COMPETITIVE MULTI-AGENT INVERSE RL

with Sub-optimal Demonstrations

Xingyu Wang, Diego Klabjan
Department of Industrial Engineering and
Management Sciences, Northwestern University

Northwestern | McCORMICK SCHOOL OF ENGINEERING

Motivation

- Learning from expert demonstrations in competitive games
  - Reward signal is not available
  - Imitation learning: recover the policy function
  - Inverse reinforcement learning (IRL)
    - Infer the reward function
- Sub-optimality in expert demonstrations
  - Common assumption relies on the optimality assumption.
  - Agents are decoupled
- Beyond the relatively small problems
  - A learning-based algorithm for IRL tasks in large-scale competitive games

Competitive Markov Decision Processes

- Zero-sum stochastic games
  $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}^f, \mathcal{A}^g, R, P, \gamma \rangle$
  - Agents: $\mathcal{N} = \{f, g\}$
  - States: a finite set of size $N_{\mathcal{S}} = |\mathcal{S}|$
    - Terminal states do not necessarily exist.
  - Actions: $\mathcal{A}^i = \times_{s \in \mathcal{S}} \mathcal{A}^i(s), i = f, g$
  - Reward: $R : \mathcal{S} \to \mathbb{R}$
  - State transition function: $P(s'|s, a^f, a^g)$
    $P : \mathcal{S} \times \mathcal{A}^f \times \mathcal{A}^g \times \mathcal{S} \to \mathbb{R}$
  - Discount Factor: $\gamma \in [0, 1)$
- Policies
  $\mathcal{F} = \times_{s \in \mathcal{S}} \Delta(\mathcal{A}^f(s)), \quad \mathcal{G} = \times_{s \in \mathcal{S}} \Delta(\mathcal{A}^g(s))$
  - $\triangle$(A): collection of all distributions on the non-empty set A

- State-value function (for agent f)

$$v^{f,g}(s_0; R) = \mathop{\mathbb{E}}_{\substack{a_t^f \sim f(a^f|s_t) \\ a_t^g \sim g(a^g|s_t) \\ s_{t+1} \sim P(s'|s_t, a_t^f, a_t^g)}} \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- Nash Equilibrium (minimax policy)
  - For any (f, g) and any s,

$$v^{f^*(R),g}(s; R) \geq v^{f^*(R),g^*(R)}(s; R) \geq v^{f,g^*(R)}(s; R)$$

  - Value of the game
    $\boldsymbol{v}^* = (v^*(s))_{s \in \mathcal{S}}$
    $v^*(s) = \max_f \min_g v^{f,g}(s; R)$
    - In zero-sum stochastic games, vector v* is the same under any Nash Eq.

IRL with Sub-optimal Demonstrations

- Inverse reinforcement learning
  - Model of the environment: i.e. state-transition-function P
  - Expert demonstrations $\mathcal{D} = \{(s_i, a_i^{E,f}, a_i^{E,g}) \mid i = 1, 2, 3, ..., N_{\mathcal{D}}\}$
    - Reward signal is not available
- Prior knowledge of the game
  - Multiple solutions exist.
  - Avoid trivial solution ($R \equiv 0$).

Adversarial Training

- Focus on one agent – similar for the other agent
- Evaluating current f with its best possible opponent g

$$F(f) = \min_{g \in \mathcal{G}} \frac{1}{N_s} \sum_{s \in \mathcal{S}} v^{f,g}(s).$$

- Find f that maximizes $F(f)$.
  - The objective function for finding minimax policy has no local optima in zero-sum stochastic games.
  - Function $F(f)$ has no local maxima
    - Very technical and long proof

Adversarial Training Algorithm

Algorithm 2 Adversarial Training Algorithm for Solving $f^*(R)$ in Zero-Sum Games (Sketch)

Require: Positive integers K_g, K_{cycle}.
Initialize: Parameters θ_f, θ_g for policy models
for $i = 1, 2, 3, \ldots.$ **do**
  if $i \,\%\, K_{\text{cycle}} \leq K_g$ **then**
    Update θ_g to optimize return for g based on PPO.
  else
    Update θ_f to optimize return for f based on PPO.

- Adversarial training algorithm
  - Maintains a best opponent model
  - Update g frequently and f occasionally
  - Policy gradient method
    - Actor-critic proximal policy optimization Actor-critic style training
      - Model outputs policy and estimated state-value function.

IRL Model

- Performance margins between experts demonstrations and Nash Eq. are reasonably tight

$$\mathbb{E}_s\left[v^{f^*(R),g^E|_{\mathcal{D}}}(s;R) - v^{f^*(R),g^*(R)}(s;R)\right],$$

$$\mathbb{E}_s\left[v^{f^*(R),g^*(R)}(s;R) - v^{f^E|_{\mathcal{D}},g^*(R)}(s;R)\right]$$

$$f^E|_{\mathcal{D}}(a|s) = \frac{\#\{(s,a,\_)\}}{\#\{(s,\_,\_)\}} \qquad g^E|_{\mathcal{D}}(a|s) = \frac{\#\{(s,\_,a)\}}{\#\{(s,\_,\_)\}}$$

Unknown outside D

- Objective function

$$\min_R \min_{f^E|_{\mathcal{D}},g^E|_{\mathcal{D}}} \mathbb{E}_s\left[v^{f^*(R),g^E|_{\mathcal{D}}}(s;R) - v^{f^E|_{\mathcal{D}},g^*(R)}(s;R)\right] \tag{5}$$

$$\text{where } f^*(R) \in \operatorname*{argmax}_{f\in\mathcal{F}} \min_{g\in\mathcal{G}} \frac{1}{N_s}\sum_{s\in S} v^{f,g}(s;R), \tag{6}$$

$$\text{and } g^*(R) \in \operatorname*{argmin}_{g\in\mathcal{G}} \max_{f\in\mathcal{F}} \frac{1}{N_s}\sum_{s\in S} v^{f,g}(s;R). \tag{7}$$

- Sample trajectories under unknown expert policies
- The minimization operator on $f^E|_{\mathcal{D}}, g^E|_{\mathcal{D}}$ encourages them to stay close to f^*, g^*
- So f^E, g^E act only at the first step
  - Equivalent to bounding the advantage of expert actions

$$(s_0^{E,g^*(R)}, a_0^f, \_) \sim \mathcal{D},$$
$$a_t^g \sim g^*(R)(a|s_t^{E,g^*(R)}) \text{ for } 0 \le t \le T,$$
$$a_t^f \sim f^*(R)(a|s_t^{E,g^*(R)}) \text{ for } 1 \le t \le T,$$
$$\text{and } s_t^{E,g^*(R)} \sim P(s'|s_{t-1}^{E,g^*(R)}, a_{t-1}^f, a_{t-1}^g) \text{ for } 1 \le t \le T.$$

Algorithm 1 Inverse Reinforcement Learning in Zero-Sum Discounted Stochastic Games

Require: Observed experts demonstrations $\mathcal{D} = \{(s_i, a_i^f, a_i^g) \mid i = 1, 2, \ldots, N_{\mathcal{D}}\}$; Positive integers K_R, I_R; Nash Equilibrium threshold τ; learning rate λ.

Initialize: Parameters θ_R for the reward function, θ_f, θ_g to parametrize $f_{\theta_f}(a|s), g_{\theta_g}(a|s)$ for Nash Equilibrium policies

for i $= 1, 2, 3, \ldots$ **do**

  Update θ_f to find Nash Equilibrium policy for f under current R_{θ_R}, return also $\hat{v}^{f,g^{\text{best}}}$

  Update θ_g to find Nash Equilibrium policy for g under current R_{θ_R}, return also $\hat{v}^{f^{\text{best}},g}$

  if $i \,\%\, K_R = 0$ and $\hat{v}^{f^{\text{best}},g} - \hat{v}^{f,g^{\text{best}}} < \tau$ **then**

    for j $= 1, 2, 3, \ldots, I_R$ **do**

      Sample one observation from $\mathcal{D}$: $(s, a^{E,f}, a^{E,g})$

      Use (8) to get $\{s_1^{f^*(R_{\theta_R}),E}, s_2^{f^*(R_{\theta_R}),E}, \ldots, s_T^{f^*(R_{\theta_R}),E}\}, \{s_1^{E,g^*(R_{\theta_R})}, s_2^{E,g^*(R_{\theta_R})}, \ldots, s_T^{E,g^*(R_{\theta_R})}\}$

      $\hat{v}^f(\bar{\theta_R}) \leftarrow R_{\bar{\theta_R}}(s) + \sum_{t=1}^{T} \gamma^{t-1} R_{\bar{\theta_R}}\left(s_t^{f^*(R_{\theta_R}),E}\right)$

      $\hat{v}^g(\bar{\theta_R}) \leftarrow R_{\bar{\theta_R}}(s) + \sum_{t=1}^{T} \gamma^{t-1} R_{\bar{\theta_R}}\left(s_t^{E,g^*(R_{\theta_R})}\right)$

      $\theta_R \leftarrow \theta_R - \lambda \nabla_{\bar{\theta_R}} \left(\hat{v}^f(\bar{\theta_R}) - \hat{v}^g(\bar{\theta_R}) + \phi(\bar{\theta_R})\right)|_{\bar{\theta_R}=\theta_R}$

π step

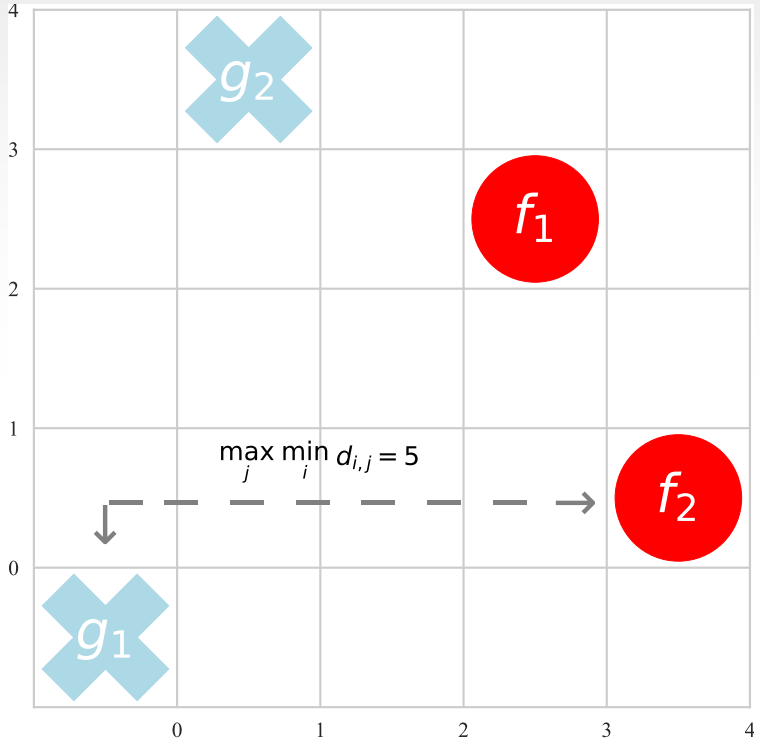
When (f^*, g^*) are accurate enough by threshold τ, move onto the R step.

Estimate "expert vs Nash Eq." performance with sampled trajectories.

Minimize performance gap

Regularization term

Experimental Study: The Chasing Game



- 2 vs 2 chasing game on a 5x5 grid
- Predator(f) and prey(g), stay or move vertically/horizontally
- Pair-wise L1 distance $d_{i,j} = |x_{f_i} - x_{g_j}| + |y_{f_i} - y_{g_j}|$
- Reward function $R_{\text{chasing}}(s) = -D(s),$
  $D(s) = \max_{j=1,2} \min_{i=1,2} d_{i,j},$
  - Encourage cooperation and accurate allocation of tasks.
  - 390,625 states, $2 * 10^7$ state-action pairs.

- **Generate sub-optimal demonstration**
  - Solve Nash Eq. in the chasing game
  - With chance ε, deflect action of f^*, g^* by 90 degrees
  - Test ε = 5%, 10%, 20%
    - When ε = 20%, the chance that none of the 4 player would take an deflected action is $(1 - 0.2)^4$ = 0.4096

Experimental Study: Nash Equilibrium

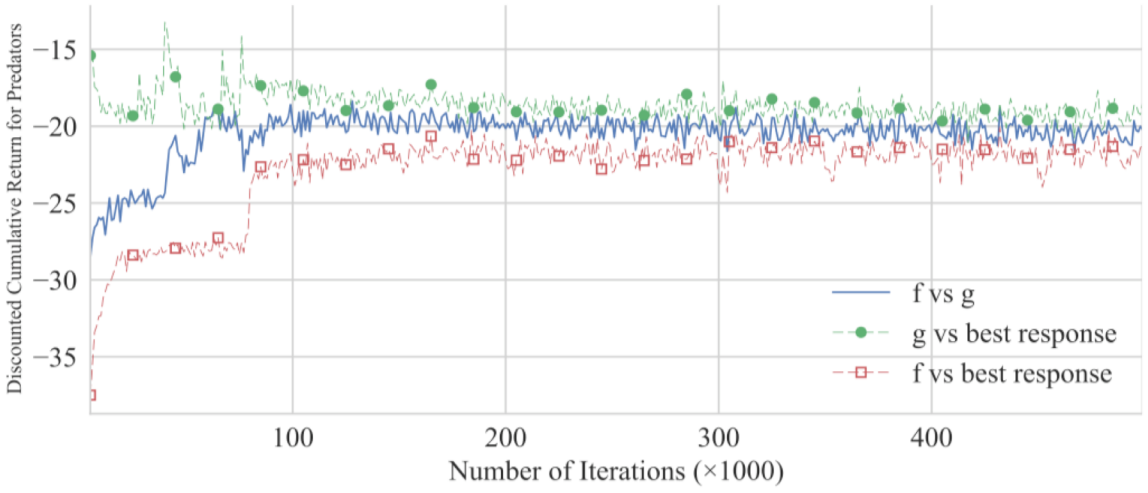


Figure 6: Performance of the proposed Nash Equilibrium algorithm in the chasing game

Table 3. Performances of solved Nash Equilibrium policies (A:Algorithm 2; B:Benchmark; R:Random)

Grid Size	$f^{\mathrm{A}}, g^{\mathrm{A}}$	$f^{\mathrm{B}}, g^{\mathrm{A}}$	$f^{\mathrm{R}}, g^{\mathrm{A}}$	$f^{\mathrm{A}}, g^{\mathrm{B}}$	$f^{\mathrm{A}}, g^{\mathrm{R}}$
5×5	-20.3	-21.2	-41.1	-20.0	-14.9
10×10	-44.7	-87.4	-94.2	-42.2	-31.8

- **Benchmark**
  - Formulate the quadratic programming problem in Filar & Vrieze (2012) as a learning-based algorithm

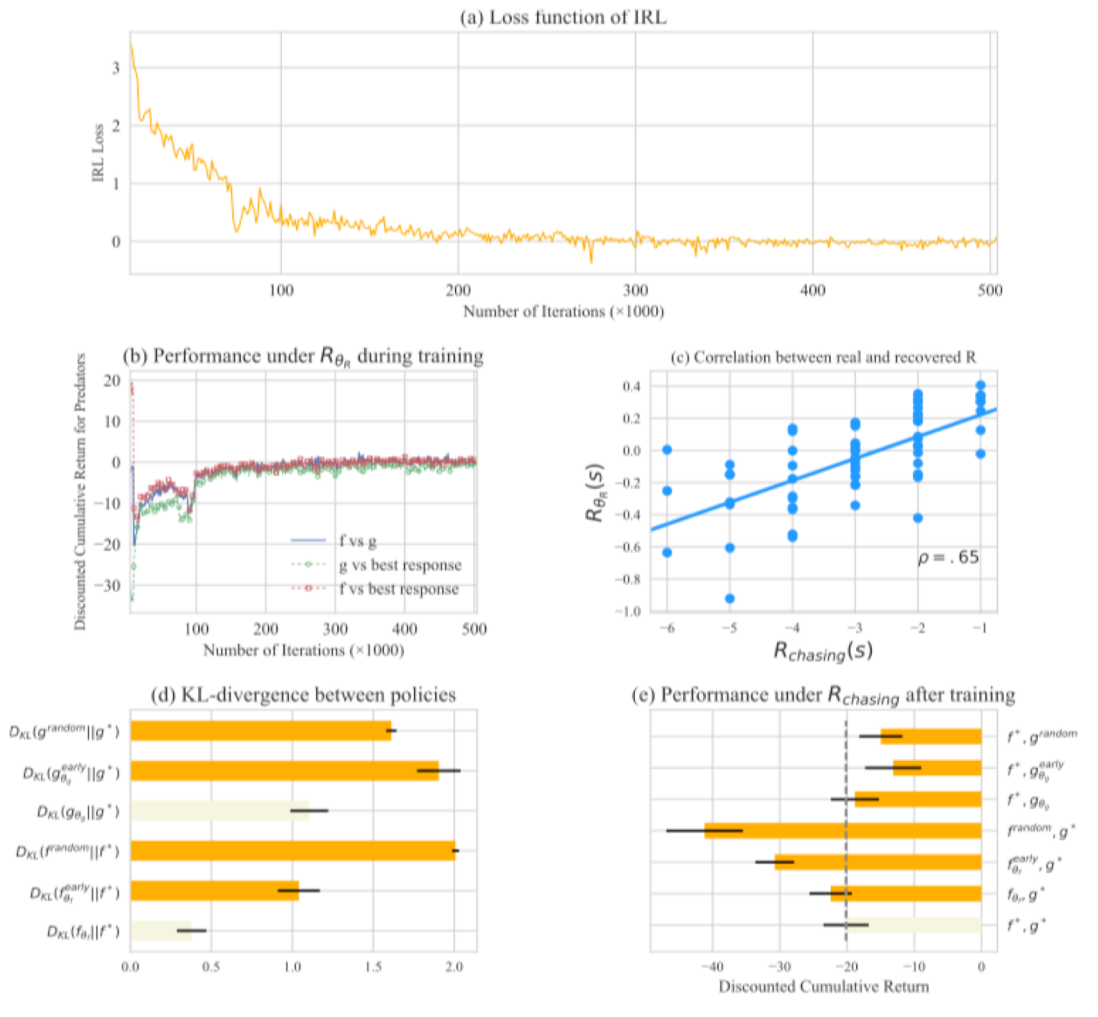
$$\text{minimize } \mathbb{E}_s[v^f(s) + v^g(s)],$$

$$R(s) + \gamma \mathbb{E}_{a^g \sim g(a|s), s' \sim p(s'|s,a^f,a^g)} v^f(s') \leq v^f(s) \text{ for any } s, a^f,$$

$$-R(s) + \gamma \mathbb{E}_{a^f \sim f(a|s), s' \sim p(s'|s,a^f,a^g)} v^g(s') \leq v^g(s) \text{ for any } s, a^g,$$

  - Constraints are implemented as Lagrangians with fixed λ.

Experimental Study: IRL



- **Regularization term**
  - Know that reward is related to distances between players
  - Assume that average distance is correlated with R

$$\bar{D}(s) = \frac{1}{4}\sum_i \sum_j d_{i,j}.$$

- **Evaluation Metrics**
  - Recovered reward function
    - Correlation between recovered R and the real R (minimax distance)
  - Recovered policies
    - Solve f^*/g^* under recovered reward function
    - Gauge KL-divergence between recovered and optimal policies
  - Performance of policies
    - Pit recovered policies against the solved Nash Eq. in the chasing game

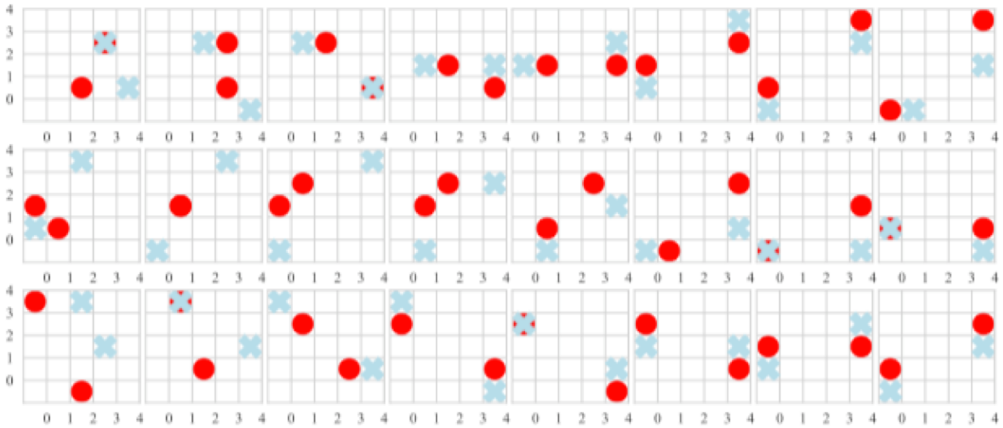


Figure 3: Trajectories generated by policy models obtained in the IRL algorithm. Each row presents a different 8-step trajectory.

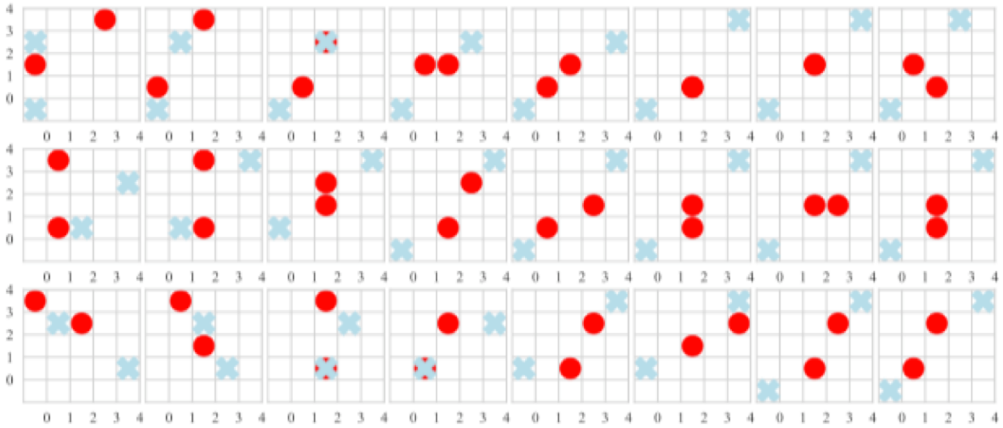


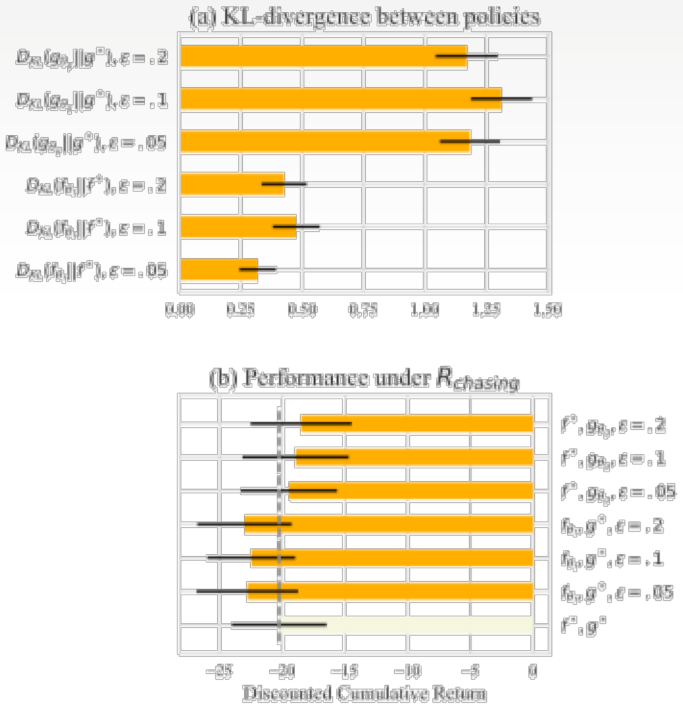
Figure 4: Trajectories generated by policy models obtained at the 20,000-th iteration. Each row presents a different 8-step trajectory. Clearly, these actions are not driven by $R_{\text{chasing}}(s) = D(s)$ and are different from the ones in Fig. 3.

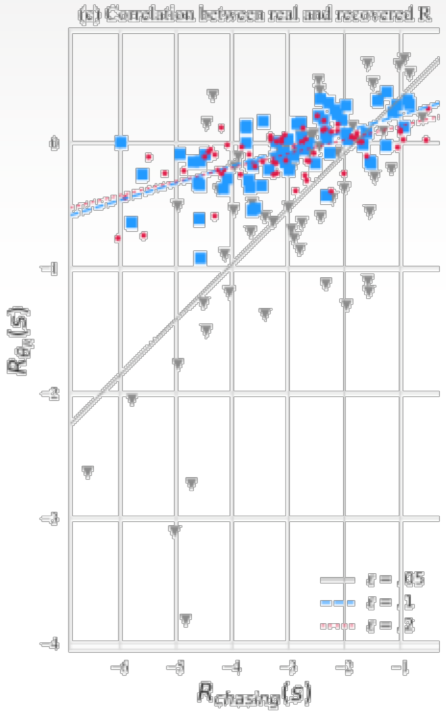
- Demonstration of recovered policies after IRL training.

- Under the regularization term
  - Recovered policies trying to maximize/minimize the averaged distance.
- Predators (f)
  - Stay at the center of the grid
- Preys (g):
  - Stay on diagonal

IRL with Different Sub-optimal Demonstrations

- Similar performances under different demonstration sets





- Benchmark comparison

Table 1. Correlations between recovered $R(s)$ and $R_{\text{chasing}}(s)$

IRL Algorithm	$\epsilon = .05$	$\epsilon = .1$	$\epsilon = .2$
Algorithm 1	**0.65**	**0.68**	**0.66**
BIRL	0.28	-0.02	0.12
DIRL	-0.31	-0.15	0.11

Table 2. Performance deterioration of recovered policies under R_{chasing} (A:Algorithm 1; B:BIRL; D:DIRL)

$\mathcal{D}_\epsilon$	f^A	f^B	f^D	g^A	g^B	g^D
.05	**11.8%**	24.1%	197.0%	**4.4%**	33.5%	38.9%
.1	**10.3%**	44.3%	100.0%	**6.9%**	33.5%	41.9%
.2	**13.3%**	68.5%	200.1%	**9.3%**	33.0%	40.9%