Intrinsic Reward Driven Imitation Learning via Generative Model

Xingrui Yu, Yueming Lyu and Ivor W. Tsang

Australian Artificial Intelligence Institute
Faculty of Engineering and Information Technology
University of Technology Sydney

July 20, 2020

Outline

- Background
- Imitation Learning
- Representative Imitation Learning Methods
- Generative Intrinsic Reward driven Imitation Learning
  - Main Idea
  - Experiments and Results
  - Conclusion and Future Direction

State

- Experience is a sequence of observations, actions, rewards

$$o_1, r_1, a_1, o_2, r_2, \cdots, a_{t-1}, o_t, r_t$$

- The state is a summary of experience

$$s_t = f(o_1, r_1, a_1, o_2, r_2, \cdots, a_{t-1}, o_t, r_t)$$

Too complex !!!

- In a fully observed environment

$$s_t = f(o_t)$$

Too simple !!!

- State $s_t \in \mathcal{S}$ can be discrete or continuous
- Action $a_t \in \mathcal{A}$ can be discrete or continuous

Markov Decision Process (MDP)

- Trajectory τ is sequence of states and actions

$$\tau = (s_1, a_1, s_2, a_2, \cdots, a_{t-1}, s_t)$$

- In a MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$

$$s_{t+1} = f(s_t, a_t)$$

- Environment transition distribution $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}_+$ (a.k.a. dynamics)

$$p(s_{t+1}|s_t, a_t)$$

- Reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$

$$r(s_t, a_t)$$

- Policy π

$$a_t \sim \pi(a_t|s_t)$$

Expected Discounted Return

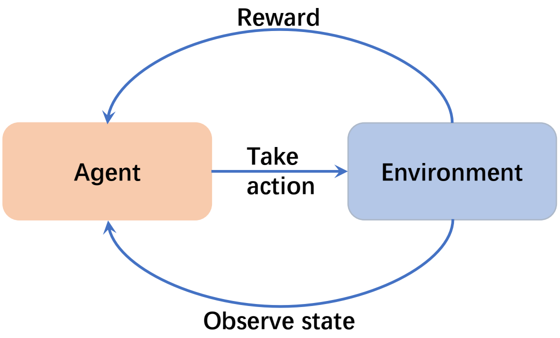
- Discount factor $\gamma \in (0, 1)$
- Expected Discounted Return of the Policy π

$$\eta(\pi) = \mathbb{E}_{\tau}\left[\sum_{t=0} \gamma^{t} r_t\right]$$

where $\tau = (s_0, a_0, \cdots, a_{T-1}, s_T)$ denotes the trajectory, $s_0 \sim \mathbb{P}_0(s_0)$, $a_t \sim \pi(a_t|s_t)$, and $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$.

Reinforcement Learning (RL)

Reinforcement Learning: Learning policies guided by sparse rewards, e.g., win a game. Agent chooses actions so as to maximize expected cumulative reward over a time horizon.



Some advanced solutions in Deep RL, e.g. DQN, REINFORCE, Actor-Critic, PPO

[Tutorial from David Silver]

Where is it successful so far?

- In simulation, where we can afford a lot of trials, easy to parallelize.
- Not in many real-world systems
  - we cannot afford to fail;
  - safety concerns;
  - reward engineering is usually difficult.

Demonstrations

"rather than having a human expert tune a system to achieve desired behavior, the expert can demonstrate desired behavior and the agent can tune itself to match the demonstration."

[Quote from Tom Mitchell.]

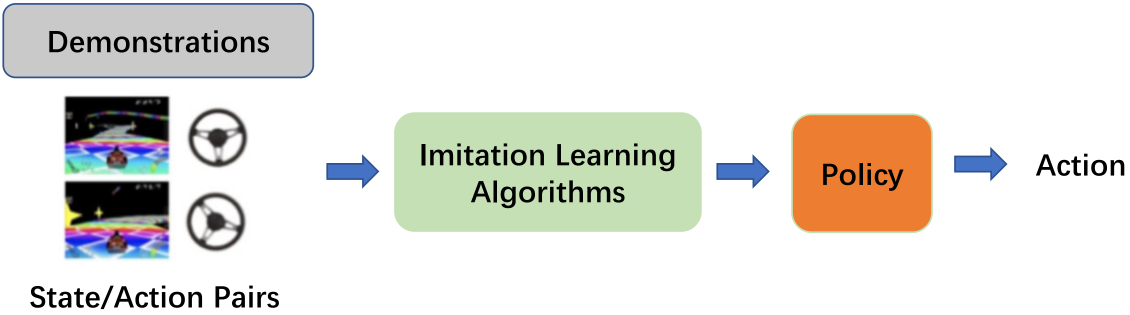
▶ **Can we transfer the knowledge from demonstrations $\mathcal{D}$ to learn a policy or reward?**

$$\mathcal{D} = \{\tau_1, \cdots, \tau_m\}$$

▶ **One-life demonstration means $\mathcal{D} = \{\tau_1\}$.**

Imitation Learning (IL)

Imitation Learning (a.k.a. learning from demonstrations):
Given: demonstrations, i.e., a set of state-action pairs played by an expert.
Goal: train a policy to mimic demonstrations without manual rewards.

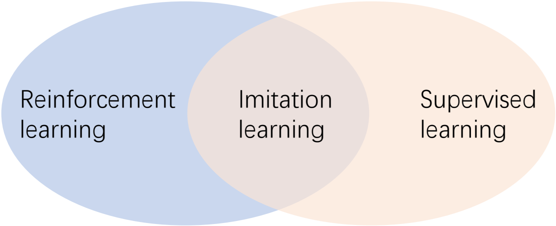


Level of Supervised Information

Comparison of goal specification:

- Reinforcement Learning
  (Weak: no specific goals, but intermediate rewards)
- **Imitation Learning**
  (Stronger: no explicit goals and rewards, but some examples how to reach them)
- Supervised Learning
  (Full: explicit goals even for intermediate steps)

Imitation learning is a fusion of reinforcement learning and supervised learning:



Representative Imitation Learning methods

- **Supervised learning**
  - Behavioral Cloning (BC), totally fails in high-dimensional environments, e.g., Atari games.
- **Supervised learning with iterative feedback actions**
  - Direct Policy Learning (DPL) via Interactive Demonstrators
  - Data Aggregation (DAgger)
- **Inverse reinforcement learning (IRL)**
  (Seeks a reward function that justifies the demonstration.)
  - Generative Adversarial Imitation Learning (GAIL)
  - Variational Adversarial Imitation Learning (VAIL)

Behavior Cloning (BC)

- BC = Supervised Learning of (s, a^*)
- Learning objective:

$$\arg\min_\theta E_{(s,a^*)\sim P^*}[L(a^*, \pi_\theta(s))]$$

  - Optimal action a^* is not available
- Given demonstrations $\mathcal{D} = \{\tau_1, \cdots, \tau_m\} = \{(s_i, a_i)\}$

$$\arg\min_\theta E_{(s_i,a_i)\sim\mathcal{D}}[L(a_i, \pi_\theta(s_i))]$$

  - Action a_i is not perfect
  - Wrongly predicted actions lead to unseen states $s \notin \mathcal{D}$
  - The learned policy cannot handle unseen states (a.k.a.catastrophic failures).

Direct Policy Learning (DPL) via Interactive Demonstrators

- DPL = Supervised Learning with interactive feedback actions
  - Fix $\mathcal{D}$, estimate π.
$$\arg\min_{\theta} E_{(s_i,a_i)\sim\mathcal{D}}[L(a_i,\pi_\theta(s_i))]$$
  - Fix π, run π to roll out $\mathcal{D}_\phi = \{s_0, s_1 \cdots\}$
  - Seek expert to label actions
  - $\mathcal{D} = \mathcal{D}_\phi$
  - Repeat
- Alternating optimization $\Rightarrow$ Unstable learning !

Data Aggregation (DAgger)

- DAgger = Supervised Learning with aggregation of interactive feedback actions
  - Fix $\mathcal{D}$, estimate π.

$$\arg\min_{\theta} E_{(s_i,a_i)\sim\mathcal{D}}[L(a_i,\pi_\theta(s_i))]$$

  - Fix π, run π to roll out $\mathcal{D}_\pi = \{s_0, s_1, \cdots\}$
  - Seek expert/trajectory optimization to label actions
  - Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$
  - Repeat
- Memorize all demonstrations, but state-action pairs are still limited.

Inverse Reinforcement Learning (IRL)

- MDP: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$
- Given $\mathcal{D} = \{\tau_1, \cdots, \tau_m\} = \{(s_0^i, a_0^i, s_1^i, a_1^i, \cdots)\} \sim \pi_E$
- **Goal:** Learn a reward function r^* so that

$$\pi_E = \arg\max_\pi E_\pi[r^*(s,a)] \text{ or } \arg\max_\pi E_\pi[r^*(s)]$$

  - Learn reward function r
  - Learn policy π given the learned reward function r
  - Compare the learned policy π with the expert policy π_E
  - Repeat
- Model-based IRL methods require given dynamics

Generative Adversarial Imitation Learning (GAIL)

- GAIL = GAN on (s_t, a_t).
- Turn IRL into a minimax problem with a uniform regularizer $H(\pi)$ on the learned policy

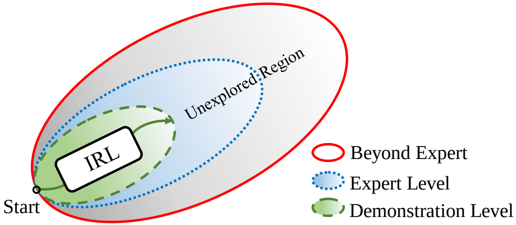
$$\min_{\pi} \max_{D} E_{\pi}[\log(D(s,a))] + E_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

- VAIL = GAIL + Information Bottleneck regularizer

- Model-free: Both GAIL and VAIL do not model dynamics $p(s_{t+1}|s_t, a_t)$.
- Learn the distribution of (s_t, a_t) by discriminator.
- Still assume $s_t \rightarrow a_t$ is reliable in demonstrations.
- What happen if the demonstrations are not perfect or even noisy ?

Grand Challenge of IL methods

Existing IL methods are restricted to the basic demonstration-level performance in imitation learning from a one-life demonstration.



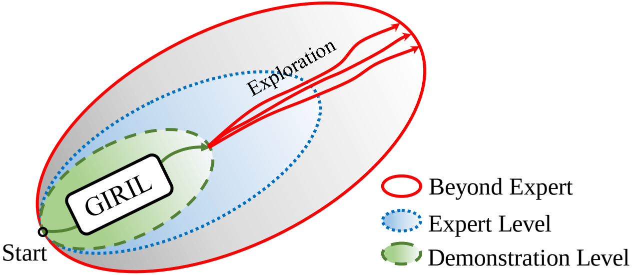
In the high-dimensional environments, e.g., Atari games
Most IL methods fail to perform as good as demonstration, even with many demonstrations.

Research Question:
"Can we develop an imitation learning method that can outperform the expert from limited demonstrations in a high-dimensional environment?"

Generative Intrinsic Reward driven Imitation Learning (GIRIL)

Main idea:

- We propose *Generative Intrinsic Reward driven Imitation Learning* (GIRIL), which seeks a family of *Intrinsic Reward* functions that enables the agent to do Sampling-based Self-supervised Exploration in the environment. This is critical to achieve better-than-expert performance[1].



[1]Here, the Demonstration-level performance is referred to the performance by a expert player until losing the first life in a game, known as one-life demonstration; while the Expert-level performance means the one after the expert player losing all available lives in a game. It is also known as one full-episode demonstration.

How to create Intrinsic Rewards that the agent outperforms the Expert?

- Hand-engineered extrinsic rewards are infeasible in complex environments.
  - Self-supervised Intrinsic curiosity reward (Pathak, et al., ICML 2017)
    $\Rightarrow$ explores actions that reduce the uncertainty in predicting the consequence of the states
    e.g. $\|\hat{s}_{t+1}(a_t, s_t) - s_{t+1}\|_2^2$.

How to create Intrinsic Rewards that the agent outperforms the Expert?

- Hand-engineered extrinsic rewards are infeasible in complex environments.
  - Self-supervised Intrinsic curiosity reward (Pathak, et al., ICML 2017)
    $\Rightarrow$ explores actions that reduce the uncertainty in predicting the consequence of the states
    e.g. $\|\hat{s}_{t+1}(a_t, s_t) - s_{t+1}\|_2^2$.
- Very limited states and actions in the trajectory of one-life demonstration.
  - Generate more states and actions than that of the Expert-level performance from the distribution of state and action dynamics of an agent $\Rightarrow$ Sampling-based Exploration.

How to create Intrinsic Rewards that the agent outperforms the Expert?

- Hand-engineered extrinsic rewards are infeasible in complex environments.
  - Self-supervised Intrinsic curiosity reward (Pathak, et al., ICML 2017)
    $\Rightarrow$ explores actions that reduce the uncertainty in predicting the consequence of the states
    e.g. $\|\hat{s}_{t+1}(a_t, s_t) - s_{t+1}\|_2^2$.
- Very limited states and actions in the trajectory of one-life demonstration.
  - Generate more states and actions than that of the Expert-level performance from the distribution of state and action dynamics of an agent $\Rightarrow$ Sampling-based Exploration.
- How to reliably learn the agent's state and action dynamics from limited demonstrations?
  - Infer the optimal action $\hat{a}_t$ from the transition of observed state pair s_t and s_{t+1}.
  - Generate the high-fidelity next state $\hat{s}_{t+1}$ from the current state s_t and action a_t in a virtuous cycle. (**Cycle Check WHAT has been learned in MDP!**)
- More reliable Intrinsic curiosity $\Rightarrow$ Better performance

Generative Intrinsic Reward Learning (GIRL)

Forward and Backward Dynamics:

- A *decoder* $p_\theta(s_{t+1}|z, s_t)$ for modeling the forward dynamics (state transition),
- and an *encoder* $q_\phi(z|s_t, s_{t+1})$ for modeling the backward dynamics (action encoding).

Variational solution by maximizing:

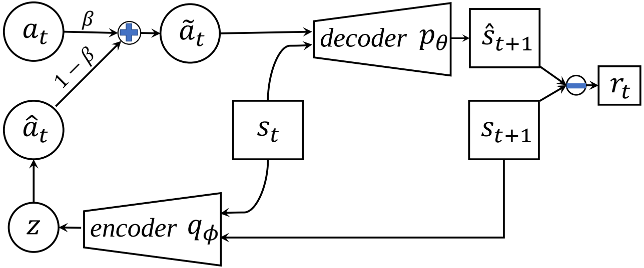
$$\begin{aligned}\mathcal{L}(s_t, s_{t+1}; \theta, \phi) = \mathbb{E}_{q_\phi(z|s_t,s_{t+1})}[\log p_\theta(s_{t+1}|z, s_t)] - \mathrm{KL}(q_\phi(z|s_t, s_{t+1})\|p_\theta(z|s_t)) \\ - \alpha \mathrm{KL}(q_\phi(\hat{a}_t|s_t, s_{t+1})\|\pi_E(a_t|s_t))]\end{aligned} \tag{1}$$

where z is the latent variable, $\pi_E(a_t|s_t)$ is the expert policy distribution, $\hat{a}_t = \mathrm{Softmax}(z)$ is the transformed latent variable, α is a positive scaling weight.

- The 1st part of (1), a Conditional VAE, models the forward and backward dynamics.
- The forward dynamics is not precise since we use limited demonstrations.
- The 2nd part of (1), the KL term, can guide the action encoding of backward dynamics.

GIRL Model and Reward

The reward inference procedure of our reward module:



Reward calculation:

$$r_t = \lambda \|\hat{s}_{t+1} - s_{t+1}\|_2^2 \quad (2)$$

where $\hat{s}_{t+1} = decoder(\beta * a_t + (1-\beta) * \mathrm{Softmax}(z), s_t)$, $\|\cdot\|_2$ denotes the L2 norm, and λ is a positive scaling weight.

GIRIL Algorithm

Algorithm 1 Generative Intrinsic Reward driven Imitation Learning (GIRIL)

Input: Expert demonstration data $\mathcal{D} = \{(s_i, a_i, s_{i+1})\}_{i=1}^{N}$.
Initialize policy π, *encoder* q_ϕ and *decoder* p_θ.
// GIRL
for $e = 1, \cdots, E$ **do**
    Sample a batch of demonstration $\tilde{\mathcal{D}} \sim \mathcal{D}$.
    Train q_Φ and p_θ to maximize the objective (1) on $\tilde{\mathcal{D}}$.
end for
// Policy Optimization
for $i = 1, \cdots, \mathrm{MAXITER}$ **do**
    Update policy via any policy gradient method, e.g. PPO on the intrinsic reward inferred by Eq. (2).
end for
Output: Policy π.

Experiments and Results

- **Atari Games**
  - *Character:* high-dimensional state space and discrete action space;
  - *Data:* a one-life demonstration with a short length for each game:

Game	Demonstration Length		# Lives available
	One-life	Full-episode	
Space Invaders	697	750	3
Beam Rider	1,875	4,587	3
Breakout	1,577	2,301	5
Q*bert	787	1,881	4
Seaquest	562	2,252	4
Kung Fu Master	1,167	3,421	4

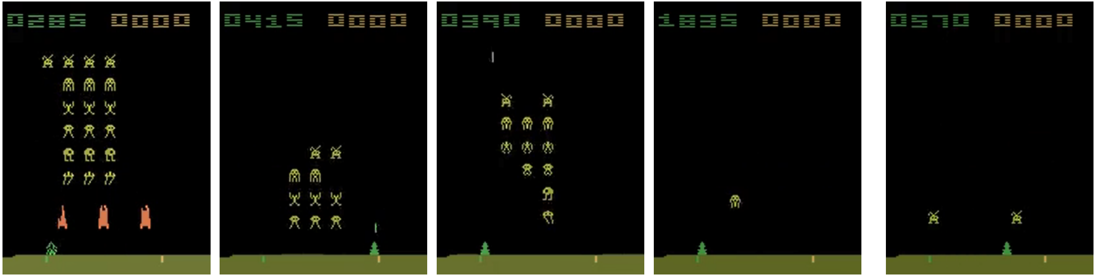
- **Continuous Control Tasks**
  - *Character:* low-dimensional state space and continuous action space;
  - *Data:* one demonstration with a fixed length of 1,000 for each task.

Baselines

- One random agent
- One supervised learning method:
  - Behavioral Cloning (BC)
- Two state-of-the-art inverse reinforcement learning methods:
  - Generative Adversarial Imitation Learning (GAIL)
  - Variational Adversarial Imitation Learning (VAIL)
- One state-of-the-art reward learning module used in exploration task:
  - Curiosity-driven Imitation Learning (CDIL)

A glance of imitation performance on the Space Invaders game:

Our method GIRIL achieves a score (1,835) that is significantly better than the expert (570).



(a) GAIL (b) VAIL (c) CDIL (d) **GIRIL (ours)** (e) Expert

GAIL: Generative Adversarial Imitation Learning.
VAIL: Variational Adversarial Imitation Learning.
CDIL: Curiosity-driven Imitation Learning, which leverages a state-of-the-art exploration method for reward learning.
Expert: the expert demonstrator.

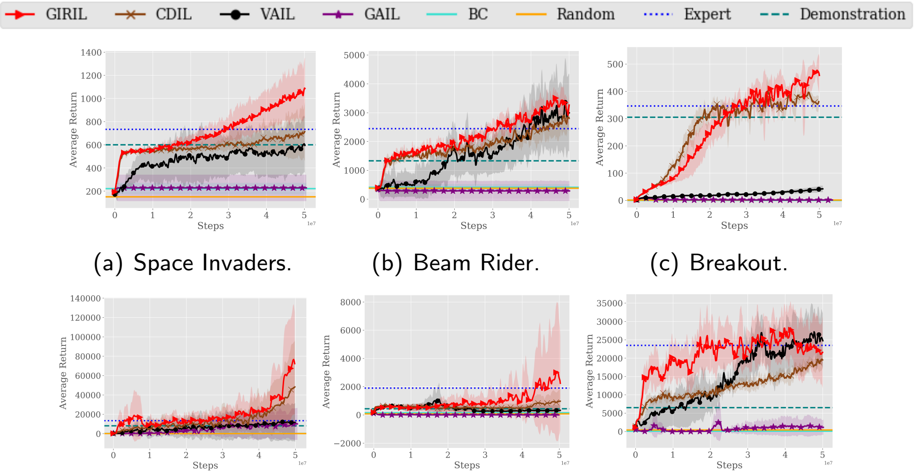
Atari Games

Quantitative Results (better-than-expert performance in bold):

	Expert	Demonstration	Imitation Learning Algorithms					Random
Game	Average	Average	GIRIL (ours)	CDIL	VAIL	GAIL	BC	Average
Space Invaders	734.1	600.0	**992.9**	668.9	549.4	228.0	186.2	151.7
Beam Rider	2,447.7	1,332.0	**3,202.3**	**2,556.9**	**2,864.1**	285.5	474.7	379.4
Breakout	346.4	305.0	**426.9**	**369.2**	36.1	1.3	0.9	1.3
Q*bert	13,441.5	8,150.0	**42,705.7**	**30,070.8**	10,862.3	8,737.4	298.4	159.7
Seaquest	1,898.8	440.0	**2,022.4**	897.7	312.9	0.0	155.2	75.5
Kung Fu Master	23,488.5	6,500.0	**23,543.6**	17,291.6	**24,615.9**	1,324.5	44.9	413.7

Our method outperforms several baselines including a state-of-the-art curiosity-based reward learning method (CDIL), two state-of-the-art IRL methods (GAIL & VAIL), and behavioral cloning (BC).

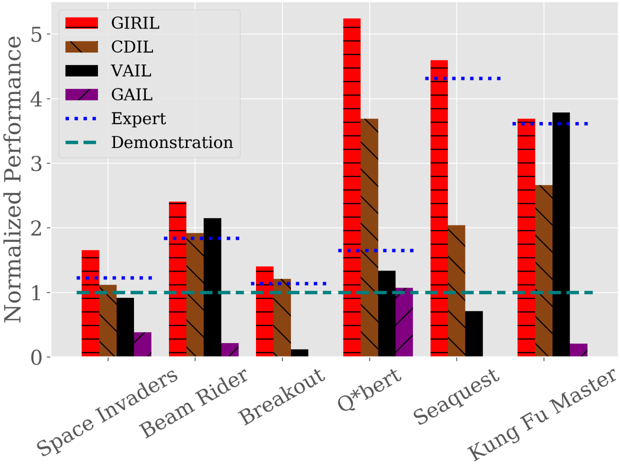
Average return vs. number of simulation steps on Atari games ($\beta = 1.0$).



(a) Space Invaders. (b) Beam Rider. (c) Breakout.

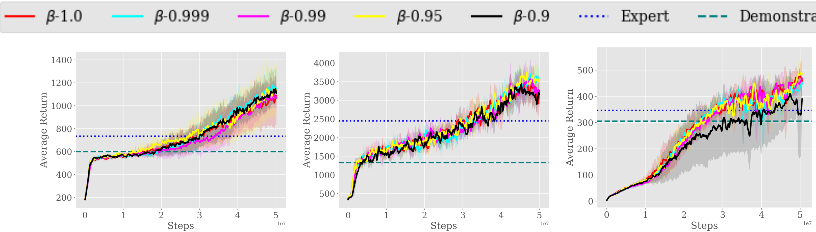
(d) Q*bert. (e) Seaquest. (f) Kung Fu Master.

Imitation learning performance improvements of our GIRIL:

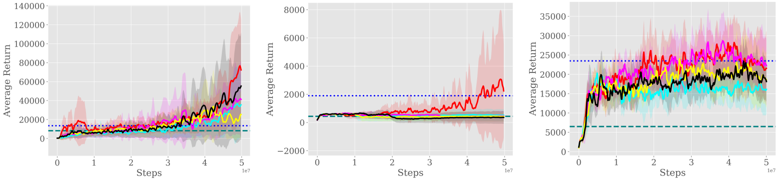


Atari Games

Parameter Analysis of our GIRIL with different β on Atari games.

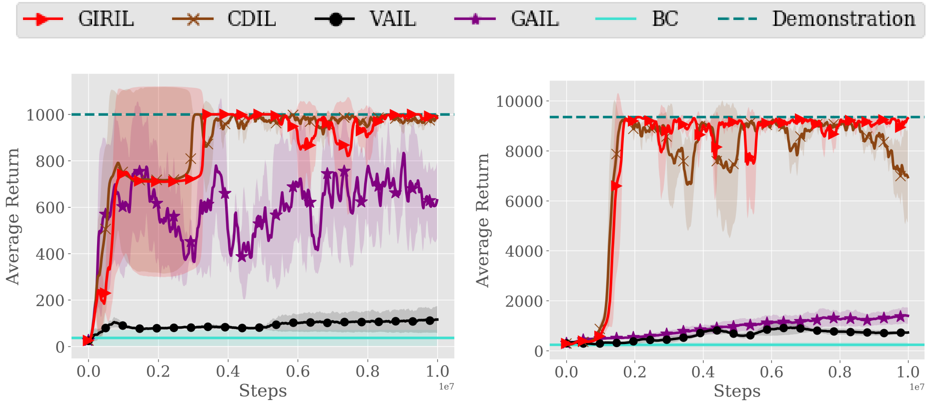


(a) Space Invaders. (b) Beam Rider. (c) Breakout.



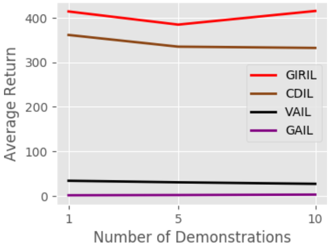
(d) Q*bert. (e) Seaquest. (f) Kung Fu Master.

Average return vs. number of simulation steps on continuous control tasks.

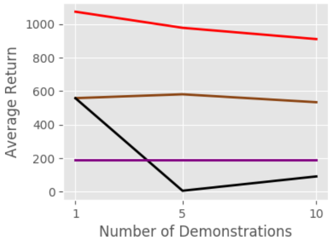


(a) InvertedPendulum. (b) InvertedDoublePendulum.

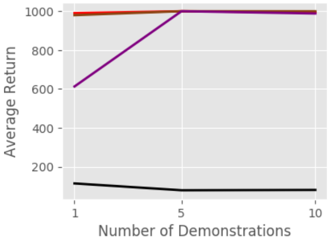
Comparison with Full-episode Demonstrations



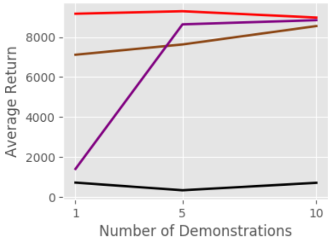
(a) Breakout.



(b) Space Invaders.



(c) InvertedPendulum.



(d) InvertedDoublePendulum.

Conclusion and Future Direction

Conclusion:

- We have proposed a novel reward learning module that combines an backward dynamics model and a forward dynamics model into one generative solution.
- It performs better forward state transition and backward action encoding, and therefore improves the dynamics modeling of MDP.
- Our GIRL generates a family of intrinsic rewards, enabling the agent to do sampling-based self-supervised exploration in the environment. (Key for better-than-expert performance.)
- Our GIRIL consistently outperforms the expert with only one incomplete demonstration in the high-dimensional Atari domain.

Future Direction

- An interesting topic for future investigation would be to apply our reward learning module to a hard exploration task.


Thank you!