Multi-Agent Adversarial Inverse Reinforcement Learning

Lantao Yu [1] **Jiaming Song** [1] **Stefano Ermon** [1]

Abstract

Reinforcement learning agents are prone to undesired behaviors due to reward mis-specification. Finding a set of reward functions to properly guide agent behaviors is particularly challenging in multi-agent scenarios. Inverse reinforcement learning provides a framework to automatically acquire suitable reward functions from expert demonstrations. Its extension to multi-agent settings, however, is difficult due to the more complex notions of rational behaviors. In this paper, we propose MA-AIRL, a new framework for multi-agent inverse reinforcement learning, which is effective and scalable for Markov games with high-dimensional state-action space and unknown dynamics. We derive our algorithm based on a new solution concept and maximum pseudolikelihood estimation within an adversarial reward learning framework. In the experiments, we demonstrate that MA-AIRL can recover reward functions that are highly correlated with ground truth ones, and significantly outperforms prior methods in terms of policy imitation.

1. Introduction

Reinforcement learning (RL) is a general and powerful framework for decision making under uncertainty. Recent advances in deep learning have enabled a variety of RL applications such as games (Silver et al., 2016; Mnih et al., 2015), robotics (Gu et al., 2017; Levine et al., 2016), automated machine learning (Zoph & Le, 2016) and generative modeling (Yu et al., 2017). RL algorithms are also showing promise in multi-agent systems, where multiple agents interact with each other, such as multi-player games (Peng et al., 2017), social interactions (Leibo et al., 2017) and multi-robot control systems (Matignon et al., 2012). However, the success of RL crucially depends on careful reward design (Amodei et al., 2016). As reinforcement learning agents are prone to undesired behaviors due to reward misspecification (Amodei & Clark, 2016), designing suitable reward functions can be challenging in many real-world applications (Hadfield-Menell et al., 2017). In multi-agents systems, since different agents may have completely different goals and state-action representations, hand-tuning reward functions becomes increasingly more challenging as we take more agents into consideration.

Imitation learning presents a direct approach to programming agents with expert demonstrations, where agents learn to produce behaviors similar to the demonstrations. However, imitation learning algorithms, such as behavior cloning (Pomerleau, 1991) and generative adversarial imitation learning (Ho & Ermon, 2016; Ho et al., 2016), typically sidestep the problem of interring an explicit representation for the underlying reward functions. Because the reward function is often considered as the most succinct, robust and transferable representation of a task (Abbeel & Ng, 2004; Fu et al., 2017), it is important to consider the problem of inferring reward functions from expert demonstrations, which we refer to as inverse reinforcement learning (IRL). IRL can offer many advantages compared to direct policy imitation, such as analyzing and debugging an imitation learning algorithm, inferring agents' intentions and re-optimizing rewards in new environments (Ng et al., 2000).

However, IRL is ill-defined, as many policies can be optimal for a given reward and many reward functions can explain a set of demonstrations. Maximum entropy inverse reinforcement learning (MaxEnt IRL) (Ziebart et al., 2008) provides a general probabilistic framework to solve the ambiguity by finding the trajectory distribution with maximum entropy that matches the reward expectation of the experts. As MaxEnt IRL requires solving an integral over all possible trajectories for computing the partition function, it is only suitable for small scale problems with known dynamics. Adversarial IRL (Finn et al., 2016a; Fu et al., 2017) scales MaxEnt IRL to large and continuous problems by drawing an analogy between a sampling based approximation of MaxEnt IRL and Generative Adversarial Networks (Goodfellow et al., 2014) with a particular discriminator structure. However, the approach is restricted to single-agent settings.

[1]Department of Computer Science, Stanford University, Stanford, CA 94305 USA. Correspondence to: Lantao Yu <lantaoyu@cs.stanford.edu>, Stefano Ermon <ermon@cs.stanford.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

In this paper, we consider the IRL problem in multi-agent environments with high-dimensional continuous state-action space and unknown dynamics. Generalizing MaxEnt IRL and Adversarial IRL to multi-agent systems is challenging. Since each agent's optimal policy depends on other agents' policies, the notion of optimality, central to Markov decision processes, has to be replaced by an appropriate equilibrium solution concept. Nash equilibrium (Hu et al., 1998) is the most popular solution concept for multi-agent RL, where each agent's policy is the best response to others. However, Nash equilibrium is incompatible with MaxEnt RL in the sense that it assumes the agents never take sub-optimal actions. Thus imitation learning and inverse reinforcement learning methods based on Nash equilibrium or correlated equilibrium (Aumann, 1974) might lack the ability to handle irrational (or computationally bounded) experts.

In this paper, inspired by logistic quantal response equilibrium (McKelvey & Palfrey, 1995; 1998) and Gibbs sampling (Hastings, 1970), we propose a new solution concept termed logistic stochastic best response equilibrium (LSBRE), which allows us to characterize the trajectory distribution induced by parameterized reward functions and handle the bounded rationality of expert demonstrations in a principled manner. Specifically, by uncovering the close relationship between LSBRE and MaxEnt RL, and bridging the optimization of joint likelihood and conditional likelihood with *maximum pseudolikelihood estimation*, we propose Multi-Agent Adversarial Inverse Reinforcemnt Learning (MA-AIRL), a novel MaxEnt IRL framework for Markov games. MA-AIRL is effective and scalable to large high-dimensional Markov games with unknown dynamics, which are not amenable to previous methods relying on tabular representation and linear or quadratic programming (Natarajan et al., 2010; Waugh et al., 2013; Lin et al., 2014; 2018). We experimentally demonstrate that MA-AIRL is able to recover reward functions that are highly correlated to the ground truth rewards, while simultaneously learning policies that significantly outperform state-of-the-art multi-agent imitation learning algorithms (Song et al., 2018) in mixed cooperative and competitive tasks (Lowe et al., 2017).

2. Preliminaries

2.1. Markov Games

Markov games (Littman, 1994) are generalizations of Markov decision processes (MDPs) to the case of N interacting agents. A *Markov game* $(\mathcal{S}, \mathcal{A}, P, \eta, \boldsymbol{r})$ is defined via a set of states $\mathcal{S}$, and N sets of actions $\{\mathcal{A}_i\}_{i=1}^N$. The function $P : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathcal{P}(\mathcal{S})$ describes the (stochastic) transition process between states, where $\mathcal{P}(\mathcal{S})$ denotes the set of probability distributions over the set $\mathcal{S}$. Given that we are in state s^t at time t and the agents take actions $(a_1, \ldots, a_N)$, the state transitions to s^{t+1} with probability $P(s^{t+1}|s^t, a_1, \ldots, a_N)$. Each agent i obtains a (bounded) reward given by a function $r_i : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathbb{R}$. The function $\eta \in \mathcal{P}(\mathcal{S})$ specifies the distribution of the initial state. We use bold variables without subscript i to denote the concatenation of all variables for all agents (*e.g.*, $\boldsymbol{\pi}$ denotes the joint policy, $\boldsymbol{r}$ denotes all rewards and $\boldsymbol{a}$ denotes actions of all agents in a multi-agent setting). We use subscript $-i$ to denote *all agents except for* i. For example, $(a_i, \boldsymbol{a}_{-i})$ represents $(a_1, \ldots, a_N)$, the actions of all N agents. The objective of each agent i is to maximize its own expected return (*i.e.*, the expected sum of discounted rewards) $\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=1}^T \gamma^t r_{i,t}\right]$, where γ is the discount factor and $r_{i,t}$ is the reward received t steps into the future. Each agent achieves its own objective by selecting actions through a stochastic policy $\pi_i : \mathcal{S} \to \mathcal{P}(\mathcal{A}_i)$. Depending on the context, the policies can be Markovian (*i.e.*, depend only on the state) or require additional coordination signals. For each agent i, we further define the expected return for a state-action pair as: $\text{ExpRet}_i^{\pi_i, \boldsymbol{\pi}_{-i}}(s_t, \boldsymbol{a}_t) = \mathbb{E}_{s^{t+1:T}, \boldsymbol{a}^{t+1:T}}\left[\sum_{l \geq t} \gamma^{l-t} r_i(s^l, \boldsymbol{a}^l) | s_t, \boldsymbol{a}_t, \boldsymbol{\pi}\right]$

2.2. Solution Concepts for Markov Games

A correlated equilibrium (CE) for a Markov game (Ziebart et al., 2011) is a joint strategy profile, where no agent can achieve higher expected reward through unilaterally changing its own policy. CE first introduced by (Aumann, 1974; 1987) is a more general solution concept than the well-known Nash equilibrium (NE) (Hu et al., 1998), which further requires agents' actions in each state to be independent, *i.e.* $\boldsymbol{\pi}(\boldsymbol{a}|s) = \Pi_{i=1}^N \pi_i(a_i|s)$. It has been shown that many decentralized, adaptive strategies will converge to CE instead of a more restrictive equilibrium such as NE (Gordon et al., 2008; Hart & Mas-Colell, 2000). To take bounded rationality into consideration, (McKelvey & Palfrey, 1995; 1998) further propose logistic quantal response equilibrium (LQRE) as a stochastic generalization to NE and CE.

Definition 1. *A logistic quantal response equilibrium for Markov game corresponds to any strategy profile satisfying a set of constraints, where for each state and action, the constraint is given by:*

$$\pi_i(a_i|s) = \frac{\exp\left(\lambda \text{ExpRet}_i^{\boldsymbol{\pi}}(s, a_i, \boldsymbol{a}_{-i})\right)}{\sum_{a_i'} \exp\left(\lambda \text{ExpRet}_i^{\boldsymbol{\pi}}(s, a_i', \boldsymbol{a}_{-i})\right)}$$

Intuitively, in LQRE, agents choose actions with higher expected return with higher probability.

2.3. Learning from Expert Demonstrations

Suppose we do not have access to the ground truth reward signal r, but have demonstrations $\mathcal{D}$ provided by an expert (N expert agents in Markov games). $\mathcal{D}$ is a set of trajectories $\{\tau_j\}_{j=1}^M$, where $\tau_j = \{(s_j^t, \boldsymbol{a}_j^t)\}_{t=1}^T$ is an expert trajectory

collected by sampling $s^1 \sim \eta(s), \boldsymbol{a}^t \sim \pi_E(\boldsymbol{a}^t|s^t), s^{t+1} \sim P(s^{t+1}|s^t, \boldsymbol{a}^t)$. $\mathcal{D}$ contains the entire supervision to the learning algorithm, *i.e.*, we assume we cannot ask for additional interactions with the experts during training. Given $\mathcal{D}$, imitation learning (IL) aims to directly learn policies that behave similarly to these demonstrations, whereas inverse reinforcement learning (IRL) (Russell, 1998; Ng et al., 2000) seeks to infer the underlying reward functions which induce the expert policies.

The MaxEnt IRL framework (Ziebart et al., 2008) aims to recover a reward function that rationalizes the expert behaviors with the *least commitment*, denoted as $\mathrm{IRL}(\pi_E)$:

$$\mathrm{IRL}(\pi_E) = \arg\max_{r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_{\pi_E}[r(s,a)] - \mathrm{RL}(r)$$
$$\mathrm{RL}(r) = \max_{\pi \in \Pi} \mathcal{H}(\pi) + \mathbb{E}_{\pi}[r(s,a)]$$

where $\mathcal{H}(\pi) = \mathbb{E}_{\pi}[-\log \pi(a|s)]$ is the policy entropy. However, MaxEnt IRL is generally considered less efficient and scalable than direct imitation, as we need to solve a forward RL problem in the inner loop. In the context of imitation learning, (Ho & Ermon, 2016) proposed to use generative adversarial training (Goodfellow et al., 2014), to learn the policies characterized by $\mathrm{RL} \circ \mathrm{IRL}(\pi_E)$ directly, leading to the Generative Adversarial Imitation Learning (GAIL) algorithm:

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\pi_E}\left[\log D_\omega(s,a)\right] + \mathbb{E}_{\pi_\theta}\left[\log(1 - D_\omega(s,a))\right]$$

where D_ω is a discriminator that classifies expert and policy trajectories, and π_θ is the parameterized policy that tries to maximize its score under D_ω. According to Goodfellow et al. (2014), with infinite data and infinite computation, at optimality, the distribution of generated state-action pairs should exactly match the distribution of demonstrated state-action pairs under the GAIL objective. The downside to this approach, however, is that we bypass the intermediate step of recovering rewards. Specifically, note that we cannot extract reward functions from the discriminator, as $D_\omega(s,a)$ will converge to 0.5 for all (s,a) pairs.

2.4. Adversarial Inverse Reinforcement Learning

Besides resolving the ambiguity that many optimal policies can explain a set of demonstrations, another advantage of MaxEnt IRL is that it can be interpreted as solving the following maximum likelihood estimation (MLE) problem:

$$p_\omega(\tau) \propto \left[\eta(s^1)\prod_{t=1}^{T} P(s^{t+1}|s^t, a^t)\right] \exp\left(\sum_{t=1}^{T} r_\omega(s^t, a^t)\right) \tag{1}$$

$$\max_{\omega} \mathbb{E}_{\pi_E}\left[\log p_\omega(\tau)\right] = \mathbb{E}_{\tau\sim\pi_E}\left[\sum_{t=1}^{T} r_\omega(s^t, a^t)\right] - \log Z_\omega$$

Here, ω are the parameters of the reward function and Z_ω is the partition function, *i.e.* an integral over all possible trajectories consistent with the environment dynamics. Z_ω is intractable to compute when the state-action spaces are large or continuous, and the environment dynamics are unknown.

Combining Guided Cost Learning (GCL) (Finn et al., 2016b) and generative adversarial training, Finn et al.; Fu et al. proposed adversarial IRL framework as an efficient sampling based approximation to the MaxEnt IRL, where the discriminator takes on a particular form:

$$D_\omega(s,a) = \frac{\exp(f_\omega(s,a))}{\exp(f_\omega(s,a)) + q(a|s)}$$

where $f_\omega(s,a)$ is the learned function, $q(a|s)$ is the probability of the adaptive sampler pre-computed as an input to the discriminator, and the policy is trained to maximize $\log D - \log(1-D)$.

To alleviate the reward shaping ambiguity (Ng et al., 1999), where many reward functions can explain an optimal policy, (Fu et al., 2017) further restricted f to a reward estimator g_ω and a potential shaping function h_ϕ:

$$f_{\omega,\phi}(s,a,s') = g_\omega(s,a) + \gamma h_\phi(s') - h_\phi(s)$$

It has been shown that under suitable assumptions, g_ω and h_ϕ will recover the true reward and value function up to a constant.

3. Method

3.1. Logistic Stochastic Best Response Equilibirum

To extend MaxEnt IRL to Markov games, we need be able to characterize the trajectory distribution induced by a set of (parameterized) reward functions $\{r_i(s, \boldsymbol{a})\}_{i=1}^N$ (analogous to Equation (1)). However existing optimality notions introduced in Section 2.2 do not *explicitly* define a tractable joint strategy profile that we can use to maximize the likelihood of expert demonstrations (as a function of the rewards); they do so *implicitly* as the solution to a set of constraints.

Motivated by Gibbs sampling (Hastings, 1970), dependency networks (Heckerman et al., 2000), best response dynamic (Nisan et al., 2011; Gandhi, 2012) and LQRE, we propose a new solution concept that allows us to characterize rational (joint) policies induced from a set of reward functions. Intuitively, our solution concept corresponds to the result of repeatedly applying a stochastic (entropy-regularized) best response mechanism, where each agent (in turns) attempts to optimize its actions while keeping the other agents' actions fixed.

To begin with, let us first consider a stateless single-shot normal-form game with N players and a reward function

$r_i : \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \to \mathbb{R}$ for each player i. We consider the following *Markov chain* over $(\mathcal{A}_1 \times \cdots \times \mathcal{A}_N)$, where the state of the Markov chain at step k is denoted $\mathbf{z}^{(k)} = (z_1, \cdots, z_N)^{(k)}$, with each random variable $z_i^{(k)}$ taking values in $\mathcal{A}_i$. The transition kernel of the Markov Chain is defined by the following equations:

$$z_i^{(k+1)} \sim P_i(a_i|\boldsymbol{a}_{-i} = \boldsymbol{z}_{-i}^{(k)}) = \frac{\exp(\lambda r_i(a_i, \boldsymbol{z}_{-i}^{(k)}))}{\sum_{a_i'} \exp(\lambda r_i(a_i', \boldsymbol{z}_{-i}^{(k)}))} \tag{2}$$

and each agent i is updated in scan order. Given all other players' actions $\boldsymbol{z}_{-i}^{(k)}$, the i-th player picks an action proportionally to $\exp(\lambda r_i(a_i, \boldsymbol{z}_{-i}^{(k)}))$, where $\lambda > 0$ is a parameter that controls the level of rationality of the agents. For $\lambda \to 0$, the agent will select actions uniformly at random, while for $\lambda \to \infty$, the agent will select actions greedily (best response). Because the Markov Chain is ergodic, it admits a unique stationary distribution which we denote $\boldsymbol{\pi}(\boldsymbol{a})$. Interpreting this stationary distribution over $(\mathcal{A}_1 \times \cdots \times \mathcal{A}_N)$ as a policy, we call this stationary joint policy a *logistic stochastic best response equilibrium* for normal-form games.

Now let us generalize the solution concept to Markov games. For each agent i, let $\{\pi_i^t\}_{t=1}^T$ denote a set of time-dependent policies. First we define the state action value function for each agent i. Starting from the base case:

$$Q_i^{\boldsymbol{\pi}^{\varnothing}}(s^T, a_i^T, \boldsymbol{a}_{-i}^T) = r_i(s^T, a_i^T, \boldsymbol{a}_{-i}^T)$$

then we recursively define:

$$\begin{aligned} Q_i^{\boldsymbol{\pi}^{t+1:T}}(s^t, a_i^t, \boldsymbol{a}_{-i}^t) = r_i(s^t, a_i^t, \boldsymbol{a}_{-i}^t) + \\ \mathbb{E}_{s^{t+1} \sim P(\cdot|s^t, \boldsymbol{a}^t)} \Big[\mathcal{H}(\pi_i^{t+1}(\cdot|s^{t+1})) + \\ \mathbb{E}_{\boldsymbol{a}^{t+1} \sim \boldsymbol{\pi}^{t+1}(\cdot|s^{t+1})} [Q_i^{\boldsymbol{\pi}^{t+2:T}}(s^{t+1}, \boldsymbol{a}^{t+1})] \Big] \end{aligned}$$

which generalizes the standard state-action value function in single-agent RL ($\boldsymbol{a}_{-i} = \varnothing$ when $N = 1$).

Definition 2. *Given a Markov game with horizon T, the logistic stochastic best response equilibrium (LSBRE) is a sequence of T stochastic policies $\{\boldsymbol{\pi}^t\}_{t=1}^T$ constructed by the following process. Consider T Markov chains over $(\mathcal{A}_1 \times \cdots \mathcal{A}_N)^{|\mathcal{S}|}$, where the state of the t-th Markov chain at step k is $\{z_i^{t,(k)} : \mathcal{S} \to \mathcal{A}_i\}_{i=1}^N$, with each random variable $z_i^{t,(k)}(s)$ taking values in $\mathcal{A}_i$. For $t \in [T, \ldots, 1]$, we recursively define the the stationary joint distribution $\pi^t(\boldsymbol{a}|s)$ of the t-th Markov chain in terms of $\{\boldsymbol{\pi}^\ell\}_{\ell=t+1}^T$ as:*

For $s^t \in \mathcal{S}, i \sim [1, \cdots, N]$, we update the state of the Markov chain as:

$$z_i^{t,(k+1)}(s^t) \sim P_i^t(a_i^t|\boldsymbol{a}_{-i}^t = \boldsymbol{z}_{-i}^{t,(k)}(s^t), s^t) = \frac{\exp\left(\lambda Q_i^{\boldsymbol{\pi}^{t+1:T}}(s^t, a_i^t, \boldsymbol{z}_{-i}^{t,(k)}(s^t))\right)}{\sum_{a_i'} \exp\left(\lambda Q_i^{\boldsymbol{\pi}^{t+1:T}}(s^t, a_i', \boldsymbol{z}_{-i}^{t,(k)}(s^t))\right)} \tag{3}$$

where parameter $\lambda \in \mathbb{R}^+$ controls the level of rationality of the agents, and $\{P_i^t\}_{i=1}^N$ specifies a set of conditional distributions. LSBRE for Markov game is the sequence of T joint stochastic policies $\{\boldsymbol{\pi}^t\}_{t=1}^T$. Each joint policy $\boldsymbol{\pi}^t : \mathcal{S} \to \mathcal{P}(\mathcal{A}_1 \times \cdots \times \mathcal{A}_N)$ is given by:

$$\boldsymbol{\pi}^t(a_1, \cdots, a_N|s^t) = P\left(\bigcap_i \{z_i^{t,(\infty)}(s^t) = a_i\}\right) \tag{4}$$

where the probability is taken with respect to the unique stationary distribution of the t-th Markov chain.

When the set of conditionals in Equation (2) are compatible (in the sense that each conditional can be inferred from the *same* joint distribution (Arnold & Press, 1989)), the above process corresponds to a *Gibbs sampler*, which will converge to a stationary joint distribution $\boldsymbol{\pi}(\boldsymbol{a})$, whose conditional distributions are consistent with the ones used during sampling, namely Equation (2). This is the case, for example, if the agents are cooperative, *i.e.*, they share the same reward function r_i. In general, $\boldsymbol{\pi}(\boldsymbol{a})$ is the distribution specified by the dependency network (Heckerman et al., 2000) defined via conditionals in Equation (2). The same argument can be made for the Markov Chains in Definition 2 with respect to the conditionals in Equation (3).

When the set of conditionals in Equation (2) and (3) are incompatible, the procedure is called a *pseudo Gibbs sampler*. As discussed in literatures on dependency networks (Heckerman et al., 2000; Chen et al., 2011; Chen & Ip, 2015), when the conditionals are *learned* from a sufficiently large dataset, the pseudo Gibbs sampler asymptotically works well in the sense that the conditionals of the stationary joint distribution are nearly consistent with the conditionals used during sampling. Under some conditions, theoretical bounds on the approximation can be obtained (Heckerman et al., 2000).

3.2. Trajectory Distributions Induced by LSBRE

Following (Fu et al., 2017; Levine, 2018), without loss of generality, in the remainder of this paper we consider the case where $\lambda = 1$. First, we note that there is a connection between the notion of LSBRE and maximum causal entropy reinforcement learning (Ziebart, 2010). Specifically, we can characterize the trajectory distribution induced by LSBRE policies with an energy-based formulation, where the probability of a trajectory increases exponentially as the sum of rewards increases. Formally, with LSBRE policies, the probability of generating a certain trajectory can be characterized with the following theorem:

Theorem 1. *Given a joint policy $\{\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t)\}_{t=1}^T$ specified by LSBRE, for each agent i, let $\{\boldsymbol{\pi}_{-i}^t(\boldsymbol{a}_{-i}^t|s^t)\}_{t=1}^T$ denote other agents' marginal distribution and $\{\pi_i^t(a_i^t|\boldsymbol{a}_{-i}^t, s^t)\}_{t=1}^T$ denote agent i's conditional distribution, both obtained from the LSBRE joint policies.*

Then the LSBRE conditional distributions are the optimal solution to the following optimization problem:

$$\min_{\hat{\pi}^{1:T}} D_{\mathrm{KL}}(\hat{p}(\tau)||\tilde{p}(\tau)) \tag{5}$$

$$\hat{p}(\tau) = \left[\eta(s^1)\cdot\prod_{t=1}^{T} P(s^{t+1}|s^t,\boldsymbol{a}^t)\cdot\boldsymbol{\pi}^t_{-i}(\boldsymbol{a}^t_{-i}|s^t))\right]\cdot \prod_{t=1}^{T}\hat{\pi}^t_i(a^t_i|\boldsymbol{a}^t_{-i},s^t)$$

$$\tilde{p}(\tau) \propto \left[\eta(s^1)\cdot\prod_{t=1}^{T} P(s^{t+1}|s^t,\boldsymbol{a}^t)\cdot\boldsymbol{\pi}^t_{-i}(\boldsymbol{a}^t_{-i}|s^t))\right]\cdot \exp\left(\sum_{t=1}^{T} r_i(s^t,a^t_i,\boldsymbol{a}^t_{-i})\right) \tag{6}$$

Proof. See Appendix A.1. □

Intuitively, for single-shot normal form games, the above statement holds obviously from the definition in Equation (2). For Markov games, similar to the process introduced in Definition 2, we can employ a dynamic programming algorithm to find the conditional policies which minimizes Equation (5). Specifically, we first construct the base case of $t = T$ as a normal form game, then recursively construct the conditional policy for each time step t, based on the policies from $t+1$ to T that have already been constructed. It can be shown that the constructed optimal policy which minimizes the KL divergence between its trajectory distribution and the trajectory distribution defined in Equation (6) corresponds to the set of conditional policies in LSBRE.

3.3. Multi-Agent Adversarial IRL

In the remainder of this paper, we assume that the expert policies form a unique LSBRE under some unknown (parameterized) reward functions, according to Definition 2. By adopting LSBRE as the optimality notion, we are able to rationalize the demonstrations by maximizing the likelihood of the expert trajectories with respect to the LSBRE stationary distribution, which is in turn induced by the $\boldsymbol{\omega}$-parameterized reward functions $\{r_i(s,\boldsymbol{a};\omega_i)\}_{i=1}^N$.

The probability of a trajectory $\tau = \{s_t,\boldsymbol{a}_t\}_{t=1}^T$ generated by LSBRE policies in a Markov game is defined by the following generative process:

$$p(\tau) = \eta(s^1)\cdot\prod_{t=1}^{T}\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t;\boldsymbol{\omega})\cdot\prod_{t=1}^{T} P(s^{t+1}|s^t,\boldsymbol{a}^t) \tag{7}$$

where $\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t;\boldsymbol{\omega})$ are the unique stationary joint distributions for the LSBRE induced by $\{r_i(s,\boldsymbol{a};\omega_i)\}_{i=1}^N$. The initial state distribution $\eta(s^1)$ and transition dynamics $P(s^{t+1}|s^t,\boldsymbol{a}^t)$ are specified by the Markov game.

As mentioned in Section 2.4, the MaxEnt IRL framework interprets finding suitable reward functions as maximum likelihood over the expert trajectories in the distribution defined in Equation (7), which can be reduced to:

$$\max_{\boldsymbol{\omega}} \mathbb{E}_{\tau\sim\boldsymbol{\pi}_E}\left[\sum_{t=1}^{T}\log\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t;\boldsymbol{\omega})\right] \tag{8}$$

since the initial state distribution and transition dynamics do not depend on the parameterized rewards.

Note that $\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t)$ in Equation (8) is the joint policy defined in Equation (4), whose conditional distributions are given by Equation (3). From Section 3.1, we know that given a set of $\boldsymbol{\omega}$-parameterized reward functions, we are able to characterize the conditional policies $\{\pi^t_i(a^t_i|\boldsymbol{a}^t_{-i},s^t)\}_{t=1}^T$ for each agent i. However direct optimization over the joint MLE objective in Equation (8) is intractable, as we cannot obtain a closed form for the stationary joint policy. Fortunately, we are able to construct an asymptotically consistent estimator by approximating the joint likelihood $\boldsymbol{\pi}^t(\boldsymbol{a}^t|s^t)$ with a product of the conditionals $\prod_{i=1}\pi^t_i(a^t_i|\boldsymbol{a}^t_{-i},s^t)$, which is termed a *pseudolikelihood* (Besag, 1975).

With the asymptotic consistency property of the *maximum pseudolikelihood estimation* (Besag, 1975; Lehmann & Casella, 2006), we have the following theorem:

Theorem 2. *Let demonstrations $\tau_1,\ldots,\tau_M$ be independent and identically distributed (sampled from LSBRE induced by some unknown reward functions), and suppose that for all $t\in[1,\ldots,T], a^t_i\in\mathcal{A}_i$, $\pi^t_i(a^t_i|\boldsymbol{a}^t_{-i},s^t;\omega_i)$ is differentiable with respect to ω_i. Then, with probability tending to 1 as $M\to\infty$, the equation*

$$\frac{\partial}{\partial\boldsymbol{\omega}}\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{i=1}^{N}\log\pi^t_i(a^{m,t}_i|\boldsymbol{a}^{m,t}_{-i},s^{m,t};\omega_i) = 0 \tag{9}$$

has a root $\hat{\boldsymbol{\omega}}_M$ such that $\hat{\boldsymbol{\omega}}_M$ tends to the maximizer of the joint likelihood in Equation (8).

Proof. See Appendix A.2. □

Theorem 2 bridges the gap between optimizing the joint likelihood and each conditional likelihood. Now we are able to maximize the objective in Equation (8) as:

$$\mathbb{E}_{\boldsymbol{\pi}_E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial\boldsymbol{\omega}}\log\pi^t_i(a^t_i|\boldsymbol{a}^t_{-i},s^t;\omega_i)\right] \tag{10}$$

To optimize the maximum pseudolikelihood objective in Equation (10), we can instead optimize the following surrogate loss which is a variational approximation to the psuedolikelihood objective (from Theorem 1):

$$\mathbb{E}_{\boldsymbol{\pi}_E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial\boldsymbol{\omega}}r_i(s^t,\boldsymbol{a}^t;\omega_i)\right] - \sum_{i=1}^{N}\frac{\partial}{\partial\boldsymbol{\omega}}\log Z_{\omega_i}$$

where Z_{ω_i} is the partition function of the distribution in Equation (6). It is generally intractable to exactly compute and optimize the partition function Z_ω, which involves an integral over all trajectories. Similar to GCL (Finn et al., 2016b) and single-agent AIRL (Fu et al., 2017), we employ *importance sampling* to estimate the partition function with adaptive samplers $\boldsymbol{q_\theta}$. Now we are ready to introduce our practical Multi-Agent Adversarial IRL (MA-AIRL) framework, where we train the $\boldsymbol{\omega}$-parameterized discriminators as:

$$\max_{\boldsymbol{\omega}} \mathbb{E}_{\boldsymbol{\pi}_E}\left[\sum_{i=1}^{N}\log\frac{\exp(f_{\omega_i}(s,\boldsymbol{a}))}{\exp(f_{\omega_i}(s,\boldsymbol{a}))+q_{\theta_i}(a_i|s)}\right]+ \mathbb{E}_{\boldsymbol{q_\theta}}\left[\sum_{i=1}^{N}\log\frac{q_{\theta_i}(a_i|s)}{\exp(f_{\omega_i}(s,\boldsymbol{a}))+q_{\theta_i}(a_i|s)}\right] \quad (11)$$

and we train the $\boldsymbol{\theta}$-parameterized generators as:

$$\max_{\theta} \mathbb{E}_{\boldsymbol{q_\theta}}\left[\sum_{i=1}^{N}\log(D_{\omega_i}(s,\boldsymbol{a}))-\log(1-D_{\omega_i}(s,\boldsymbol{a}))\right] = \mathbb{E}_{\boldsymbol{q_\theta}}\left[\sum_{i=1}^{N} f_{\omega_i}(s,\boldsymbol{a})-\log(q_{\theta_i}(a_i|s))\right] \quad (12)$$

Specifically, for each agent i, we have a discriminator with a particular structure $\frac{\exp(f_\omega)}{\exp(f_\omega)+q_\theta}$ for a binary classification, and a generator as an adaptive importance sampler for estimating the partition function. Intuitively, q_θ is trained to minimize the KL divergence between its trajectory distribution and that induced by the reward functions, for reducing the variance of importance sampling, while f_ω in the discriminator is trained to estimate the reward function. At optimality, f_ω will approximate the advantage function for the expert policy and q_θ will approximate the expert policy.

3.4. Solving Reward Ambiguity in Multi-Agent IRL

For single-agent reinforcement learning, Ng et al. shows that for any state-only potential function $\phi : \mathcal{S} \to \mathbb{R}$, potential-based reward shaping defined as:

$$r'(s^t, a^t, s^{t+1}) = r(s^t, a^t, s^{t+1}) + \gamma\Phi(s^{t+1}) - \Phi(s^t)$$

is a necessary and sufficient condition to guarantee invariance of the optimal policy in both finite and infinite horizon MDPs. In other words, given a set of expert demonstrations, there is a class of reward functions, all of which can explain the demonstrated expert behaviors. Thus without further assumptions, it would be impossible to identify the ground-truth reward that induces the expert policy within this class. Similar issues also exist when we consider multi-agent scenarios. Devlin & Kudenko show that in multi-agent systems, using the same reward shaping for one or more agents will not alter the set of Nash equilibria. It is possible to extend this result to other solution concepts such as CE and LSBRE. For example, in the case of LSBRE, after specifying the level of rationality λ, for any $\pi_i \neq \pi_i^{\text{LSBRE}}$, we have:

$$\mathbb{E}_{\pi_i^{\text{LSBRE}},\boldsymbol{\pi}_{-i}^{\text{LSBRE}}}[r_i(s,\boldsymbol{a})] \geq \mathbb{E}_{\pi_i,\boldsymbol{\pi}_{-i}^{\text{LSBRE}}}[r_i(s,\boldsymbol{a})] \quad (13)$$

since each individual LSBRE conditional policy is the optimal solution to the corresponding entropy regularized RL problem (See Appendix (A.1)). It can be also shown that any policy that satisfies the inequality in Equation (13) will still satisfy the inequality after reward shaping (Devlin & Kudenko, 2011).

To mitigate the reward shaping effect and recover reward functions with higher linear correlation to the ground truth reward, as in (Fu et al., 2017), we further assume the functions f_{ω_i} in Equation (11) have a specific structure:

$$f_{\omega_i,\phi_i}(s^t, a^t, s^{t+1}) = g_{\omega_i}(s^t, a^t) + \gamma h_{\phi_i}(s^{t+1}) - h_{\phi_i}(s^t)$$

where g_ω is a reward estimator and h_ϕ is a potential function. We summarize the MA-AIRL training procedure in Algorithm 1.

Algorithm 1 Multi-Agent Adversarial IRL

Input: Expert trajectories $\mathcal{D}_E = \{\tau_j^E\}$; Markov game as a black box with parameters $(N, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \eta, \boldsymbol{P}, \gamma)$
Initialize the parameters of policies $\boldsymbol{q}$, reward estimators $\boldsymbol{g}$ and potential functions $\boldsymbol{h}$ with $\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\phi}$.
repeat
  Sample trajectories $\mathcal{D}_\pi = \{\tau_j\}$ from $\boldsymbol{\pi}$:
    $s^1 \sim \eta(s), \boldsymbol{a}^t \sim \boldsymbol{\pi}(\boldsymbol{a}^t|s^t), s^{t+1} \sim P(s^{t+1}|s^t, \boldsymbol{a}^t)$
  Sample state-action pairs $\mathcal{X}_\pi$, $\mathcal{X}_E$ from $\mathcal{D}_\pi$, $\mathcal{D}_E$.
  for $i = 1, \ldots,$ N **do**
    Update ω_i, ϕ_i to increase the objective in Eq. 11:
    $\mathbb{E}_{\mathcal{X}_E}[\log D(s, a_i, s')] + \mathbb{E}_{\mathcal{X}_\pi}[\log(1 - D(s, a_i, s'))]$
  end for
  for $i = 1, \ldots,$ N **do**
    Update reward estimates $\hat{r}_i(s, a_i, s')$ with $g_{\omega_i}(s, a_i)$.
    or $(\log D(s, a_i, s') - \log(1 - D(s, a_i, s')))$
    Update θ_i with respect to $\hat{r}_i(s, a_i, s')$.
  end for
until Convergence
Output: Learned policies $\boldsymbol{\pi_\theta}$ and reward functions $\boldsymbol{g_\omega}$.

4. Related Work

A vast number of methods and paradigms have been proposed for single-agent imitation learning and inverse reinforcement learning. However, multi-agent scenarios are less commonly investigated, and most existing works assume specific reward structures. These include fully cooperative games (Barrett et al., 2017; Le et al., 2017; Šošić et al., 2017; Bogert & Doshi, 2014), two player zero-sum games (Lin

et al., 2014), and rewards as linear combinations of pre-specified features (Reddy et al., 2012; Waugh et al., 2013). Recently, Song et al. proposed MA-GAIL, a multi-agent extension of GAIL which works on general Markov games.

While both MA-AIRL and MA-GAIL are based on adversarial training, the methods are inherently different. MA-GAIL is based on the notion of Nash equilibrium, and is motivated via a specific choice of Lagrange multipliers for a constraint optimization problem. MA-AIRL, on the other hand, is derived from MaxEnt RL and LSBRE, and aims to obtain an MLE solution for the joint trajectories; we connect this with a set of conditionals via pseudolikelihood, which are then solved with the adversarial reward learning framework. From a reward learning perspective, the discriminators' outputs in MA-GAIL will converge to uninformative uniform distribution, while MA-AIRL allows us to recover reward functions from the optimal discriminators.

5. Experiments

We seek to answer the following questions via empirical evaluation: (1) Can MA-AIRL efficiently recover the expert policies for each individual agent from the expert demonstrations (policy imitation)? (2) Can MA-AIRL effectively recover the underlying reward functions, for which the expert policies form a LSBRE (reward recovery)?

Task Description To answer these questions, we evaluate our MA-AIRL algorithm on a series of simulated particle environments (Lowe et al., 2017). Specifically, we consider the following scenarios: *cooperative navigation*, where three agents cooperate through physical actions to reach three landmarks; *cooperative communication*, where two agents, a speaker and a listener, cooperate to navigate to a particular landmark; and *competitive keep-away*, where one agent tries to reach a target landmark, while an adversary, without knowing the target a priori, tries to infer the target from the agent's behaviors and prevent it from reaching the goal through physical interactions.

In our experiments, for generality, the learning algorithms will not leverage any prior knowledge on the types of interactions (cooperative or competitive). Thus for all the tasks described above, the learning algorithms will take a decentralized form and we will not utilize additional reward regularization, besides penalizing the $\ell 2$ norm of the reward parameters to mitigate overfitting (Ziebart, 2010; Kalakrishnan et al., 2013).

Training Procedure In the simulated environments, we have access to the ground-truth reward functions, which enables us to accurately evaluate the quality of both recovered policies and reward functions. We use a multi-agent version of ACKTR (Wu et al., 2017; Song et al., 2018), an efficient model-free policy gradient algorithm for training the experts as well as the adaptive samplers in MA-AIRL. The supervision signals for the experts come from the ground-truth rewards, while the reward signals for the adaptive samplers come from the discriminators. Specifically, we first obtain expert policies induced by the ground-truth rewards, then we use them to generate demonstrations, from which the learning algorithms will try to recover the policies as well as the underlying reward functions. We compare MA-AIRL against the state-of-the-art multi-agent imitation learning algorithm, MA-GAIL (Song et al., 2018), which is a generalization of GAIL to Markov games. Following (Li et al., 2017; Song et al., 2018), we use behavior cloning to pretrain MA-AIRL and MA-GAIL to reduce sample complexity for exploration, and we use 200 episodes of expert demonstrations, each with 50 time steps, which is close to the amount of time steps used in (Ho & Ermon, 2016)[1].

5.1. Policy Imitation

Although MA-GAIL achieved superior performance compared with behavior cloning (Song et al., 2018), it only aims to recover policies via distribution matching. Moreover, the training signal for the policy will become less informative as training progresses; according to (Goodfellow et al., 2014) with infinite data and computational resources the discriminator outputs will converge to 0.5 for all state-action pairs, which could potentially hinder the robustness of the policy towards the end of training. To empirically verify our claims, we compare the quality of the learned policies in terms of the expected return received by each agent.

In the cooperative environment, we directly use the ground-truth rewards from the environment as the oracle metric, since all agents share the same reward. In the competitive environment, we follow the evaluation procedure in (Song et al., 2018), where we place the experts and learned policies in the same environment. A learned policy is considered "better" if it receives a higher expected return while its opponent receives a lower expected return. The results for cooperative and competitive environments are shown in Tables 1 and 2 respectively. MA-AIRL consistently performs better than MA-GAIL in terms of the received reward in all the considered environments, suggesting superior imitation learning capabilities to the experts.

5.2. Reward Recovery

The second question we seek to answer is concerned with the reward recovering problem as in inverse reinforcement learning: is the algorithm able to recover the ground truth reward functions with expert demonstrations being the only source of supervision? To answer this question, we evaluate the statistical correlations between the ground truth rewards

[1]The codebase for this work can be found at https://github.com/ermongroup/MA-AIRL.

Table 1. Expected returns in cooperative tasks. Mean and variance are taken across different random seeds used to train the policies.

Algorithm	Nav. ExpRet	Comm. ExpRet
Expert	-43.195 $\pm$ 2.659	-12.712 $\pm$ 1.613
Random	-391.314 $\pm$ 10.092	-125.825 $\pm$ 3.4906
MA-GAIL	-52.810 $\pm$ 2.981	-12.811 $\pm$ 1.604
MA-AIRL	**-47.515** $\pm$ 2.549	**-12.727** $\pm$ 1.557

Table 2. Expected returns of the agents in competitive task. Agent #1 represents the agent trying to reach the target and Agent #2 represents the adversary. Mean and variance are taken across different random seeds.

Agent #1	Agent #2	Agent #1 ExpRet
Expert	Expert	-6.804 $\pm$ 0.316
MA-GAIL	Expert	-6.978 $\pm$ 0.305
MA-AIRL	Expert	**-6.785** $\pm$ 0.312
Expert	MA-GAIL	-6.919 $\pm$ 0.298
Expert	MA-AIRL	**-7.367**$\pm$ 0.311

(which the learning algorithms have no access to) and the inferred rewards for the same state-action pairs.

Specifically, we consider two types of statistical correlations: *Pearson's correlation coefficient* (PCC), which measures the linear correlation between two random variables; and *Spearman's rank correlation coefficient* (SCC), which measures the statistical dependence between the rankings of two random variables. Higher SCC suggests that two reward functions have higher monotonic relationships and higher PCC suggests higher linear correlations. For each trajectory, we compare the ground-truth return from the environment with the supervision signals from the discriminators, which correspond to g_ω in MA-AIRL and $\log(D_\omega)$ in MA-GAIL.

Tables 3 and 4 provide the SCC and PCC statistics for cooperative and competitive environments respectively. In the cooperative case, compared to MA-GAIL, MA-AIRL achieves a much higher PCC and SCC, which could facilitate policy learning. The statistical correlations between reward signals gathered from discriminators for each agent are also quite high, suggesting that while we do not reveal the agents are cooperative, MA-AIRL is able to discover high correlations between the agents' reward functions. In the competitive case, the reward functions learned by MA-AIRL also significantly outperform MA-GAIL in terms of SCC and PCC statistics. In Figure 1, we further show the changes of PCC statistics with respect to training time steps for MA-GAIL and MA-AIRL. The reward functions recovered by MA-GAIL initially have a high correlation with the ground truth, yet that dramatically decreases as training continues, whereas the functions learned by MA-AIRL maintains a high correlation throughout the course of training, which is in line with the theoretical analysis that in MA-GAIL, reward signals from the discriminators will become less informative towards convergence.

Table 3. Statistical correlations between the learned reward functions and the ground-truth rewards in cooperative tasks. Mean and variance are taken across N independently learned reward functions for N agents.

Task	Metric	MA-GAIL	MA-AIRL
Nav.	SCC	0.792 $\pm$ 0.085	**0.934** $\pm$ 0.015
	PCC	0.556 $\pm$ 0.081	**0.882** $\pm$ 0.028
Comm.	SCC	0.879 $\pm$ 0.059	**0.936** $\pm$ 0.080
	PCC	0.612 $\pm$ 0.093	**0.848** $\pm$ 0.099

Table 4. Statistical correlations between the learned reward functions and the ground-truth rewards in competitive task.

Algorithm	MA-GAIL	MA-AIRL
SCC #1	0.424	**0.534**
SCC #2	0.653	**0.907**
Average SCC	0.538	**0.721**
PCC #1	0.497	**0.720**
PCC #2	0.392	**0.667**
Average PCC	0.445	**0.694**

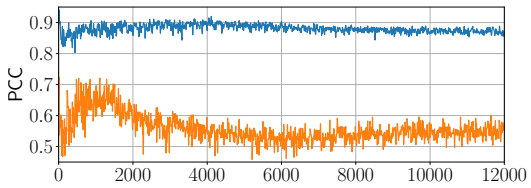

Figure 1. PCC w.r.t. the training epochs in cooperative navigation, with MA-AIRL (blue) and MA-GAIL (orange).

6. Discussion and Future Work

We propose MA-AIRL, the first multi-agent MaxEnt IRL framework that is effective and scalable to Markov games with high-dimensional state-action space and unknown dynamics. We derive our algorithm based on a solution concept termed LSBRE and we employ maximum pseudolikelihood estimation to achieve tractability. Experimental results demonstrate that MA-AIRL is able to imitate expert behaviors in high-dimensional complex environments, as well as learn reward functions that are highly correlated with the ground truth rewards. An exciting avenue for future work is to include reward regularization to mitigate overfitting and leverage prior knowledge of the task structure.