Multi-Agent Learning from Learners

Mine Melodi Caliskan [*1] **Francesco Chini** [*1] **Setareh Maghsudi** [1]

Abstract

A large body of the "Inverse Reinforcement Learning" (IRL) literature focuses on recovering the reward function from a set of demonstrations of an expert agent who acts optimally or noisily optimally. Nevertheless, some recent works move away from the optimality assumption to study the "Learning from a Learner (LfL)" problem, where the challenge is inferring the reward function of a learning agent from a sequence of demonstrations produced by progressively improving policies. In this work, we take one of the initial steps in addressing the multi-agent version of this problem and propose a new algorithm, MA-LfL (Multi-agent Learning from a Learner). Unlike the state-of-the-art literature, which recovers the reward functions from trajectories produced by agents in some equilibrium, we study the problem of inferring the reward functions of interacting agents in a general sum stochastic game without assuming any equilibrium state. The MA-LfL algorithm is rigorously built on a theoretical result that ensures its validity in the case of agents learning according to a multi-agent soft policy iteration scheme. We empirically test MA-LfL and we observe high positive correlation between the recovered reward functions and the ground truth.

1. Introduction

The "Inverse Reinforcement Learning (IRL)" problem corresponds to inferring the reward function of a reinforcement learning (RL) agent from a set of trajectories. Learning the reward function, as compared to directly learning the policy of the demonstrator, allows to have a more succinct description of the task performed by the agent and this knowledge is better suited to be transferred to new environments. This is even more important when the demonstrator is not an expert, especially when it is under an ongoing policy learning process. Learning the reward functions, which do not change during the learning process, is also crucial in a multi-agent setting. Consider for instance the case of lane change in a highway for autonomous cars (Fig. 1). Here the environment contains multiple private agents, which can observe each others' states and actions but cannot access any other information such as the policies and rewards of others. Inferring the reward functions of other agents can be useful to model and predict their behaviour.

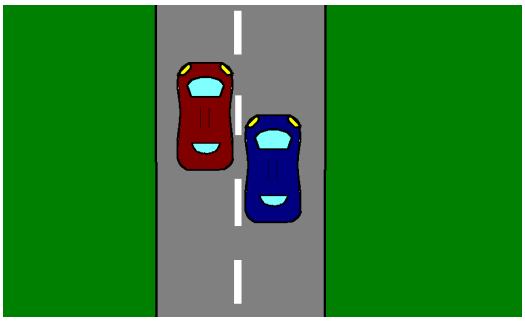


Figure 1. Two autonomous cars from different companies might optimize different reward functions which are not directly accessible. For example, one company might prioritize speed and another one safety or energy efficiency. They share the same environment (road) and learning the reward function of each other can help them to predict the other agent behaviour.

Initial work on IRL typically assumes the reward function to be linear w.r.t. a set of features (Abbeel & Ng, 2004). However recent approaches to IRL have been relaxing this assumption (Ho & Ermon, 2016; Fu et al., 2017). Early IRL literature also assumes the observed agent to be an expert, i.e., to behave optimally or noisy optimally (Ng et al., 2000; Ziebart et al., 2008). Recent work has relaxed the optimality assumption (Brown et al., 2019), (Tangkaratt et al., 2020) and in (Jacq et al., 2019), the authors have introduced the "Learning from a Learner (LfL)" problem, where the challenge is to infer the reward function of a learning agent from trajectories produced by a sequence of progressively improving policies. For another approach to

[*]Equal contribution [1]Department of Computer Science, University of Tuebingen, Tübingen, Germany. Correspondence to: Mine Melodi Caliskan <mine.caliskan@uni-tuebingen.de>, Francesco Chini <francesco.chini1990@gmail.com>, Setareh Maghsudi <setareh.maghsudi@uni-tuebingen.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the LfL problem see also (Ramponi et al., 2020).

The IRL problem has also been studied in the multi-agent case (Natarajan et al., 2010), where the goal is to recover the reward functions of a set of agents interacting in a stochastic game. In this setting, the agents are usually assumed to be in a certain equilibrium, such as Nash or correlated equilibrium (Reddy et al., 2012). This is quite restrictive considering that in many real-world applications, such as autonomous cars, multi-agent systems will likely not be in any equilibrium.

Here we introduce and study the multi-agent version of the LfL problem. We address the problem of recovering the reward functions of agents learning in a general-sum stochastic game. We do not assume the agents to be in any equilibrium but rather to be independently learning according to a multi-agent soft policy iteration scheme. To address this problem, we propose a new algorithm, MA-LfL (Multi-agent Learning from a Learner), which builds upon the single agent LfL algorithm (Jacq et al., 2019). Our algorithm, which we present both in offline and online settings, allows each agent to recover the reward functions of other agents while improving its own policy with respect to its own reward function. Moreover the recovered reward functions can be used by the agents to predict the next policy improvements of the other agents. We include error bounds both for the reward recovery and the policy improvement predictions. These are novel contributions even in the single agent case.

2. Related Work

Our work stems from (Jacq et al., 2019), where the authors introduce the LfL framework. The framework enables an *Observer* to learn the reward function of a *Learner*, who learn to solve a Markov Decision Process. The motivation there is to train the *Observer* with the recovered reward in order to potentially outperform the *Learner*. In our multi-agent setting all agents are *Observers* and *Learners* simultaneously. Our motivation is not to make the agents imitate (Yu et al., 2019; Torabi et al., 2018) or outperform each other (Jacq et al., 2019). Rather, we focus on modeling the agents during an ongoing learning process and we allow the agents to be heterogeneous, namely to have different action spaces and different reward functions.

The majority of the state-of-the-art research assumes specific reward structures, ranging from fully cooperative games (Natarajan et al., 2010; Barrett et al., 2017; Le et al., 2017; Šošić et al., 2017), to zero-sum games (Lin et al., 2017). We do not assume any of these restrictions as we allow the agents to interact in a general-sum stochastic game.

Multi-agent Adversarial Inverse Reinforcement Learning (MA-AIRL) (Yu et al., 2019) and Multi-Agent Generative Adversarial Imitation Learning (MA-GAIL) (Song et al., 2018) are frameworks with adversarial learning and they estimate policies and reward functions. In both works there are no strong assumptions on the reward structure. However in (Song et al., 2018) the agents are assumed to be in a Nash equilibrium. In (Yu et al., 2019) the agents are assumed to be in a logistic stochastic best response equilibrium (LSBRE), an equilibrium concept which is a stochastic generalization of Nash and correlated equilibrium. This reflects the assumption that the agents act sub-optimally, significantly relaxing the assumptions of early works on multi-agent IRL (Natarajan et al., 2010; Reddy et al., 2012). We take a step further by assuming the agents to be in a learning process rather than in an equilibrium.

3. Problem Setting

We consider the problem of N agents with a entropy-regularized objective acting in a Markov Game.

Definition 3.1. A Markov game (Littman, 1994) $\mathcal{M}$ for N agents is a tuple $(\mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, T, \{\mathcal{R}^i\}_{i=1}^N, P_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}^i$ is the action set of agent $i \in \{1, \cdots, N\}$, $T : \mathcal{S} \times \mathcal{A}^1 \times \ldots \mathcal{A}^N \to \mathcal{P}(\mathcal{S})$ is the transition function, $R^i \colon \mathcal{S} \times \mathcal{A}^1 \times \ldots \mathcal{A}^N \to \mathbb{R}$ is the reward function of agent $i \in \{1, \cdots, N\}$, $P_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution, and $0 \leq \gamma < 1$ is the discount factor.

Definition 3.2. A *policy* for agent i is a map $\pi^i \colon \mathcal{S} \to \mathcal{P}(\mathcal{A}^i)$, where $\mathcal{P}(\mathcal{A}^i)$ denotes the set of probability measures over $\mathcal{A}^i$s. Given policies $\pi^1, \ldots, \pi^N$, we use $\boldsymbol{\pi}$ to denote the joint policy $\boldsymbol{\pi} \colon \mathcal{S} \to \mathcal{P}\left(\mathcal{A}^1 \times \cdots \times \mathcal{A}^N\right)$, where $\boldsymbol{\pi}\left(a^1, \ldots, a^N | s\right) = \prod_{i=1}^N \pi^i\left(a^i | s\right)$. Moreover, $\boldsymbol{a} = \left(a^1, \ldots, a^N\right)$ is the the joint action profile of all agents. Besides, $\boldsymbol{a}^{-i} = \left(a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^N\right)$ and $\boldsymbol{\pi}^{-i}\left(\boldsymbol{a}^{-i} | s\right) = \prod_{j \neq i} \pi^j\left(a^j | s\right)$ respectively denote the joint action and the joint policy of the opponents of agent i.

Remark 3.3. Note that we do not assume the existence of a centralized actor. The symbol $\boldsymbol{\pi}$ only denotes the product of N individual policies.

Assumption 3.4. In our setting, agents have access to states s and actions $\boldsymbol{a}$ of all agents. However each agent i can only observe its own reward R^i.

3.1. Entropy-regularized objective

In a standard stochastic game, the objective of each agent i is to find a policy π^i that maximizes the expected total discounted reward. Formally,

$$\mathcal{J}(\pi^i) = \mathop{\mathbb{E}}_{\substack{a^i \sim \pi^i \\ \boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}}} \left[\sum_{t \geq 0} \gamma^t R^i\left(s_t, \boldsymbol{a}_t^{-i}, a_t^i\right) \right].$$

Remark 3.5. Note that the reward of every agent i, R^i, depends also on the actions of other agents. Consequently,

also the objective depends on the joint policy $\boldsymbol{\pi}^{-i}$ of other agents.

Assumption 3.6. We assume that the objective is entropy regularized; i.e., each agent i maximizes

$$\mathcal{J}_{\text{soft}}(\pi^i) = \mathop{\mathbb{E}}_{\pi^i, \boldsymbol{\pi}^{-i}} \left[\sum_{t \geq 0} \gamma^t \left(R_t^i + \alpha \mathcal{H}_t \right) \right], \quad (1)$$

where $R_t^i = R^i(s_t, \boldsymbol{a}_t)$, $\mathcal{H}_t = \mathcal{H}\left(\pi^i(\cdot|s_t)\right) = -\mathop{\mathbb{E}}_{a^i \sim \pi(\cdot|s)} \left[\ln \pi^i(a^i|s)\right]$ is the Shannon entropy and $\alpha > 0$ is a coefficient that controls the the degree of regularization.

Entropy regularization has been introduced in the RL literature as an approach to tackle the exploration-exploitation dilemma (Haarnoja et al., 2017; 2018).

Definition 3.7. Given a joint policy $\boldsymbol{\pi}$, the soft Q-value function for agent i is defined as

$$Q_{\text{soft}}^{\boldsymbol{\pi}, i}(s, \boldsymbol{a}) = R_0^i + \mathop{\mathbb{E}}_{\boldsymbol{\pi}} \left[\sum_{t > 0} \gamma^t \left(R_t^i + \alpha \mathcal{H}_t \right) \right], \quad (2)$$

for every $s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$, $R_t^i = R^i(s_t, \boldsymbol{a}_t)$, $\mathcal{H}_t = \mathcal{H}\left(\pi^i(\cdot|s_t)\right)$.

Remark 3.8. It is straightforward to show that $Q_{\text{soft}}^{\boldsymbol{\pi}, i}$ satisfies the following Bellman equation

$$Q_{\text{soft}}^{\boldsymbol{\pi}, i}(s, \boldsymbol{a}) = R^i(s, \boldsymbol{a}) + \gamma \mathop{\mathbb{E}}_{\boldsymbol{\pi}} [Q_{\text{soft}}^{\boldsymbol{\pi}, i}(s', \boldsymbol{a}') + \alpha \mathcal{H}\left(\pi^i(\cdot|s')\right)]. \quad (3)$$

4. Multi-agent Soft Policy Iteration

Our MA-LfL algorithm is built on the assumption that the agents are learning according to a multi-agent soft policy iteration (MA-SPI), which we derive from SPI in the single agent case. Before introducing the proposed MA-LfL algorithm, in this section we explain in detail the MA-SPI algorithm. Similar to many policy iteration algorithms (Sutton & Barto, 2018), it consists of a policy evaluation step and a policy improvement one and is an on-policy algorithm.

4.1. Reducing a Markov Game to a Single Agent Markov Decision Process

Let us recall here the statement of the theorem that underlies the single agent SPI algorithm, which guarantees that it improves policies monotonically.

Theorem 4.1 (Theorem 4 in Appendix A of (Haarnoja et al., 2017))**.** *Given a policy π in a entropy regularized Markov Decision Process, define a new policy π_{new} as*

$$\pi_{new}(\cdot|s) \propto \exp\left(\frac{Q_{soft}^{\pi}(s, \cdot)}{\alpha} \right), \quad (4)$$

for every state s, where α is the entropy coefficient. Then it follows that $Q_{soft}^{\pi_{new}}(s, a) \geq Q_{soft}^{\pi}(s, a)$, for every state-action pair (s, a).

We report here the statement of the theorem that underlies the single agent LfL algorithm of Jacq et al. (2019).

Theorem 4.2 (Theorem 2 in (Jacq et al., 2019))**.** *Let π and π_{new} two consecutive policies in an entropy regularized Markov Decision Process, with entropy coefficient α, such that π_{new} is the single agent soft policy improvement given by* (4)*. Then the following reward function*

$$\overline{R}(s, a) = \alpha \ln \pi_{new}(a|s) + \alpha\gamma \mathop{\mathbb{E}}_{s' \sim P} [D_{KL}\left(\pi(\cdot|s') \| \pi_{new}(\cdot|s')\right)]$$

coincides with the actual reward function R, up to a shaping -which will be defined in Section 4.4. Namely,

$$\overline{R}(s, a) = R(s, a) + g(s) - \gamma \mathop{\mathbb{E}}_{s' \sim P} [g(s')],$$

where g is a function defined on the state space.

Definition 4.3. Let $\mathcal{M} = (\mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, T, \{\mathcal{R}^i\}_{i=1}^N, P_0, \gamma)$ be a Markov game and let $\boldsymbol{\pi}^{-i}$ a joint policy for all agents except for agent i. We define the single agent Markov decision process $\widetilde{\mathcal{M}^i} = \left(\widetilde{\mathcal{S}}, \widetilde{\mathcal{A}}, \widetilde{P}, \widetilde{R}, \widetilde{P}_0, \widetilde{\gamma}\right)$, where

- $\widetilde{\mathcal{S}} = \mathcal{S}$;
- $\widetilde{\mathcal{A}} = \mathcal{A}^i$;
- $\widetilde{P}(s'|s, a) = P(s'|s, \boldsymbol{a}^{-i}, a)\boldsymbol{\pi}^{-i}(\boldsymbol{a}^{-i})$;
- $\widetilde{R}(s, a) = \mathop{\mathbb{E}}_{\boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}} [R^i(s, \boldsymbol{a}^{-i}, a)]$;
- $\widetilde{\gamma} = \gamma$.

For agent i, a policy π^i defines a policy for the MDP $\widetilde{\mathcal{M}^i}$. Moreover, if $\boldsymbol{\pi}^{-i}$ remains fixed, then the entropy regularized objective in Eq 1 is equal to the entropy regularized objective for $\widetilde{\mathcal{M}^i}$, i.e.,

$$\widetilde{\mathcal{J}}_{\text{soft}}(\pi^i) = \mathop{\mathbb{E}}_{\pi^i} \left[\sum_{t \geq 0} \gamma^t \left(\widetilde{R}(s_t, a_t) + \alpha \mathcal{H}\left(\pi^i(\cdot|s_t)\right) \right) \right].$$

4.2. Policy Evaluation

Given a joint policy $\boldsymbol{\pi} = \prod_{i=1}^N \pi^i$, each agent i learns the expectation of $Q_{\text{soft}}^{\boldsymbol{\pi}, i}$ with respect to π^{-i} during the run of some episodes. From the perspective of agent i, during the evaluation phase, the other agents can be thought of being part of the environment, by absorbing the policy $\boldsymbol{\pi}^{-i}$ into the dynamics. Therefore, during the evaluation phase, the Markov game is equivalent to a Markov Decision Process

$\widetilde{\mathcal{M}}^i$ for agent i and the expectation of $Q^{\boldsymbol{\pi},i}_{\text{soft}}$ is in fact the soft Q function $\widetilde{Q}^{\pi^i}_{\text{soft}}$ w.r.t. to $\widetilde{\mathcal{M}}^i$. Hence, agent i learns $\widetilde{Q}^{\pi^i}_{\text{soft}}(s,a^i) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},i}_{\text{soft}}(s,\boldsymbol{a}^{-i},a^i)\right]$ via temporal difference learning based on the Bellman equation Eq (3):

$$\widetilde{Q}^{\pi^i}_{\text{soft}}(s,a^i) = \tilde{R}^i\left(s,a^i\right) + \gamma\mathbb{E}_{\boldsymbol{\pi}}\left[\widetilde{Q}^{\pi^i}_{\text{soft}}(s',a^i_{\text{new}}) + \alpha\mathcal{H}\left(\pi^i\left(\cdot|s'\right)\right)\right] \quad (5)$$

where $\tilde{R}^i\left(s,a^i\right) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}[R^i(s,\boldsymbol{a}^{-i},a^i)]$.

4.3. Policy Improvement

Definition 4.4. Given a policy π^i for agent i and $\boldsymbol{\pi}^{-i}$ for the opponents, the soft policy improvement for agent i is defined as

$$\pi^i_{\text{new}}(a^i|s) \propto \exp\left(\frac{1}{\alpha}\widetilde{Q}^{\pi^i}_{\text{soft}}(s,a^i)\right), \quad (6)$$

where $\widetilde{Q}^{\pi^i}_{\text{soft}}(s,a^i) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},i}_{\text{soft}}\left(s,\boldsymbol{a}^{-i},\cdot\right)\right]$. In the following we will will use the notation $\text{SPI}_{\boldsymbol{\pi}^{-i}}(\pi^i)$ to denote the soft policy improvement π^i_{new}.

Assumption 4.5. We assume all the agent to update their policies simultaneously

$$\boldsymbol{\pi}_{\text{new}} = \prod_{i=1}^{N}\text{SPI}_{\boldsymbol{\pi}^{-i}}(\pi^i).$$

Lemma 4.6. *Let $\widetilde{Q}^{\pi^i}_{soft}$ be the soft Q-value function for a policy π^i as a policy for the MDP $\widetilde{\mathcal{M}}^i$. Formally,*

$$\widetilde{Q}^{\pi^i}_{soft}(s,a^i) = \widetilde{R}(s,a^i) + \mathbb{E}_{\pi^i}\left[\sum_{t>0}\gamma^t\left(\widetilde{R}_t + \alpha\mathcal{H}_t\right)\right],$$

where $\widetilde{R}_t = \widetilde{R}(s_t,a_t)$ and $\mathcal{H}_t = \mathcal{H}\left(\pi^i(\cdot|s_t\right)$. Then we have

$$\widetilde{Q}^{\pi^i}_{soft}(s,a^i) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},i}_{soft}(s,\boldsymbol{a}^{-i},a^i)\right],$$

where $\boldsymbol{\pi} = \boldsymbol{\pi}^{-i}\pi^i$ and $Q^{\boldsymbol{\pi},i}_{soft}$ is the soft Q-value function of $\boldsymbol{\pi}$ for agent i in the Markov game $\mathcal{M}$.

Proof. The proof follows immediately from the definition of $\widetilde{\mathcal{M}}^i$ given above. □

Theorem 4.7 (Soft-policy Improvement Theorem)**.** *Let π^i be a policy for agent i and $\boldsymbol{\pi}^{-i}$ a joint policy for other agents and let $\pi^i_{new} = SPI_{\boldsymbol{\pi}^{-i}}(\pi^i)$ as defined in* (6)*.*

Then for every $a^i \in \mathcal{A}^i$, we have

$$\tilde{Q}^{\pi_{new},i}_{soft}(s,a^i) \geq \tilde{Q}^{\pi,i}_{soft}(s,a^i)$$

where $\tilde{Q}^{\pi,i}_{soft}(s,a^i) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},i}_{soft}\left(s,\boldsymbol{a}^{-i},a^i\right)\right]$ and $\boldsymbol{\pi}_{new} = \pi^i_{new}\boldsymbol{\pi}^{-i}$.

Proof. As explained above, when the policies $\boldsymbol{\pi}^{-i}$ for the other agents are held fixed, which is guaranteed by Assumption 4.5, the Markov game reduces to a MDP $\widetilde{\mathcal{M}}^i$ for agent i. Therefore the proof follows directly from (6) and Theorem 4.1. □

Remark 4.8. As a consequence of Theorem 4.7 below, $\pi^i_{\text{new}}(a^i|s)$ is the greedy improvement for agent i w.r.t. the opponents joint policy $\boldsymbol{\pi}^{-i}$, namely

$$\pi^i_{\text{new}}(a^i|s) = \arg\max_{\pi^i_{\text{new}}}\left\{\mathbb{E}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}\\ a^i\sim\pi^i_{\text{new}}}}\left[Q^{\boldsymbol{\pi},i}_{\text{soft}}(s,\boldsymbol{a}^{-i},a^i)\right]\right\}.$$

In other words the soft policy update is guaranteed to be an improvement for agent i in the case where the other agents do not change their policy π^{-i}. However in our setting we assume all the agents to update their policies simultaneously, therefore there is no guarantee that the policy update is an actual improvement.

4.4. Invariance Under Reward Shaping

The classical IRL problem is ill-posed (Ng et al., 2000); that is, the solution is not unique, as several different reward functions explain the behavior of an optimal agent. Similar difficulties arise in the LfL setting (Jacq et al., 2019) because the single-agent Soft Policy Iteration algorithm is invariant under reward shaping. Naturally, the multi-agent setting inherits the same issue. More precisely, if we transform the reward function of an agent i by adding a *shaping*, then the soft policy improvement Eq (6) is still the same. A function $sh\colon \mathcal{S}\times\mathcal{A}^1\times\cdots\times\mathcal{A}^N\to\mathbb{R}$ is called *shaping* if there exists a function $g\colon\mathcal{S}\to\mathbb{R}$ such that $sh(s,\boldsymbol{a}) = g(s) - \gamma\,\mathbb{E}_{s'\sim P(s,\boldsymbol{a})}\left[g(s')\right]$.

Lemma 4.9 (SPI invariance under shaping)**.** *Let $R^i_1\colon\mathcal{S}\times\mathcal{A}^1\times\cdots\times\mathcal{A}^N\to\mathbb{R}$ and $R^i_2\colon\mathcal{S}\times\mathcal{A}^1\times\cdots\times\mathcal{A}^N\to\mathbb{R}$ two reward functions for agent i such that for every $s\in\mathcal{S}$, $\boldsymbol{a} = (a^1,\ldots,a^N)\in\mathcal{A}^1\times\cdots\times\mathcal{A}^N$: $R^i_1(s,\boldsymbol{a}) = R^i_2(s,\boldsymbol{a}) + sh(s,\boldsymbol{a})$. Let $SPI^1_{\boldsymbol{\pi}^{-i}}$ and $SPI^2_{\boldsymbol{\pi}^{-i}}$ be the soft policy improvement operators induced respectively by R^i_1 and R^i_2. Then for every policy π^i*

$$SPI^1_{\boldsymbol{\pi}^{-i}}(\pi^i) = SPI^2_{\boldsymbol{\pi}^{-i}}(\pi^i).$$

Proof. The proof is a simple extension of the proofs of Lemma 1 and Theorem 1 in (Jacq et al., 2019). □

5. Multi-Agent Learning from Learners

In this section, we explain how each agent i can recover an estimation R^j_i of the reward function R^j of each other agent after a certain number of MA-SPI steps, as described

Algorithm 1 Multi-agent Soft Policy Iteration (MA-SPI)

Initialization $\pi^i \leftarrow$ Uniformly random policy, for $i = 1, \dots, N$.
for $h = 1$ to H **do**
  Initialize $\widetilde{Q}^{\pi^i}_{\text{soft}} \leftarrow 0$, for $i = 1, \dots, N$
  for each episode **do**
    $t \leftarrow 0$
    $s_0 \sim P_0$
    while s_t **not** terminal **do**
      Each agent i chooses $a^i_t \sim \pi^i(\cdot|s_t)$
      Each agent i observes $R^i(s_t, \boldsymbol{a}_t)$
      $s_{t+1} \sim P(s_t, \boldsymbol{a}_t)$
      Each agent i chooses $a^i_{t+1} \sim \pi^i(\cdot|s_{t+1})$
      Each agent i updates $\widetilde{Q}^{\pi^i}_{\text{soft}}$ according to Eq (5)
      $t \leftarrow t + 1$
    end while
    Each agent simultaneously i updates $\pi^i \leftarrow \text{SPI}(\pi^i)$ using Eq (6)
  end for
end for

in Section 4. We call this algorithm *Multi Agent Learning from a Learner* (MA-LfL), as it is a multi-agent extension of the LfL algorithm developed in (Jacq et al., 2019). The pseudocode is given in Algorithm 2.

The core of the proposed MA-LfL algorithm is the theorem below, which states the following: From the observation of one soft policy improvement for an agent i, namely observing two consecutive policies π^i and $\text{SPI}_{\boldsymbol{\pi}^{-i}}(\pi^i)$, it is possible to recover the expectation w.r.t. $\boldsymbol{\pi}^{-i}$ of the reward function R^i, up to a shaping.

Theorem 5.1 (Recovering reward up to shaping)**.** *Let $\boldsymbol{\pi}^{-i}$ be a joint policy for all the agents except i. Besides, π^i is a policy for i and $\pi^i_{new} = SPI_{\boldsymbol{\pi}^{-i}}(\pi^i)$ is the soft policy improvement given by Eq (6). Then*

$$\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\overline{R^i}(s, \boldsymbol{a}^{-i}, a^i)\right] = \alpha \ln \pi^i_{new}(a^i|s) + \alpha\gamma \mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}\\ s'\sim P(\cdot|s,\boldsymbol{a}^{-i},a^i)}} \left[D_{KL}(\pi^i(\cdot|s')|\pi^i_{new}(\cdot|s'))\right], \tag{7}$$

where $\overline{R^i}(s, \boldsymbol{a}^{-i}, a^i) = R^i(s, \boldsymbol{a}^{-i}, a^i) + sh(s, \boldsymbol{a}^{-i}, a^i)$, $sh\colon \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathbb{R}$ is a shaping.

Proof. As in the proof of Theorem 4.7, if $\boldsymbol{\pi}^{-i}$ remains fixed, for agent i, the Markov game $\mathcal{M}$ reduces to the Markov decision process $\widetilde{\mathcal{M}}^i$ as defined in Section 4.1. From Lemma 4.6, we have that $\pi^i_{\text{new}}(\cdot|s) \propto \exp\left(\frac{1}{\alpha}\widetilde{Q}^{\pi^i}_{\text{soft}}(s, \cdot)\right)$. Therefore, using Theorem 4.2, we can recover the reward $\widetilde{R}$ of $\widetilde{\mathcal{M}}^i$ up to a shaping. Formally,

$$\widetilde{\overline{R}}(s, a^i) = \alpha \ln \pi^i_{\text{new}}(a^i|s) + \alpha\gamma \mathop{\mathbb{E}}_{s'\sim\widetilde{P}}\left[D_{\text{KL}}\left(\pi^i(\cdot|s')\|\pi^i_{\text{new}}(\cdot|s')\right)\right], \tag{8}$$

where

$$\widetilde{\overline{R}}(s, a^i) = \widetilde{R}(s, a^i) + g(s) - \gamma \mathop{\mathbb{E}}_{s'\sim\widetilde{P}}[g(s')], \tag{9}$$

for some function $g\colon \mathcal{S} \to \mathcal{S}$.

From the definition of the Markov decision process $\widetilde{\mathcal{M}}^i$ in Section 4.1, and Eq (9), we rewrite Eq (8) as

$$\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\overline{R^i}(s, \boldsymbol{a}^{-i}, a^i)\right] = \alpha \ln \pi^i_{new}(a^i|s) + \alpha\gamma \mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}\\ s'\sim P(\cdot|s,\boldsymbol{a})}} \left[D_{\text{KL}}\left(\pi^i(\cdot|s')\|\pi^i_{\text{new}}(\cdot|s')\right)\right].$$

□

Remark 5.2. As mentioned in Remark 4.8, the MA-SPI algorithm is not guaranteed to improve agents' policies. However our reward recovering MA-LfL algorithm only relies on the way agents are updating their policies and it is not affected on whether the agents are actually improving.

5.1. Estimating Other Agent Policies

Theorem 5.1 allows each agent to extract information about the reward functions of each other agents given their policies. In practice, agents can only observe the actions of each other; therefore, to apply Theorem 5.1, they must learn the policies from the observed trajectories. Every agent uses the entropy regularized maximum likelihood estimation (MLE) method to estimate other agents' policies from the observed trajectories.

Let $\boldsymbol{\pi} = \prod_{i=1}^N \pi^i$ be a joint policy for the agents, and $\mathcal{D}$ represent a set of trajectories $\mathcal{D} = \{\tau_1, \dots, \tau_K\}$ produced by $\boldsymbol{\pi}$. Each trajectory τ_k is a sequence of states and actions $\tau_k = \{s_{k,0}, \boldsymbol{a}_{k,0}, s_{k,1}, \boldsymbol{a}_{k,1}, \dots\}$. Each agent j learns a parameterized approximation $\hat{\pi}^i_j = \hat{\pi}^i_{\theta^i_j}$ of the policy π^i of agent i by maximizing the entropy regularized likelihood $\mathcal{L}(\theta^i_j)$. Formally,

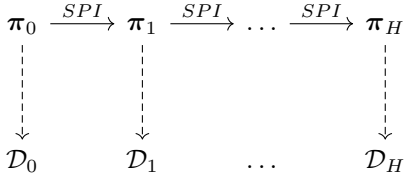
$$\mathcal{L}(\theta^i_j) = -\sum_{k=1}^K \sum_{(s,a^i)\in\tau_k} \ln\left(\hat{\pi}^i_j(a^i|s)\right) + \lambda\mathcal{H}\left(\hat{\pi}^i_j(\cdot|s)\right), \tag{10}$$

where $\lambda > 0$ is the entropy regularization parameter.

5.2. Estimating Rewards from Trajectories

Now we discuss how each agent learns the rewards of other agents after a specific number of MA-SPI steps. Let

$\{\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_H\}$ be the $H+1$ joint policies for N agents obtained while performing the MA-SPI algorithm (Algorithm 1) for H rounds; i.e., for every for $h = 1, \ldots H$, let $\boldsymbol{\pi}_h = \text{SPI}(\boldsymbol{\pi}_{h-1})$. Moreover, let $\mathcal{D}_h$ be the set of trajectories produced by the agents with the joint policy $\boldsymbol{\pi}_h$ during the h-th MA-SPI.



Let $\hat{R}^i_j = \hat{R}^i_{\phi^i_j}$ be a parametrization of reward R^i that agent j is attempts to learn. As explained in Section 5.1, agent j can learn $\{\pi^i_0, \pi^i_1, \ldots, \pi^i_H\}$ from $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_H\}$ respectively. From those learned policies, agent j computes H targets $\{Y_1, \ldots, Y_H\}$ according to Theorem 5.1, defined as

$$Y^i_h(s, a^i) = \alpha \ln \pi^i_h\left(a^i|s\right) + \alpha\gamma \mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i} \\ s'\sim P(\cdot|s,\boldsymbol{a})}} \left[D_{\text{KL}}\left(\pi^i_{h-1}(\cdot|s')\|\pi^i_h(\cdot|s')\right)\right], \quad (11)$$

for every $h = 1, \ldots, H$.

Recall that from each improvement $\pi^i_h \xrightarrow{SPI} \pi^i_{h+1}$, Theorem 5.1 allows inferring the expectation of $R^i + sh_h$, where sh_h is a shaping function. Observe that we use the index h because for different improvements, we might have different shapings. Since sh_h is a shaping, by definition (see Section 4.4), there exists a function $g_h \colon \mathcal{S} \to \mathbb{R}$ such that

$$sh_h(s, \boldsymbol{a}) = g_h(s) - \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,\boldsymbol{a})}\left[g_h(s')\right].$$

Definition 5.3. Let g_{ψ_h} be a parametrization of g_h. We define the loss function for the parameters ϕ^i_j of $\hat{R}^i_j = \hat{R}^i_{\phi^i_j}(s, \boldsymbol{a}^{-i}, a^i)$ as

$$\mathcal{L}^i_j(\phi^i_j) = \min_{\psi_0,\ldots,\psi_H} \sum_{h=1}^{H} \sum_{(s,\boldsymbol{a},s')\in\mathcal{D}_h} (\hat{R}^i_j + sh_{\psi_h}(s, s') - Y_h)^2, \quad (12)$$

where $sh_{\psi_h}(s, s') = g_{\psi_h}(s) - \gamma g_{\psi_h}(s')$.

Remark 5.4. The optimization of the loss function above is directly affected by the policy inference success, because the target values Y_h's are produced by the inferred policies.

5.3. Semi-online MA-LfL

In the previous section, we explained how agents learn the reward functions of other agents in a offline manner from a collection of sets of trajectories generated during a number of MA-SPI iterations. However, MA-LfL can also be performed *semi-online*, meaning that each agent maintains an estimation of the reward functions of the opponent which is updated after each MA-SPI step. Since entropy regularized maximum likelihood estimation can be performed online, each agent can learn the policies of the opponents in a streaming manner, during each MA-SPI step.

Algorithm 2 Multi-agent Learning from a Learner (MA-LfL)

Run Algorithm 1 and generate sets of trajectories $\{\mathcal{D}_h\}_{h=1}^H$ using $\{\pi_h\}_{h=1}^H$
for $h = 1$ to H **do**
  for each agent $j, i = 1, \ldots, N, j \neq i$ **do**
    Agent j learns estimate $\hat{\pi}^i_h$ from D_h via Eq (10)
    Agent j computes targets Y^i_h using Eq (11)
  end for
end for
Each agent j computes $\hat{R}^i_j$ via Eq (12)
Return $\hat{R}^i_j$ for each $j, i = 1, \ldots, N, j \neq i$.

Now, consider the h-th MA-SPI step. Then agent j updates the parameters ϕ^i_j of $\hat{R}^i_j$ using the gradient $\nabla\mathcal{L}^j_h$, as in a mini-batch gradient descent, where $\mathcal{L}^j_h$ is the following loss function

$$\mathcal{L}^j_h(\phi^i_j) = \sum_{(s,\boldsymbol{a},s')\in\mathcal{D}_h} \left(\hat{R}^i_j(s, \boldsymbol{a}^{-i}, a^i) + sh_{\psi_h}(s, s') - Y_h\right)^2,$$

where $sh_{\psi_h}(s, s') = g_{\psi_h}(s) - \gamma g_{\psi_h}(s')$.

After the h-th iteration of MA-SPI, to predict the next soft policy improvement of the opponents using (6), each agent j has to estimate $\widetilde{Q}^{\pi^i}_{\text{soft}}(s, a^i) = \mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},i}_{\text{soft}}(s, \boldsymbol{a}^{-i}, a^i)\right]$ for each other agent i. That is doable with an off-line version of TD-learning or Monte Carlo (Sutton & Barto, 2018) from the trajectories in $\mathcal{D}_h$, using the current estimations $\hat{R}^i_j$ and $\hat{\pi}^i$ instead of R^i and π^i.

Remark 5.5. When performing MA-LfL online, the agents must assess the quality of their current estimations of other agents' reward functions. One way to do so is to use the current rewards estimations to predict future soft policy improvements followed by observing the actual ones. If the agents have valid estimations of the reward functions, their predictions of the soft policy improvements will be close to the actual ones. We will provide more details on how to bound the reward estimation error in the next Section 6.

6. Error Bound Analysis

In this section we provide a bound on the error on the recovered reward functions in terms of the policies improvement prediction error in Theorem 6.3 and Theorem 6.1. Con-

versely, we also provide a bound on the policy improvement prediction error in terms of the error on the recovered rewards in Theorem 6.5. In Appendix A we state and prove the single-agent version of these results for the LfL framework of (Jacq et al., 2019) and in Appendix B we extend the proofs to our multi-agent setting.

6.1. Reward Recovery Error Bound

Theorem 6.1. *Let R^i be the reward function for agent i and let $\hat{R}^i_j$ be the reward estimation of R^i learned by agent j. Let π^i a policy for agent i and $\boldsymbol{\pi}^{-i}$ a joint policy for the other agents. Let $\pi^i_{new} = SPI_{\boldsymbol{\pi}^{-i}}(\pi^i)$ be the soft policy improvement as defined in Theorem 4.7 and let $\hat{\pi}^i_{new}$ be the soft policy improvement predicted by agent j using $\hat{R}^i_j$, namely $\hat{\pi}^i_{new} \propto \exp\left(\frac{1}{\alpha}\mathbb{E}_{\boldsymbol{\pi}^{-i}}[Q^{\boldsymbol{\pi},i,\hat{R}^i_j}_{soft}]\right)$. If*

$$\sup_{a^i\in\mathcal{A}^i\, s\in\mathcal{S}} |\ln \pi^i_{new}(a^i|s) - \ln \hat{\pi}^i_{new}(a^i|s)| < \delta,$$

then there exists a shaping sh such that for every $s \in \mathcal{S}$ and $a^i \in \mathcal{A}^i$

$$|\mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\hat{R}^i_j(s,\boldsymbol{a}^{-i},a^i) - (R^i+sh)(s,\boldsymbol{a}^{-i},a^i)\right]| < \varepsilon,$$

where

$$\varepsilon = \delta\alpha(1+\gamma),$$

and α is the entropy coefficient and γ is the discount factor.

Proof. See Appendix B. □

Corollary 6.2. *Consider the case in which R^i and $\hat{R}^i$ are state-only dependent reward functions. If*

$$\sup_{a^i\in\mathcal{A}^i s\in\mathcal{S}} \left|\ln \pi^i_{new}(a^i|s) - \ln \hat{\pi}^i_{new}(a^i|s)\right| < \delta$$

there exists a shaping $sh\colon \mathcal{S}\times\mathcal{A}^i \to \mathcal{S}$ such that depends only on the state and the actions of agent i such that

$$\sup_{s\in\mathcal{S}a^i\in\mathcal{A}^i} \left|\hat{R}^i_j(s) - \left(R^i(s) + sh(s,a^i)\right)\right| < \varepsilon,$$

where $\varepsilon = \delta\alpha(1+\gamma)$, and α is the entropy coefficient and γ is the discount factor.

Proof. Follows directly from Theorem 6.1. □

In the special case of state-only dependent reward function, we can provide another error bound that depends on the KL-divergence between the predicted and the actual soft policy improvement.

Theorem 6.3. *Let us assume the reward function R^i for agent i and its estimation $\hat{R}^i_j$ maintained by agent j to be state-only dependent. Let π^i be a policy for the agent i, $\pi^i_{new} = SPI_{\boldsymbol{\pi}^{-i}}(\pi^i)$ its soft policy improvement and let $\hat{\pi}^i_{new}$ be the soft policy improvement predicted by the agent j. Let us assume $\mathcal{A}^i$ to be a finite set and let $|\mathcal{A}^i|$ be its cardinality. If*

$$\sup_s D_{KL}(\pi^i_{new}(\cdot|s)\|\hat{\pi}^i_{new}(\cdot|s)) < \delta,$$

then there exists a shaping $sh\colon \mathcal{S}\times\mathcal{A}^i \to \mathbb{R}$, that depends only on states and on the actions of agent i, such that

$$\sup_{s\in\mathcal{S}} |\hat{R}^i(s) - \mathbb{E}_{a^i\sim\pi^i(\cdot|s)}(R^i(s) + sh(s,a^i))| < \varepsilon,$$

where

$$\varepsilon = \delta\left(1 + \gamma|\mathcal{A}^i|e^{\frac{\Delta^i}{\alpha(1-\gamma)}}\right),$$

α is the entropy coefficient, γ is the discount factor and Δ^i is the maximum gap for R^i, namely $\Delta^i = \sup_{s\in\mathcal{S}} R^i(s) - \inf_{s\in\mathcal{S}} R^i(s)$.

Proof. See Appendix B. □

Remark 6.4. The assumptions in Theorem 6.3 on the finiteness of the action space $\mathcal{A}^i$ and the fact that R^i is a function only of the state are not so restrictive. Moreover the error bound on the estimation error for the reward is express in terms of the bound on the KL-divergence, which can be learned in practice by agent j.

6.2. Policy Improvement Prediction Error Bound

The recovered rewards allow the agents to predict the soft policy improvements of each other agents. The following theorem provides a bound on the KL-divergence between the actual improvement and the predicted improvement.

Theorem 6.5. *Let R^i be the reward function for agent i and let $\hat{R}^i_j$ be an estimation recovered by agent j. Let $\pi^i_{new} = SPI_{\boldsymbol{\pi}^{-i}}(\pi^i)$ be the actual policy improvement of the policy π^i, and $\hat{\pi}^i_{new}$ the soft policy improvement predicted by agent j using $\hat{R}^i_j$. Let $\delta > 0$ be such that for all $s \in \mathcal{S}, a^i \in \mathcal{A}^i$*

$$\left|\mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\hat{R}(s,a) - (R+sh)(s,a)\right]\right| < \delta$$

for a shaping sh. Then

$$\sup_{s\in\mathcal{S}} D_{KL}(\pi^i_{new}(\cdot|s)|\hat{\pi}^i_{new}(\cdot|s)) < \varepsilon,$$

where

$$\varepsilon = \delta\left(\frac{1}{\alpha(1-\gamma)} + \frac{|\mathcal{A}^i|}{e^{\frac{\inf R^i}{\alpha}}}\right),$$

and α is the entropy coefficient and γ is the discount factor.

Proof. See Appendix B. □

7. Experiments

We test MA-LfL experimentally in a 3×3 deterministic grid world environments. The agents always start at the top-left cell and try to reach to the bottom-right cell. Our experimental setting involves two agents, i.e., $N = 2$. We emphasize that our theoretical results hold regardless of the number of agents. We assume the transition function is deterministic and known. The action space includes five actions: move up, down, left, and right, or stay.

We use two different reward functions in order to demonstrate our algorithm achieves reward recovery in general-sum games: M_{hom}: Homogeneous reward function as a combination of Manhattan disjoint distance Eq (13) and M_{het}: Heterogenous reward function as a combination of Manhattan joint and disjoint distance Eq (14).

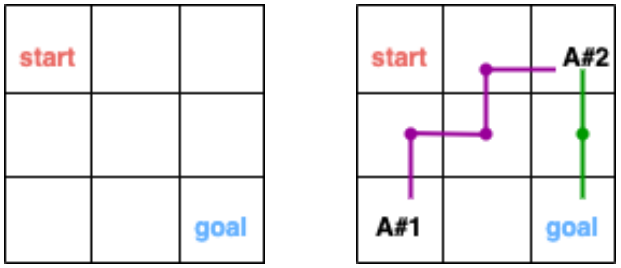
Definition 7.1. Let $\boldsymbol{p}_g = (x_g, y_g)$ be a goal location and $\boldsymbol{p}_i(t) = (x_i, y_i)$, $\boldsymbol{p}_j(t) = (x_j, y_j)$ be the positions of agent i and agent j at time t. Then we define

$$M^i_{\text{hom}}(t) = -\|\boldsymbol{p}_i(t) - \boldsymbol{p}_g\|_1 + \|\boldsymbol{p}_i(t) - \boldsymbol{p}_j\|_1 \tag{13}$$

and

$$M^i_{\text{het}}(t) = \begin{cases} -\|\boldsymbol{p}_i(t) - \boldsymbol{p}_g\|_1 - \|\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)\|_1 & \text{A\#1} \\ -\|\boldsymbol{p}_i(t) - \boldsymbol{p}_g\|_1 + \|\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)\|_1 & \text{A\#2} \end{cases} \tag{14}$$

for both agents $i = 1, 2$.



(a) The grid world.

(b) Example reward calculation.

Figure 2. (a) Agents start at the top-left grid and the goal location is the bottom-right. Every time agents arrive to the goal location, their states are reset as the start cell. (b) Purple lines indicates the Manhattan distance between two agents and green lines indicates the Manhattan distance between Agent #2 and the goal location.

In the M_{hom} setting, the agents try to minimize the distance between themselves and the goal while at the same time trying to stay as far away from each other as possible. In M_{het}, similarly, both agents try to minimize the distance between themselves and the goal, however, while one agent tries to stay as close to the other as possible the other agent tries to stay away.

We measure the performance of MA-LfL by computing the correlation between the recovered rewards with the ground-truth ones. In all cases in our experiments, agents have no access to the other agents' policies or rewards, and they use state-action models for estimating the reward functions. However, since all the experiments consist of simulations, we have access to the ground-truth reward functions that we use for the evaluation. We use statistical correlation metrics Pearson's correlation coefficient (PCC) for linear correlation and Spearman's correlation coefficient (SCC) for rank correlation to compare the estimated reward functions with the actual rewards. In our experiments, we demonstrate recovery of rewards MA-LfL achieves using MA-SPI in both the heterogeneous and the homogeneous reward cases. We present our results in Table. 1.

Metric	M_{hom}	M_{het}
PCC #1	0.48 ± 0.06	0.45 ± 0.04
PCC #2	0.59 ± 0.02	0.42 ± 0.02
$\hat{P}$	0.54 ± 0.03	0.44 ± 0.01
SCC #1	0.44 ± 0.14	0.51 ± 0.02
SCC #2	0.60 ± 0.04	0.43 ± 0.03
$\hat{S}$	0.52 ± 0.06	0.47 ± 0.01

Table 1. Pearson's correlation coefficients (PCC) and Spearman's correlation coefficients (SCC) of Agent 1 and Agent 2 between true reward functions and estimated reward functions. $\hat{P}$ and $\hat{S}$ are the averaged scores of PCC and SCC over both agents. Mean and variance are taken from the experiments with different random seeds.

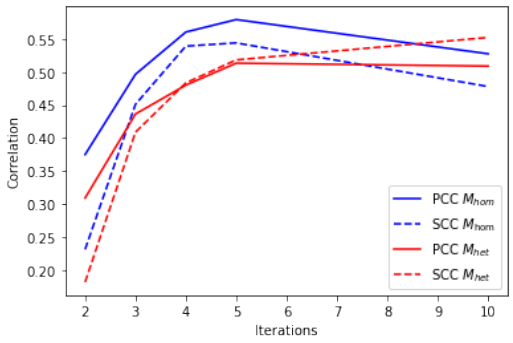


Figure 3. The quality of recovered rewards has a logarithmic growth rate as the agents improve their policies. In our experiments, we observed that agents were able to recover rewards using only 10 iterations in a 3×3 grid world.

As a baseline for correlation coefficients, we calculated correlation between estimated joint and disjoint rewards to disjoint and joint ground truths respectively. Results are given in Table 2.

Estimated Reward	Manh. Disjoint	Manh. Joint
Manhattan Disjoint	**PCC: 0.55** **SCC: 0.57**	PCC: 0.4 SCC: 0.37
Manhattan Joint	PCC: 0.32 SCC: 0.33	**PCC: 0.47** **SCC: 0.51**

Table 2. Cross-correlation between ground truths and estimations of two reward functions. All correlations are positive due to their very similar structure, however the correlations between the recovered rewards and their correspondent ground truths are higher.

8. Discussion on Generalization to Different Frameworks

Even though reward recovering in MA-LfL is based on the assumption of agents are using MA-SPI to optimize their policies, MA-LfL could potentially be used when agents optimize their policies with different models as demonstrated in single-agent case (Jacq et al., 2019).

We expect MA-LfL to perform well with learning frameworks that have similar characteristics to SPI. SPI is an on-policy algorithm which it makes it easier for an observing agent to infer the policies from trajectories generated with a fixed policy. Off-policy algorithms such as SAC for continuous environments and Soft Q-learning for discrete environments might be desirable by practitioners because of sample efficiency, but the fact that constant updates of the policies after each step requires some care to compensate potential errors in inferring the policy of other agents from the generated trajectories. Another important characteristic of SPI is that it optimizes a stochastic policy which encourages exploration while agents optimize their own policy, especially in the sparse reward cases. Since PPO maintains both characteristics, it would be reasonable to expect MA-LfL to perform well under this learning framework as an alternative to SPI, in continuous and high-dimensional environments.

9. Conclusion

We propose MA-LfL, a multi-agent algorithm that allows inverse reinforcement learning in an entropy-regularized reinforcement learning setting. The input data of our algorithm are trajectories produced by agents that are not assumed to be in any equilibrium, but rather are learning according to a multi-agent soft policy iteration (MA-SPI). The reward functions recovered by MA-LfL in our experiments show high correlation with the ground truth ones.

Some of the potential applications of our MA-LfL algorithm are: imitation learning from multi-agent systems which have not yet reached an equilibrium, allowing the use of MARL algorithms that explicitly use the knowledge of all the agents reward functions in scenarios where those are not accessible, the promotion of fairness or to further collaboration in social dilemmas such as the Prisoner's Dilemma by letting each agent being aware of other agents' rewards. However, since MA-LfL allows agents to recover rewards only up to a shaping, some care is required, especially in scenarios with many agents.

As a future work, it would be valuable to study generalization of MA-LfL over different learning frameworks and more experimental investigations would provide useful insights. Investigating the scalability and performance on partially observable scenarios would also be worthful.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF) under Grant 01IS20051 and the Cyber Valley under Grant CyVy-RF-2021-20.

We thank to Glenn Angrabeit for his support on the experiments. We would also like to thank the anonymous ICML reviewers for their valuable feedback on the manuscript.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pp. 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

Barrett, S., Rosenfeld, A., Kraus, S., and Stone, P. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017.

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361. PMLR, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Jacq, A., Geist, M., Paiva, A., and Pietquin, O. Learning from a learner. In *International Conference on Machine Learning*, pp. 2990–2999. PMLR, 2019.

Le, H. M., Yue, Y., Carr, P., and Lucey, P. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pp. 1995–2003. PMLR, 2017.

Lin, X., Beling, P. A., and Cogill, R. Multiagent inverse reinforcement learning for two-person zero-sum games. *IEEE Transactions on Games*, 10(1):56–68, 2017.

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

Natarajan, S., Kunapuli, G., Judah, K., Tadepalli, P., Kersting, K., and Shavlik, J. Multi-agent inverse reinforcement learning. In *2010 ninth international conference on machine learning and applications*, pp. 395–400. IEEE, 2010.

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

Ramponi, G., Drappo, G., and Restelli, M. Inverse reinforcement learning from a gradient-based learner. *Advances in Neural Information Processing Systems*, 33:2458–2468, 2020.

Reddy, T. S., Gopikrishna, V., Zaruba, G., and Huber, M. Inverse reinforcement learning for decentralized non-cooperative multiagent systems. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1930–1935, 2012. doi: 10.1109/ICSMC.2012.6378020.

Song, J., Ren, H., Sadigh, D., and Ermon, S. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 31, 2018.

Šošić, A., KhudaBukhsh, W. R., Zoubir, A. M., and Koeppl, H. Inverse reinforcement learning in swarm systems. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 1413–1421, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tangkaratt, V., Han, B., Khan, M. E., and Sugiyama, M. Variational imitation learning with diverse-quality demonstrations. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9407–9417. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/tangkaratt20a.html.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4950–4957, 2018.

Yu, L., Song, J., and Ermon, S. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 7194–7201. PMLR, 2019.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A. Proofs of the error bounds for the single-agent case

For the sake of clarity, we start by presenting a single agent version of the theorems in Section 6. In Appendix B, we will discuss how to extend them to the multi-agent case. In this section, we use terminology of *Learner* and *Observer* as in (Jacq et al., 2019), where the Learner is the RL agent and the Observer is the IRL algorithm.

A.1. Single-agent setting

In the single-agent LfL setting (Jacq et al., 2019), an agent, called the *Learner*, is learning to solve a Markov Decision Process $M = (\mathcal{S}, \mathcal{A}, P, R, P_0, \gamma)$ via soft-policy iteration. Namely, the Learner starts with a policy π_0 and it subsequantially improves to $\pi_1 \propto \exp \frac{Q^{\pi_0}_{\text{soft}}(s,a)}{\alpha}$, then π_1 will be improved to $\pi_2 \propto \exp \frac{Q^{\pi_1}_{\text{soft}}(s,a)}{\alpha}$ and so on. The *Observer*, namely the IRL algorithm, perceives trajectories generated by the policies $\pi_0, \pi_1, \ldots, \pi_K$ of the Learner agent and infers its reward function R.

A.2. Reward recovery error bounds

Let π be a policy for the Learner and let $\hat{R}$ an estimation of the reward R maintained by the Observer. In the following we denote by $Q^{\pi,R}_{\text{soft}}$ the actual soft Q function for π, and $Q^{\pi,\hat{R}}_{\text{soft}}$ the soft Q function for π computed w.r.t. $\hat{R}$. Namely

$$Q^{\pi,R}_{\text{soft}}(s,a) = R(s,a) + \gamma \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t (R(s_t,a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right]$$

and

$$Q^{\pi,\hat{R}}_{\text{soft}}(s,a) = \hat{R}(s,a) + \gamma \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t (\hat{R}(s_t,a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right].$$

Note that the Observer can learn $Q^{\pi,\hat{R}}_{\text{soft}}$ from the trajectories produced by the policy π of the Learner using its $\hat{R}$ of R. The Observer can then use $Q^{\pi,\hat{R}}_{\text{soft}}$ to predict the future soft policy improvement of the Learner. Theorem A.2 and Theorem A.1 quantify the error of the reward estimation in terms of the soft policy improvement prediction error.

Theorem A.1. *Let R be the actual reward and let $\hat{R}$ be an estimation recovered by the Observer. Then if*

$$\sup_{a \in \mathcal{A}, s \in \mathcal{S}} |\ln \pi_{new}(a|s) - \ln \hat{\pi}_{new}(a|s)| < \delta,$$

then there exists a shaping sh such that

$$\sup_{a \in \mathcal{A}, s \in \mathcal{S}} \left| \hat{R}(s,a) - (R + sh)(s,a) \right| < \varepsilon,$$

where $\varepsilon = \delta\alpha(1+\gamma)$.

Proof. From the definition of soft policy improvement, we have that for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$

$$\ln \pi_{\text{new}}(a|s) - \ln \hat{\pi}_{\text{new}}(a|s) = \frac{1}{\alpha} \left(Q^{\pi,R}_{\text{soft}}(s,a) - Q^{\pi,\hat{R}}_{\text{soft}}(s,a) + f(s) \right),$$

where $f(s) = \ln \hat{Z}(s) - \ln Z(s)$, and $Z(s)$ and $\hat{Z}(s)$ are the normalizing terms $Z(s) = \sum_{\tilde{a} \in \mathcal{A}} e^{\frac{Q^{\pi,R}_{\text{soft}}(s,\tilde{a})}{\alpha}}$ and $\hat{Z}(s) = \sum_{\tilde{a} \in \mathcal{A}} e^{\frac{Q^{\pi,\hat{R}}_{\text{soft}}(s,\tilde{a})}{\alpha}}$. From Lemma 1 in (Jacq et al., 2019), we have that there exists a shaping $sh \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that

$$\ln \pi_{\text{new}}(a|s) - \ln \hat{\pi}_{\text{new}}(a|s) = \frac{1}{\alpha} \left(Q^{\pi,R+sh}_{\text{soft}}(s,a) - Q^{\pi,\hat{R}}_{\text{soft}}(s,a) \right).$$

From the soft Bellman equations for $Q_{\text{soft}}^{\pi,R+sh}$ and $Q^{\pi,\hat{R}}$, we have

$$\begin{aligned}\left|\hat{R}(s,a)-(R+sh)(s,a)\right| &= \left|\left[Q_{\text{soft}}^{\pi,\hat{R}}(s,a)-Q_{\text{soft}}^{\pi,R+sh}(s,a)\right]-\gamma \mathop{\mathbb{E}}_{\substack{s'\sim P\\ a'\sim\pi}}[Q_{\text{soft}}^{\pi,\hat{R}}(s',a')-Q_{\text{soft}}^{\pi,R+sh}(s',a')]\right| \\ &= \alpha\left|\ln\hat{\pi}_{\text{new}}(a|s)-\ln\pi_{\text{new}}(a|s)-\gamma \mathop{\mathbb{E}}_{\substack{s'\sim P\\ a'\sim\pi}}[\ln\hat{\pi}_{\text{new}}(a'|s')-\ln\pi_{\text{new}}(a'|s')]\right| \\ &\leq \alpha\delta\,(1+\gamma)\,.\end{aligned}$$

Therefore

$$\sup_{a,s}\left|\hat{R}(s,a)-(R+sh)(s,a)\right| \leq \alpha\delta\,(1+\gamma)\,.$$

□

In the same spirit as Theorem A.1, the following theorem provides an error bound on the recovered reward function in terms on the soft policy prediction error. Here the error bound does depend also on the size of the action space $|\mathcal{A}|$ and on the gap $\Delta = \sup R - \inf R$. However, instead of assuming a strong bound on the difference between the logarithm of the predicted improvement and the logarithm of the actual one, here it is enough to use a bound on the KL-divergence.

Theorem A.2 (Single-agent LfL)**.** *Let R be the actual reward and let $\hat{R}$ be an estimation recovered by the Observer. Let us assume $\mathcal{A}$ to be finite and let $\Delta = \sup_{a\in\mathcal{A},s\in\mathcal{S}} R(s,a) - \inf_{a\in\mathcal{A},s\in\mathcal{S}} R(s,a)$. Let π be a policy for the Learner. Let $\hat{\pi}_{new} \propto \exp\frac{Q_{soft}^{\pi,\hat{R}}}{\alpha}$ be the soft policy improvement predicted by the Observer. Let $\delta > 0$ be such that*

$$\sup_{s\in\mathcal{S}} D_{KL}(\pi_{new}(\cdot|s)|\hat{\pi}_{new}(\cdot|s)) < \delta,$$

then there exists a shaping $sh\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ such that for every $s\in\mathcal{S}$

$$|\mathbb{E}_{a\sim\pi_{new}}\left[\hat{R}(s,a)\right] - \mathbb{E}_{a\sim\pi_{new}}\left[(R+sh)(s,a)\right]| < \varepsilon,$$

where

$$\varepsilon = \delta\left(1+|\mathcal{A}|e^{\frac{\Delta}{\alpha(1-\gamma)}}\right).$$

Proof. From the definition of soft policy improvement, we have that for every $s\in\mathcal{S}$ and $a\in\mathcal{A}$

$$\ln\pi_{\text{new}}(a|s) - \ln\hat{\pi}_{\text{new}}(a|s) = \frac{1}{\alpha}\left(Q_{\text{soft}}^{\pi,R}(s,a) - Q_{\text{soft}}^{\pi,\hat{R}}(s,a) + f(s)\right),$$

where $f(s) = \ln\hat{Z}(s) - \ln Z(s)$, and $Z(s)$ and $\hat{Z}(s)$ are the normalizing terms $Z(s) = \sum_{\tilde{a}} e^{\frac{Q_{\text{soft}}^{\pi,R}(s,\tilde{a})}{\alpha}}$ and $\hat{Z}(s) = \sum_{\tilde{a}} e^{\frac{Q_{\text{soft}}^{\pi,\hat{R}}(s,\tilde{a})}{\alpha}}$. From Lemma 1 in (Jacq et al., 2019), we have that there exists a shaping $sh\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ such that

$$\ln\pi_{\text{new}}(a|s) - \ln\hat{\pi}_{\text{new}}(a|s) = \frac{1}{\alpha}\left(Q_{\text{soft}}^{\pi,R+sh}(s,a) - Q_{\text{soft}}^{\pi,\hat{R}}(s,a)\right)$$

From the Bellman equation for the soft Q function, we have

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}[\hat{R}(s,a)-(R+sh)(s,a)]\right| \\
&=\left|\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}\left[Q^{\pi,R+sh}_{\text{soft}}(s,a)-Q^{\pi,\hat{R}}_{\text{soft}}(s,a)-\gamma\mathbb{E}_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\left[Q^{\pi,\hat{R}}_{\text{soft}}(s',a')-Q^{\pi,R+sh}_{\text{soft}}(s',a')\right]\right]\right| \\
&\leq\left|\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}\left[Q^{\pi,\hat{R}}_{\text{soft}}(s,a)-Q^{\pi,R+sh}_{\text{soft}}(s,a)\right]\right|+\gamma\left|\mathop{\mathbb{E}}_{\substack{a\sim\pi_{\text{new}}\\ s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\left[Q^{\pi,R+sh}_{\text{soft}}(s',a')-Q^{\pi,\hat{R}}_{\text{soft}}(s,a)\right]\right| \\
&=\left|\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}[\ln\pi_{\text{new}}(a|s)-\ln\hat{\pi}_{\text{new}}(a|s)]\right|+\gamma\left|\mathop{\mathbb{E}}_{\substack{a\sim\pi_{\text{new}}\\ s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}[\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}_{\text{new}}(a'|s')]\right| \\
&=D_{\text{KL}}\left(\pi_{\text{new}}(\cdot|s)\|\hat{\pi}_{\text{new}}(\cdot|s)\right)+\gamma\left|\mathop{\mathbb{E}}_{\substack{a\sim\pi_{\text{new}}\\ s'\sim P(s,a)\\ a'\sim\pi(\cdot|s')}}[\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}_{\text{new}}(a'|s')]\right|
\end{aligned}$$

Let us now analyze the second term on right-hand side. Our goal is to bound it with the expectation w.r.t. to π_{new} so we can use our assumption on the KL-divergence between π_{new} and $\hat{\pi}_{\text{new}}$.

$$\begin{aligned}
\left|\mathbb{E}_{a'\sim\pi(\cdot|s')}[\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}(a'|s')]\right| &=\left|\sum_{a'\in\mathcal{A}}\pi(a'|s')\left(\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}_{\text{new}}(a'|s')\right)\right| \\
&=\left|\sum_{a'\in\mathcal{A}}\pi_{\text{new}}(a'|s')(\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}_{\text{new}}(a'|s'))\left(\frac{\pi(a'|s')}{\pi_{\text{new}}(a'|s')}\right)\right| \\
&\overset{(*)}{\leq}|\mathcal{A}|e^{\frac{\Delta}{\alpha(1-\gamma)}}\sup_s\left|\sum_{a'\in\mathcal{A}}\pi_{\text{new}}(a'|s')(\ln\pi_{\text{new}}(a'|s')-\ln\hat{\pi}_{\text{new}}(a'|s'))\right| \\
&\leq|\mathcal{A}|e^{\frac{\Delta}{\alpha(1-\gamma)}}\delta.
\end{aligned}$$

The inequality $(*)$ follows from the following observation. Observe that

$$\frac{\pi(a'|s')}{\pi_{\text{new}}(a'|s')}=\pi(a'|s')\sum_{\tilde{a}\in\mathcal{A}}e^{\frac{Q^{\pi,R}_{\text{soft}}(s',\tilde{a})-Q^{\pi,R}_{\text{soft}}(s',a')}{\alpha}}\leq|\mathcal{A}|e^{\frac{\Delta}{\alpha(1-\gamma)}}.$$

The last inequality follows from the fact that $Q^{\pi,R}_{\text{soft}}(s,a)-Q^{\pi,R}_{\text{soft}}(s',a')\leq\frac{1}{1-\gamma}\Delta$, for every $s,s'\in\mathcal{S}$ and $a,a'\in\mathcal{A}$. □

A.3. Soft Policy Improvement prediction error bound

As discuss in the previous section, the Observer can use the recovered reward to predict the next soft policy improvements of the Learner. Here we prove an error bound of the prediction in terms of the reward estimation error.

Theorem A.3 (Single agent LfL.)**.** *Let R be the actual reward and let $\hat{R}$ be an estimation recovered by the Observer. Let π_{new} is the actual policy improvement of the policy π and $\hat{\pi}_{new}$ is the predicted policy improvement of the actual policy π using recovered reward $\hat{R}$. If there exist $\delta>0$ and a shaping $sh\colon\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ such that*

$$\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left|\hat{R}(s,a)-(R+sh)(s,a)\right|<\delta$$

then

$$\sup_{s\in\mathcal{S},a\in\mathcal{A}}D_{KL}(\pi_{new}|\hat{\pi}_{new})<\varepsilon$$

where

$$\varepsilon=\delta\left(\frac{1}{\alpha(1-\gamma)}+\frac{|\mathcal{A}|}{e^{\inf R}}\right).$$

Proof. For every $s \in \mathcal{S}$

$$\begin{aligned}|D_{\text{KL}}\left(\pi_{\text{new}}(\cdot|s)|\hat{\pi}_{\text{new}}(\cdot|s)\right)| &= \mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}[\ln \pi_{\text{new}}(a|s) - \ln \hat{\pi}(a|s)] \\ &= \frac{1}{\alpha}\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}\left[Q_{\text{soft}}^{R+sh,\pi}(s,a) - \alpha \ln Z(s) - Q_{\text{soft}}^{\hat{R},\pi}(s,a) + \alpha \ln \hat{Z}(s)\right] \\ &= \frac{1}{\alpha}\mathop{\mathbb{E}}_{a\sim\pi_{\text{new}}}\left[\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t((R+sh)(s_t,a) - \hat{R}(s_t,a_t))\right]\right] - (\ln Z(s) - \ln \hat{Z}(s)),\end{aligned}$$

where $Z(s)$ and $\hat{Z}(s)$ are the normalizing factors

$$Z(s) = \sum_{\tilde{a}\in\mathcal{A}} e^{\frac{Q_{\text{soft}}^{R+sh,\pi}(s,\tilde{a})}{\alpha}} \qquad \hat{Z}(s) = \sum_{\tilde{a}\in\mathcal{A}} e^{\frac{Q_{\text{soft}}^{\hat{R},\pi}(s,\tilde{a})}{\alpha}}.$$

Therefore, using the assumption and the property that $\ln\left(\frac{x}{y}\right) = \ln\left(\frac{x-y}{y}+1\right) \leq \frac{x-y}{x}$, we have

$$\begin{aligned}|D_{\text{KL}}\left(\pi_{\text{new}}(\cdot|s)|\hat{\pi}_{\text{new}}(\cdot|s)\right)| &\leq \frac{\delta}{\alpha(1-\gamma)} + \frac{\hat{Z}(s) - Z(s)}{Z(s)} \\ &\leq \delta\left(\frac{1}{\alpha(1-\gamma)} + \frac{|\mathcal{A}|}{e^{\frac{\inf R}{\alpha}}}\right).\end{aligned}$$

□

B. Proofs of the error bounds in the multi-agent case

Here we include the proofs of Theorem 6.1, Theorem 6.3 and Theorem6.5 which are the multi-agent extensions of Theorem A.1, Theorem A.2 and Theorem A.3 in Appendix A.

Proof of Theorem 6.1. Similar to the proof of Theorem A.1, we can write

$$\ln \pi_{\text{new}}(a^i|s) - \ln \hat{\pi}_{\text{new}}(a^i|s) = \frac{1}{\alpha}\left(\mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q_{\text{soft}}^{\boldsymbol{\pi},R^i+sh}(s,\boldsymbol{a}^{-i},a^i) - Q_{\text{soft}}^{\boldsymbol{\pi},\hat{R}}(s,\boldsymbol{a}^{-i},a^i)\right]\right), \tag{15}$$

for a certain shaping $sh\colon \mathcal{S}\times\mathcal{A}^1\times\ldots\mathcal{A}^N \to \mathbb{R}$.

Using (15) and the Bellman equation (3), we can write

$$\begin{aligned}&\left|\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\hat{R}_j^i(s,\boldsymbol{a}^{-i},a^i) - (R^i+sh)(s,\boldsymbol{a}^{-i},a^i)\right]\right| \\ &\qquad = \alpha\Big|\ln\hat{\pi}_{\text{new}}^i(a^i|s) - \ln\pi_{\text{new}}(a^i|s) - \gamma\mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}\\ s'\sim P(\cdot|s,\boldsymbol{a})\\ \tilde{a}^i\sim\pi^i(\cdot|s')}}\left[\ln\hat{\pi}_{\text{new}}^i(\tilde{a}^i|s') - \ln\pi_{\text{new}}(\tilde{a}^i|s')\right]\Big| \leq \alpha\delta(1+\gamma).\end{aligned} \tag{16}$$

□

Proof of Theorem 6.3. As in the proof of Theorem A.2, there exists a shaping $sh\colon \mathcal{S}\times\mathcal{A}^i \to \mathbb{R}$, that depends only on the state and on the action of agent i, such that

$$\ln \pi_{\text{new}}(a^i|s) - \ln \hat{\pi}_{\text{new}}(a^i|s) = \frac{1}{\alpha}\left(\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q_{\text{soft}}^{\boldsymbol{\pi},R^i+sh}(s,\boldsymbol{a}^{-i},a^i) - Q_{\text{soft}}^{\boldsymbol{\pi},\hat{R}}(s,\boldsymbol{a}^{-i},a^i)\right]\right).$$

Therefore, following the same idea as in the proof of Theorem A.2, we can write

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}[\hat{R}^i(s)-(R^i+sh)(s,a^i)]\right| \\
&\quad\le\left|\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\ln\pi^i_{\text{new}}(a^i|s)-\ln\hat{\pi}^i_{\text{new}}(a^i|s)\right]\right|+\gamma\left|\mathop{\mathbb{E}}_{\substack{a^i\sim\pi^i_{\text{new}}\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}(\cdot|s)\\ s'\sim P(\cdot|s,\boldsymbol{a})\\ \tilde{a}^i\sim\pi^i(\cdot|s')}}\left[\ln\pi^i_{\text{new}}(\tilde{a}^i|s')-\ln\hat{\pi}^i_{\text{new}}(\tilde{a}^i|s')\right]\right| \\
&\quad= D_{\text{KL}}(\pi^i_{\text{new}}(\cdot|s)\|\hat{\pi}_{\text{new}}(\cdot|s))+\gamma\left|\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}\\ s'\sim P(\cdot|s,\boldsymbol{a}^{-i},a^i)}}\left[\mathop{\mathbb{E}}_{\tilde{a}^i\sim\pi^i}\left[\ln\pi^i_{\text{new}}(\tilde{a}^i|s')-\ln\hat{\pi}_{\text{new}}(\tilde{a}^i|s')\right]\right]\right]\right|. \quad (17)
\end{aligned}$$

As in the proof of Theorem A.2, we would like the most inner expectation of the second term in the right-hand side to be w.r.t. π^i_{new}, in order to express it in terms of the KL-divergence. To achieve that, we can similarly bound the ratio $\frac{\pi^i}{\pi^i_{\text{new}}}$ as follows

$$\frac{\pi^i(a^i|s')}{\pi^i_{\text{new}}(a^i|s')}=\pi(a^i|s')\sum_{a^i\in\mathcal{A}^i}e^{\frac{\mathbb{E}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}[Q^{\boldsymbol{\pi},R^i}_{\text{soft}}(s',\boldsymbol{a}^{-i},a^i)-Q^{\boldsymbol{\pi},R^i}_{\text{soft}}(s',\boldsymbol{a}^{-i},a^i)]}{\alpha}}\le|\mathcal{A}^i|e^{\frac{\Delta^i}{\alpha(1-\gamma)}},$$

where $\Delta^i=\sup_{s\in\mathcal{S}}R^i(s)-\inf_{s\in\mathcal{S}}R^i(s)$. This allows us to conclude that

$$\left|\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\hat{R}^i(s)-(R^i+sh)(s,a^i)\right]\right|\le\delta\left(1+\gamma|\mathcal{A}^i|e^{\frac{\Delta^i}{\alpha(1-\gamma)}}\right).$$

□

Proof of Theorem 6.5. Similarly to the proof of Theorem A.3, we have

$$\begin{aligned}
D_{\text{KL}}\left(\pi^i_{\text{new}}(\cdot|s)|\hat{\pi}^i_{\text{new}}(\cdot|s)\right)&=\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\ln\pi^i_{\text{new}}(a^i|s)-\ln\hat{\pi}^i(a^i|s)\right] \\
&=\frac{1}{\alpha}\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[Q^{\boldsymbol{\pi},R^i+sh}_{\text{soft}}(s,\boldsymbol{a}^{-i},a^i)-Q^{\boldsymbol{\pi},\hat{R}^i}_{\text{soft}}(s,\boldsymbol{a}^{-i},a^i)\right]\right]+\alpha\ln\hat{Z}(s)-\alpha\ln Z(s) \\
&=\frac{1}{\alpha}\mathop{\mathbb{E}}_{a^i\sim\pi^i_{\text{new}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{a}\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t((R^i+sh)(s_t,\boldsymbol{a}_t)-\hat{R}(s_t,\boldsymbol{a}_t))\right]\right]-(\ln Z(s)-\ln\hat{Z}(s)),
\end{aligned}$$

where $Z(s)$ and $\hat{Z}(s)$ are the normalizing terms.

Following the same argument in the proof of Theorem A.3 we get

$$D_{\text{KL}}\left(\pi^i_{\text{new}}(\cdot|s)|\hat{\pi}^i_{\text{new}}(\cdot|s)\right)\le\delta\left(\frac{1}{\alpha(1-\gamma)}+\frac{|\mathcal{A}^i|}{e^{\frac{\min R^i}{\alpha}}}\right).$$

□

C. Experiments

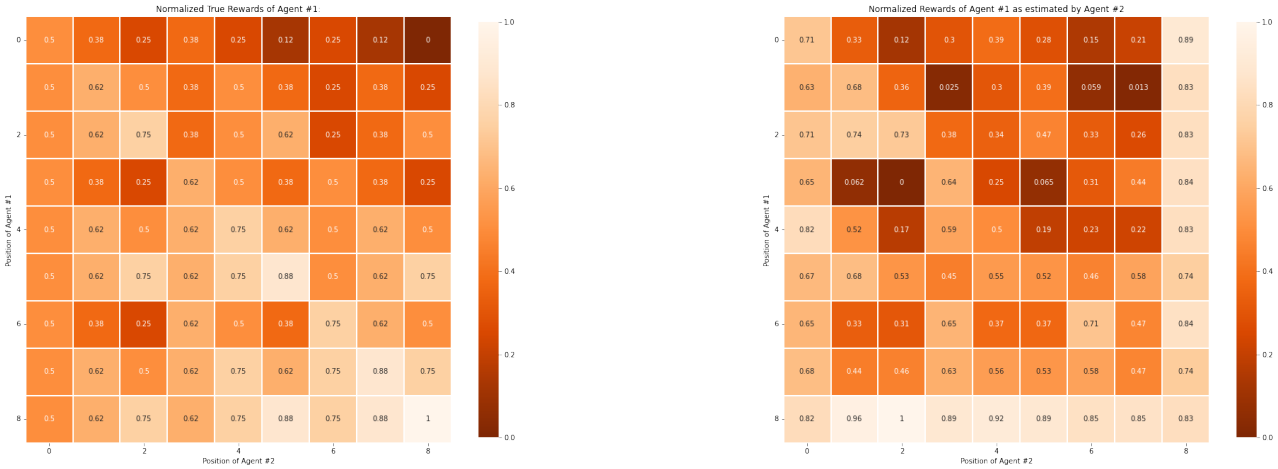
In this section, we provide the details of our experimental evaluations. We execute all experiments under a Conda environment using Python with a computation unit GPU-2080i and the source code is available at GitHub [1].

In Fig. 4 and Fig. 5 we present the visualizations of heterogeneous and homogeneous reward cases respectively.

[1] https://github.com/melodiCyb/multiagent-learning-from-learners

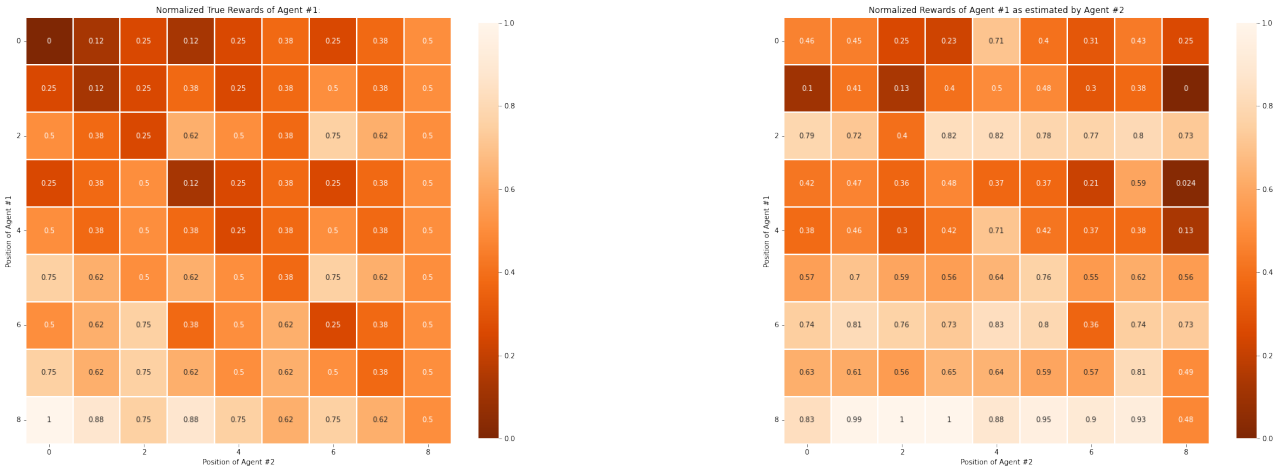
Parameter	Value
Alpha	3
Beta	0.1
Gamma	0.9
Episode Length	1000
Iteration #	10
Episode #	3000
Entropy Coefficient	0.3
Adam Learning Rate	0.1
Adam Epoch #	10
Reward Adam Epoch #	1000
Reward Adam Learning Rate	0.01

Table 3. Parameters to reproduce results for MA-LfL in Grid World scenario in Section 7 Table 1.



(a) Normalized true rewards w.r.t M_{het}. (b) Normalized recovered rewards w.r.t M_{het}.

Figure 4. M_{het} case for Agent $\#1$ with MA-SPI.



(a) Normalized true rewards w.r.t M_{hom}. (b) Normalized recovered rewards w.r.t M_{hom}.

Figure 5. M_{hom} case for Agent $\#1$ with MA-SPI.