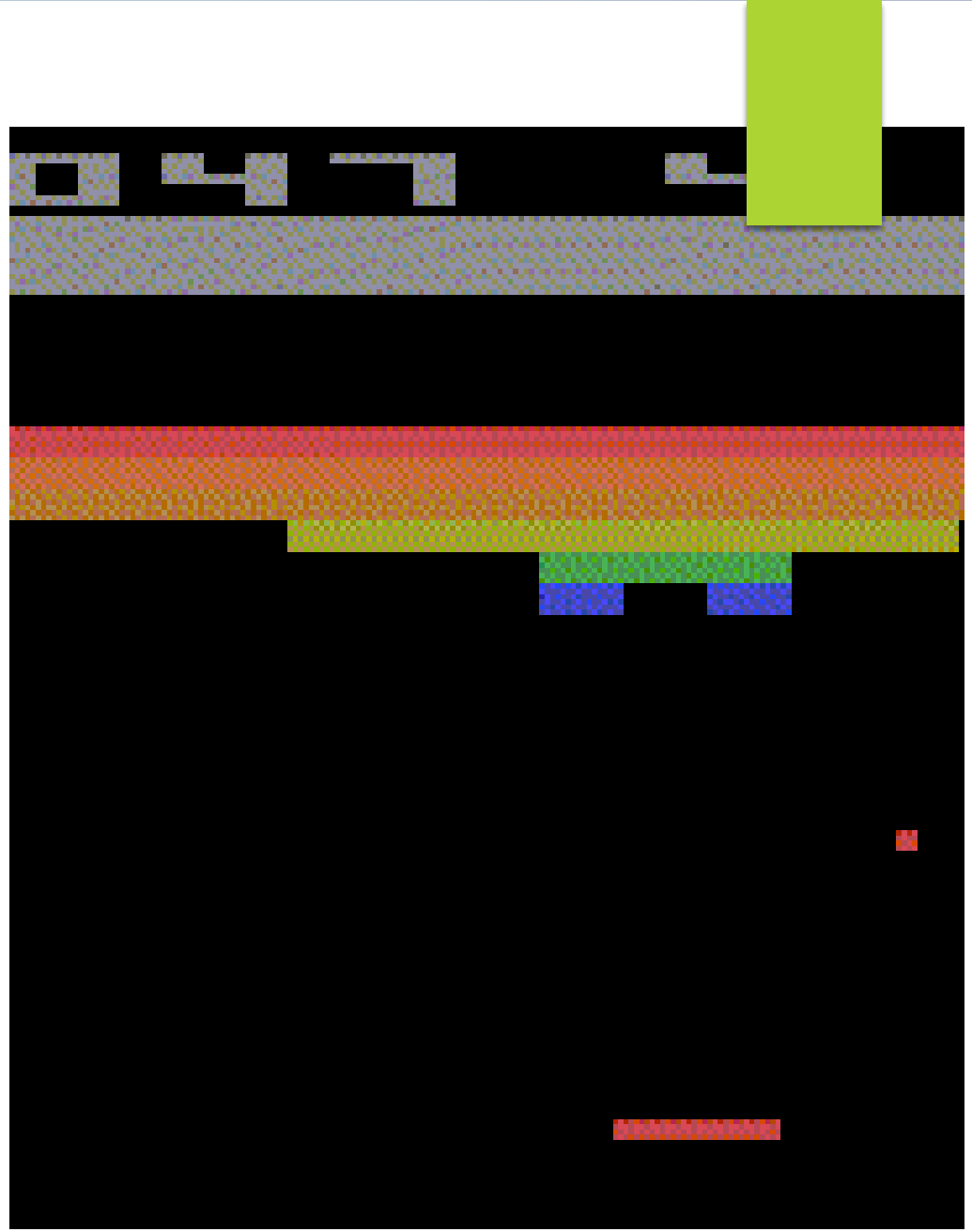
Behavioral Cloning Interpretation

- **"Do exactly what the demonstrator would have done"**
- ***"Learn to miss the ball in this situation"***
- Issue
  - Imitating a failed demonstration
  - Poor generalization to states/situations not experienced by the demonstrator
  - Ultimately learning a way, but not the right way of doing it.



Inverse RL Interpretation

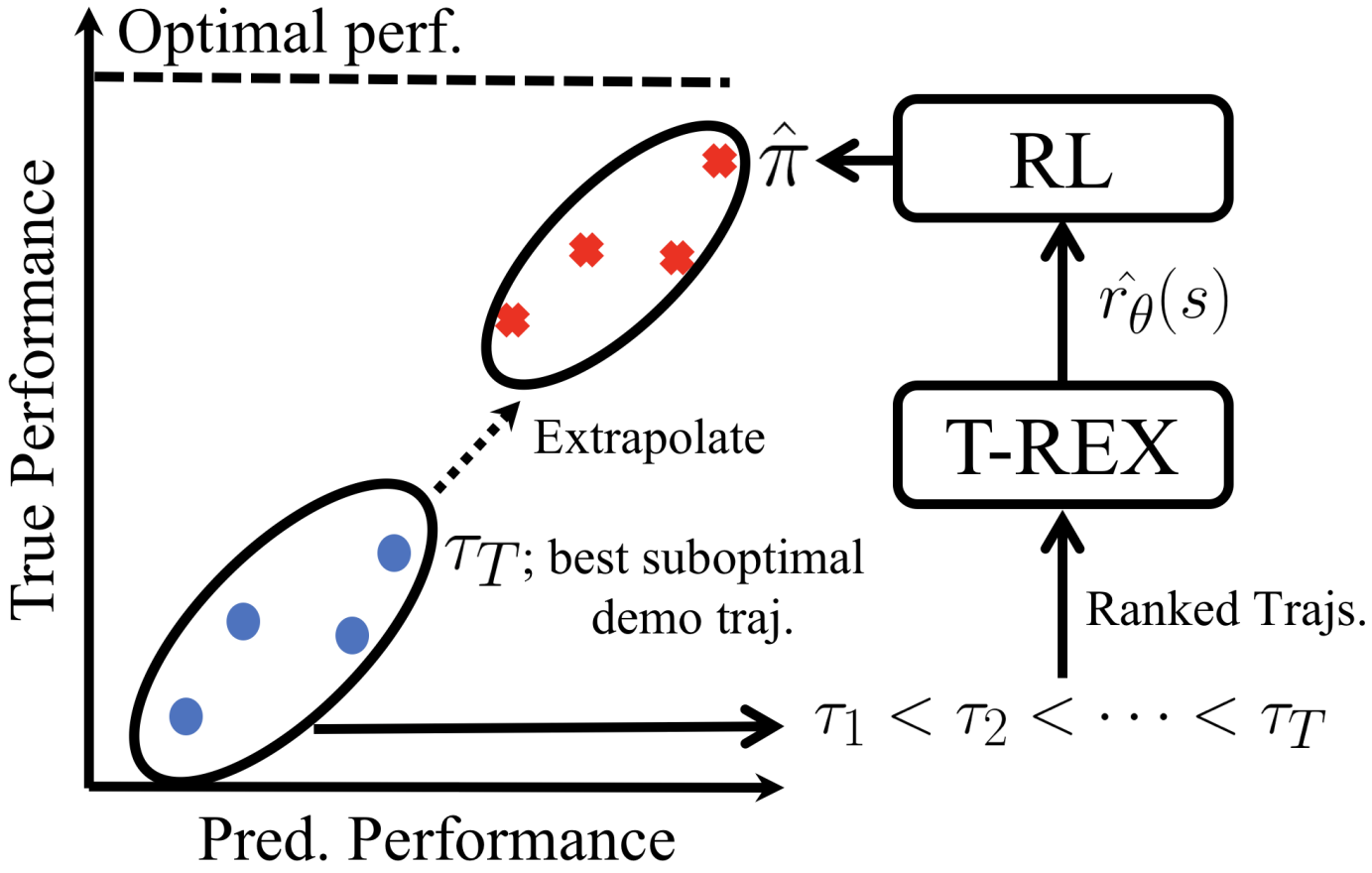
- **"Find a way to reward the agent for its actions in such a way that the demonstration appears optimal"**
- **"Knocking off this specific set of bricks is the best thing to do"**
- Issue
  - *Reward ambiguity* – many reward functions make demonstration appear optimal
  - *Semantically incorrect* – A reward function learn to make a poor demonstrator appear optimal might induce suboptimal behavior
  - Suboptimal behavior



T-REX

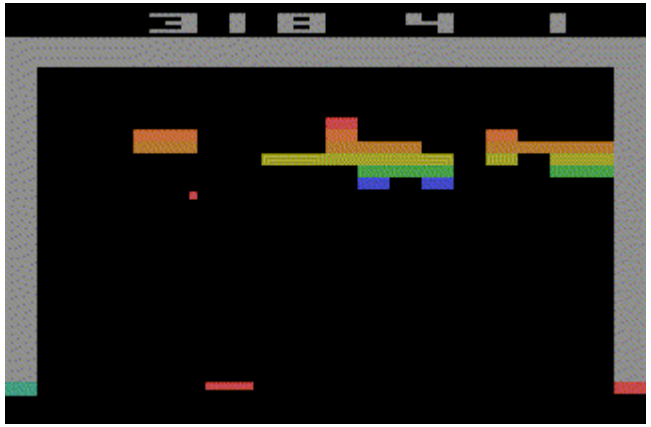
- **Question: Given a sequence of ranked demonstrations, what reward function explains how one demonstration is better than another**
- Premise: Ranking demonstrations is easier (in general) than producing optimal demonstrations
  - Music, sports, difficult skills
- Benefits
  - No assumptions on absolute performance of demonstrator
  - Can learn from suboptimal demonstrations
  - Reduced Ambiguity – 0-reward function
  - Has access to different areas of the state space (different performance qualities

T-REX Overview



TREX Interpretation

- **“Find a reward function that supports the ranked demonstrations”**
- **“Knocking off more bricks is better”**
- Outcome
  - Can often learn a better reward function and outperform the demonstrator!



T-REX Method

$$\mathrm{P}\big(\hat{J}_\theta(\tau_i) < \hat{J}_\theta(\tau_j)\big) \approx \frac{\exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}{\exp \sum_{s \in \tau_i} \hat{r}_\theta(s) + \exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}, \tag{2}$$

$$\mathcal{L}(\theta) = -\sum_{\tau_i \prec \tau_j} \log \frac{\exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}{\exp \sum_{s \in \tau_i} \hat{r}_\theta(s) + \exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}. \tag{3}$$

Related Work

- Deep Reinforcement Learning from Human Preferences (Christiano et al. 2017) – loss function
- Reward learning from human preferences and demonstrations in Atari (Ibarz et al. 2018)
- Deep IRL: Guided Cost Learning (Finn, Levine, & Abbeel. 2016), Generative Adversarial Imitation Learning (Ho and Ermon, 2016), Adversarial Inverse RL (Fu, Huo, & Levine) – *none applied to video games.*
- Imitating Atari games from Youtube videos (Aytar et al. 2018)
  - Does not outperform demonstrator

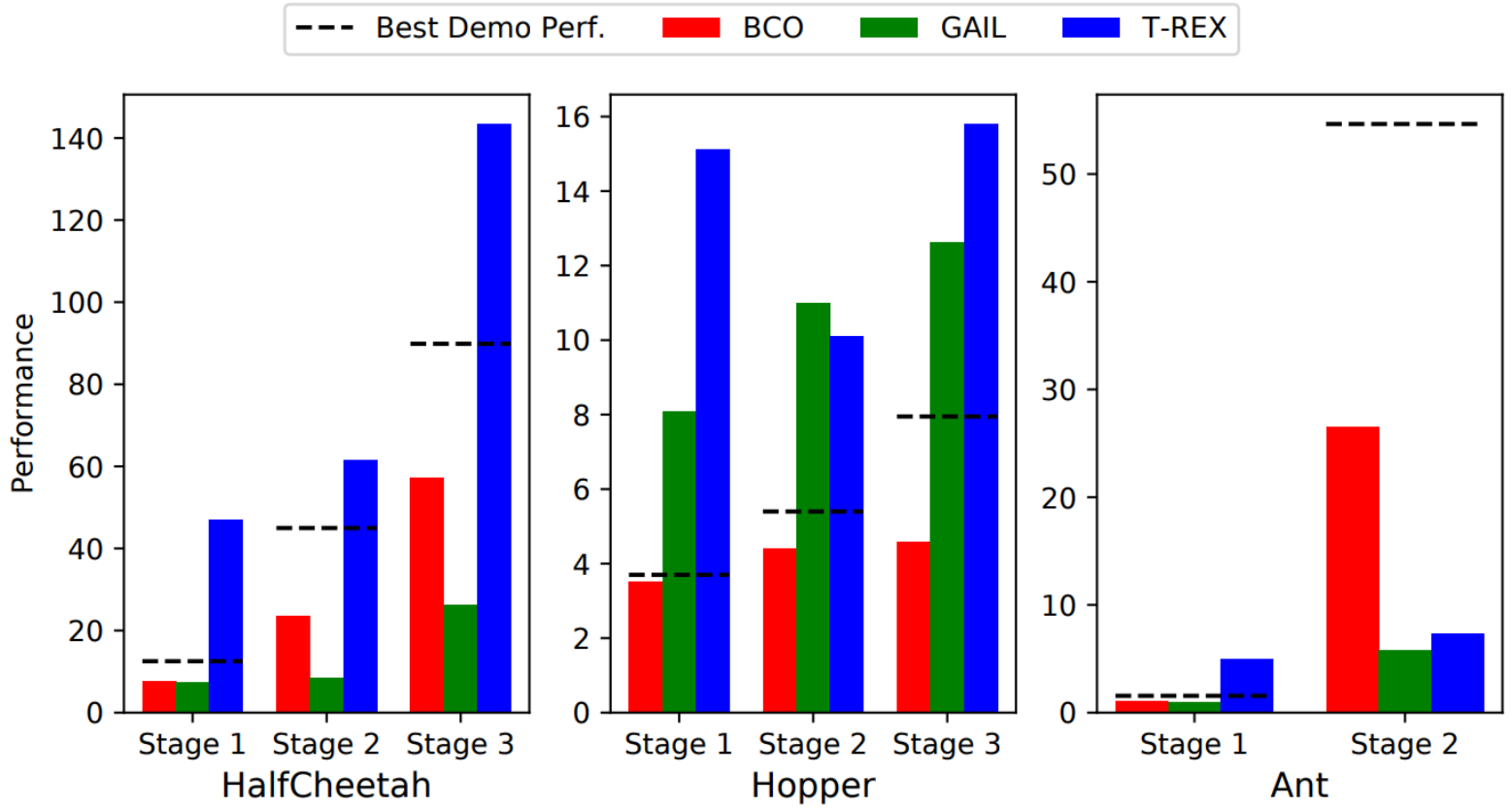
Extended work on learning from suboptimal demonstrations

- Learning from failed demonstrations (Shiarlis et al. 2016)
- Preference-based IRL (Wirth et al. 2016)

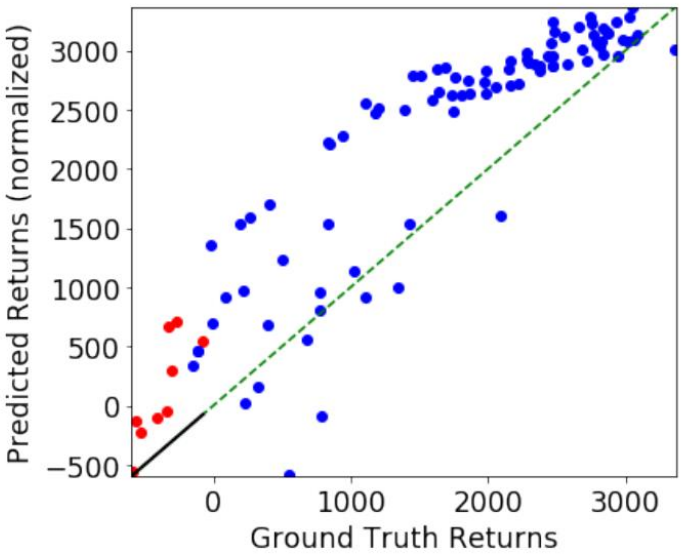
Experiments

- Mujoco & Atari
- Trained PPO (Proximal Policy Optimization) agents
  - Checkpoint policies during training
  - Generate demonstrations from checkpointed policies
  - Ranked episodes according to ground-truth rewards
- Atari Experiments – Masked the score
- Ran T-REX on the ranked trajectories
- Additional experiments on human demonstration data, noisy rankings, and human rankings through Amazon Mechanical Turk

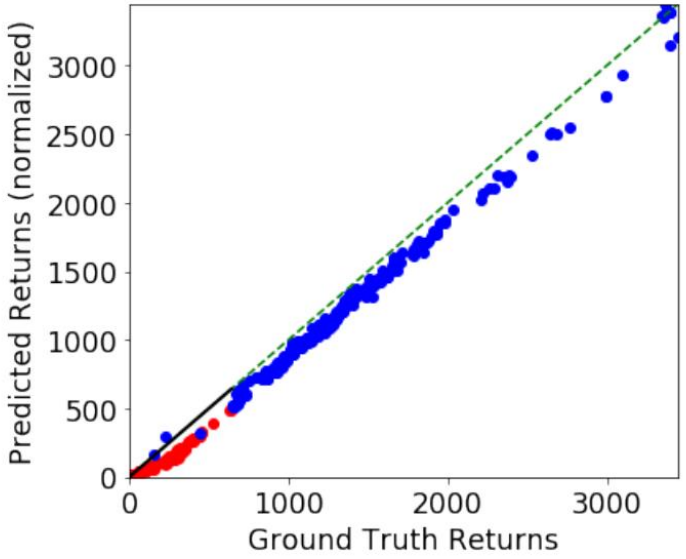
Performance - Mujoco



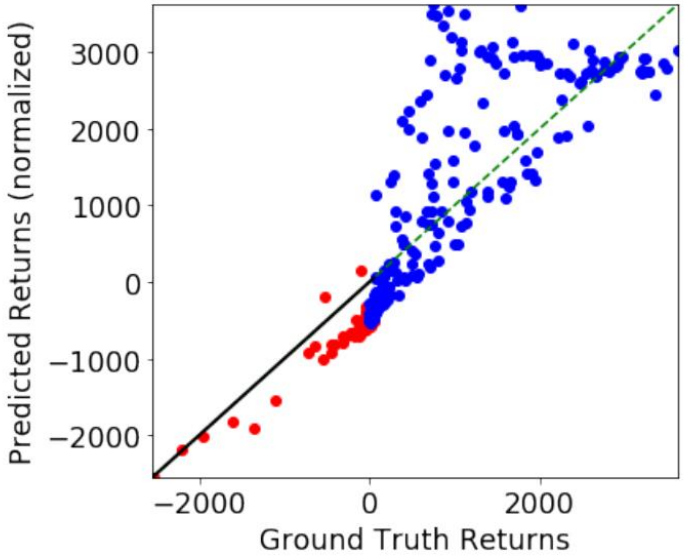
Reward Extrapolation - Mujoco



(a) HalfCheetah



(b) Hopper

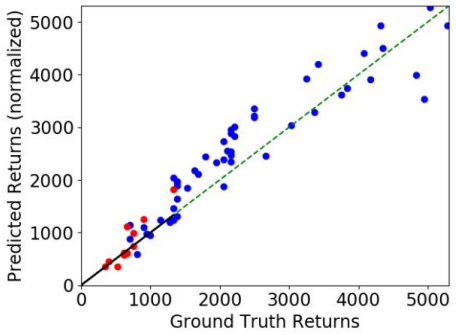


(c) Ant

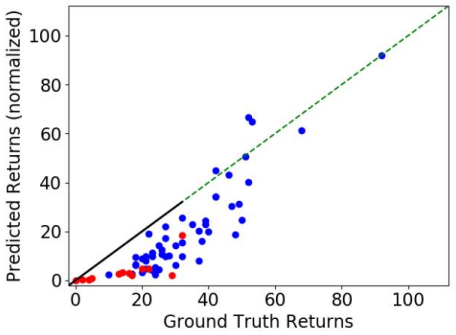
Performance - Atari

	Ranked Demonstrations		LfD Algorithm Performance		
Game	Best	Average	T-REX	BCO	GAIL
Beam Rider	1,332	686.0	**3,335.7**	568	355.5
Breakout	32	14.5	**221.3**	13	0.28
Enduro	84	39.8	**586.8**	8	0.28
Hero	**13,235**	6,742.0	0	2,167	0
Pong	-6	-15.6	**-2.0**	-21	-21
Q*bert	800	627	**32,345.8**	150	0
Seaquest	600	373.3	**747.3**	0	0
Space Invaders	600	332.9	**1,032.5**	88	370.2

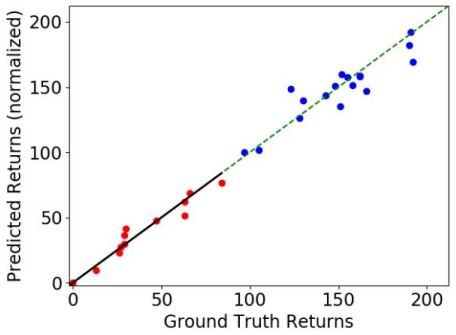
Reward Extrapolation - Atari



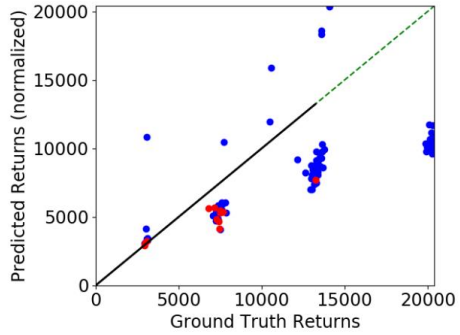
(a) Beam Rider



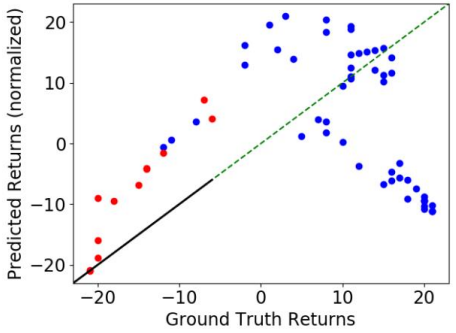
(b) Breakout



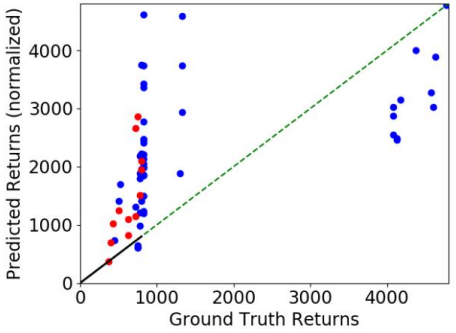
(c) Enduro



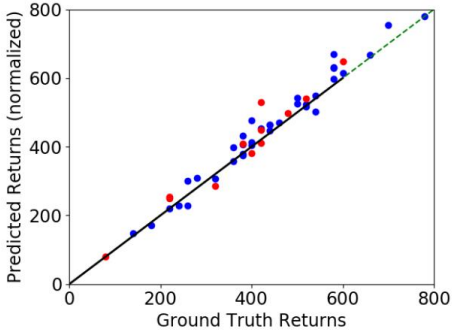
(d) Hero



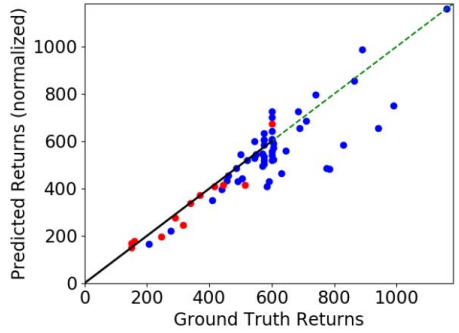
(e) Pong



(f) Q*bert

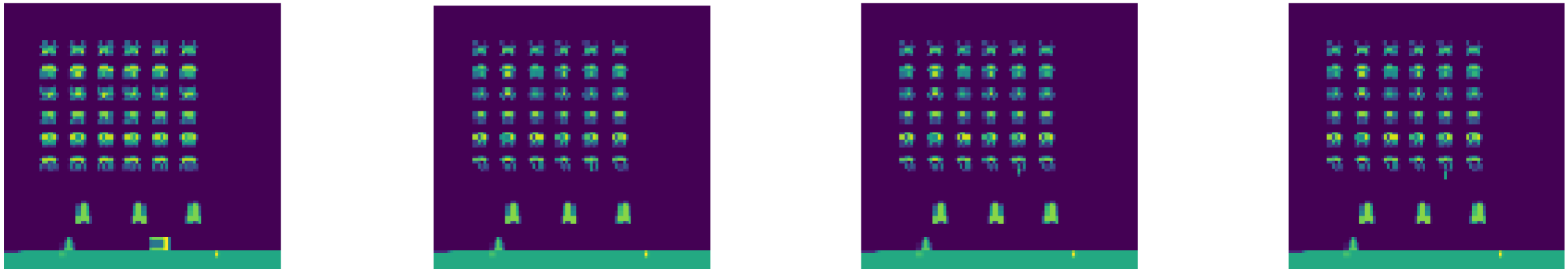


(g) Seaquest



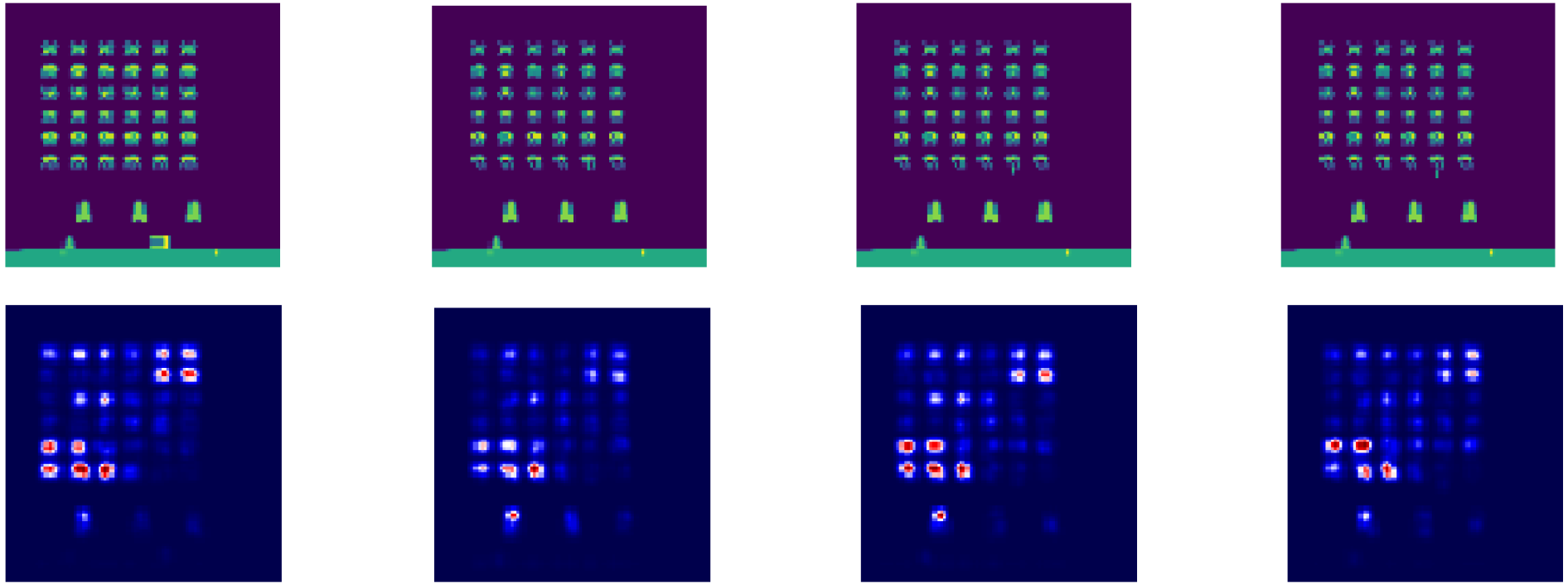
(h) Space Invaders

Space Invaders – Minimum Reward



(c) Space Invaders observation with minimum predicted reward

Space Invaders – Minimum Reward



(d) Space Invaders reward model attention on minimum predicted reward

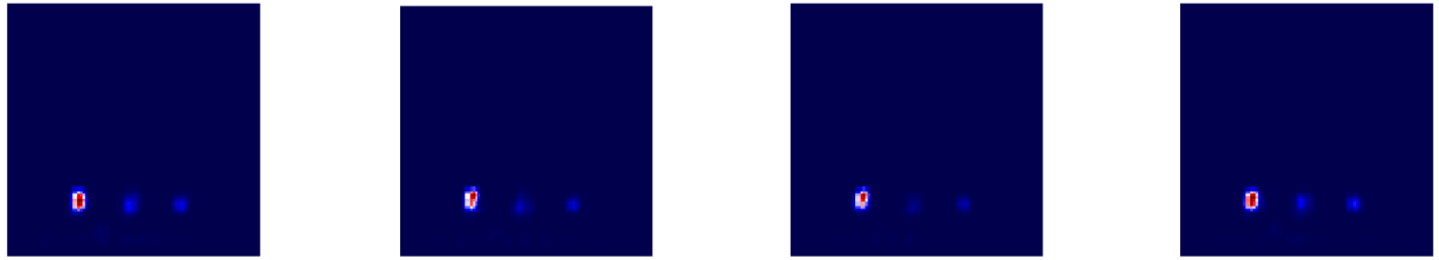
Space Invaders – Maximum Reward

(a) Space Invaders observation with maximum predicted reward

Space Invaders – Maximum Reward

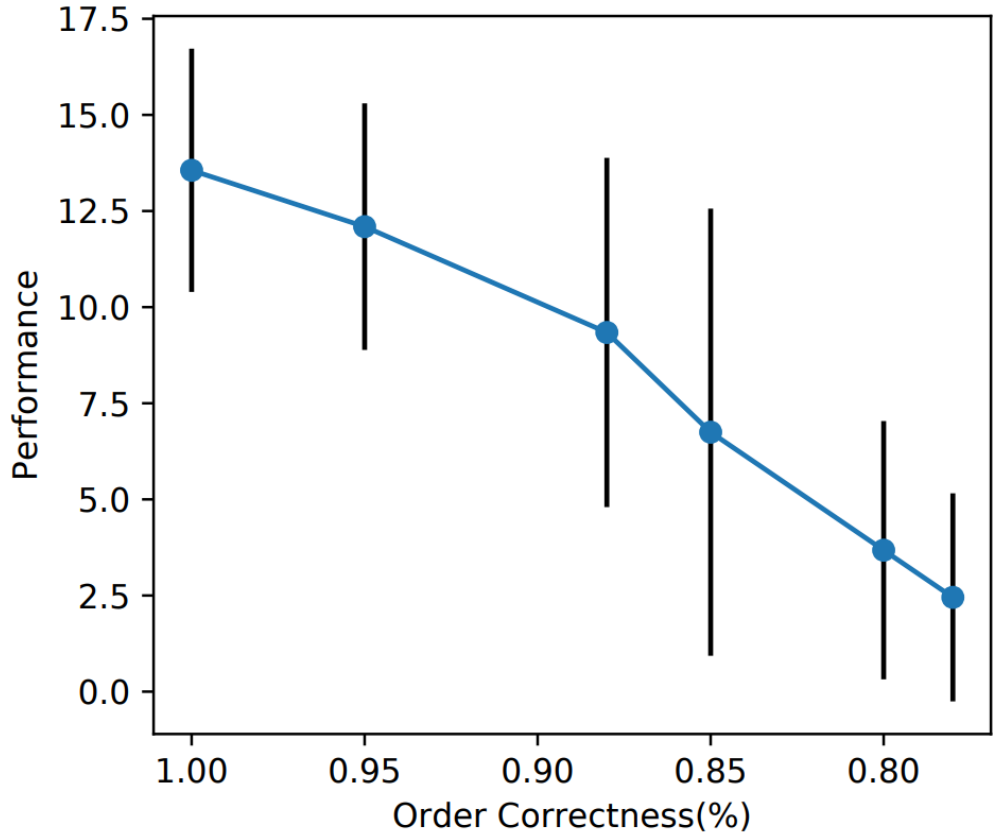
Note that the agent never observed a demonstration that successfully destroyed all the aliens! = Successful Extrapolation!

(a) Space Invaders observation with maximum predicted reward



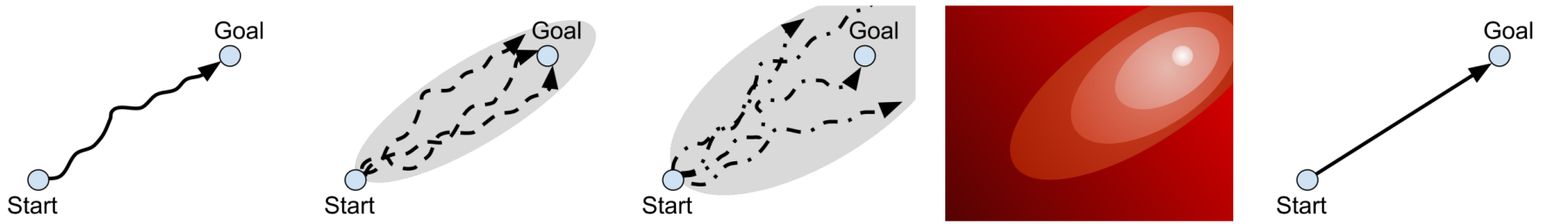
(b) Space Invaders reward model attention on maximum predicted reward

Ranking Noise



DREX: T-REX Without Rankings! (Brown, Goo, and Niekum. 2019)

- Step 1 – Take in demonstrations without rankings
- Step 2 – Perform behavioral cloning on the demonstrations
- Step 3 – Add noise to the learned policy, implicitly generated ranked demonstrations
- Perform T-REX on these ranked "demonstrations"



(a) Demonstration (b) Small noise (c) Larger noise (d) Learned reward function from ranking: $(a) \succ (b) \succ (c)$ (e) Optimized policy

(Brown, Goo, and Niekum. 2019)

Summary

- T-REX is a reward-learning from demonstration algorithm that can learn from suboptimal, ranked demonstrations
- T-REX is the first algorithm (to our knowledge) to successfully outperform the demonstrator on video games
- T-REX is often able to accurately predict rewards for states that lie beyond the performance range of the demonstrator
- T-REX is robust to ranking noise, human rankings, and human demonstrations
- D-REX removes the need for rankings
- Limitation: Unclear in practice whether extrapolation will occur