ASYNCHRONOUS MULTI-AGENT GENERATIVE ADVERSARIAL IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning aims to inversely learn a policy from expert demonstrations, which has been extensively studied in the literature for both single-agent setting with Markov decision process (MDP) model, and multi-agent setting with Markov game (MG) model. However, existing approaches for general multi-agent Markov games are not applicable to multi-agent *extensive* Markov games, where agents make asynchronous decisions following a certain order, rather than simultaneous decisions. We propose a novel framework for asynchronous multi-agent generative adversarial imitation learning (AMAGAIL) under general extensive Markov game settings, and the learned expert policies are proven to guarantee subgame perfect equilibrium (SPE), a more general and stronger equilibrium than Nash equilibrium (NE). The experiment results demonstrate that compared to state-of-the-art baselines, our AMAGAIL model can better infer the policy of each expert agent using their demonstration data collected from asynchronous decision-making scenarios (i.e., extensive Markov games).

1 INTRODUCTION

Imitation learning (IL) also known as learning from demonstrations allows agents to imitate expert demonstrations to make optimal decisions without direct interactions with the environment. Especially, inverse reinforcement learning (IRL) (Ng et al. (2000)) recovers a reward function of an expert from collected demonstrations, where it assumes that the demonstrator follows an (near-)optimal policy that maximizes the underlying reward. However, IRL is an ill-posed problem, because a number of reward functions match the demonstrated data (Ziebart et al. (2008; 2010); Ho & Ermon (2016); Boularias et al. (2011), where various principles, including maximum entropy, maximum causal entropy, and relative entropy principles, are employed to solve this ambiguity (Ziebart et al. (2008; 2010); Boularias et al. (2011); Ho & Ermon (2016); Zhang et al. (2019)).

Going beyond imitation learning with single agent discussed above, recent works including Song et al. (2018),Yu et al. (2019), investigated a more general and challenging scenario with demonstration data from multiple interacting agents. Such interactions are modeled with multi-agent Markov games (Littman & Szepesvári (1996)) rather than Markov decision processes, where all agents make simultaneous decisions at each step. However, these works fail to model and characterize extensive-form Markov games (Fudenberg & Levine (1983)), with agents making asynchronous decisions over steps, which are common in many real world scenarios, for example, multiplayer games (Knutsson et al. (2004)), such as Go game, and many card games. Players take turns to play, thus influence each others' decisions. The order in which agents make decisions has significant impacts of the game equilibrium.

In this paper, we propose a novel framework, asynchronous multi-agent generative adversarial imitation learning (AMAGAIL): A group of experts provide demonstration data when playing an extensive Markov game (EMG) with in general an asynchronous decision-making process, and AMAGAIL inversely learns the decision-making policy of each expert. We introduce a *player function* governed by the environment to capture the participation order and dependency of agents when making decisions, which could be deterministic (i.e., taking turns) or stochastic (i.e., by chance) to participate. A player function of an agent is a probability function: given the perfectly known agent participation history, i.e. at each previous state in the history we know which agent(s) participant(s), it provides the probability of the agent participating in the next state. With EMG model, our

framework generalizes MAGAIL (Song et al. (2018)) from the Markov games to extensive Markov games, and the learned expert policies are proven to guarantee subgame perfect equilibrium (SPE) (Fudenberg & Levine (1983)), a more general and stronger equilibrium than Nash equilibrium (NE) (guaranteed in MAGAIL Song et al. (2018)). The experiment results demonstrate that compared to GAIL (Ho & Ermon (2016)) and MAGAIL (Song et al. (2018)), our AMAGAIL model can better infer the policy of each expert agent using their demonstration data collected from asynchronous decision-making scenarios (i.e., extensive Markov games).

2 PRELIMINARIES

2.1 EXTENSIVE MARKOV GAMES

Markov games (MGs) (Littman (1994)) are the cases of N interacting agents to make simultaneous decisions at each time step with strategies only depending on the current state. Extensive-form games (EFGs) (Fudenberg & Levine (1983)) allow N agents to make asynchronous decisions with strategies conditioned on the entire history of the game. Motivated by MGs (Littman (1994)) and EFGs (Fudenberg & Levine (1983)), we propose the framework of *extensive Markov games* (EMGs) to generalize the scenarios of multi-agent asynchronous decision making with memoryless stationary policies. *An extensive Markov game* is denoted as a tuple $(\mathcal{S}, \mathcal{A}, Y, \zeta, P, \eta, \boldsymbol{r})$ with a set of states $\mathcal{S}$ and N sets of actions $\{\mathcal{A}_i\}_{i=1}^N$. At each time step t with a state $s_t \in \mathcal{S}$, if an agent i takes an action, the indicator variable $I_{i,t} = 1$; otherwise, $I_{i,t} = 0$. As a result, the participation vector $\boldsymbol{I}_t = [I_{1,t}, \cdots, I_{N,t}]$ indicates active vs inactive agents at step t. The set of all possible participation vectors is denoted as $\mathcal{I}$, namely, $\boldsymbol{I}_t \in \mathcal{I}$. Moreover, $h_{t-1} = [\boldsymbol{I}_0, \cdots, \boldsymbol{I}_{t-1}]$ represent the participation history from step 0 to $t-1$. The player function Y (governed by the environment) describes the probability of an agent i to make an action at a step t, given the participation history h_{t-1}, namely, $Y(i|h_{t-1})$. Clearly, when the player function $Y(i|h_{t-1}) = 1$ for all agents i's at time step t, an extensive Markov game boils down to a Markov game (Littman (1994); Song et al. (2018)), where all agents take actions at all steps. ζ defines the distribution of the initial participation vector $\boldsymbol{I}_0$. Note that, the player function can be naturally extended to a higher-order form when the condition includes both previous participation history and previous state-action history; thus, it can be adapted to non-Markov processes. Let ϕ denotes no participation, determined by player function Y, the transition process to the next state follows a transition function: $P : \mathcal{S} \times \mathcal{A}_1 \cup \{\phi\} \times \cdots \times \mathcal{A}_N \cup \{\phi\} \mapsto \mathcal{P}(\mathcal{S})$. Agent i obtains a (bounded) reward given by a function $r_i : \mathcal{S} \times \mathcal{A}_i \mapsto \mathbb{R}$. Agent i aims to maximize its own total expected return $R_i = \sum_{t=0}^{\infty} \gamma^t r_{i,t}$, where $\gamma \in [0, 1]$ is the discount factor. Actions are chosen through a stationary and stochastic policy $\pi_i : \mathcal{S} \times \mathcal{A}_i \mapsto [0, 1]$. In this paper, bold variables without subscript i denote the concatenation of variables for all the agents, e.g., all actions as $\boldsymbol{a}$, the joint policy defined as $\boldsymbol{\pi}(\boldsymbol{a}|s) = \prod_{i=1}^N \pi_i(a_i|s)$, $\boldsymbol{r}$ as all rewards. Subscript $-i$ denotes all agents except for i, then $(a_i, \boldsymbol{a}_{-i})$ represents the action of all N agents $(a_1, \cdots, a_N)$. We use expectation with respect to a policy π to denote an expectation with respect to the trajectories it generates. For example, $\mathbb{E}_{\pi_i,Y}[r_i(s, a)] \triangleq \mathbb{E}_{s_t,a_i\sim\pi_i,I_{i,t}\sim Y}[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_i)]$, denotes the following sample process for the right hand side: $s_0 \sim \eta$, $I_{i,0} \sim \zeta$, $I_{i,t} \sim Y(i|h_{t-1})$, $a_i \sim \pi_i(a_i|s_t)$, $s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$, for $\forall i \in [N]$.

2.2 SUBGAME PERFECT EQUILIBRIUM FOR EXTENSIVE MARKOV GAMES

In Markov games (MGs), all agents make simultaneous decisions at any time step t, with the same goal of maximizing its own total expected return. Thus, agents' optimal policies are interrelated and mutually influenced. Nash equilibrium (NE) has been employed as the solution concept to resolve the dependency across agents, where no agents can achieve a higher expected reward by unilaterally changing its own policy (Song et al. (2018)). However, in extensive Markov games (EMGs) allowing asynchronous decisions, there exist situations where agents encounter states (subgames) resulted from other agents' "trembling-hand" actions. Since the procedure of finding NE does not guarantee all these states and subgames taken into consideration, when trapped in the situations, agents are not able to make optimal decisions based on their polices under NE. To address this problem, Selten firstly proposed subgame perfect equilibrium (SPE) (Selten (1965)). SPE ensures NE of every possible subgame of the original game. It has been shown that in a finite or infinite extensive-form game with either discrete or continued time, best-response strategies all converge to SPE, rather than NE (Selten (1965); Abramsky & Winschel (2017); Xu (2016)).

2.3 MULTI-AGENT IMITATION LEARNING IN MARKOV GAMES

In Markov games, MAGAIL (Song et al. (2018)) was proposed to learn experts' policies constrained by Nash equilibrium. Since there may exist multiple Nash equilibrium solutions, a maximum causal entropy regularizer is employed to resolve the ambiguity. Thus, the optimal policies can be found by solving the following multi-agent reinforcement learning problem.

$$\mathbf{MARL}(\mathbf{r}) = \arg\max_{\boldsymbol{\pi}} \sum_{i=1}^{N} (\beta H_i(\pi_i) + \mathbb{E}_{\pi_i, \pi_{E_{-i}}}[r_i]), \tag{1}$$

where $H_i(\pi_i)$ is the γ-discounted causal entropy of policy $\pi_i \in \Pi$, $H_i(\pi_i) \triangleq \mathbb{E}_{\pi_i}[-\log \pi_i(a_i|s)] = \mathbb{E}_{s_t, a_i \sim \pi_i}[-\sum_{t=0}^{\infty} \gamma^t \log \pi_i(a_i|s_t)]$, and β is a weight to the entropy regularization term. In practice, the reward function is unknown. MAGAIL applies multi-agent IRL (MAIRL) below to recover reward experts' functions, with ψ as a convex regularizer,

$$\mathbf{MAIRL}_{\psi}(\boldsymbol{\pi}_E) = \arg\max_{\mathbf{r}} -\psi(\mathbf{r}) + \sum_{i=1}^{N} (\mathbb{E}_{\boldsymbol{\pi}_E}[r_i]) - \big(\max_{\boldsymbol{\pi}} \sum_{i=1}^{N} (\beta H_i(\pi_i)) + \mathbb{E}_{\pi_i, \boldsymbol{\pi}_{E_{-i}}}[r_i]\big). \tag{2}$$

Moreover, Generative Adversarial Imitation Learning (GAIL) algorithm employs generative adversarial networks (GANs) to inversely learn the experts' policies captured by $\mathbf{MARL} \circ \mathbf{MAIRL}_{\psi}(\pi_E)$:

$$\min_{\theta} \max_{w} \mathbb{E}_{\boldsymbol{\pi}_\theta}\Big[\sum_{i=1}^{N} \log D_{w_i}(s, a_i)\Big] + \mathbb{E}_{\boldsymbol{\pi}_E}\Big[\sum_{i=1}^{N} \log(1 - D_{w_i}(s, a_i))\Big]. \tag{3}$$

D_{w_i} is a discriminator for agent i that classifies the experts' vs policy trajectories. π_θ represent the learned experts' parameterized policies, which generate trajectories with maximized the scores from D_{w_i} for $i \in [N]$.

3 ASYNCHRONOUS MULTI-AGENT IMITATION LEARNING

Extending multi-agent imitation learning to extensive Markov games is challenging, because of the asynchronous decision making and dynamic state (subgame) participating. In this section, we will tackle this problem using subgame perfect equilibrium (SPE) solution concept.

3.1 ASYNCHRONOUS MULTI-AGENT REINFORCEMENT LEARNING

In an extensive Markov game (EMG), Nash equilibrium needs to be guaranteed at each state $s \in \mathcal{S}$ [1], namely, we apply subgame perfect equilibrium (SPE) solution concept instead. Formally, a set of agent policies $\{\pi_i\}_{i=1}^{N}$ is an SPE if at each state $s \in \mathcal{S}$ (also considered as a root node of a subgame), no agent can achieve a higher reward by unilaterally changing its policy on the root node or any other descendant nodes of the root node, i.e., $\forall i \in [N], \forall \hat{\pi}_i \neq \pi_i, \mathbb{E}_{\pi_i, \boldsymbol{\pi}_{-i}, Y}[r_i] \geq \mathbb{E}_{\hat{\pi}_i, \boldsymbol{\pi}_{-i}, Y}[r_i]$. Therefore, our constrained optimization problem is (Filar & Vrieze (2012), Theorem 3.7.2)

$$\min_{\boldsymbol{\pi}, \boldsymbol{v}} f_r(\boldsymbol{\pi}, \boldsymbol{v}) = \sum_{i=1}^{N} \Big(\sum_{s \in \mathcal{S}, h \in \mathcal{H}} v_i(s|h) - \mathbb{E}_{a_i \sim \pi_i(\cdot|s)}[q_i(s, a_i|h)] \Big) \tag{4}$$

$$\textbf{s.t. } v_i(s|h) \geq q_i(s, a_i|h) \; \forall i \in [N], s \in \mathcal{S}, a_i \in \mathcal{A}_i, h \in \mathcal{H}, \tag{5}$$

$$\boldsymbol{v} \triangleq [v_1; \cdots; v_N]. \tag{6}$$

For an agent i with a probability of taking action a at state s_t given a history h_{t-1}, its Q-function is

$$q_i(s_t, a_i|h_{t-1}) = \mathbb{E}_{\boldsymbol{\pi}_{-i}}[Y(i|h_{t-1}) r_i(s_t, a_i) + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t-1}) \sum_{s_{t+1} \in \mathcal{S}} P(s_{t+1}|s_t, \boldsymbol{a}_{s_t}) v_i(s_{t+1}|h_t)], \tag{7}$$

[1] Note that in a Markov game, where each agent makes simultaneous decisions at each time step t, subgame perfect equilibrium (SPE) is equivalent to Nash equilibrium, since the Nash equilibrium at each state s (i.e., a subgame) is the same.

where $Pr(\boldsymbol{I}_t|h_{t-1}) = \prod_{i:I_{i,t}=1} Y(i|h_{t-1}) \prod_{j:I_{j,t}=0}(1 - Y(j|h_{t-1}))$, the probability of participation vector $\boldsymbol{I}_t$ given history h_{t-1}. The constraints in eq. (5) guarantee SPE, i.e., $(v_i(s|h) - q_i(s, a_i|h))$ is non-negative for any $i \in [N]$. Consistent with MAGAIL (Song et al. (2018)) the objective has a global minimum of zero under SPE, and $\boldsymbol{\pi}$ forms SPE if and only if $f_r(\boldsymbol{\pi}, \boldsymbol{v})$ reaches zero while being a feasible solution.

We use **AMA-RL(r)** to denote the set of policies that form SPE under reward function $\boldsymbol{r}$, and can maximize γ-discounted causal entropy of policies:

$$\textbf{AMA-RL}(\boldsymbol{r}) = \arg \min_{\boldsymbol{\pi} \in \Pi, \boldsymbol{v}} f_r(\boldsymbol{\pi}, \boldsymbol{v}) - H(\boldsymbol{\pi}), \tag{8}$$

$$\textbf{s.t. } v_i(s|h) \geq q_i(s, a_i|h) \; \forall i \in [N], s \in \mathcal{S}, a_i \in \mathcal{A}_i, \forall h \in \mathcal{H}, \tag{9}$$

where q_i is defined in eq. (7). Our objective is to define a suitable inverse operator AMAIRL in analogy to MAIRL in eq. (2). The key idea of MAIRL is to choose a reward that creates a *margin* between a set of experts and every other set of policies. However, the *constraints* in SPE optimization eq. (8) can make this challenging. To that end, we derive an equivalent Lagrangian formulation of eq. (8) to defined a margin between the expected rewards of two sets of policies to capture the "difference".

3.2 Asynchronous Multi-Agent Inverse Reinforcement Learning

The SPE constraints in eq. (9) state that no agent i can obtain a higher expected reward via 1-step temporal (TD) difference learning. We replace 1-step constraints with (t+1)-step constraints with the solution remaining the same as AMARL. The general idea is consistent with MAGAIL (Song et al. (2018)). The detailed derivation is in Appx A.1. The updated (t+1)-step constraints is

$$\hat{v}_i(s^{(0)}; \boldsymbol{\pi}, \boldsymbol{r}, \zeta) \geq Q_i^{(t)}(\{s^{(j)}, a_i^{(j)}\}_{j=0}^{t}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}), \tag{10}$$
$$\forall t \in \mathbb{N}^+, i \in [N], s^{(j)} \in \mathcal{S}, a_i^{(j)} \in \mathcal{A}_i, h_{t-1} \in \mathcal{H}.$$

By implementing the (t+1)-step formulation (eq. 10), we aim to construct the Lagrangian dual of the primal in eq. 8. Since for any policy $\boldsymbol{\pi}$, $f_r(\boldsymbol{\pi}, \hat{\boldsymbol{v}}) = 0$ given $\hat{v}_i$ defined as in Theorem 1 in Appx A.1 (proved in Lemma 1 in Appx A.2), we just focus on the constraints (eq. 10) to get the dual problem

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\pi}} L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}, \lambda) \triangleq \sum_{i=1}^{N} \sum_{h_{t-1} \in \mathcal{H}} \sum_{\tau_i \in \mathcal{T}_i^t} \lambda(\tau_i; h_{t-1})(Q_i^{(t)}(\tau_i; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) - \hat{v}_i(s^{(0)}; \boldsymbol{\pi}, \boldsymbol{r}, \zeta)), \tag{11}$$

where $\mathcal{T}_i^t$ is the set of all length-t trajectories of the form $\{s^{(j)}, a_i^{(j)}\}_{j=0}^{t}$, with $s^{(0)}$ as initial state, λ is a vector of $N \cdot |\mathcal{T}_i^{(t)}| \cdot |\mathcal{H}|$ Lagrange multipliers, and $\hat{v}_i$ is defined as in Theorem 1 in Appx A.1.

Theorem 2 illustrates that a specific λ is able to recover the difference of the sum of expected rewards between not all optimal and all optimal policies.

Theorem 2 *For any two policies $\boldsymbol{\pi}^*$ and $\boldsymbol{\pi}$, let*

$$\lambda_\pi^*(\tau_i; h_{t-1}) = \eta(s^{(0)}) Pr(h_{t-1}) \prod_{j=0}^{t-1} (\sum_{\boldsymbol{a}_{-i}^j} \boldsymbol{\pi}_{-i}^*(\boldsymbol{a}_{-i}|s^{(j)}) P(s^{(j+1)}|s^{(j)}, \boldsymbol{a}^{(j)})) \prod_{s^{(j)}: I_{i,j}=1} \pi_i(a_i^{(j)}|s^{(j)})$$

be the probability of generating the sequence τ_i using policy π_i and $\boldsymbol{\pi}_{-i}^*$ and h_{t-1}, where $Pr(h_{t-1}) = Pr(\boldsymbol{I}_0) \prod_{k=1}^{t-1} Pr(\boldsymbol{I}_k|h_{k-1})$ is the probability of history h_{t-1}. Then

$$\lim_{t \to \infty} L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}^*, \lambda_\pi^*) = \sum_{i=1}^{N} \mathbb{E}_{\pi_i} \mathbb{E}_{\boldsymbol{\pi}_{-i}^*, Y}[r_i(s^{(j)}, a_i^{(j)})] - \mathbb{E}_{\boldsymbol{\pi}^*, Y}[r_i(s^{(j)}, a_i^{(j)})]$$

where the dual function is $L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}^*, \lambda_\pi^*)$ and each multiplier can be considered as the probability of generating a trajectory of agent $i \in N$, $\tau_i \in \mathcal{T}_i^t$, and $h_{t-1} \in \mathcal{H}$.

Theorem 2 (proved in Appx A.3) provides a horizon to establish AMAIRL objective function with regularizer ψ.

$$\textbf{AMA-IRL}_{\psi}(\boldsymbol{\pi_E}) = \arg\max_{\boldsymbol{r}} -\psi(\boldsymbol{r}) + \sum_{i=1}^{N}(\mathbb{E}_{\boldsymbol{\pi_E},Y}[r_i]) - (\max_{\boldsymbol{\pi}} \sum_{i=1}^{N}(\beta H_i(\pi_i) + \mathbb{E}_{\pi_i,\boldsymbol{\pi}_{E_{-i}},Y}[r_i])), \tag{12}$$

where $H_i(\pi_i) = \mathbb{E}_{\pi_i,\boldsymbol{\pi}_{E_{-i}}}[-\log \pi_i(a|s)]$ is the discounted causal entropy for policy π_i when other agents follow $\boldsymbol{\pi}_{E_{-i}}$, and β is a hyper-parameter controlling the strength of the entropy regularization term as in GAIL (Ho & Ermon (2016)).

Corollary 2.1. *If $I = 1$ for all $i \in [N]$ then* $\textbf{AMA-IRL}_{\psi}(\boldsymbol{\pi_E}) = \textbf{MAIRL}_{\psi}(\boldsymbol{\pi_E})$; *furthermore, if $N = 1$, $\beta = 1$ then* $\textbf{AMA-IRL}_{\psi}(\boldsymbol{\pi_E}) = \textbf{IRL}_{\psi}(\pi_E)$.

3.3 Asynchronous Multi-Agent Occupancy Measure Matching

We first define the **extensive occupancy measure** in extensive Markov games:

Definition 1 *For a policy $\pi_i \in \Pi$, define its extensive occupancy measure $\rho^p_{\pi_i} : \mathcal{S} \times \mathcal{A}_i \cup \{0\} \mapsto \mathbb{R}$ as*

$$\rho^p_{\pi_i}(s,a) = \pi_i(a|s)(\eta(s)\zeta(i) + \sum_{t=1}^{\infty}\sum_{h_{t-1}} \gamma^t Pr(s_t = s|\pi_i, \boldsymbol{\pi}_{E_{-i}})Y(i|h_{t-1}))$$

if $a \in \mathcal{A}_i$, and if $a_i \in \{0\}$

$$\rho^p_{\pi_i}(s,0) = \eta(s)(1-\zeta(i)) + \sum_{t=1}^{\infty}\sum_{h_{t-1}} \gamma^t Pr(s_t = s|\pi_i, \boldsymbol{\pi}_{E_{-i}})(1 - Y(i|h_{t-1})).$$

The occupancy measure can be interpreted as the distribution of state-action pairs that an agent i encounters under the participating and nonparticipating situations. Notably, when $\eta(i) = 1$, $Y(i|h_{t-1}) = 1$ for all $t \in \{1, ..., \infty\}$, $h_{t-1} \in \mathcal{H}$, extensive occupancy measure in EMG turns to the occupancy measure defined in MAGAIL and GAIL, i.e., $\rho^p_{\pi_i} = \rho_{\pi_i}$. With the additively separable regularization ψ, for each agent i, π_{E_i} is the unique optimal response to other experts $\pi_{E_{-i}}$. Therefore we obtain the following theorem (see proof of Theorem 3 in Appendix A.4):

Theorem 3 *Assume $\psi(\boldsymbol{r}) = \sum_{i=1}^{N}\psi_i(r_i)$, ψ_i is convex for each $i \in [N]$, and that* $\textbf{AMA-RL}(r)$ has a unique solution[2] for all $\boldsymbol{r} \in \textbf{AMA-IRL}_{\psi}(\boldsymbol{\pi}_E)$, then

$$\textbf{AMA-RL} \circ \textbf{AMA-IRL}_{\psi}(\boldsymbol{\pi}_E) = \arg\min_{\boldsymbol{\pi}} \sum_{i=1}^{N}\sum_{h\in\mathcal{H}} -\beta H_i(\pi_i) + \psi_i^*(\rho^p_{\pi_i,\boldsymbol{\pi}_{E_{-i}}} - \rho^p_{\boldsymbol{\pi}_E}) \tag{13}$$

where π_i, E_{-i} denotes π_i for agent i, and $\pi_{E_{-i}}$ for other agents.

In practice, we are only able to calculate $\rho^p_{\boldsymbol{\pi}_E}$ and $\rho^p_{\boldsymbol{\pi}}$. As following MAGAIL (Song et al. (2018)), we match the occupancy measure between $\rho^p_{\boldsymbol{\pi}_E}$ and $\rho^p_{\boldsymbol{\pi}}$ rather than $\rho^p_{\boldsymbol{\pi}_E}$ and $\rho^p_{\pi_i,\boldsymbol{\pi}_{E_{-i}}}$.

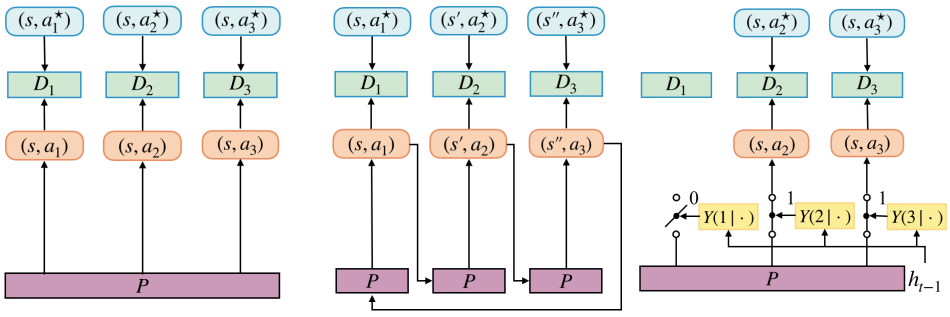
4 Practical Asynchronous Multi-Agent Imitation Learning

In this section, we propose practical algorithms for asynchronous multi-agent imitation learning, and provide common scenarios depending on player function structures.

4.1 Asynchronous Multi-Agent Generative Adversarial Imitation Learning

The selected ψ_i in Proposition 1 (in Appendix A.5) contributes to the corresponding generative adversarial model where each agent i has a generator π_{θ_i} and a discriminator, D_{w_i}. When the generator is allowed to behave, the produced behavior will receive a score from discriminator. The

[2]The set of subgame perfect equilibrium is not always convex, so we have to assume $\textbf{AMA-RL}(r)$ returns a unique solution



(a) Synchronous participation (b) Deterministic participation (c) Stochastic participation

Figure 1: AMAGAIL with three player function structures. (a) **Synchronous participation:** The player function is equal to 1, all agents take actions at all time steps. (b) **Deterministic participation:** In this example, three agents take turns to make actions with a fixed order. (c) **Stochastic participation:** Three agents all have stochastic player functions (i.e., yellow boxes), thus, each agent has certain probability to make an action w.r.t the player function given the participation history h_{t-1}; in this example, only agents #2 and #3 happen to make actions, and agent #1 does not.

generator attempts to train the agent to maximize its score and fool the discriminator. We optimize the following objective:

$$\min_{\theta} \max_{w} \mathbb{E}_{\boldsymbol{\pi}_\theta}\left[\sum_{i=1}^{N} \log D_{w_i}(s, a_i)\right] + \mathbb{E}_{\boldsymbol{\pi}_E}\left[\sum_{i=1}^{N} \log(1 - D_{w_i}(s, a_i))\right]. \tag{14}$$

In practice, the input of AMA-GAIL is the demonstrations $\mathcal{Z}$ provided by N expert agents in the same environment. The demonstrations $\mathcal{Z} = \{(s_t, \boldsymbol{a})\}_{t=0}^{T}$ are collected by sampling $s_0 \sim \eta(s)$, $\boldsymbol{I}_0 \sim \zeta, \boldsymbol{I}_t \sim Y, \boldsymbol{a} \sim \boldsymbol{\pi}^*(\cdot|s_t), s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$. The assumptions include knowledge of $N, \gamma, \mathcal{S}, \mathcal{A}$, transition P, initial state distribution and agent distribution η, ζ as well as player function Y as a black box, and no additional experts interactions with environment during training process.

In the RL process of finding each agent's policy π_{θ_i}, we follow MA-GAIL (Song et al. (2018)) to apply Multi-agent Actor-Critic with Kronecker-factors (MACK) and use the advantage function with the baseline V_ϕ for variance reduction. The summarized algorithm is presented in Algorithm 1 in Appendix B.

4.2 Player Function Structures

In EMGs, the order in which agents make decisions is determined by the player function Y. Below, we discuss three representative structures of player function Y, including synchronous participation, deterministic participation, and stochastic participation.

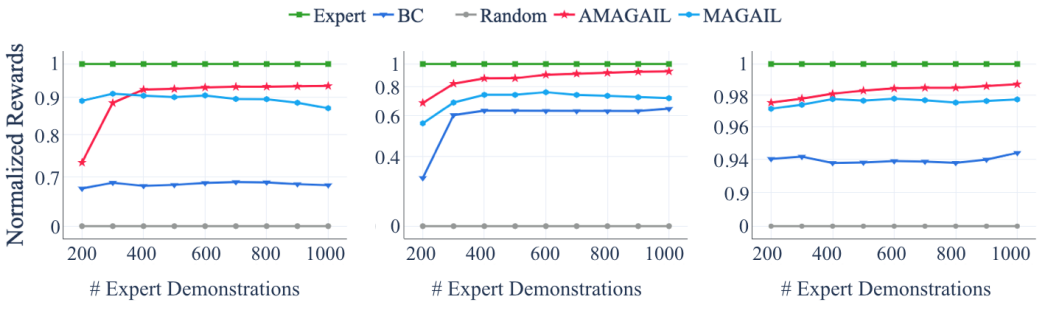
Synchronous participation. When $Y(i|h_{t-1}) = 1$ holds for all agents $i \in [N]$ at every step t (as shown in Fig. 1a), agents make simultaneous actions, and an extensive Markov game boils down to a Markov game.

Deterministic participation. When the player function $Y(i|h_{t-1})$ is deterministic for all agents $i \in [N]$, it can only take 1 or 0 at each step t. Many board games, e.g., Go, and chess, have deterministic player functions, where agents take turns to play. Fig. 1b shows an example of deterministic participation structure.

Stochastic participation. When the player function is stochastic, namely, $Y(i|h_{t-1}) \in (0, 1)$ for some agent $i \in [N]$ at certain time step t, the agent i will make an action by chance. As illustrated in Fig. 1c, three agents all have stochastic player functions at step t, and agent #1 does not take an action at step t, while agent #2 and #3 happen to take actions.

5 Experiments

We evaluate AMAGAIL with both stochastic and deterministic player function structures under cooperative and competitive games. Our implementations are based on OpenAI baselines (Dhariwal et al. (2017)) and decentralized Multi-agent generative adversarial imitation learning (Song et al.



(a) Deterministic Cooperative Navigation (b) Stochastic Cooperative Navigation (c) Deterministic Cooperative Reaching

Figure 2: Average true reward from cooperative tasks. Performance of experts and random policies are normalized to one and zero respectively. We use inverse log scale for better comparison.

(2018)). The results are collected by averaging over 5 random seeds (refer to Appx C for implementation details).

We choose to use the particle environment (Lowe et al. (2017)) as the basic game setting, and make modifications to allow general and asynchronous player functions. The modified games include: **Deterministic Cooperative Navigation**: Three agents (agent #1, #2 and #3) need to cooperate to get close to three randomly allocated landmarks through physical actions. They get high rewards if they are close to the landmarks and are penalized for any collision with each other. Ideally, each agent should cover a single distinct landmark. In this process, the agents must follow a deterministic participation order to act and make decisions - repeatedly having all three agents act in the first round, then only agent #1& #2 act in the second round, and the third round only agent #1 acts. **Stochastic Cooperative Navigation**: This game is the same with deterministic cooperative navigation except that all three agents have a stochastic player function. Each agent has 50% chance to act at each step t. **Deterministic Cooperative Reaching**: This game has three agents aiming to cooperatively get to a single landmark with minimum collision before they reach the landmark. In this game, agents follow a deterministic player function, same as that in deterministic cooperative navigation game, to make actions. **Stochastic Predator-Prey**: There are three slower cooperating agents (or we call adversaries) chasing a faster agent in an environment of two landmarks; the faster agent acts first, then the three adversaries each with a stochastic player function of 50% chance to act and try to catch the agent. The adversaries and the agent need to avoid two randomly placed landmarks. The adversaries collect rewards when touching the agent, where the agent is penalized. Note that, an agent that does not participate in a round of a game does not get a reward.

In the above simulated environments, we have the true reward functions. This enables us to both find expert policies in each game and give direct evaluations of recovered policy quality. Multi-agent ACKTR (Wu et al. (2017); Song et al. (2018)) is used to train the experts and generate expert demonstrations. Evaluation on quality of recovered policies is done by generating a set of trajectories and calculate agents' average true reward. We compare our AMAGAIL with two baselines - behavior cloning (BC) (Pomerleau (1991)) and decentralized Multi-agent generative adversarial imitation learning (MAGAIL) (Song et al. (2018)). Behavior cloning (BC) utilizes the maximum likelihood estimation for each agent independently to approach their policies. Decentralized multi-agent generative adversarial imitation learning (MAGAIL) treats each agent with a unique discriminator working as the agent's reward signal and a unique generator as the agent's policy. It follows the maximum entropy principle to make an agent's occupancy measures of state-action pairs to match those from demonstrations.

5.1 Performances with Deterministic and Stochastic Play Functions

We use the *deterministic cooperative navigation* (with a deterministic player function), *stochastic cooperative navigation* (with a stochastic player function) and *deterministic cooperative reaching* games to compare the performance of AMAGAIL under different types of player functions with other state-of-the-art approaches. Figure 2 shows the normalized rewards, when running policies learned by BC, MAGAIL and AMAGAIL, respectively.

When there is only a small amount of expert demonstrations, the normalized rewards of all three methods increase as demonstrations increase. After a sufficient amount of demonstrations are collected, i.e., more than 400, AMAGAIL has higher rewards than BC and MAGAIL. This happens

Table 1: Average agent rewards in stochastic predator-prey. We compare behavior cloning (BC) and multi-agent GAIL (MAGAIL) methods. Best marked in bold (high vs. low rewards preferable depending on the agent vs. adversary role).

Task	Stochastic Predator-Prey				
Agent	Behavior Cloning			MAGAIL	AMAGAIL
Adversary	BC	MAGAIL	AMAGAIL	Behavior Cloning	
Rewards	-5.0 ± 10.8	-9.0 ± 13.1	$\mathbf{-14.0 \pm 19.4}$	-3.6 ± 8.5	$\mathbf{-2.1 \pm 6.9}$

because at certain time steps there are non-participating agents (based on the player function), however, BC and MAGAIL would consider the non-participation as an action the agent can take, which is in fact governed by the environment. On the other hand, AMAGAIL can model and characterize such no participation events correctly, thus more accurately learn the expert policies. In the stochastic cooperative navigation game, AMAGAIL performs consistently better than MAGAIL and BC. However, in the deterministic cooperative navigation game, with demonstration number of 200, AMAGAIL does not perform as well as MAGAIL. This is because of the game setting, namely, two players who are actively searching for landmarks are sufficient to gain a high reward in the deterministic cooperative navigation game. Therefore, the last agent, i.e., player #3, does not have any motivation to promote its own reward, since the rewards are shared among all agents, based on the cooperative setting. This contributes to good performance of MAGAIL with limited 200 demonstrations. While, for AMAGAIL, player #3 causes a problem, because of player #3's $\frac{2}{3}$ absence rate, AMAGAIL does not have enough state-action pairs to mimic player #3 and leads to a systematic drawback. This problem changes when we adjust the game setup from 3 landmarks to 1 landmark, i.e., all agents need to act actively to reach the landmark. This is captured in the deterministic cooperative reaching game. In this scenario, an inactive player will lower down the overall reward. As shown in Fig 2, AMAGAIL outperforms BC and MAGAIL consistently, even with a small amount of demonstration data.

5.2 Performance with Complex Player Interaction

We use the *stochastic predator-prey* game to show AMAGAIL's performance on recovering the complicated player interactions. Since there are two competing sides in this game, we cannot directly compare each methods' performance via expected reward. Therefore, we use the Song et al. (2018)'s evaluation paradigm and compare with baselines by letting (agents trained by) BC play against (adversaries trained by) other methods, and vice versa. From table 1, AMAGAIL performs better than MAGAIL and BC. More details for all the particle environments are in the Appx C.

6 Related Work and Conclusion

Imitation learning (IL) aims to learn a policy from expert demonstrations, which has been extensively studied in the literature for single agent scenarios (Finn et al. (2016); Ho & Ermon (2016)). Behavioral cloning (BC) uses the observed demonstrations to directly learn a policy (Pomerleau (1991); Torabi et al. (2018)). Apprenticeship learning and inverse reinforcement learning (IRL) ((Ng et al. (2000); Syed & Schapire (2008); Ziebart et al. (2008; 2010); Boularias et al. (2011))) seek for recovering the underlying reward based on expert trajectories in order to further learn a good policy via reinforcement learning. The assumption is that expert trajectories generated by the optimal policy maximize the unknown reward. Generative adversarial imitation learning (GAIL) and conditional GAIL (cGAIL) incorporate maximum casual entropy IRL (Ziebart et al. (2010)) and the generative adversarial networks (Goodfellow et al. (2014)) to simultaneously learn non-linear policy and reward functions (Ho & Ermon (2016); Zhang et al. (2019); Baram et al. (2017)). A few recent studies on multi-agent imitation learning, such as MAGAIL (Song et al. (2018) and MAAIRL (Yu et al. (2019)), model the interactions among agents as Morkov games, where all agents make simultaneous actions at each step t. These works fail to characterize a more general and practical interaction scenario, i.e., extensive-form Markov games, (Fudenberg & Levine (1983)), where agents make asynchronous decisions over steps. In this paper, we make the first attempt to propose an asynchronous multi-agent generative adversarial imitation learning (AMAGAIL) framework, which models the asynchronous decision-making process as an extensive Markov game and develops a player function to capture the participation dynamics of agents. Experimental results demonstrate that our proposed AMAGAIL can accurately learn the experts' policies from their asynchronous trajectory data, comparing to state-of-the-art baselines.

REFERENCES

Samson Abramsky and Viktor Winschel. Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *Mathematical Structures in Computer Science*, 27(5):751–761, 2017.

Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 390–399. JMLR. org, 2017.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 182–189, 2011.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. *GitHub, https://github.com/openai/baselines, 2017*, 2017.

Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pp. 49–58, 2016.

Drew Fudenberg and David Levine. Subgame-perfect equilibria of finite-and infinite-horizon games. *Journal of Economic Theory*, 31(2):251–268, 1983.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.

Bjorn Knutsson, Honghui Lu, Wei Xu, and Bryan Hopkins. Peer-to-peer support for massively multiplayer games. In *IEEE INFOCOM 2004*, volume 1. IEEE, 2004.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Michael L Littman and Csaba Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, volume 96, pp. 310–318, 1996.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417, 2015.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Reinhard Selten. Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. *Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics*, (H. 2):301–324, 1965.

Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 7461–7472, 2018.

Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pp. 1449–1456, 2008.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5279–5288, 2017.

Zibo Xu. Convergence of best-response dynamics in extensive-form games. *Journal of Economic Theory*, 162:21–54, 2016.

Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. *arXiv preprint arXiv:1907.13220*, 2019.

Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. Unveiling taxi drivers strategies via cgail-conditional generative adversarial imitation learning. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. *International Conference on Machine Learning (ICML)*, 2010.

A APPENDIX A. PROOFS

A.1 TIME DIFFERENCE LEARNING

Theorem 1. *For a certain policy $\boldsymbol{\pi}$ and reward $\boldsymbol{r}$, let $\hat{v}_i(s^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ be the unique solution to the Bellman equation:*

$$\hat{v}_i(s^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) = \mathbb{E}_{\boldsymbol{\pi}}\Big[Y(i|h_{t-1}) r_i(s^{(t)}, a_i^{(t)}) + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t-1}) \sum_{s^{(t+1)} \in \mathcal{S}} P(s^{(t+1)}|s^{(t)}, \boldsymbol{a}^{(t)}) v_i(s^{(t+1)})\Big],$$
$$t \in \mathbb{N}^+, \forall s^{(t)} \in \mathcal{S}, h_{t-1} \in \mathcal{H}.$$

Denote $\hat{q}_i^{(t)}(\{s^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, s^{(t)}, a_i^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ as the discounted expected return for the i-th agent conditioned on visiting the trajectory $\{s^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, s^{(t)}$ in the first $t-1$ steps and choosing action $a_i^{(t)}$ at the t-th step, when other agents using policy $\boldsymbol{\pi}_{-i}$:

$$\begin{aligned}
&\hat{q}_i^{(t)}(\{s^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, s^{(t)}, a_i^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \\
=&\sum_{j=0}^{t-1} \gamma^j r_i(s^{(j)}, a_i^{(j)}) I_{i,j} \\
&+ \gamma^t \mathbb{E}_{\boldsymbol{\pi}_{-i}}[Y(i|h_{t-1}) r_i(s^{(t)}, a_i^{(t)}) + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t-1}) \sum_{s^{(t+1)} \in \mathcal{S}} P(s^{(t+1)}|s^{(t)}, \boldsymbol{a}^{(t)}) v_i(s^{(t+1)}; \boldsymbol{\pi}, \boldsymbol{r}, h_t)].
\end{aligned}$$

Then $\boldsymbol{\pi}$ is subgame perfect equilibrium if and only if:

$$\begin{aligned}
\hat{v}_i(s^{(0)}; \boldsymbol{\pi}, \boldsymbol{r}, \zeta) \geq& \mathbb{E}_{\boldsymbol{\pi}_{-i}}[\hat{q}_i^{(t)}(\{s^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, s^{(t)}, a_i^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})] \\
\triangleq& Q_i^{(t)}(\{s^{(j)}, a_i^{(j)}\}_{j=0}^{t}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \\
&\forall t \in \mathbb{N}^+, i \in [N], s^{(j)} \in \mathcal{S}, a_i^{(j)} \in \mathcal{A}_i, h_{t-1} \in \mathcal{H}.
\end{aligned} \tag{15}$$

Theorem 1 illustrates that if we replace the 1-step constraints with $(t+1)$-step constraints, we still get the same solution as AMA-RL($\boldsymbol{r}$) in terms of a subgame perfect equilibrium solution.

A.2 EXISTENCE AND EQUIVALENCE OF $\mathbf{v}$ AND NASH EQUILIBRIUM

Lemma 1 *By definition of $\hat{v}_i(s^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ in Theorem 1 and $\hat{q}_i(s^{(t)}, a_i; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ in eq. 7. Then for any $\boldsymbol{\pi}$, $f_r(\boldsymbol{\pi}, \hat{\boldsymbol{v}}) = 0$. Furthermore, $\boldsymbol{\pi}$ is subgame perfect equilibrium under $\boldsymbol{r}$ if and only if $\hat{v}_i(s; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \geq \hat{q}_i(s, a_i; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ for all $i \in [N]$, $s \in \mathcal{S}$, $a_i \in \mathcal{A}_i$ and $h_{t-1} \in \mathcal{H}$.*

Proof We have

$$\begin{aligned}
&\hat{v}_i(s^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \\
=&\mathbb{E}_{\boldsymbol{\pi}}\Big[Y(i|h_{t-1}) r_i(s^{(t)}, a_i^{(t)}) + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t-1}) \sum_{s^{(t+1)} \in \mathcal{S}} P(s^{(t+1)}|s^{(t)}, \boldsymbol{a}^{(t)}) v_i(s^{(t+1)})\Big] \\
=&\mathbb{E}_{\pi_i} \mathbb{E}_{\boldsymbol{\pi}_{-i}}\Big[Y(i|h_{t-1}) r_i(s^{(t)}, a_i^{(t)}) + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t=1}) \sum_{s^{(t+1)} \in \mathcal{S}} P(s^{(t+1)}|s^{(t)}, \boldsymbol{a}^{(t)}) v_i(s^{(t+1)})\Big] \\
=&\mathbb{E}_{\pi_i}[q_i(s^{(t)}, a_i^{(t)}; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})].
\end{aligned}$$

which utilizes the fact that a_i and $\mathbf{a}_{-i}$ are independent at s. Therefore, we can easily get $f_r(\boldsymbol{\pi}, \hat{\boldsymbol{v}}) = 0$.

If $\boldsymbol{\pi}$ is a subgame perfect equilibrium, and existing one or more of the constrains does not hold, so agent i can receive a strictly higher expected reward for rest of the states, which is against the subgame perfect equilibrium assumption.

If the constraints hold, i.e., for all i and (s, a_i), $\hat{v}_i(s; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \geq \hat{q}_i(s, a_i; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})$ then

$$\hat{v}_i(s; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}) \geq \mathbb{E}_{\pi_i}[\hat{q}_i(s, a_i; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1})] = \hat{v}_i(s; \boldsymbol{\pi}, \boldsymbol{r}, h_{t-1}).$$

Value iteration, thus, over $\hat{v}_i(s;\boldsymbol{\pi},\boldsymbol{r},h_{t-1})$ converges. If one can find another policy $\boldsymbol{\pi}'$ so that $\hat{v}_i(s;\boldsymbol{\pi},\boldsymbol{r},h_{t-1}) < \mathbb{E}_{\pi_i}[\hat{q}_i(s,a_i;\boldsymbol{\pi},\boldsymbol{r},h_{t-1})]$, then at least one violation exists in the constraints since π_i' is a convex combination over action a_i. Therefore, for any policy π_i' and action a_i for any agent i, $\mathbb{E}_{\pi_i}[\hat{q}_i(s,a_i;\boldsymbol{\pi},\boldsymbol{r},h_{t-1})] \geq \mathbb{E}_{\pi_i'}[\hat{q}_i(s,a_i;\boldsymbol{\pi},\boldsymbol{r},h_{t-1})]$ always hold, so π_i is the optimal reply to $\boldsymbol{\pi}_{-i}$, and $\boldsymbol{\pi}$ constitutes a subgame perfect equilibrium once it repeats this argument for all agents. Notably, by assuming $f_r(\boldsymbol{\pi},\boldsymbol{v}) = 0$ for some $\boldsymbol{v}$; if $\boldsymbol{v}$ satisfies the assumptions, then $\boldsymbol{v} = \hat{\boldsymbol{v}}$. ■

A.3 Proof to Theorem 2

Proof We use $Q^*, \hat{q}^*, \hat{v}^*$ to denote the $Q, \hat{q}$ and $\hat{v}$ quantities defined for policy $\boldsymbol{\pi}^*$. For the two terms in $L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}^*, \lambda_\pi^*)$ we have:

$$L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}^*,\lambda_\pi^*) \triangleq \sum_{i=1}^{N}\sum_{h_{t-1}\in\mathcal{H}}\sum_{\tau_i\in\mathcal{T}_i^t}\lambda^*(\tau_i;h_{t-1})(Q_i^*(\tau_i;\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1}) - \hat{v}_i^*(s^{(0)};\boldsymbol{\pi}^*,\boldsymbol{r},\zeta))$$

For agent i, τ_i and h_{t-1} we have,

$$\lambda_\pi^*(\tau_i;h_{t-1})\cdot Q_i^*(\tau_i;\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1}) = Pr(\tau_i;h_{t-1})\cdot Q_i^*(\tau_i;\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1}).$$

For any agent i, we note that

$$\begin{aligned}
&\sum_{h_{t-1}\in\mathcal{H}}\sum_{\tau_i\in\mathcal{T}_i}\lambda_\pi^*(\tau_i;h_{t-1})\cdot Q_i^*(\tau_i;\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1})\\
=&\mathbb{E}_{\pi_i}\mathbb{E}_{\boldsymbol{\pi}^*_{-i}}[\sum_{j=0}^{t-1}\gamma^j r_i(s^{(j)},a_i^{(j)})I_{i,j} + \gamma^t\mathbb{E}_{\boldsymbol{\pi}^*_{-i}}[Y(i|h_{t-1})r_i(s^{(t)},a_i^{(t)})+\\
&\gamma\sum_{\boldsymbol{I}_t}Pr(\boldsymbol{I}_t|h_{t-1})\sum_{s^{(t+1)}\in\mathcal{S}}P(s^{(t+1)}|s^{(t)},\boldsymbol{a}^{(t)})v_i(s^{(t+1)};\boldsymbol{\pi}^*,\boldsymbol{r},h_t)]]\\
=&\mathbb{E}_{\pi_i}\mathbb{E}_{\boldsymbol{\pi}^*_{-i},Y}[\sum_{j=0}^{t-1}\gamma^j r_i(s^{(j)},a_i^{(j)})I_{i,j} + \gamma^t\hat{q}_i^*(\{s^{(j)},\boldsymbol{a}^{(j)}\}_{j=0}^{t-1},s^{(t)},a_i^{(t)};\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1})]
\end{aligned}$$

which is using π_i for agent i for the first t steps and using π_i^* for the remaining steps, whereas other agents follow $\boldsymbol{\pi}^*_{-i}$. As $t\to\infty$, this converges to $\mathbb{E}_{\pi_i}\mathbb{E}_{\boldsymbol{\pi}^*_{-i},Y}[r_i(s^{(j)},a_i^{(j)})]$ as $\gamma^t\to 0$ and $\hat{q}_i^*(\{s^{(j)},\boldsymbol{a}^{(j)}\}_{j=0}^{t-1},s^{(t)},a_i^{(t)};\boldsymbol{\pi}^*,\boldsymbol{r},h_{t-1})$ is bounded. Moreover, for $\hat{v}_i^*(s^{(0)};\boldsymbol{\pi}^*,\boldsymbol{r},\zeta)$ we have

$$\sum_{h_{t-1}\in\mathcal{H}}\sum_{\tau_i\in\mathcal{T}_i^t}\lambda^*(\tau_i;h_{t-1})\hat{v}_i^*(s^{(0)};\boldsymbol{\pi}^*,\boldsymbol{r},\zeta) = \mathbb{E}_{s^{(0)}\sim\eta}[\hat{v}_i^*(s^{(0)};\boldsymbol{\pi}^*,\boldsymbol{r},\zeta)] = \mathbb{E}_{\boldsymbol{\pi}^*,Y}[r_i(s^{(j)},a_i^{(j)})].$$

Combining the two we have,

$$\lim_{t\to\infty}L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}^*,\lambda_\pi^*) = \sum_{i=1}^{N}\mathbb{E}_{\pi_i}\mathbb{E}_{\boldsymbol{\pi}^*_{-i},Y}[r_i(s^{(j)},a_i^{(j)})] - \mathbb{E}_{\boldsymbol{\pi}^*,Y}[r_i(s^{(j)},a_i^{(j)})]$$

which describes the differences in expected rewards. ■

A.4 Proof to Theorem 3

Proof For a single agent i where other agents have policy $\boldsymbol{\pi}_{E_{-i}}$, we give the following analysis and definition.

For a policy $\pi_i\in\Pi$, its occupancy measure defined in Def. 3.3 allows us to write $E_{\pi_i,Y}[r_i(s,a)] = \sum_{s,a}\rho_{\pi_i}^p(s,a)r_i(s,a)$ for any reward function r_i. A basic result is that the set of valid occupancy measures $\mathcal{D}_i \triangleq \{\rho_{\pi_i}^p : \pi_i\in\Pi\}$ can be written as a feasible set of affine constraints:

$$\begin{aligned}
&\mathcal{D}_i\\
=&\{\rho_{\pi_i}^p : \rho_{\pi_i}^p \geq 0 \text{ and}\\
&\sum_{a\in\mathcal{A}_i\cup\{\phi\}}\rho_{\pi_i}^p(s,a) = \eta(s) + \gamma\sum_{s',\boldsymbol{a}}P(s|s',\boldsymbol{a})\boldsymbol{\pi}_{E_{-i}}(\boldsymbol{a}_{-i}|s')\rho_{\pi_i}^p(s',a_i)\ \forall s\in\mathcal{S}\}.
\end{aligned}$$

Therefore, the proof of MAA-RL∘MAA-IRL can be derived in a similar fashion with GAIL (Ho & Ermon (2016)) and MAGAIL (Song et al. (2018)). ■

A.5 PROPOSITION 1

Proposition 1: *If* $\beta = 0$ *and* $\psi(\boldsymbol{r}) = \sum_{i=1}^{N} \psi_i(r_i)$ *where* $\psi_i(r_i) = \mathbb{E}_{\pi_E,Y}[g(r_i)]$ *if* $r_i > 0; +\infty$ *otherwise, and*

$$g(x) = \begin{cases} -x - \log(1 - e^x) & if\ r_i > 0 \\ +\infty & o.w. \end{cases}$$

then

$$\arg\min_{\pi} \sum_{i=1}^{N} \psi_i^*(\rho^p_{\pi_i,\boldsymbol{\pi}_{E-i}} - \rho^p_{\boldsymbol{\pi}_E}) = \arg\min_{\pi} \sum_{i=1}^{N} \psi_i^*(\rho^p_{\pi_i,\boldsymbol{\pi}_{-i}} - \rho^p_{\boldsymbol{\pi}_E}) = \boldsymbol{\pi}_E.$$

Theorem 3 and Proposition 1 discuss the differences from the single agent scenario similar in Song et al. (2018). On the one hand, in Theorem 3 we make the assumption that **AMA-RL**($\boldsymbol{r}$) has a unique solution, which is always true in the single agent case due to convexity of the space of the optimal policies. On the other hand, in Proposition 1 we remove the entropy regularizer because here the causal entropy for π_i may depend on the policies of the other agents, so the entropy regularizer on two sides are not the same quantity. Specifically, the entropy for the left hand side conditions on $\boldsymbol{\pi}_{E_{-i}}$ and the entropy for the right hand side conditions on $\boldsymbol{\pi}_{-i}$ (which would disappear in the single-agent case).

B APPENDIX B. ALGORITHM

Algorithm 1 Asynchronous Multi-Agent GAIL (AMAGAIL)

Input: Initial parameters of policies, discriminators and value (baseline) estimators, θ_0, w_0, ϕ_0; state-action pair demonstrations $\mathcal{Z} = \{(s_j, \boldsymbol{a})\}_{j=0}^{M}$; batch size B; extensive Markov game as a block box $(N, \mathcal{S}, \mathcal{A}, P, \eta, \zeta, Y, r, \gamma)$.
Output: Learned policies π_θ and reward functions D_w.

for u = 0,1,2,... **do**
  Generate state-action pairs of batch size B from π_u through the process: $s_0 \sim \eta(s)$, $\boldsymbol{I}_0 \sim \zeta, \boldsymbol{I}_t \sim Y, \boldsymbol{a} \sim \boldsymbol{\pi}^*(\cdot|s_t), s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$; denote the generated state-action pairs as $\mathcal{X}$.
  Sample state-action pairs from $\mathcal{Z}$ of batch size B; denote the demonstrated state-action pairs as $\mathcal{X}_E$.
  for i = 1,...,N **do**
    Update w_i to increase the objective
    $\mathbb{E}_{\mathcal{X}}[\log D_{w_i}(s, a_i)] + \mathbb{E}_{\mathcal{X}_E}[\log(1 - D_{w_i}(s, a_i))]$
  end for
  for i = 1,...,N **do**
    Compute value estimate V^* and advantage estimate A_i for $(s, a) \in \mathcal{X}$.
    Update ϕ_i to decrease the objective
    $\mathbb{E}_{\mathcal{X}}[(V_\phi(s, a_{-i}) - V^*(s, a_{-i}))^2]$
    Update θ_i by policy gradient with the setting step sizes:
    $\mathbb{E}_{\mathcal{X}}[\nabla_{\theta_i} \pi_{\theta_i}(a_i|s_i) A_i(s, a)]$
  end for
end for
Return learned policies π_θ and reward functions D_w.

C APPENDIX C. EXPERIMENT DETAILS

C.1 HYPERPARAMETERS

For the particle environment, we follow the setting of MAGAIL (Song et al. (2018)) to use two layer MLPs with 128 cells in each layer for the policy generator network, value network and the

Table 2: Performance in stochastic cooperative navigation.

#Expert Episodes	200	400	600	800	1000
Expert			-12.5 ± 6.0		
Random			-61.6 ± 20.0		
Behavior Cloning	-45.8 ± 12.0	-30.7 ± 9.9	-30.8 ± 10.4	-30.9 ± 10.5	-30.1 ± 9.8
MAGAIL	-34.4 ± 13.5	-25.4 ± 8.9	-24.5 ± 8.3	-25.8 ± 8.4	-26.6 ± 8.4
AMAGAIL	-26.1 ± 8.8	-19.0 ± 8.5	-17.5 ± 8.2	-16.6 ± 7.9	-16.0 ± 7.3

Table 3: Performance in deterministic cooperative navigation.

#Expert Episodes	200	400	600	800	1000
Expert			-13.8 ± 6.8		
Random			-61.6 ± 16.5		
Behavior Cloning	-29.3 ± 11.0	-29.0 ± 10.8	-28.8 ± 10.8	-28.7 ± 10.6	-29.0 ± 10.8
MAGAIL	-19.0 ± 7.6	-18.3 ± 7.5	-18.3 ± 7.3	-18.8 ± 7.3	-20.0 ± 8.0
AMAGAIL	-26.6 ± 7.8	-17.5 ± 7.0	-17.2 ± 6.9	-17.1 ± 6.9	-17.0 ± 7.0

discriminator. We use a batch size of 1,000. The policy is trained using K-FAC optimizer (Martens & Grosse (2015)) with parameters the same in Song et al. (2018).

C.2 Detailed Results

Below we list the exact performance (average over agents) in tables 2, 3 and 4. The means and standard deviations are computed over 1,000 epidodes. The policies in the cooperative tasks are trained with varying number of expert demonstrations. The policies in the competitive task are trained on a dataset with 1,000 expert trajectories.

The environment for each episode is drastically different (e.g. location of landmarks are randomly sampled), which leads to the seemingly high standard deviation across episodes.

Table 4: Performance in deterministic cooperative reaching.

#Expert Episodes	200	400	600	800	1000
Expert			-78.1 ± 16.4		
Random			-140.7 ± 30.3		
Behavior Cloning	-81.8 ± 17.2	-82.0 ± 17.3	-81.9 ± 17.0	-82.0 ± 17.5	-81.6 ± 17.1
MAGAIL	-79.9 ± 17.2	-79.5 ± 16.9	-79.5 ± 16.8	-79.6 ± 16.8	-79.5 ± 17.1
AMAGAIL	-79.6 ± 16.7	-79.3 ± 17.2	-79.1 ± 16.9	-79.0 ± 16.8	-79.0 ± 16.90