Provably Efficient Imitation Learning from Observations

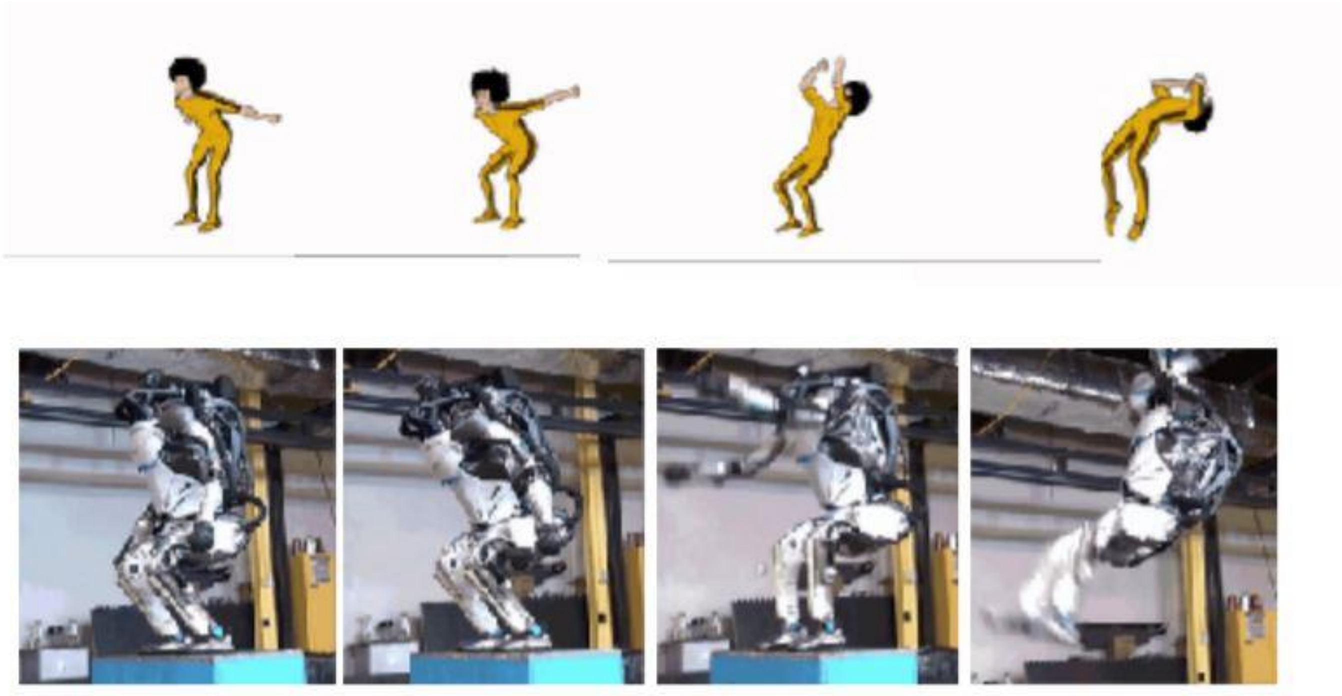
Wen Sun Anirudh Vemula Byron Boots J.Andrew Bagnell

Presented by: Zichu Liu

Motivation: Imitation Learning from Observations (ILFO)

Trajectories of Observations

Learning From Observations



No interactive expert, no expert action, no reset, no cost signals

Finite time horizon (T-step) Episodic MDP

Forward Adversarial Imitation Learning (FAIL)

$$\pi_0(a|x_0), \pi_1(a|x_1), \cdots\cdots, \pi_T(a|x_T)$$

Decomposition into T subtasks

At time T, $\{\pi_1, \dots, \pi_{T-1}\}$ are learned already and fixed.

A state distribution $\upsilon_T(x)$ is induced by $\{\pi_1, \dots, \pi_{T-1}\}$

Expert policy π^* naturally induces a distribution μ_T^*

We want to learn a policy $\pi_T \in \Pi_T$ such that the resulting observation distribution from $\{\pi_1, \dots, \pi_{T-1}, \pi_T\}$ at time step T is close to the expert's observation distribution μ_T^* at time step T

Divergence: Integral Probability Metrics (IPM)

$$d_{\mathcal{F}}(P_1, P_2) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim P_1}[f(x)] - E_{x \sim P_2}[f(x)] \right)$$

$\mathcal{F} = \left\{f : \left||f|\right|_\infty \leq 1\right\}$: Total Variation distance

$\mathcal{F} = \{f : \left||f|\right|_L \leq 1\}$: Wasserstein distance

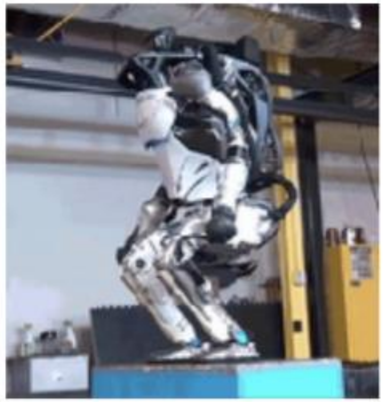
$\mathcal{F} = \{f : \left||f|\right|_H \leq 1\}$: Maximum mean discrepancy

Learning the First Policy π_0

$\sim \mu_1^*(x)$

Expert Distribution

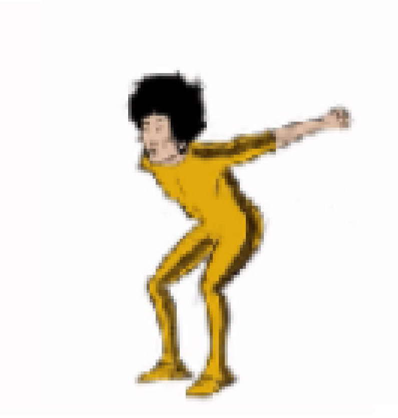
$\min_{\pi_0 \in \Pi} \max_{f \in \mathcal{F}} f($  $) - f($  $)$



$\sim v_1(x) = \sum_{x_0, a_0} P(x_0)\, \pi_0(a_0|x_0) P(x|x_0, a_0)$

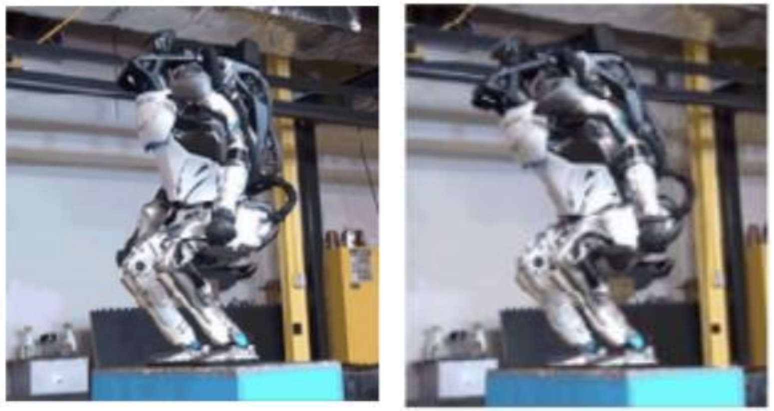
Learner Distribution

Learning the Second Policy π_1



$\sim \mu_2^*(x)$

Expert Distribution

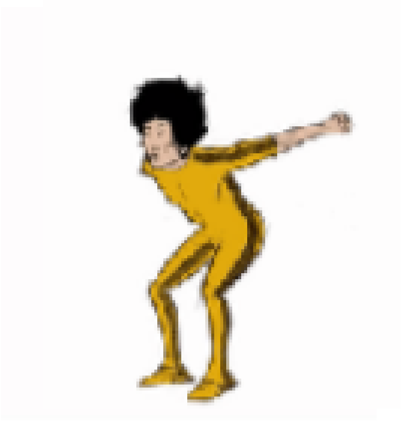


Learner Distribution

$\min_{\pi_1 \in \Pi} \max_{f \in \mathcal{F}} f($  $) - f($  $)$

$\sim v_2(x) = \sum_{x_1, a_1} v_1(x_1)\, \pi_1(a_1 | x_1) P(x | x_1, a_1)$

Learning the Second Policy π_1



$\sim \mu_2^*(x)$

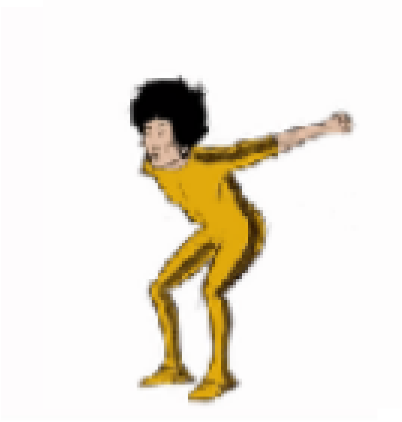
Expert Distribution

$\min_{\pi_1 \in \Pi} \max_{f \in \mathcal{F}} f($  $) - f($  $)$

$\sim v_2(x) = \sum_{x_1,a_1} \boxed{v_1(x_1)} \pi_1(a_1|x_1) P(x|x_1,a_1)$

Learner Distribution

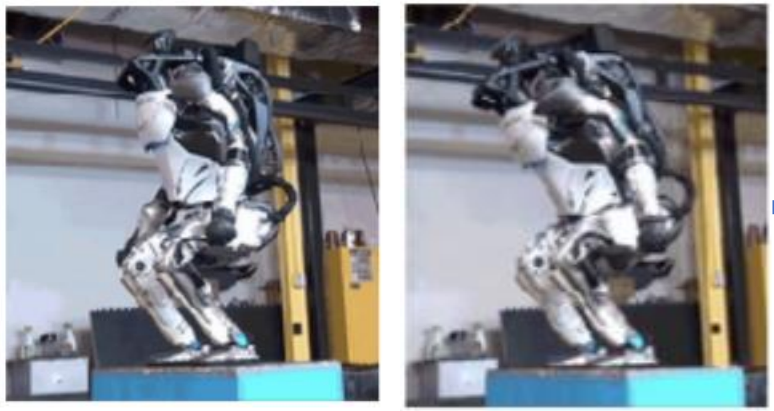
Learning the Second Policy π_1



$\sim\mu_2^*(x)$

Expert Distribution

$\min_{\pi_1\in\Pi}\max_{f\in\mathcal{F}} f($  $) - f($  $)$



$\sim v_2(x) = \sum_{x_1,a_1} v_1(x_1)\,\pi_1(a_1|x_1)P(x|x_1,a_1)$

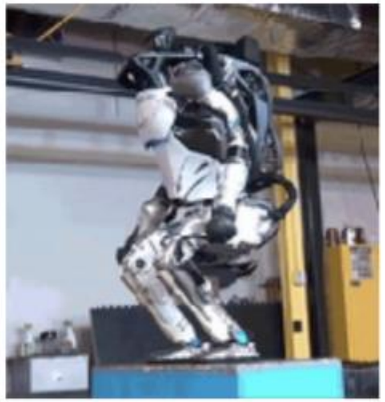
Learner Distribution

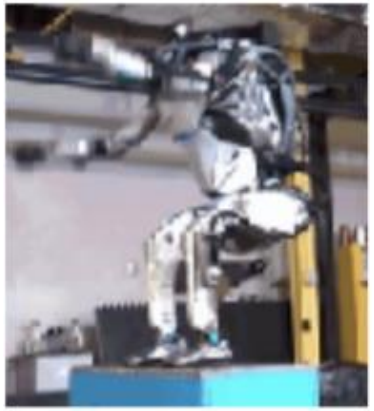
Learning the Third Policy π_2

$\sim \mu_3^*(x)$

Expert Distribution

$\min_{\pi_2 \in \Pi}$ $\max_{f \in \mathcal{F}} f($  $) - f($  $)$





$\sim v_3(x)$

Learning π_T

Given the distribution v_T induced by $\{\pi_1, \cdots, \pi_T\} \in \Pi$, the observation distribution at time step $T+1$ as

$$v_{T+1}(x) = \sum_{x_T, a_{T-1}} v_T(x_T)\,\pi(a_T|x_T)P(x|x_T, a_T)$$

Expert distribution at time step $T+1$ is denoted as μ^*_{T+1}

π_T is obtained via minimizing the divergence between v_{T+1} and μ^*_{T+1}

$$\pi_T = \underset{\pi \in \Pi}{\operatorname{argmin}} \max_{f \in \mathcal{F}} f(v_{T+1}) - f(\mu^*_{T+1})$$

Learning π_T

However, the divergence $\max_{f \in \mathcal{F}} f(v_{T+1}) - f(\mu^*_{T+1})$ is not directly measurable since we do not have access to μ^*_{T+1} but only samples from μ^*_{T+1}.

Learning π_T

To estimate this divergence, we draw a dataset

$$\mathcal{D} = \{(x_T^i, a_T^i, x_{T+1}^i)\}$$

such that $x_T^i \sim \upsilon_T, \mathrm{a}_\mathrm{T}^\mathrm{i} \sim \mathrm{U(A)}, \mathrm{x}_{\mathrm{T+1}}^\mathrm{i} \sim \mathrm{P}(\cdot\,|x_T^i, a_T^i)$

Observations from expert $\mathcal{D}^* = \left\{\tilde{x}_{T+1}^i\right\}_{i=1}^{N'} \sim \mu_{T+1}^*$

Learning π_T

Empirical estimation of divergence:

$$\max_{f \in \mathcal{F}} \left(\frac{K}{N} \sum^{N} \pi\left(a_T^i \middle| x_T^i\right) f\left(x_{T+1}^i\right) - \frac{1}{N'} \sum^{N'} f\left(\tilde{x}_{T+1}^i\right) \right)$$

Where the importance weight $K\pi(a_T^i | x_T^i)$ is used to account for the fact that we draw actions uniformly from A but want to evaluate π.

Learning π_T

Now define the utility function of the two-player game:

$$u(\pi, f) = \frac{K}{N}\sum^{N}\pi\left(a_T^i \middle| x_T^i\right)f\left(x_{T+1}^i\right) - \frac{1}{N'}\sum^{N'} f\left(\tilde{x}_{T+1}^i\right)$$

Then we have the two-player game with solution (π^*, f^*):

$$f^* = \underset{\mathrm{f} \in \mathcal{F}}{\operatorname{argmax}}\, u(\pi^*, f)$$

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\, u(\pi, f^*)$$

Algorithm 1 Min-Max Game $(\mathcal{D}^\star, \mathcal{D}, \Pi, \mathcal{F}, T)$

1: Initialize $\pi^0 \in \Pi$
2: **for** $n = 1$ to T **do**
3: $\quad f^n = \arg\max_{f \in \mathcal{F}} u(\pi^n, f)$ (LP Oracle)
4: $\quad u^n = u(\pi^n, f^n)$
5: $\quad \pi^{n+1} = \arg\min_{\pi \in \Pi} \sum_{t=1}^{n} u(\pi, f^t) + \phi(\pi)$ (Regularized CS Oracle)
6: **end for**
7: **Output**: $\pi^{n^\star}$ with $n^\star = \arg\min_{n \in [T]} u^n$

Algorithm 2 FAIL($\{\Pi_h\}_h, \{\mathcal{F}_h\}_h, \epsilon, n, n', T$)

Set $\boldsymbol{\pi} = \emptyset$
for $h = 1$ to $H - 1$ **do**
  Extract expert's data at $h + 1$: $\tilde{\mathcal{D}} = \{\tilde{x}^i_{h+1}\}_{i=1}^{n'}$
  $\mathcal{D} = \emptyset$
  for $i = 1$ to n **do**
    Reset $x_1^{(i)} \sim \rho$
    Execute $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{h-1}\}$ to generate state x^i_h
    Execute $a^i_h \sim U(\mathcal{A})$ to generate x^i_{h+1} and add $(x^i_h, a^i_h, x^i_{h+1})$ to $\mathcal{D}$
  end for
  Set π_h to be the return of Algorithm 1 with inputs $\left(\tilde{\mathcal{D}}, \mathcal{D}, \Pi_h, \mathcal{F}_{h+1}, T\right)$
  Append π_h to $\boldsymbol{\pi}$
end for

Assumption: (Realizability and Capacity of Function Class)

Assume Π and $\mathcal{F}$ are finite and contain π_t^* and f_t^*, i.e.,

$$\pi_t^* \in \Pi \text{ and } f_t^* \in \mathcal{F}, \forall t \in [0, T]$$

Convergence Result

Theorem 3.1. *Given* $\epsilon \in (0,1], \delta \in (0,1]$, *set* $T = \Theta\left(\frac{4K^2}{\epsilon^2}\right)$, $N = N' = \Theta\left(\frac{K \log(|\Pi_h||\mathcal{F}_{h+1}|/\delta)}{\epsilon^2}\right)$, *Algorithm 1 outputs* π *such that with probability at least* $1 - \delta$,

$$\left| d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1}) - \min_{\pi' \in \Pi_h} d_{\mathcal{F}_{h+1}}(\pi'|\nu_h, \mu^\star_{h+1}) \right| \leq O(\epsilon).$$

Convergence Result

Given $\epsilon \in (0,1], \delta \in (0,1]$, algorithm 1 outputs π such that with probability $1-\delta$,

$$|\{\max_{\mathrm{f}} f(\mathcal{v}_{T+1}) - f(\mu^*_{T+1})\} - \{\min_{\pi'} \max_{\mathrm{f}} f(\mathcal{v}_{T+1}) - f(\mu^*_{T+1})\}| < O(\epsilon)$$

$$\left(\left|Div(\pi) - \min_{\pi'} Div(\pi')\right| < O(\epsilon)\right)$$

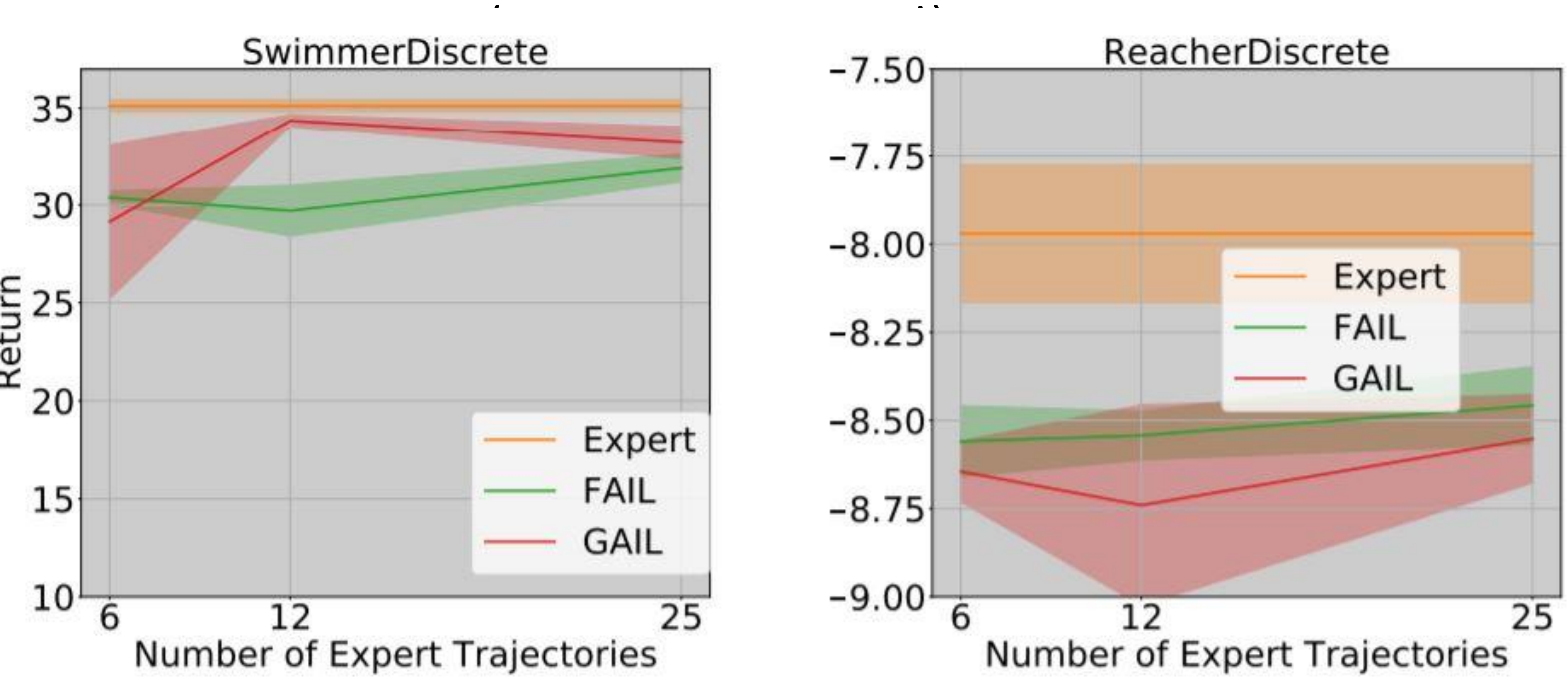
Simulation:

Model	T	Dense/Sparse Reward Task
Swimmer	100	Dense
Reacher	50	Dense/Sparse
FetchReach	50	Sparse

Simulation:

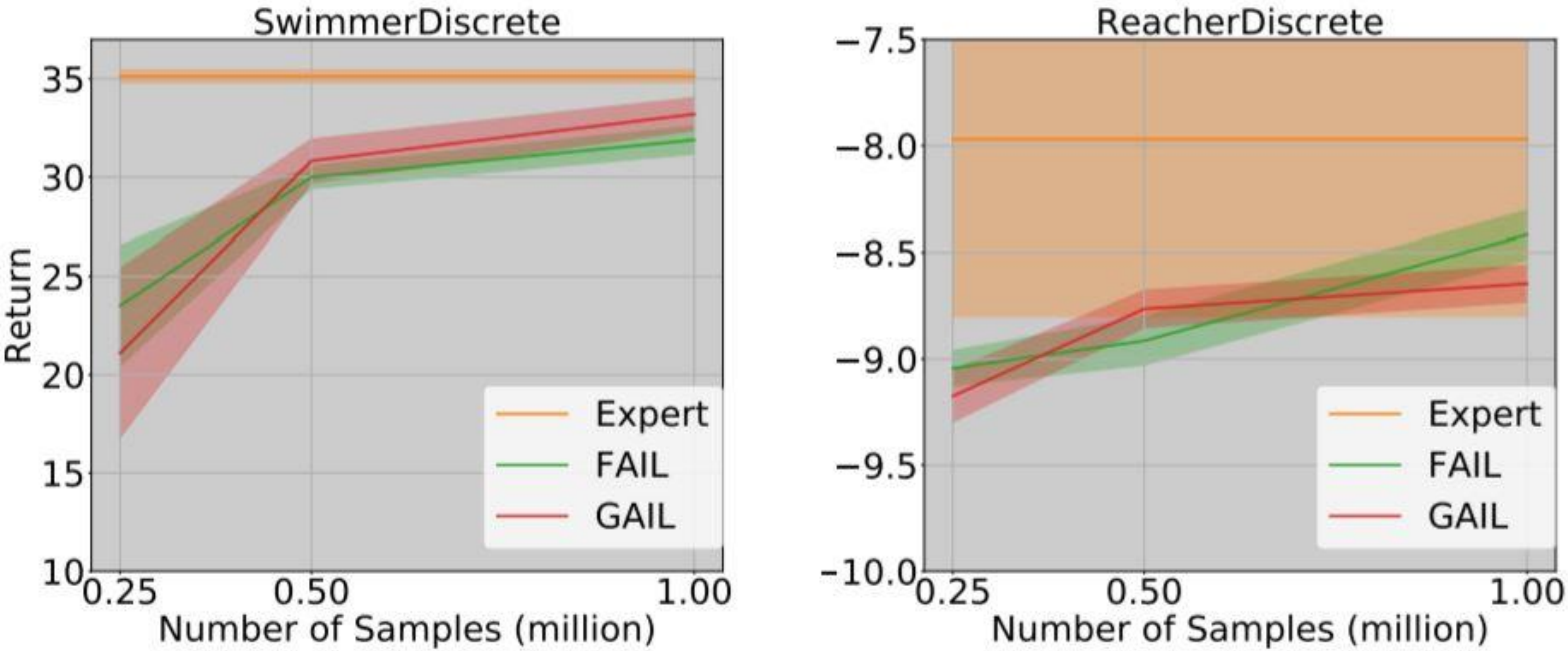
Compare FAIL with modified GAIL:

The modified version of GAIL uses RL methods to minimize the divergence between the learner's average state distribution and expert's average state distribution.



SwimmerDiscrete
Return
Number of Expert Trajectories
Expert
FAIL
GAIL
ReacherDiscrete
Number of Expert Trajectories
Expert
FAIL
GAIL

Experiment (Dense Reward)



Experiment (Sparse Reward)

Experiment (Sparse Reward)

Summary

- This paper point out a new direction of imitation learning research: imitation learning from observation alone. (ILFO)
- Propose FAIL, an algorithm that is theoretically guaranteed to solve the ILFO problems.
- Modify GAIL to solve ILFO problem, experimentally demonstrate that GAIL and FAIL work equivalently well in problems with dense reward, and FAIL outperforms GAIL on sparse reward MDPs.