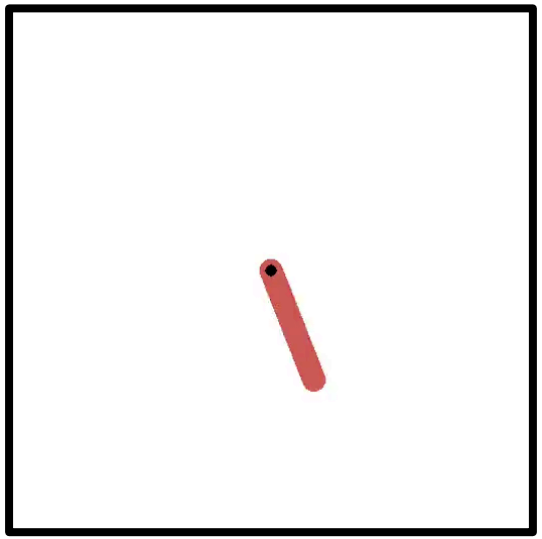
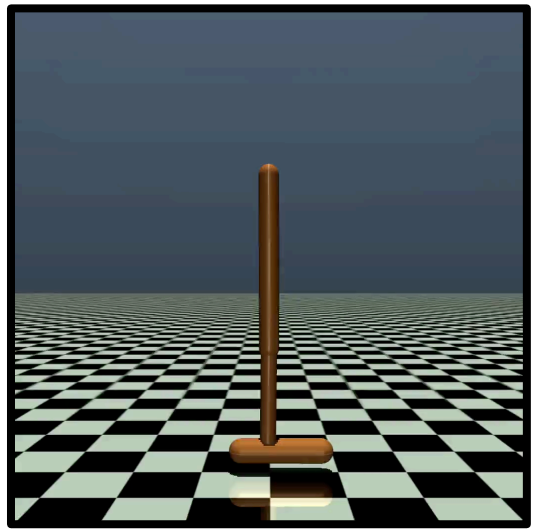
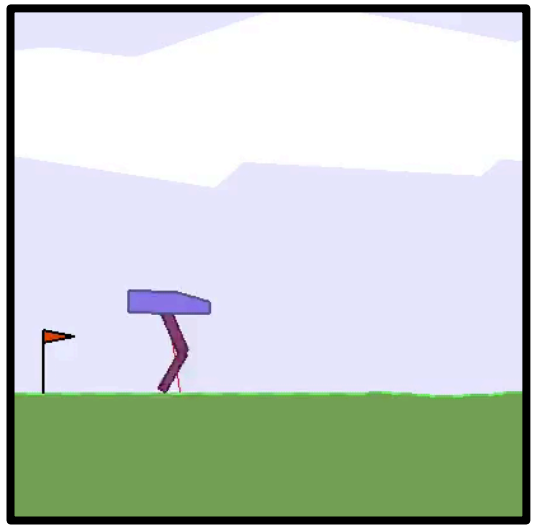
Injective State-Image Mapping facilitates Visual Adversarial Imitation Learning

Subhajit Chaudhury, Daiki Kimura, Asim Munawar and Ryuki Tachibana

IBM Research AI, Tokyo

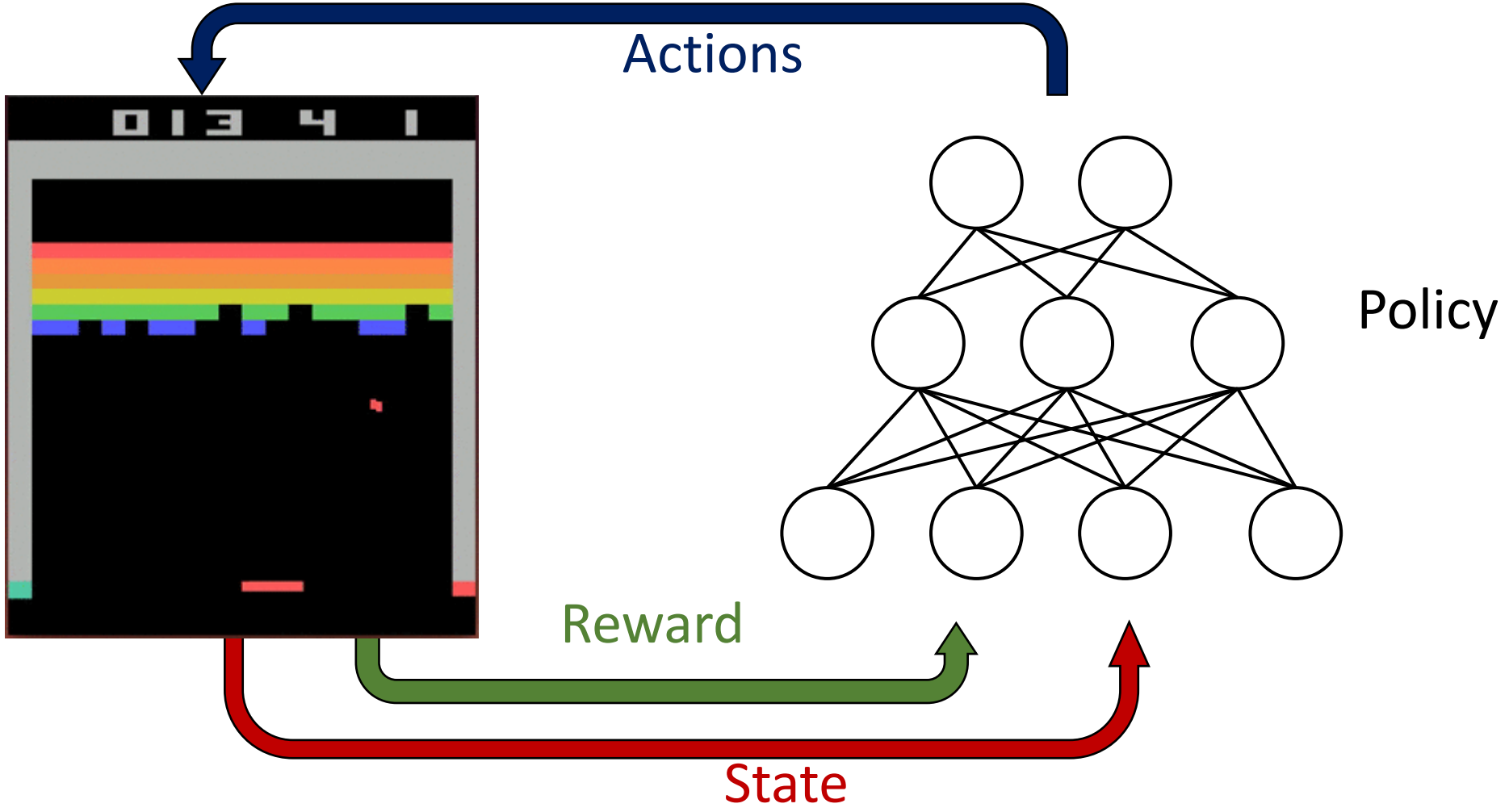






Reinforcement learning

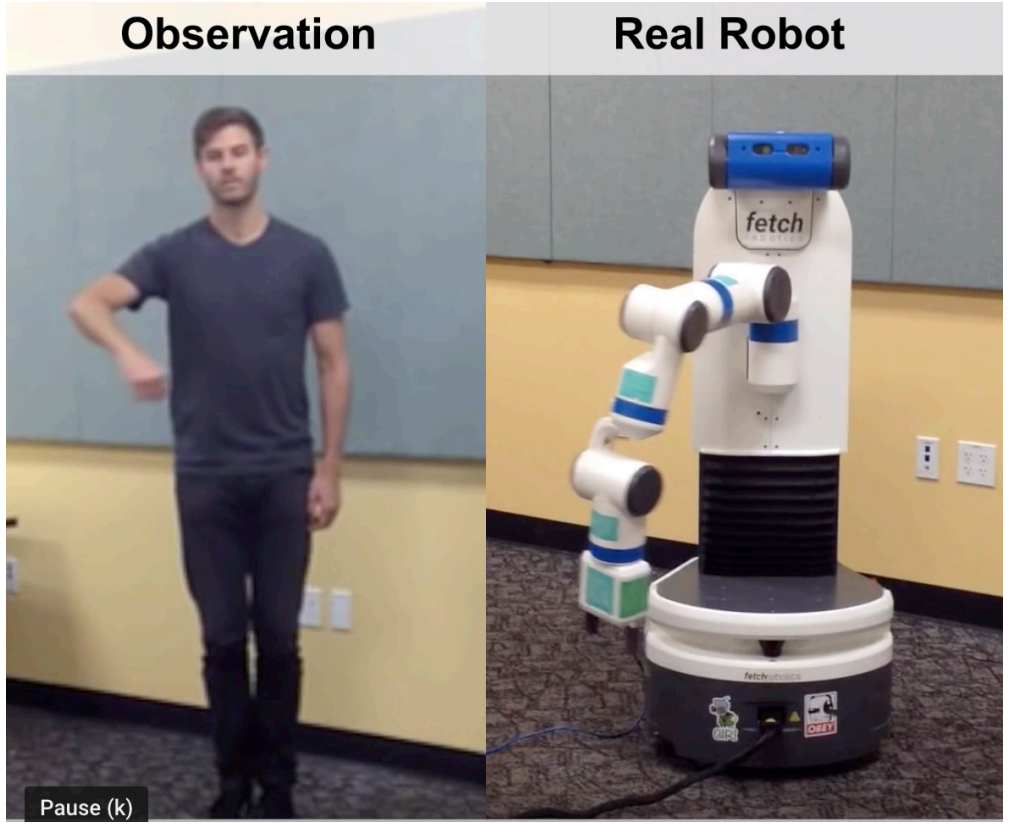
- Learn control policies to achieve a target.
- Use reward signals to optimize model parameters from high-dimensional inputs



Source: Google images

Imitation learning

- Non-trivial to craft good reward signals for goals.
- Imitation learning mimics an expert demonstrations (e.g. humans, video demos , etc.)
- Natural way of learning in nature by imitating other expert, e.g. Swimming.
- *Applications*: Robotic manipulation, games, virtual reality, etc.



Time-contrastive networks , Sermanet+, ICRA2018

Imitation learning: Problem overview

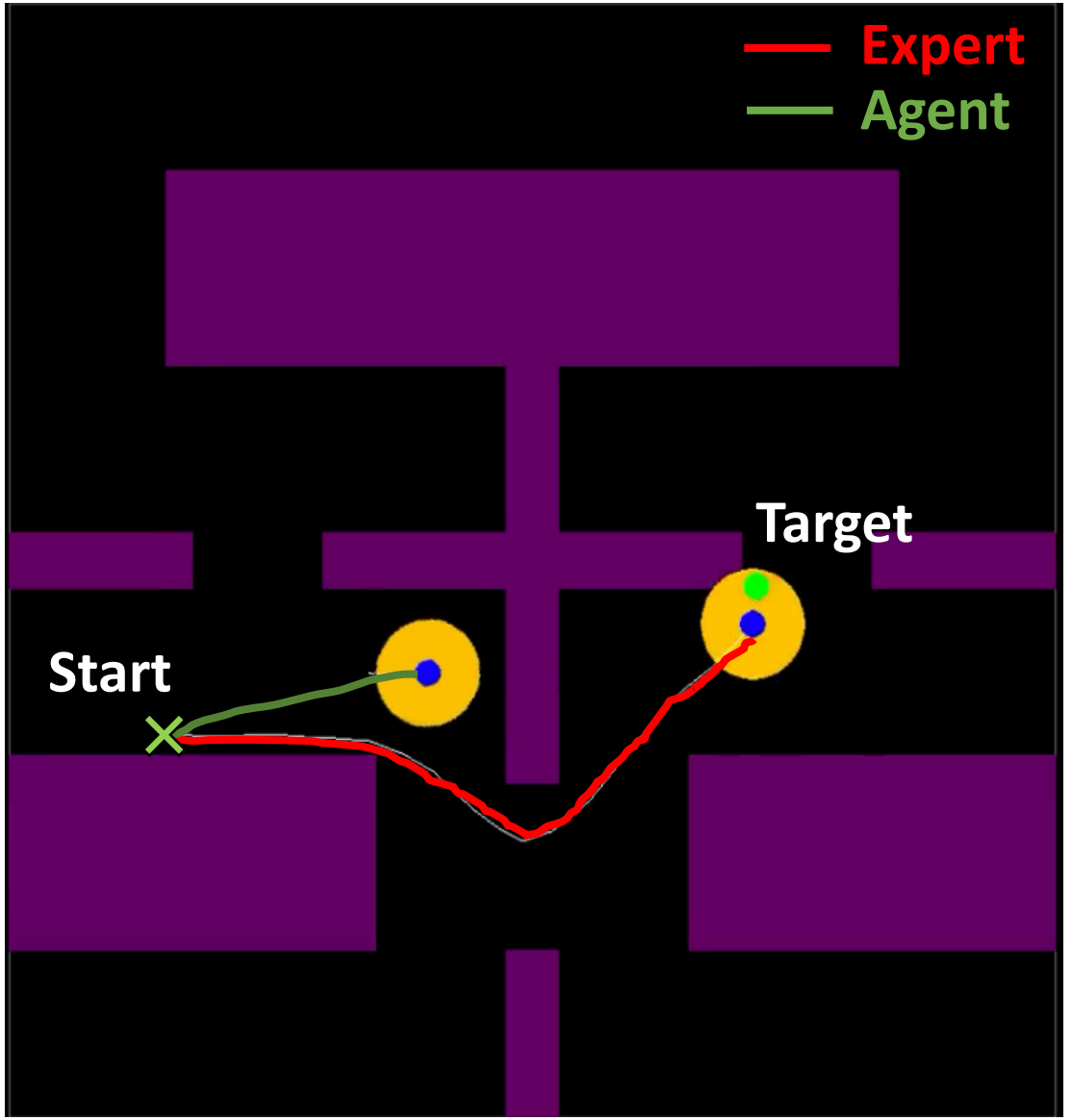
- **Goal**: To learn a policy π that can perform similarly to an expert π_E from finite sample expert trajectories only.

  - Given expert trajectories:
    $\rho_{\pi_E}(s,a) = \{(s_0^e, a_0^e, s_1^e, a_1^e, \dots)\} \sim \pi_E$

  - Let the agents trajectories are given as:
    $\rho_{\pi}(s,a) = \{(s_0^a, a_0^a, s_1^a, a_1^a, \dots)\} \sim \pi$

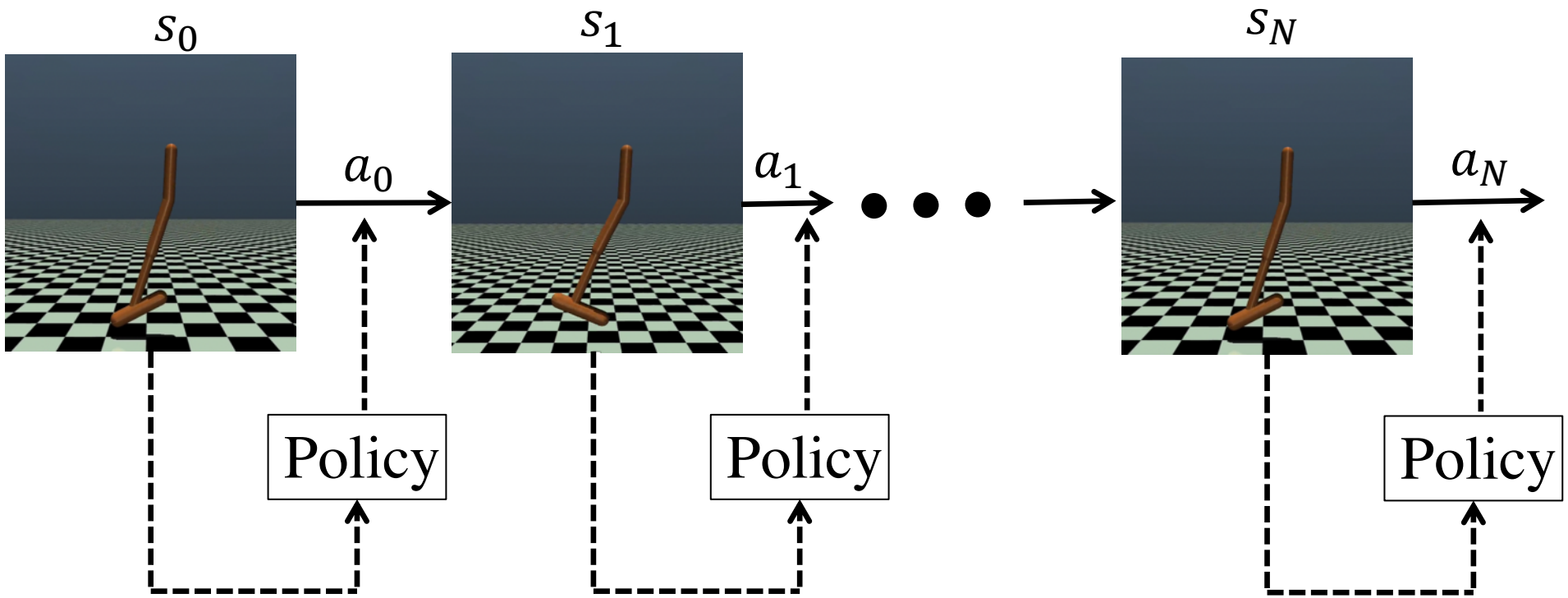
- Find π such that,

$$\min_{\pi \in \Pi} \quad \boldsymbol{d}(\rho_{\pi}(s,a), \rho_{\pi_E}(s,a))$$



Imitation learning: Behavior cloning

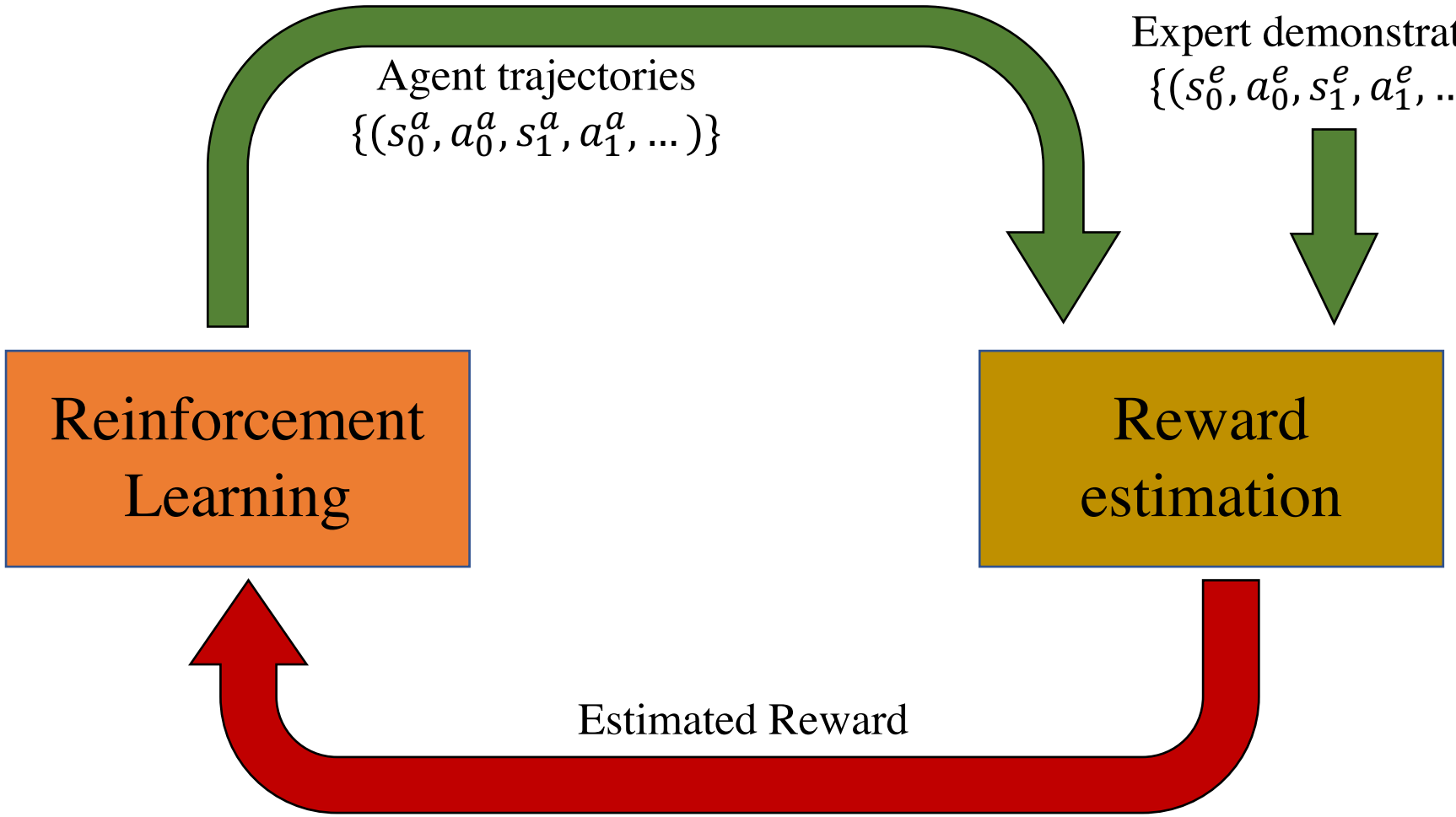
- Supervised learning from states to single-step actions.



- Drawbacks:
  - Cannot recover from deviations unseen in the training expert trajectories.
  - Require a large amount of training samples to learn a good policy.

Inverse reinforcement Learning

- Use comparison between the expert and the agent's current trajectory to generate reward signal.



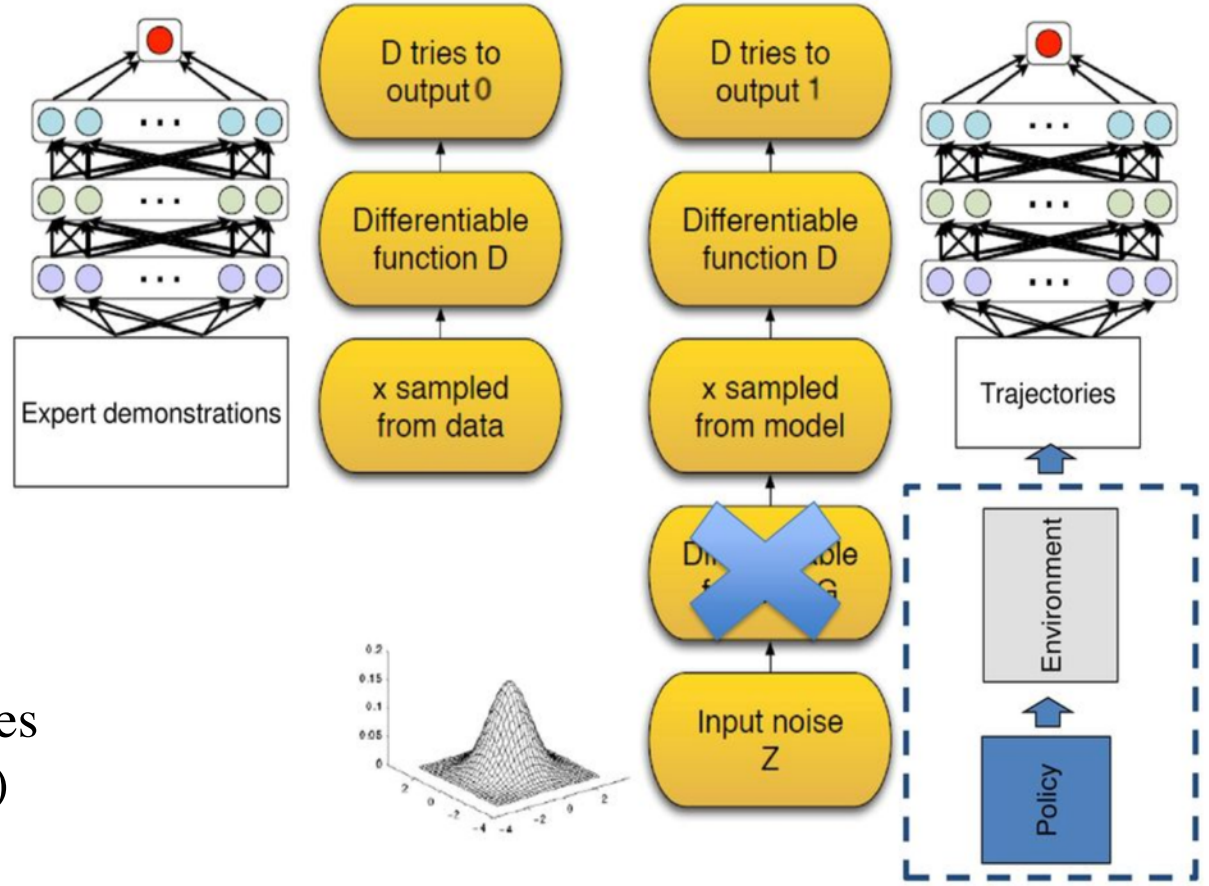
- Peng+ Siggraph2018:
  MSE Loss: $\sum_i ||s_i^e - s_i^a||_2^2$
- Sermanet+, ICRA2018:
  TCN embeddings + MSE loss,
  $\sum_i ||f(s_i^e) - f(s_i^a)||_2^2$

Inverse reinforcement Learning: GAIL

- Update policy until the expert and agent's policy cannot be discriminated.

- Advantages:
  - Learn from few demonstrations.
  - Can recover from unseen deviations in the training set.

- Drawbacks:
  - Require both state and actions.
  - Works on small dimensional low-level state spaces and actions. (joint angles, velocities and torques)



Inverse reinforcement Learning: GAIL

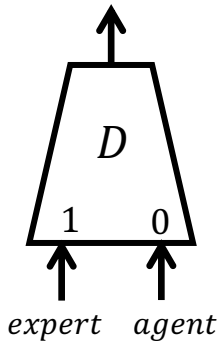
- Occupancy measure of a policy is given as, $\rho_\pi(s,a) = \underbrace{\pi(a|s)\sum_{t=0}\gamma^t P(s_t = s|\pi)}$

roughly the distribution of state and actions visited by expert policy.

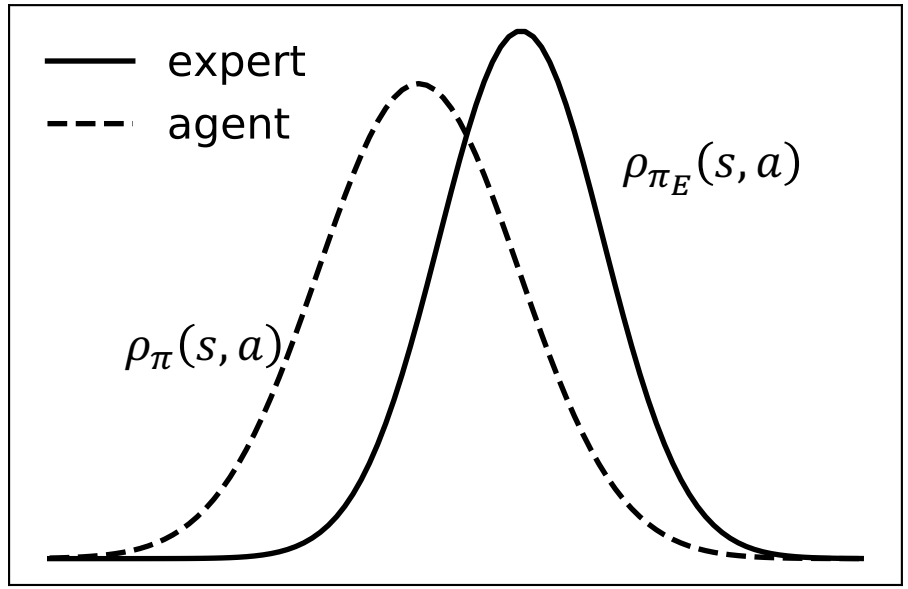
- GAIL minimizes the JS divergence between the agent's and expert's occupancy measure.

GAIL objective (Ho+ Nips2016):

$$\pi^*, D^* = \min_{\pi\in\Pi}\max_{D\in(0,1)} E_{\pi_E}[\log D(s,a)] + E_\pi[\log(1 - D(s,a))]$$



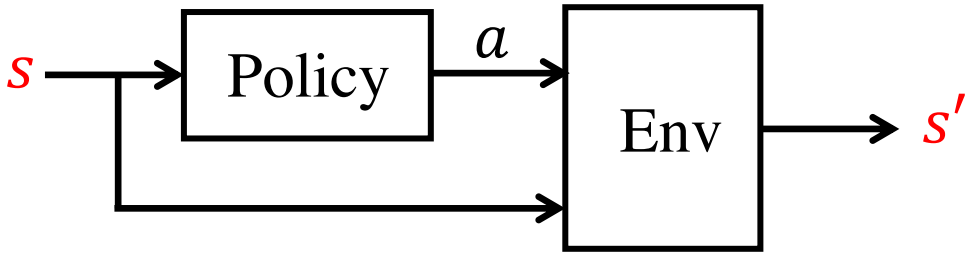
$$\pi^* = \min_{\pi\in\Pi} JS(\rho_\pi(s,a), \rho_{\pi_E}(s,a))$$



Inverse reinforcement Learning from Observations

- The occupancy measure can be extended to represent state visitation frequency (removes the dependency on expert actions)

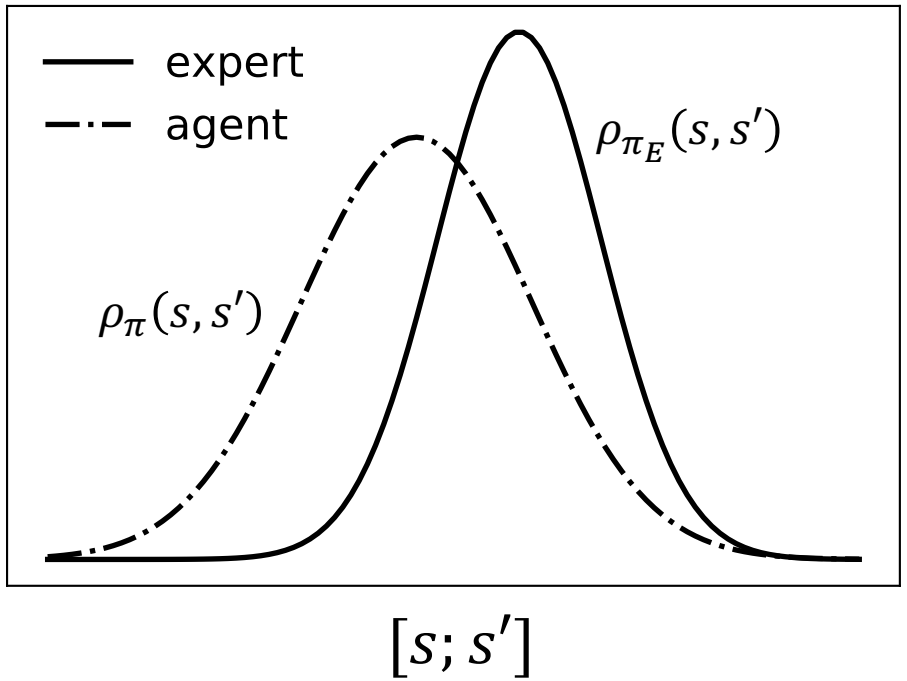
$$\rho_\pi(s, s') = \sum_a p(s'|a, s)\, \pi(a|s) \sum_{t=0} \gamma^t\, P(s_t = s|\pi)$$



GAIFO objective (Torabi+ Arxiv2018) :

$$\pi^*, D^* = \min_{\pi \in \Pi} \max_{D \in (0,1)} E_{\pi_E}[\log D(s, s')] + E_\pi[\log(1 - D(s, s'))]$$

$$\pi^* = \min_{\pi \in \Pi} JS(\rho_\pi(s, s'), \rho_{\pi_E}(s, s'))$$



Visual Imitation learning

- Collecting low-level information in real-world is difficult.
- Visual imitation from video demonstrations is more natural.

Source: Google images

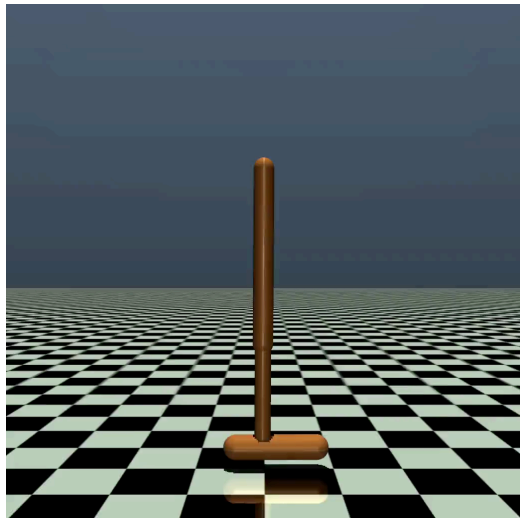
Visual Imitation: problem setting

- Learn from only expert video demonstrations without access to low-level joints information.

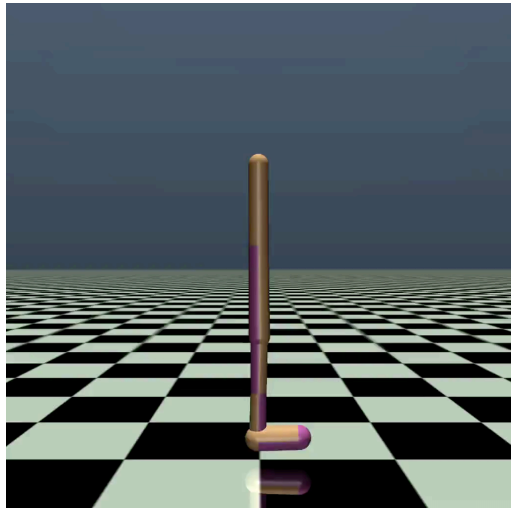
- Challenges:
  - Task relevant feature extraction.
  - Comparison metric between frames.

Previous methods estimate these steps separately

(Peng+ Siggraph2018, Sermanet+, ICRA2018)



Hopper



Walker2d

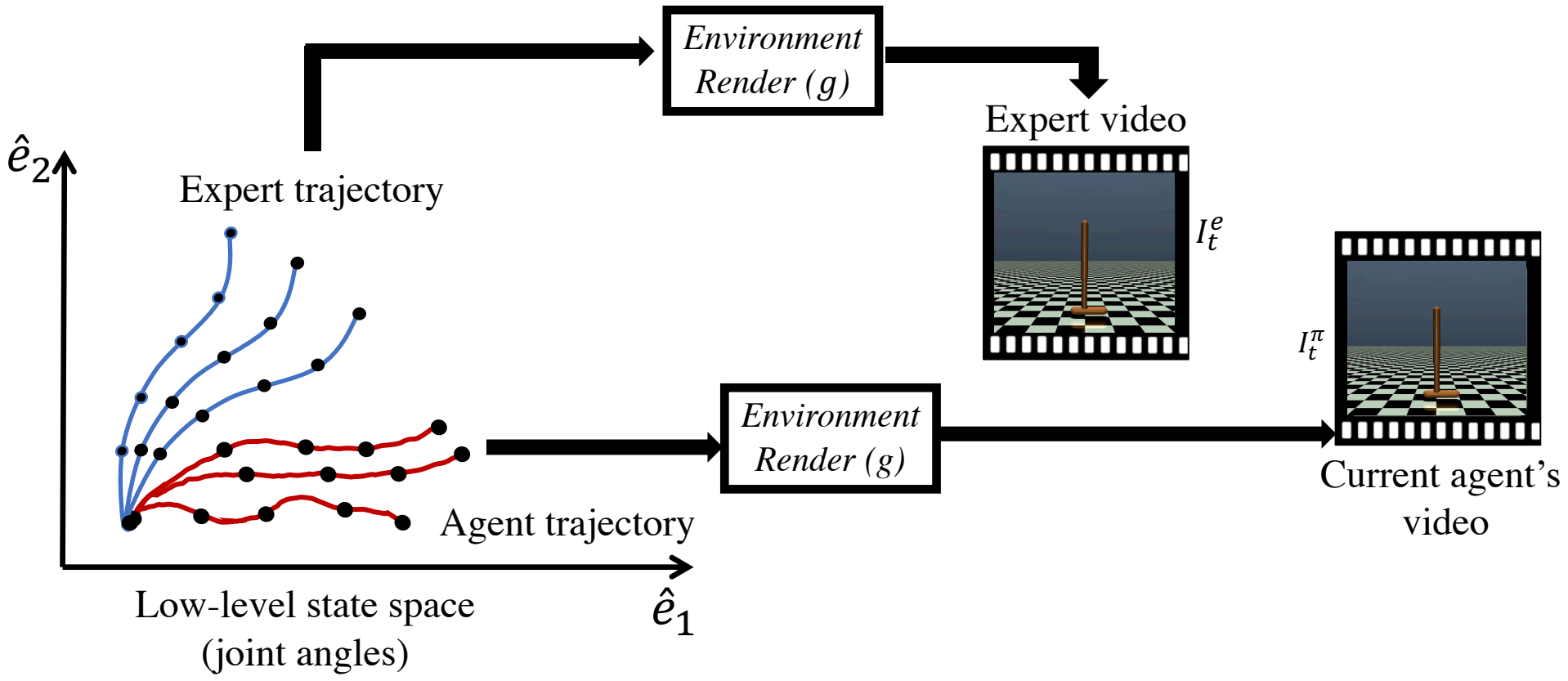
Expert videos

Our solution: Jointly learn feature extraction and comparison metric for robust reward estimation using GANs.

Visual Imitation: Proposed method

- Consider a render mapping $g : \mathcal{S} \rightarrow \mathcal{I}$ from low-level joint states to expert video frames.

If there exists an injective mapping $g : \mathcal{S} \rightarrow \mathcal{I}$, adversarial video learning matches the state visitation frequency.



Visual Imitation: Proposed method

- Proof: Consider an adversarial learning setting from consecutive video frames (I, I') as below:

$$\pi^*, D^* = \min_{\pi \in \Pi} \max_{D \in (0,1)} \underbrace{E_{\pi_E}[\log D(I, I')] + E_{\pi}[\log(1 - D(I, I'))]}$$

$D(I, I')$ directly compares adjacent video frames in expert and agent videos for reward estimation

Using Bayes rule, $D(I, I') = p(E|I, I') = \frac{p(I, I'|E)p(E)}{p(I,I')} = \frac{1}{1+\frac{p(I, I'|A)}{p(I, I'|E)}}$

(Assuming expert and agent samples are equally likely)

Visual Imitation: Proposed method

- Proof: Consider an adversarial learning setting from consecutive video frames (I, I') as below:

$$\pi^*, D^* = \min_{\pi \in \Pi} \max_{D \in (0,1)} E_{\pi_E}[\log D(I, I')] + E_{\pi}[\log(1 - D(I, I'))] \quad \text{and} \quad D(I, I') = \frac{1}{1 + \frac{p(I, I'|A)}{p(I, I'|E)}}$$

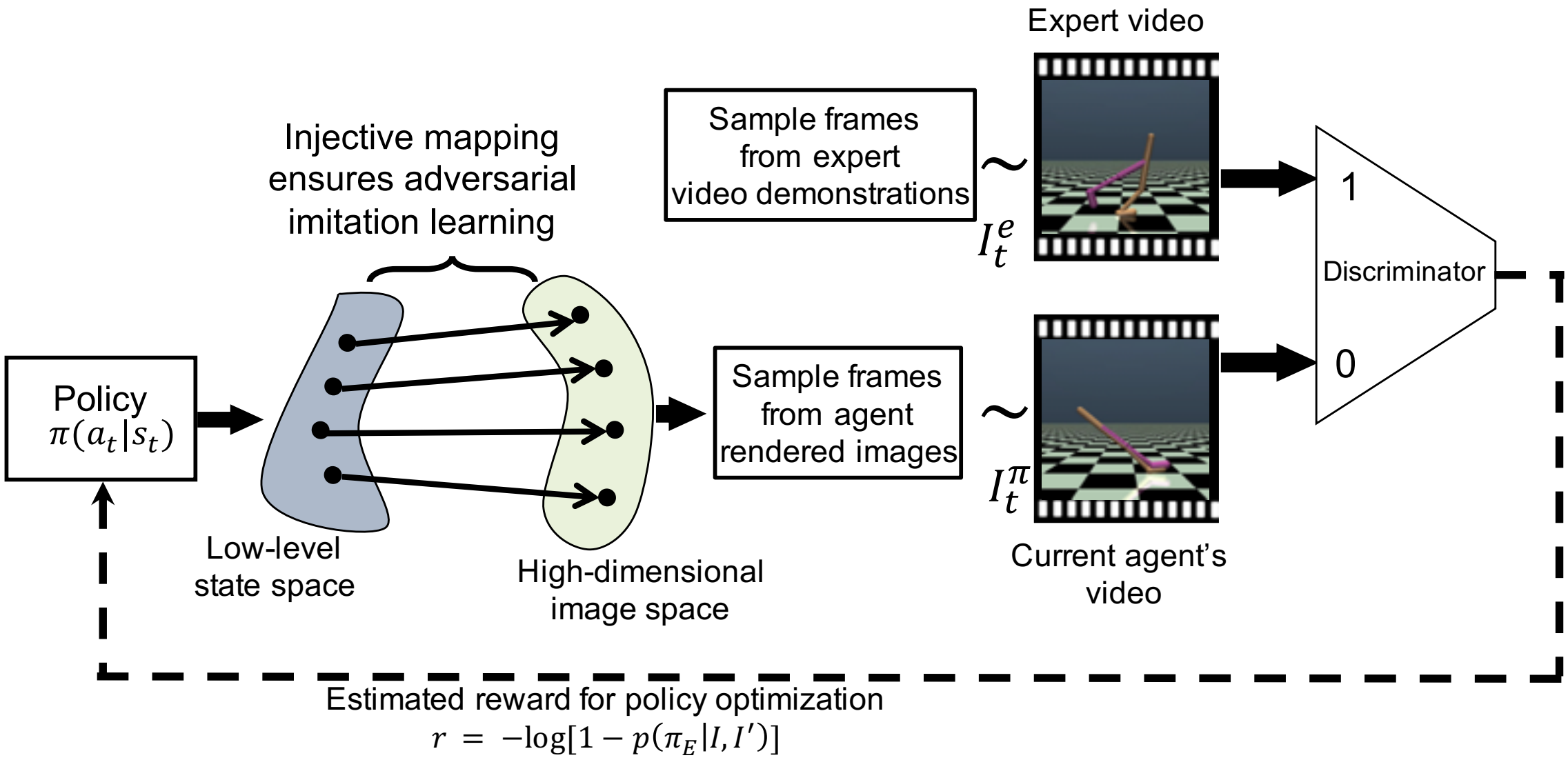
Using Bayes rule, $\frac{p(I, I'|A)}{p(I, I'|E)} = \frac{p(s, s'|A)\,|\det J|^{-1}}{p(s, s'|E)\,|\det J|^{-1}} = \frac{p(s, s'|A)}{p(s, s'|E)}$

(if $J = \frac{dI}{dS}$ is non-singular, implying the mapping g should be injective
- Example: no occlusion,
no motion along the axis on camera)

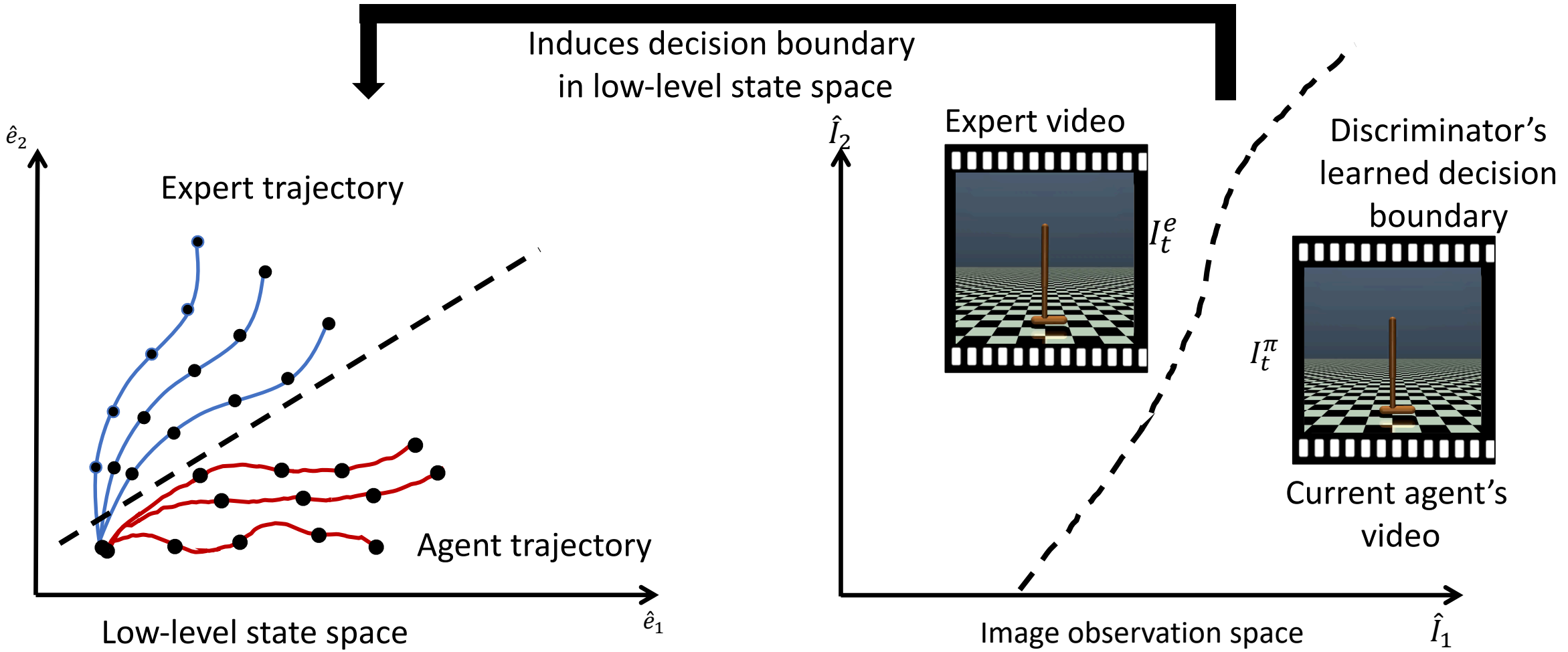
$$\Rightarrow D(I, I') = D(s, s')$$

$$\Rightarrow \pi^*, D^* \text{ is the solution for } \min_{\pi \in \Pi} \max_{D \in (0,1)} \underbrace{E_{\pi_E}[\log D(s, s')] + E_{\pi}[\log(1 - D(s, s'))]}_{\text{This is similar to learning from low-level trajectories}}$$

Visual Imitation: Proposed method



Visual Imitation: Proposed method

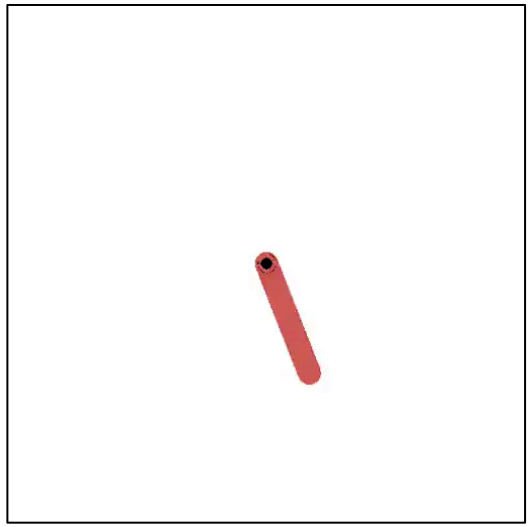


Visual Imitation: Results

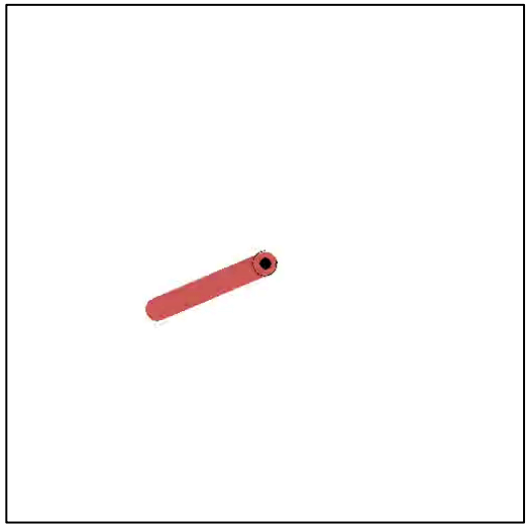
(Comparison with other methods)

Pendulum environment

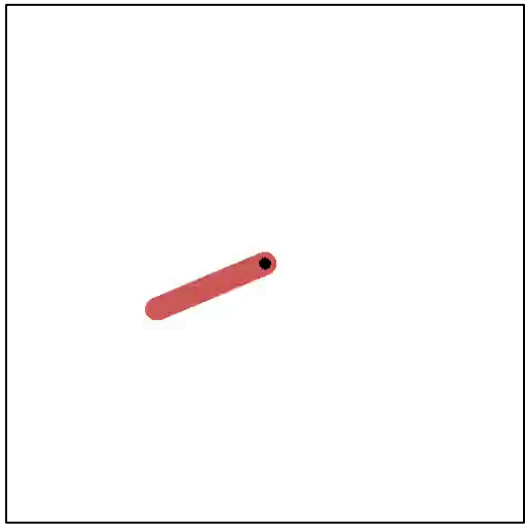
Number of trajectories: 1



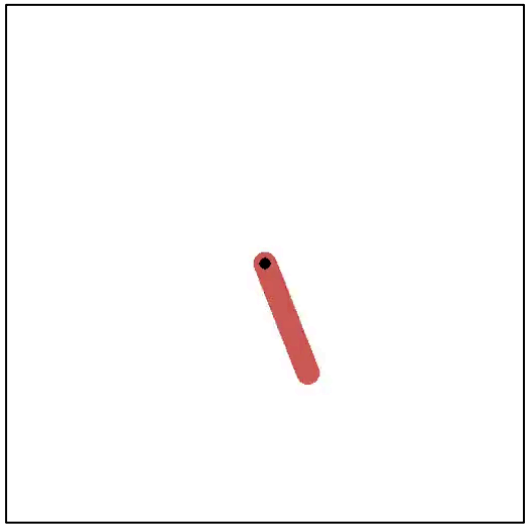
GAIL
(Learned from low level states-actions)



DeepMimic+Pixel



DeepMimic +
Single View TCN



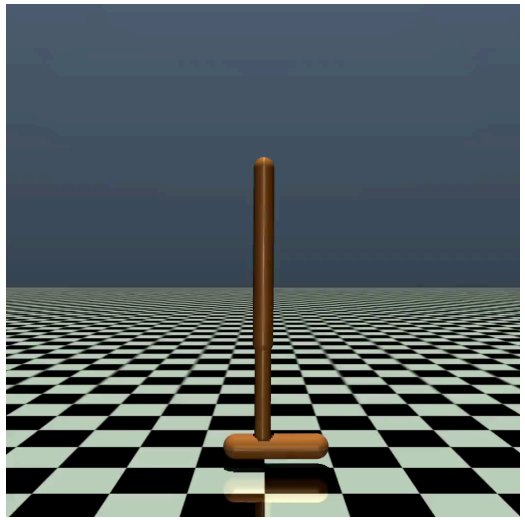
VIGAN (Proposed)

Visual Imitation: Results

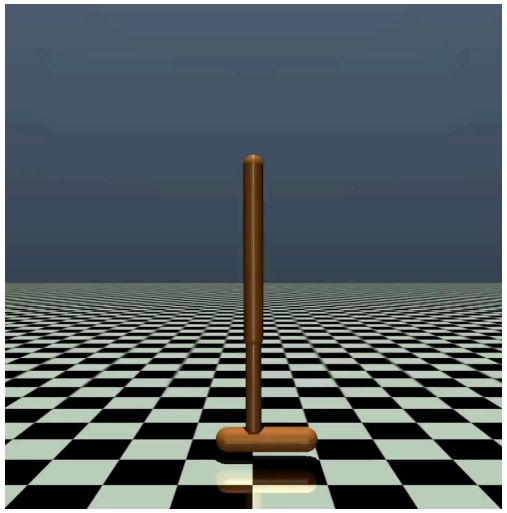
(Comparison with other methods)

Hopper environment

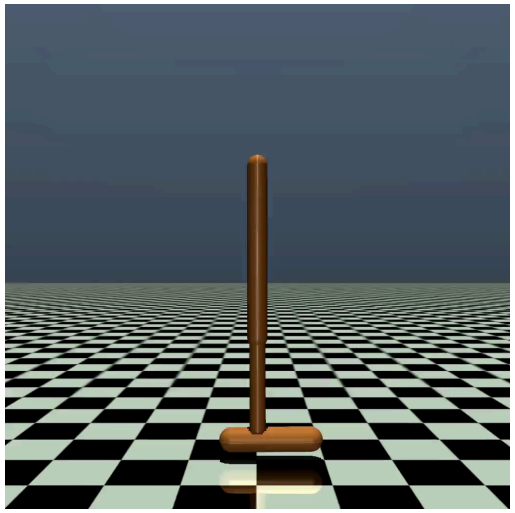
Number of trajectories: 1



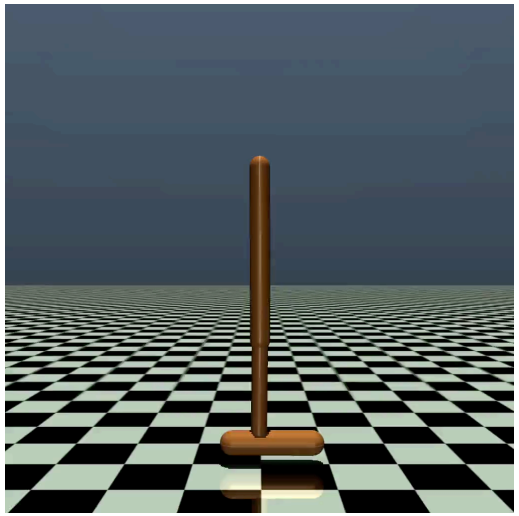
GAIL
(Learned from low level states-actions)



DeepMimic+Pixel



DeepMimic +
Single View TCN



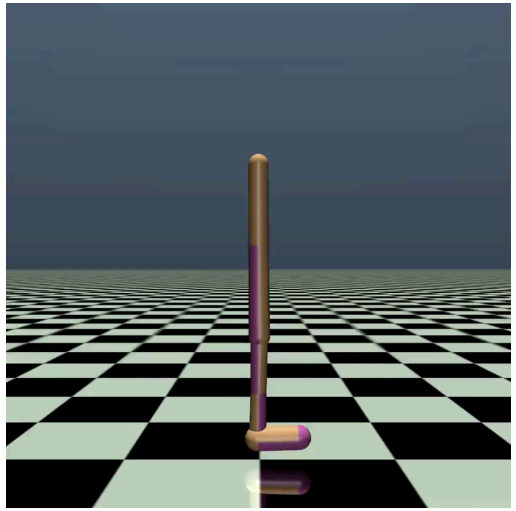
Proposed

Visual Imitation: Results

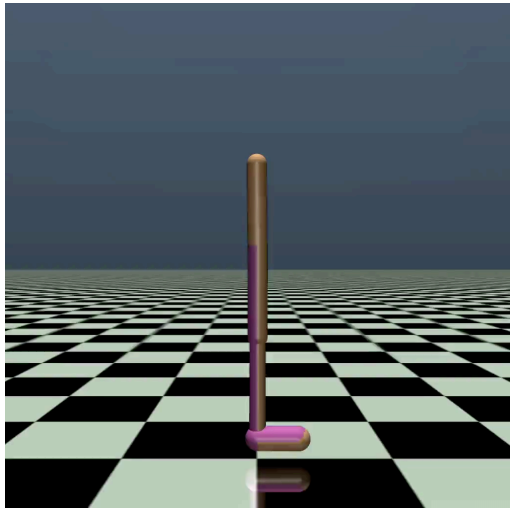
(Comparison with other methods)

Walker2d environment

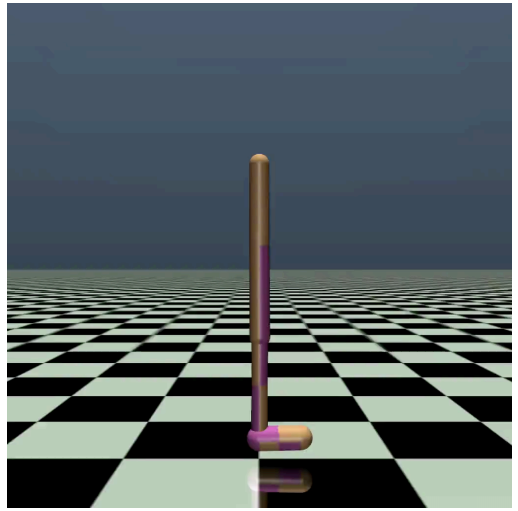
Number of trajectories: 4



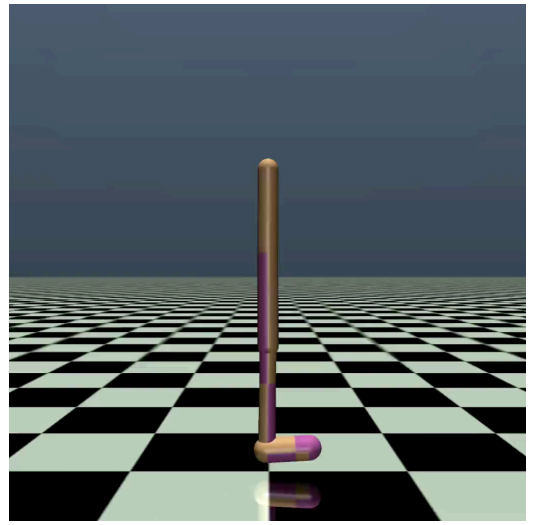
GAIL
(Learned from low level states-actions)



DeepMimic+Pixel



DeepMimic +
Single View TCN



Proposed

Visual Imitation: Results

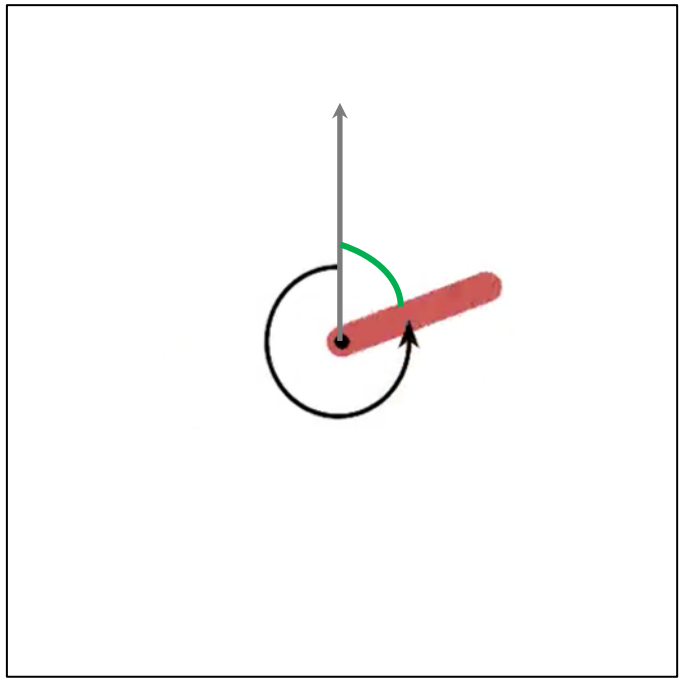
(Comparison with other methods)

Quantitative results shows the superiority of our method for all standard environments

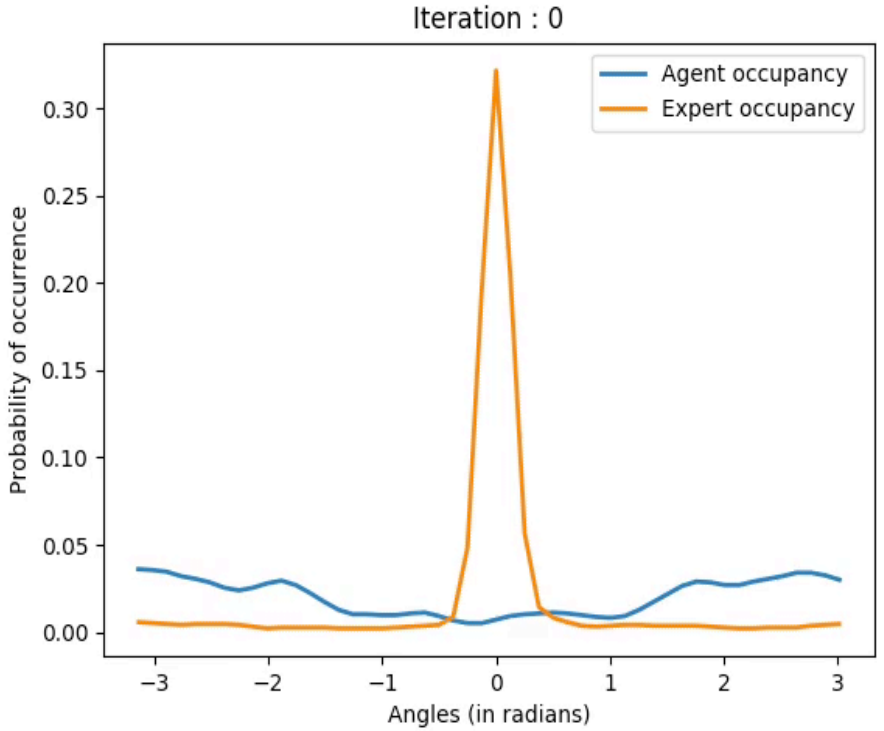
Env	#Traj	GAIL	DM+Pix	DM+TCN	Proposed
CartPole	1	*200.0*	191.6	200.0	200.0
	5	*200.0*	200.0	200.0	200.0
	10	*200.0*	200.0	200.0	200.0
Pendulum	1	*-242.0*	−1203.4	−732.3	**−194.4**
	5	*-278.6*	−1247.9	**−205.6**	−287.5
	10	*-313.0*	−1298.3	−209.2	**−177.4**
Hopper	1	*3607.1*	1012.1	619.3	**3053.6**
	4	*3159.6*	1008.3	610.6	**2513.3**
	11	*3466.6*	979.8	624.5	**2490.3**
	25	*3733.5*	990.0	605.5	**2812.2**
Walker2d	1	*5673.6*	537.3	747.8	**2505.9**
	4	*5160.5*	729.9	3659.5	**4211.2**
	11	*4920.7*	846.3	629.4	**4340.4**
	25	*5596.6*	495.5	3356.4	**5606.4**

Convergence of JS divergence of low-level states by VIGAN

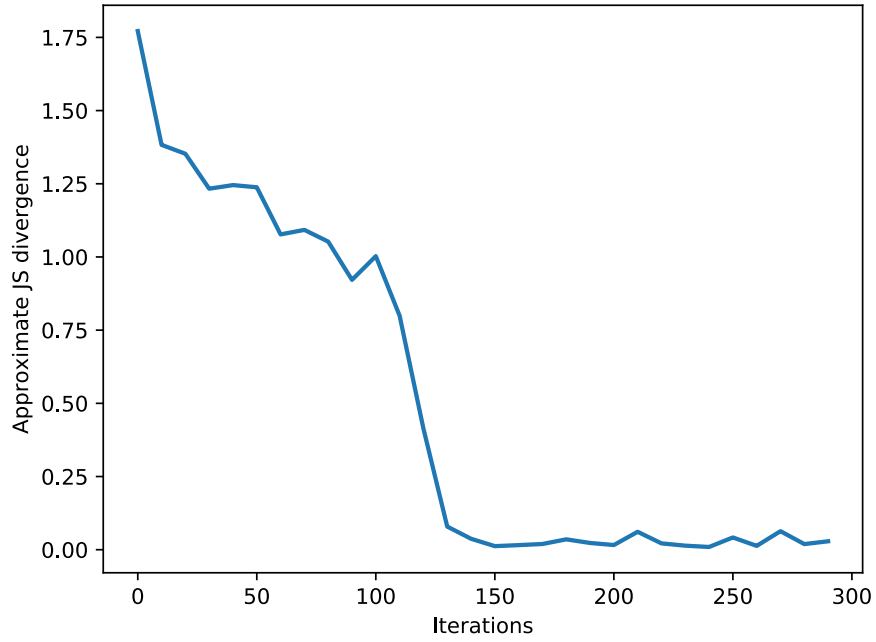
Imitation from expert video demonstration results in occupancy measure matching of the agent's pendulum angle to the expert.



Computing angle θ for Pendulum-v0 environment



Matching occupancy measure of angle θ by proposed video imitation learning.

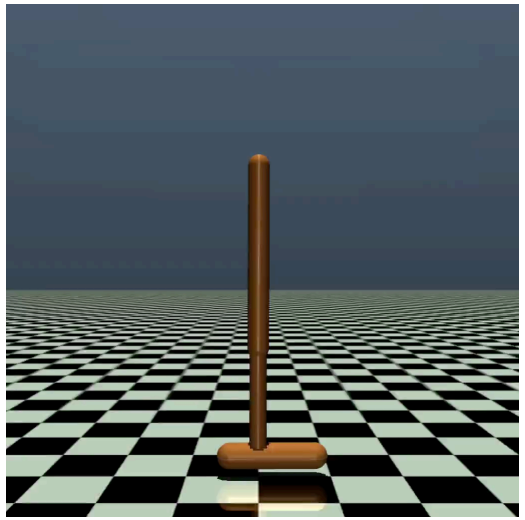


Approximate JS divergence

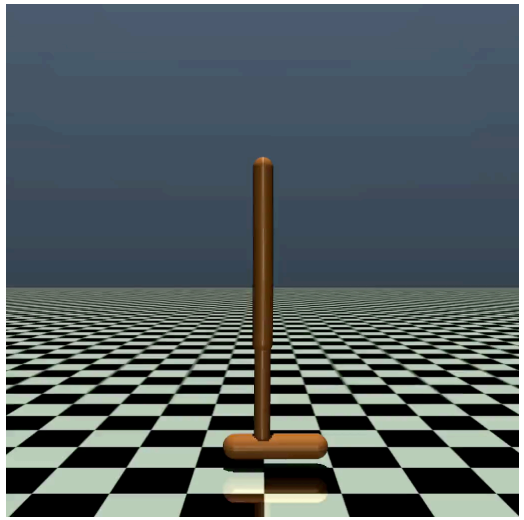
Visual Imitation: Results

(Robustness to noisy expert data)

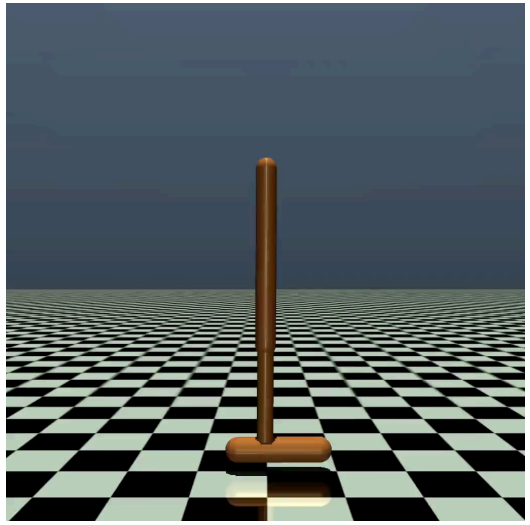
Robustness of our method to input shaking noise on Hopper.



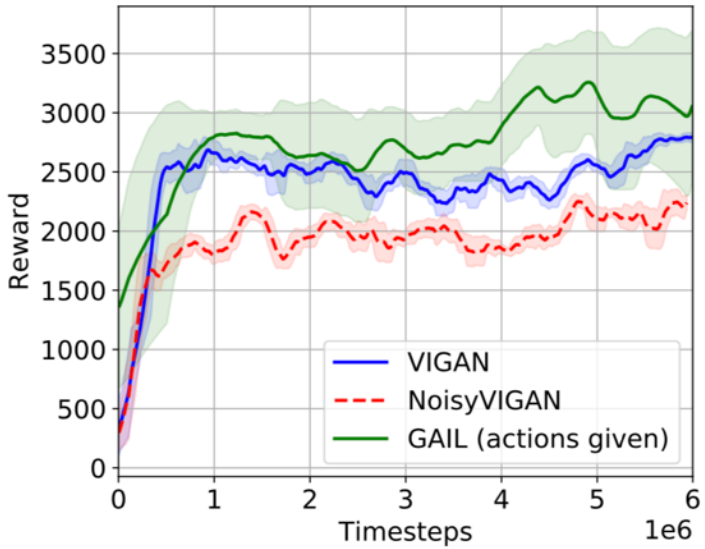
Noisy video demo
(Shaking camera)



Proposed
(1 trajectory)



Proposed
(10 trajectories)



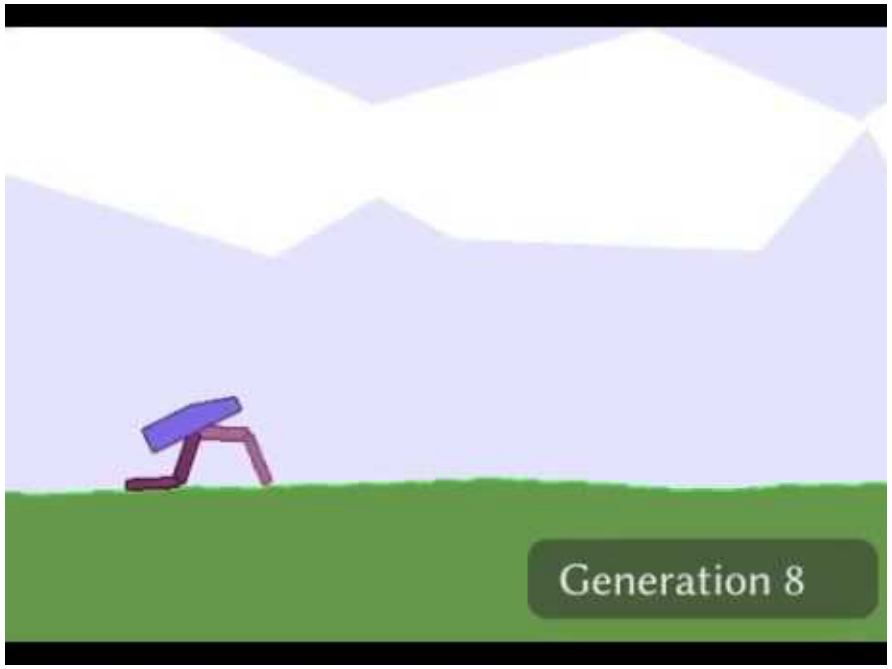
(a) Noisy demonstrations (Hopper)

Visual Imitation: Results

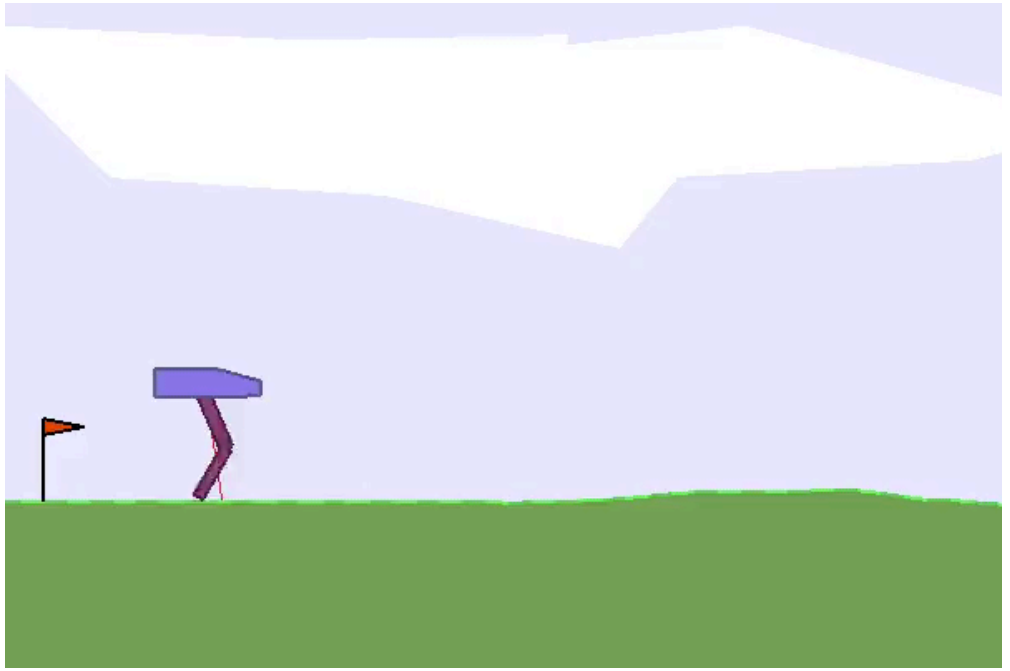
(Learning from YouTube videos)

Learned policy from YouTube videos with different walking styles.

Video 1 : https://www.youtube.com/watch?v=uwz8JzrEwWY (we used clip from 0:46 to 0:55, Generation 512)



Video 1: Original YouTube video



Proposed
(Imitation from Video 1)

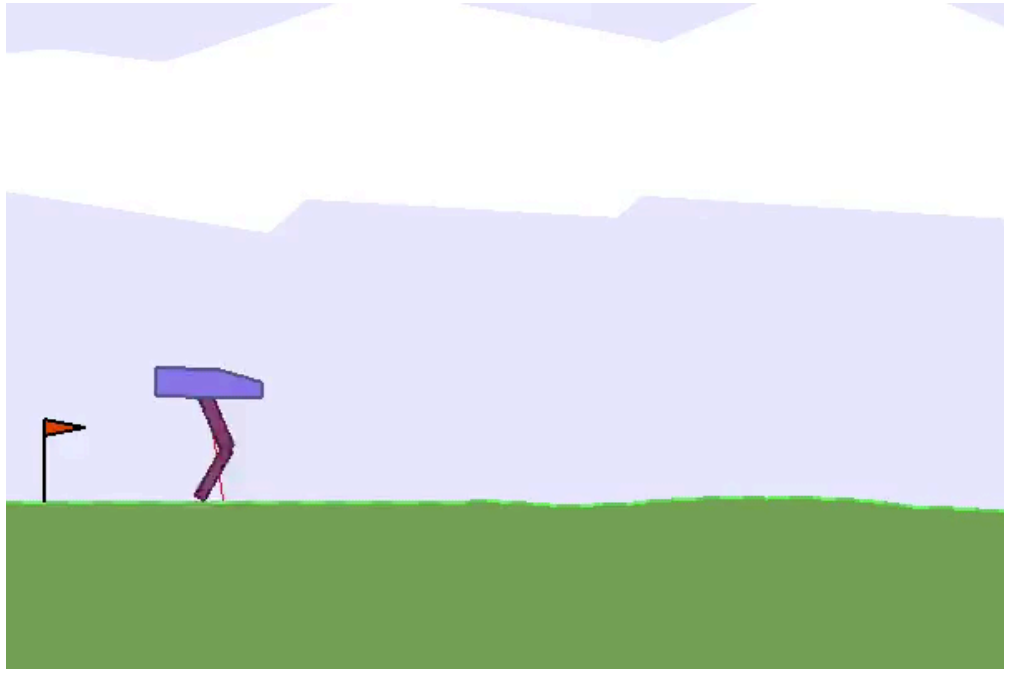
Visual Imitation: Results

(Learning from YouTube videos)

Learned policy from YouTube videos with different walking styles.

Video 2 : https://www.youtube.com/watch?v=nWd2cN5oriM (we used the entire 85 second video)

Video 2: Original YouTube video

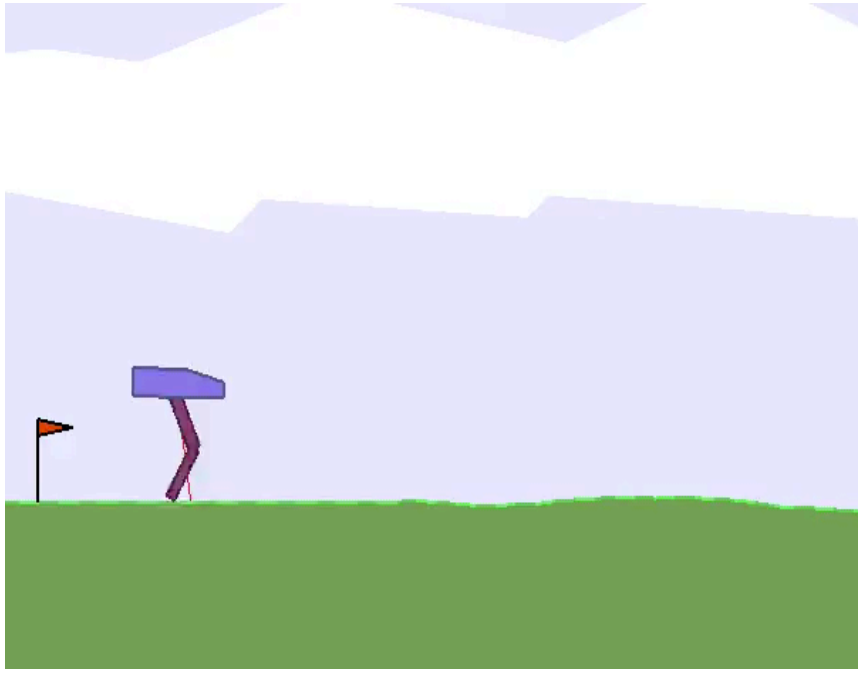


Proposed
(Imitation from Video 2)

Visual Imitation: Results

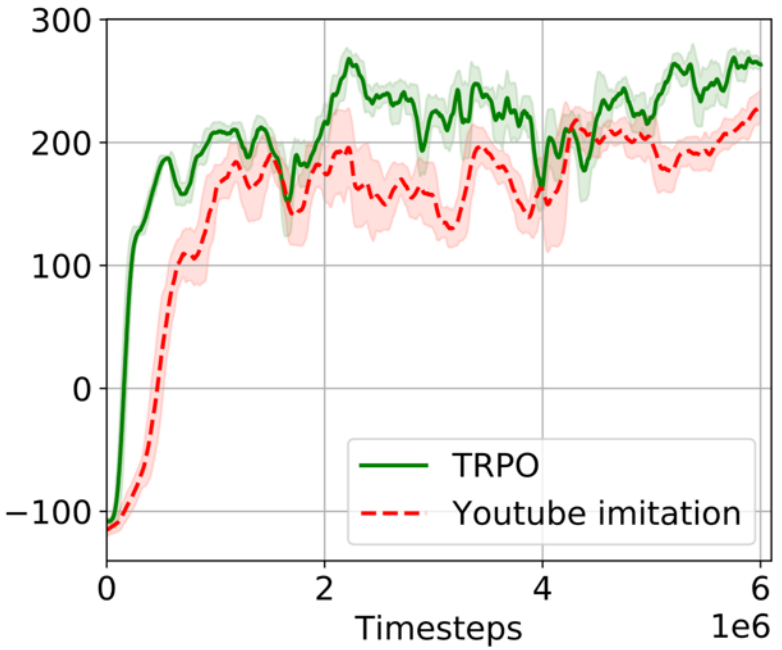
(Learning from YouTube videos)

Comparison with learning from true reward



Proposed
(Imitation from Video 2)

TRPO
(Learned from default reward)



Our method performs
similar to TRPO

Thank you

Questions?