Imitation Learning From Inconcurrent Multi-Agent Interactions

Xin Zhang*,[1], Weixiao Huang*,[1], Yanhua Li[1], Renjie Liao[2], Ziming Zhang[1]

***Abstract*— Multi-agent imitation learning (MA-IL) aims to inversely learn policies for all agents using demonstrations collected from an expert group. However, this problem has only been studied in the setting of Markov games (MGs) *allowing participants for concurrent actions*, and do not work for general MGs, with agents inconcurrently making decisions in different turns. In this work, we propose *i*MA-IL, a novel multi-agent imitation learning framework for general (inconcurrent) Markov games. The learned policies are proven to guarantee subgame perfect equilibrium (SPE), a stronger equilibrium than Nash equilibrium (NE). The experiment results demonstrate that compared to state-of-the-art baselines, our *i*MA-IL model can better infer the policy of each expert agent using their demonstration data collected from inconcurrent decision-making scenarios.**

Inconcurrent MG	Our *i*MA-IL Model	MAGAIL & MAAIRL
Actions	$a_1, a_2, \cdots$	$a_1, a_2, \cdots$, **No**
Participation	Yes, **No**	Yes
Transition	$s_1, s_2, \cdots$	$s_1, s_2, \cdots$

By agents　By environment

Fig. 1: Decision-making process in an inconcurrent MG.

I. Introduction

Reinforcement learning (RL) requires a predefined reward function or reinforcement signal [20], [13], [21], [24] as the objective for the reinforcement learner to efficiently explore and learn a good policy. However, it is hard to manually specify an appropriate and informative reward function in a complex learning environment [9], [3]. Moreover, in scenarios with multiple agents interacting with each other using shared or competing rewards, the reward specification problem becomes more challenging.

Imitation Learning (IL) or Learning from Demonstrations (LfD) [1], [4], [11] aims to tackle the reward specification problem by directly learning from expert demonstrations. Especially, inverse reinforcement learning (IRL) [17], [31], [30], [11], [29] recovers a reward function from expert demonstrations, with an assumption that the demonstrator follows an (near-)optimal policy when generating the data. Recent works [25], [28] have investigated a more general scenario with demonstration data from multiple interacting agents. Such interactions are modeled by extending Markov decision processes on individual agents to multi-agent Markov games (MGs) [15]. However, these works only work for *concurrent* MGs, with all agents making simultaneous decisions in each turn, and do not work for general MGs, allowing agents to make inconcurrent decisions in different turns, which is common in many real world scenarios. For example, in multiplayer games [12], such as Go game, and many card games, players take turns to play, thus influence each other's decision. The order in which agents make decisions has a significant impact on the game equilibrium. Fig. 1 illustrates the decision-making process in an inconcurrent MG, where the environment not only governs the state transition, but also agents' participation. As a result, directly applying concurrent MG based approaches, i.e., MAGAIL [25] and MAAIRL [28] would implicitly model the agent participation as an action the agent can choose, thus leads to learner policies with poor performances.

In this paper, we propose a novel framework, inconcurrent multi-agent imitation learning (*i*MA-IL): A group of experts provide demonstration data when playing a Markov game (MG) with an inconcurrent decision-making process, and *i*MA-IL inversely learns each expert's decision-making policy. We introduce a *player function* governed by the environment to capture the participation order and dependency of agents when making decisions. The participation order could be deterministic (i.e., agents take turns to act) or stochastic (i.e., agents need to take actions by chance). With the general MG model, our framework generalizes MAGAIL [25] from the concurrent Markov games to (inconcurrent) Markov games, and the learned expert policies are proven to guarantee subgame perfect equilibrium (SPE) [8], a stronger equilibrium than the Nash equilibrium (NE) (guaranteed in MAGAIL [25]). The experiment results show that compared to GAIL [11] and MAGAIL [25], our *i*MA-IL can better infer the policy of each expert agent using their demonstration collected from inconcurrent decision-making scenarios.

II. Preliminaries

A. Markov Games

Markov games (MGs) [14] are the cases of N interacting agents, with each agent making a sequence of decisions whose strategies only depend on the current state. A *Markov game*[1] is denoted as a tuple $(N, \mathcal{S}, \mathcal{A}, Y, \zeta, P, \eta, \boldsymbol{r}, \gamma)$ 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 the indicator variable $I_{i,t} = 1$, an agent i is allowed to take an action; otherwise, $I_{i,t} = 0$, the agent

*Authors are of equal contribution.

[1]Xin Zhang, Weixiao Huang, Yanhua Li and Ziming Zhang are with Worcester Polytechnic Institute (WPI), USA, `xzhang17,whuang2,yli15,zzhang15@wpi.edu`.

[2]Renjie Liao is with the University of Toronto and Vector Institute, Toronto, ON M5G 1M1, Canada, `rjliao@cs.toronto.edu`.

[1]Note that Markov games defined in MAGAIL ([25]) assume concurrent participation.We follow the rich literature [5], [10] to define Markov games, which allow both concurrent and inconcurrent decision-making processes.

i does not take an action. 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 being allowed to make an action at a step t, given the participation history h_{t-1}, namely, $Y(i|h_{t-1})$. ζ defines the participation probability of an agent at the initial time step $\zeta : [N] \mapsto [0, 1]$. 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. The initial states are determined by a distribution $\eta : \mathcal{S} \mapsto [0, 1]$. 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 $r_i : \mathcal{S} \times \mathcal{A}_i \mapsto \mathbb{R}^2$. 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 $\pi_i : \mathcal{S} \times \mathcal{A}_i \mapsto [0, 1]$. We denote expert policy of an agent i as π_{E_i}, and its learner policy as π_i. 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 to denote an expectation with respect to the trajectories it generates. For example, $\mathbb{E}_{\boldsymbol{\pi},Y}[r_i(s, a_i)] \triangleq \mathbb{E}_{s_t, \boldsymbol{a} \sim \boldsymbol{\pi}, \boldsymbol{I}_t \sim Y}[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_i)]$, denotes the following sample process as $s_0 \sim \eta$, $\boldsymbol{I}_0 \sim \zeta$, $\boldsymbol{I}_t \sim Y$, $\boldsymbol{a} \sim \boldsymbol{\pi}(\cdot|s_t)$, $s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$, for $\forall i \in [N]$. Clearly, when player function $Y(i|h_{t-1}) = 1$ holds for all agents i's at any time step t, it is a MG with concurrent participation of all agents as is introduced [14], [25]. To distinguish our work from MAGAIL and be consistent with the literature [5] and [10], we refer the game setting discussed in MAGAIL as concurrent Markov games (cMGs), and that of our work as Markov games (MGs).

B. Subgame Perfect Equilibrium for Markov Games

In concurrent Markov games (cMGs), 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 a solution concept to resolve the dependency across agents, where no agents can achieve a higher expected reward by unilaterally changing its own policy [25]. However, Markov games (MGs) allowing inconcurrent decisions (e.g., turn-based games such as the Go game) views a Nash equilibrium a weaker solution [23]. An inconcurrent MG is modeled as a tree: each non-terminal node represents a state in the game, each leaf node represents an outcome, and a node with its following nodes forms a subgame [23]. This model reflects the action sequential dependency in inconcurrent MGs. In such a game setting, the Nash equilibrium focuses on participants' final outcomes (i.e., root-node Nash) and overlooks the action sequential dependency. Therefore, it cannot rule out the "non-credible threats", i.e., outcomes that will not be reached by rational players [23]. Instead, the subgame perfect equilibrium (SPE) traverses through the game tree and finds Nash equilibrium at each node (subgame). This solution set of every node (subgame) Nash forms an SPE. It has been shown that in a finite or infinite extensive-form game with either discrete or continuous time, best-response strategies converge to SPE, rather than NE [22], [2], [27].

III. Inconcurrent Multi-Agent Imitation Learning

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

A. Inconcurrent Multi-Agent Reinforcement Learning

In a Markov game (MG), the Nash equilibrium needs to be guaranteed at each state (subgame) $s \in \mathcal{S}$[3], 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 ([7], Theorem 3.7.2)

$$\begin{aligned} \min_{\boldsymbol{\pi}, \boldsymbol{v}} f_r(\boldsymbol{\pi}, \boldsymbol{v}) &= \sum_{i=1}^{N} \sum_{s \in \mathcal{S}, h \in \mathcal{H}} v_i(s|h) - \mathbb{E}_{a_i \sim \pi_i(\cdot|s)}[q_i(s, a_i|h)] \\ \textbf{s.t. } & v_i(s|h) \geq q_i(s, a_i|h) \ \forall i \in [N], s \in \mathcal{S}, a_i \in \mathcal{A}_i, h \in \mathcal{H}, \\ & \boldsymbol{v} \triangleq [v_1; \cdots; v_N]. \end{aligned} \tag{1}$$

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

$$\begin{aligned} & q_i(s_t, a_i|h_{t-1}) = \mathbb{E}_{\boldsymbol{\pi}_{-i}, Y}[Y(i|h_{t-1}) r_i(s_t, a_i) \\ & + \gamma \sum_{\boldsymbol{I}_t \in \mathcal{I}} Pr(\boldsymbol{I}_t|h_{t-1}) \sum_{s_{t+1} \in \mathcal{S}} P(s_{t+1}|s_t, \boldsymbol{a}_{s_t}) v_i(s_{t+1}|h_t)], \end{aligned} \tag{2}$$

where $Pr(\boldsymbol{I}_t|h_{t-1}) = \prod_{i: I_{i,t}=1} Y(i|h_{t-1}) \prod_{j: I_{j,t}=0} (1 - Y(j|h_{t-1}))$ is the probability of participation vector $\boldsymbol{I}_t$ given history h_{t-1}. The constraints in eq. (1) guarantee an SPE, i.e., $(v_i(s|h) - q_i(s, a_i|h))$ is non-negative for any $i \in [N]$. Consistent with MAGAIL [25] 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.

[2]Because of the inconcurrent setting, the rewards only depend on agents' own actions.

[3]Note that in a concurrent 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.

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

$$i\textbf{MA-RL}(\boldsymbol{r}) = \arg\min_{\boldsymbol{\pi}\in\Pi,\boldsymbol{v}} f_r(\boldsymbol{\pi},\boldsymbol{v}) - H(\boldsymbol{\pi}), \tag{3}$$

$$\textbf{s.t. } v_i(s|h) \geq q_i(s,a_i|h)\ \forall i\in[N], s\in\mathcal{S}, a_i\in\mathcal{A}_i, \forall h\in\mathcal{H}, \tag{4}$$

where q_i is defined in eq. (2). Our objective is to define a suitable inverse operator *i*MA-IRL. 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. (3) can make this challenging. To that end, we derive an equivalent Lagrangian formulation of eq. (3) to defined a margin between the expected rewards of two sets of policies to capture the "difference".

B. Inconcurrent Multi-Agent Inverse Reinforcement Learning

The SPE constraints in eq. (4) 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 *i*MA-RL. The general idea is consistent with MAGAIL [25]. The updated (t+1)-step constraints are

$$\hat{v}_i(s^{(0)};\boldsymbol{\pi},\boldsymbol{r},\zeta) \geq Q_i^{(t)}(\{s^{(j)},a_i^{(j)}\}_{j=0}^t;\boldsymbol{\pi},\boldsymbol{r},h_{t-1}), \tag{5}$$
$$\forall t\in\mathbb{N}^+, i\in[N], s^{(j)}\in\mathcal{S}, a_i^{(j)}\in\mathcal{A}_i, h_{t-1}\in\mathcal{H}.$$

By implementing the (t+1)-step formulation eq. (5), we aim to construct the Lagrangian dual of the primal in eq. (3). 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 VI-A , we just focus on the constraints in eq. (5) to get the dual problem

$$\max_{\lambda\geq 0}\min_{\boldsymbol{\pi}} L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi},\lambda) \triangleq \sum_{i=1}^N \sum_{h_{t-1}\in\mathcal{H}} \sum_{\tau_i\in\mathcal{T}_i^t} \lambda(\tau_i;h_{t-1})(Q_i^{(t)}(\tau_i;\boldsymbol{\pi},\boldsymbol{r},h_{t-1}) - \hat{v}_i(s^{(0)};\boldsymbol{\pi},\boldsymbol{r},\zeta)), \tag{6}$$

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 VI-A.

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*

$$\lambda_\pi^*(\tau_i;h_{t-1}) = \eta(s^{(0)})Pr(h_{t-1})\prod_{j=0}^{t-1}(\sum_{\boldsymbol{a}_{-i}^j}\boldsymbol{\pi}_{-i}^*(\boldsymbol{a}_{-i}|s^{(j)}) P(s^{(j+1)}|s^{(j)},\boldsymbol{a}^{(j)}))\prod_{s^{(j)}:I_{i,j}=1}\pi_i(a_i^{(j)}|s^{(j)}) \tag{7}$$

be the probability of generating the sequence τ_i using policy π_i, $\boldsymbol{\pi}_{-i}^*$ 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 provides a horizon to establish *i*MA-IRL objective function with regularizer ψ.

$$i\textbf{MA-IRL}_\psi(\boldsymbol{\pi_E}) = \arg\max_{\boldsymbol{r}} -\psi(\boldsymbol{r}) + \sum_{i=1}^N(\mathbb{E}_{\boldsymbol{\pi}_E,Y}[r_i]) - (\max_{\boldsymbol{\pi}}\sum_{i=1}^N(\beta H_i(\pi_i) + \mathbb{E}_{\pi_i,\boldsymbol{\pi}_{E_{-i}},Y}[r_i])), \tag{8}$$

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 [11].

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

C. Inconcurrent Multi-Agent Occupancy Measure Matching

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

Definition 1 *For an agent $i\in[N]$ with a policy $\pi_i\in\Pi$, define its inconcurrent occupancy measure $\rho_{\pi_i}^p:\mathcal{S}\times\mathcal{A}_i\cup\{\phi\}\mapsto\mathbb{R}$ as $\rho_{\pi_i}^p(s,a)=$*

$$\begin{cases} \pi_i(a|s)(\eta(s)\zeta(i) + \sum_{t=1}^\infty\sum_{h_{t-1}}\gamma^t Pr(s_t=s|\pi_i,\boldsymbol{\pi}_{E_{-i}})Y(i|h_{t-1})), \\ \quad \text{if } a\in\mathcal{A}_i, \\ \eta(s)(1-\zeta(i)) + \sum_{t=1}^\infty\sum_{h_{t-1}}\gamma^t Pr(s_t=s|\pi_i,\boldsymbol{\pi}_{E_{-i}})(1-Y(i|h_{t-1})), \\ \quad \text{if } a\in\{\phi\}. \end{cases}$$

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 $\zeta(i)=1$, $Y(i|h_{t-1})=1$ for all $t\in\{1,...,\infty\}$, $h_{t-1}\in\mathcal{H}$, inconcurrent occupancy measure in MG turns to the occupancy measure defined in MAGAIL and GAIL, i.e., $\rho_{\pi_i}^p=\rho_{\pi_i}$. With the additively separable regularization ψ, for each agent i, π_{E_i} is the unique optimal response to other experts $\boldsymbol{\pi}_{E_{-i}}$. Therefore we obtain the following theorem:

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 i***MA-RL**(r) *has a unique solution*[4] *for all $\boldsymbol{r}\in i$***MA-IRL**$_\psi(\boldsymbol{\pi}_E)$, *then*

$$i\textbf{MA-RL}\circ i\textbf{MA-IRL}_\psi(\boldsymbol{\pi}_E) = \arg\min_{\boldsymbol{\pi}}\sum_{i=1}^N\sum_{h\in\mathcal{H}} -\beta H_i(\pi_i) + \psi_i^*(\rho_{\pi_i,\boldsymbol{\pi}_{E_{-i}}}^p - \rho_{\boldsymbol{\pi}_E}^p) \tag{9}$$

[4]The set of subgame perfect equilibrium is not always convex, so we have to assume *i***MA-RL**(r) returns a unique solution.

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 [25], we match the occupancy measure between $\rho^p_{\boldsymbol{\pi}_E}$ and $\rho^p_{\boldsymbol{\pi}}$ rather than $\rho^p_{\boldsymbol{\pi}_E}$ and $\rho^p_{\pi_i, \boldsymbol{\pi}_{E_{-i}}}$.

IV. Practical Inconcurrent Multi-Agent Imitation Learning

In this section, we propose practical algorithms for inconcurrent multi-agent imitation learning, and introduce three representative scenarios with different player functions.

A. Inconcurrent Multi-Agent Generative Adversarial Imitation Learning

The selected ψ_i in Proposition 1 (in Appx VI-B) 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 generator attempts to train the agent to maximize its score and fool the discriminator. We optimize the following objective:

$$\min_\theta \max_w \mathbb{E}_{\boldsymbol{\pi}_\theta, Y}\Big[\sum_{i=1}^N \log D_{w_i}(s, a_i)\Big] + \mathbb{E}_{\boldsymbol{\pi}_E, Y}\Big[\sum_{i=1}^N \log(1 - D_{w_i}(s, a_i))\Big]. \quad (10)$$

In practice, the input of *i*MA-IL is $\mathcal{Z}$, the demonstration data from N expert agents in the same environment, where the demonstration data $\mathcal{Z} = \{(s_t, \boldsymbol{a})\}_{t=0}^T$ are collected by sampling $s_0 \sim \eta$, $\boldsymbol{I}_0 \sim \zeta, \boldsymbol{I}_t \sim Y, \boldsymbol{a} \sim \boldsymbol{\pi}^*(\cdot|s_t), s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$. The assumptions include knowledge of $N, \gamma, \mathcal{S}, \mathcal{A}$. Transition P, initial state distribution η, agent distribution ζ, player function Y are all considered as black boxes, and no additional expert interactions with environment during training process are allowed. In the RL process of finding each agent's policy π_{θ_i}, we follow MAGAIL [25] to apply Multi-agent Actor-Critic with Kronecker-factors (MACK) and use the advantage function with baseline V_ν for variance reduction.

B. Player Function Structures

In MGs, 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 concurrent participation, deterministic participation, and stochastic participation.

Concurrent participation. When $Y(i|h_{t-1}) = 1$ holds for all agents $i \in [N]$ at every step t (as shown in Fig. 2a), agents make simultaneous actions, and a general Markov game boils down to a simple concurrent Markov game.

Deterministic participation. When the player function $Y(i|h_{t-1})$ is deterministic for all agents $i \in [N]$, it can only output 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. 2b 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 time step t, the agent i will make an action by chance. As illustrated in Fig. 2c, 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.

V. Experiments

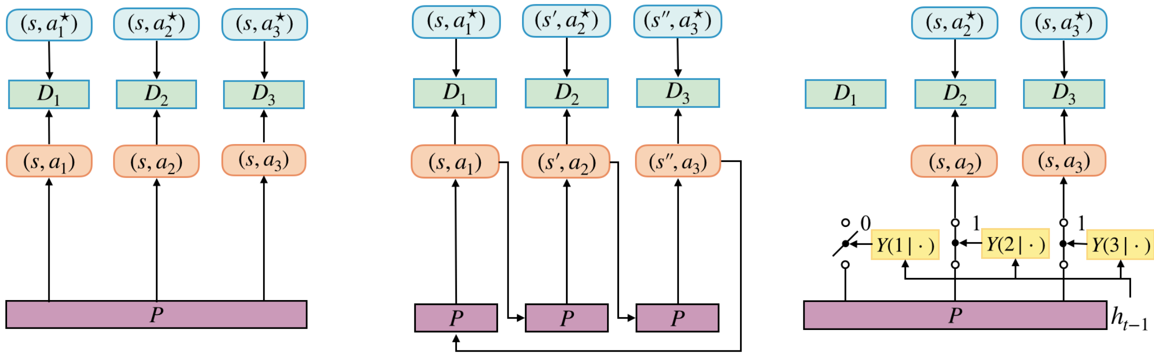
We evaluate *i*MA-IL with both stochastic and deterministic player function structures under cooperative games. We compared our *i*MA-IL with two baselines, including Behavior Cloning (BC) by OpenAI [6] and decentralized Multi-agent generative adversarial imitation learning (MAGAIL) [25]. The results are collected by averaging over 5 random seeds.

We use the particle environment [16] as a basic setting, and customize it into four games to allow different inconcurrent player function structures. **Deterministic Cooperative Navigation**: Three agents (agent #1, #2 and #3) need to cooperate to get close to three randomly placed 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 take actions, i.e., in the first round all three agents act, in the second round only agent #1 and #2 act, in the third round only agent #1 acts, and repeat these rounds until the game is completed. **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 round t.

In these game environments, agents are first trained with Multi-agent ACKTR [26], [25], thus the true reward functions are available, which enable us to evaluate the quality of recovered policies. When generating demonstrations from well-trained expert agents, a "null" (no-participation) as a placeholder action is recorded for each no-participation round in the trajectory. The quality of a recovered policy is evaluated by calculating agents' average true reward of a set of generated trajectories. We compare our *i*MA-IL with two baselines - behavior cloning (BC) [18] and decentralized Multi-agent generative adversarial imitation learning (MAGAIL) [25]. 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 match agents' occupancy measures from recovered policies to demonstration data.

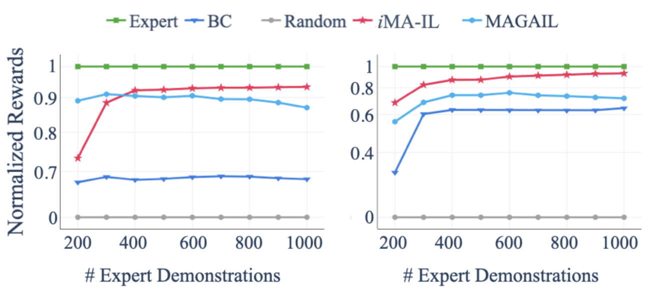
We compare *i*MA-IL with baselines under *deterministic cooperative navigation*, and *stochastic cooperative navigation* games. Fig. 3 show the normalized rewards, when learning policies with BC, MAGAIL and *i*MA-IL, respectively.

When there is only a small amount of expert demonstrations, the normalized rewards of BC and *i*MA-IL increase, especially, when less demonstration data are used, i.e.,



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

Fig. 2: *i*MA-IL with three player function structures. (a) **Concurrent 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 a 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.



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

Fig. 3: 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.

less than 400 demonstrations. After a sufficient amount of demonstrations are used, i.e., more than 400, *i*MA-IL has higher rewards than BC and MAGAIL. This makes sense since at certain time steps there exist non-participating agents (based on the player functions), but BC and MAGAIL models consider the non-participation as an action the agent can choose, where in reality it is governed by the environment. On the other hand, with the introduced player function Y, *i*MA-IL characterizes such no participation events correctly, thus more accurately learns the expert policies.

The normalized awards of BC are roughly unchanged in Fig. 3(a), and in Fig. 3(b) after 400 demonstrations, which seems contradictory to that of [19], [25], and can be explained as follows. In Fig. 3(b) (stochastic cooperative navigation), the performance of BC is low when using less demonstrations, but increases rapidly as more demonstrations are used, and finally converges to the "best" performance around 0.65 with 300 demonstrations. In Fig. 3(a), deterministic cooperative navigation is easier to learn compared with the stochastic cooperative navigation game shown in Fig. 3(b), since there is no randomness in the player function. The performance with only 200 demonstrations is already stabilized at 0.7. In the stochastic cooperative navigation game (Fig. 3(b)), *i*MA-IL performs consistently better than MAGAIL and BC. However, in the deterministic cooperative navigation game (Fig. 3(a)), with 200 demonstration, *i*MA-IL does not perform as well as MAGAIL. This is due to the game setting, namely, two players actively searching for landmarks are sufficient to gain a high reward in this game. The last agent, player #3, learned to be "lazy", without any motivation to promote the total shared reward among all agents. In this case, it is hard for *i*MA-IL to learn a good policy of player #3 with small amount of demonstration data, because player #3's has $\frac{2}{3}$ absence rate, given the pre-defined deterministic participation function. Hence, *i*MA-IL does not have enough state-action pairs to learn player #3. This gets improved when there are sufficient data, say, more than 400 demonstrations.

VI. Conclusion

In this paper, we make the first attempt to propose an inconcurrent multi-agent generative adversarial imitation learning (*i*MA-IL) framework, which models the inconcurrent decision-making process as a Markov game and develops a player function to capture the participation dynamics of agents. Experimental results demonstrate that our proposed *i*MA-IL can accurately learn the experts' policies from their inconcurrent trajectory data, comparing to SOTA baselines.

Acknowledgement

Xin Zhang, Weixiao Huang, and Yanhua Li were supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871. Ziming Zhang was supported in part by NSF CCF-2006738.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] 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.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[4] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20. Citeseer, 1997.

[5] Krishnendu Chatterjee, Rupak Majumdar, and Marcin Jurdziński. On nash equilibria in stochastic games. In *International Workshop on Computer Science Logic*, pages 26–40. Springer, 2004.

[6] 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.

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

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

[9] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In *Advances in neural information processing systems*, pages 6765–6774, 2017.

[10] Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1, 2013.

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

[12] 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.

[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

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

[16] 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*, pages 6379–6390, 2017.

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

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

[19] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[20] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[21] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[22] 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.

[23] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[24] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

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

[26] 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*, pages 5279–5288, 2017.

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

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

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

[30] 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.

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

APPENDIX

A. 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:*

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

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:

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

B. 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

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