Imitation Learning From Inconcurrent Multi-Agent Interactions

Xin Zhang\textsuperscript{*1}, Weixiao Huang\textsuperscript{*1}, Yanhua Li\textsuperscript{1}, Renjie Liao\textsuperscript{2}, Ziming Zhang\textsuperscript{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 iMA-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 iMA-IL model can better infer the policy of each expert agent using their demonstration data collected from inconcurrent decision-making scenarios.

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, but also 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 (iMA-IL): A group of experts provide demonstration data when playing a Markov game (MG) with an inconcurrent decision-making process, and iMA-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 iMA-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 is denoted as a tuple \( (N, S, \mathcal{A}, Y, \mathcal{Z}, P, \eta, r, \gamma) \) with a set of states \( S \) and \( N \) sets of actions \( \{A_i\}_{i=1}^{N} \). At each time step \( t \) with a state \( s_t \in S \), 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.
\textsuperscript{1}Xin Zhang, Weixiao Huang, Yanhua Li and Ziming Zhang are with Worcester Polytechnic Institute (WPI), USA, xzhang17, whuang2, ylii15, zzhang15@wpi.edu.
\textsuperscript{2}Renjie Liao is with the University of Toronto and Vector Institute, Toronto, ON MSG 1M1, Canada, rjliiao@cs.toronto.edu.

\textsuperscript{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.
work from MAGAIL and be consistent with the literature [5].

Agent $i$'s own actions.

inconcurrent decisions (e.g., turn-based games such as the Go game) resolves 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) $s \in S^{MULTI}$, 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 $s \in S$ (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., $\forall i \in [N], \forall \pi_{i} \neq \pi_i, \mathbb{E}_{\pi_{i},\pi_{-i},Y}[r_{t}] \geq \mathbb{E}_{\pi_{i},\pi_{-i},Y}[r_{t}]$. Therefore, our constrained optimization problem is ([7], Theorem 3.7.2)

\[
\min_{\pi,v} f_{\pi}(\pi,v) = \sum_{i=1}^{N} \sum_{s,t \in S, h \in H} v_i(s|h) - \mathbb{E}_{\pi_i,s_t,h}[q_i(s,a_i|h)]
\]

s.t.

\[
v_i(s|h) \geq q_i(s,a_i|h) \forall i \in [N], s \in S, a_i \in A_i, h \in H,
\]

\[
v \triangleq [v_1, \ldots ; v_N].
\]

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

\[
q_i(s_t,a_t|h_{t-1}) = \mathbb{E}_{\pi_{-i},Y} [Y(i|h_{t-1})r_i(s_t,a_t) + \gamma \sum_{I_t \in I} \mathbb{P}(I_t|h_{t-1}) \sum_{s_t+1 \in S} P(s_{t+1}|s_t,a_t) v_i(s_{t+1}|h_{t+1})].
\]

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].

2Because of the inconcurrent setting, the rewards only depend on agents’ own actions.
We use $\ddot{\text{iMA-RL}}(r)$ to denote the set of policies that form an SPE under reward function $r$, and can maximize $\gamma$-discounted causal entropy of policies:

$$\ddot{\text{iMA-RL}}(r) = \arg \min_{\pi \in \Pi, v} f_t(\pi, v) - H(\pi),$$

subject to $v_i(s(h)) \geq q_i(s, a_i|h) \ \forall i \in [N], s \in S, a_i \in A_i, \forall h \in \mathcal{H},$

where $q_i$ is defined in eq. (2). Our objective is to define a suitable inverse operator $\ddot{\text{iMA-IRL}}$. 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 define 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 $\ddot{\text{iMA-RL}}$. The general idea is consistent with MAGAIL [25].

The updated $(t+1)$-step constraints are:

$$\dot{v}_i(s^{(t)}; \pi, r, \zeta) \geq Q_i^{(t)}(s^{(t)}, a^{(t)}); \pi, r, h_{t-1}),$$

where $s^{(t)} = (s(t), a^{(t)})$, with $s(0)$ as initial state, $\lambda$ is a vector of $N \cdot |T^*_i| \cdot |\mathcal{H}|$ Lagrange multipliers, and $\dot{v}_i$ is defined as in Theorem 1 in Appx VI-A.

Theorem 2 illustrates that a specific $\lambda$ can 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 $\pi^*$ and $\pi$, let

$$\lambda^*_\pi(h_{t-1}) = \eta \sum_{s^{(t)}}(s^{(t)}) \prod_{t=1}^{t-1} \sum_{a_{t-1}^{(t)}} \pi^*_{a_{t-1}^{(t)}|s^{(t-1)}}$$

$$P(s^{(t+1)}|s^{(t)}, a^{(t)}) \prod_{a^{(t)}, h_{t-1}=1} \pi(a^{(t)}|s^{(t)})$$

be the probability of generating the sequence $\tau_i$ usung policy $\pi_i$, $\pi^*_i$ and $h_{t-1}$, where $P(h_{t-1}) = Pr(I_0) \prod_{k=1}^{t-1} Pr(I_k|h_{k-1})$ is the probability of history $h_{t-1}$.

Then

$$\lim_{t \to \infty} L^{(t+1)}(\pi^*, \lambda^*_\pi) =$$

$$\sum_{s} \sum_{a_i} \sum_{h_{t-1}} \sum_{s^{(t)}} \sum_{a^{(t)}} \pi^*_{a^{(t)}|s^{(t)}, h_{t-1}}$$

$$- \sum_{s} \sum_{a_i} \sum_{h_{t-1}} \sum_{s^{(t)}} \sum_{a^{(t)}} \pi(a^{(t)}|s^{(t)}, h_{t-1})$$

where the dual function is $L^{(t+1)}(\pi^*, \lambda^*_\pi)$ and each multiplier can be considered as the probability of generating a trajectory of agent $i \in [N], \pi_i \in \mathcal{T}^i$, and $h_{t-1} \in \mathcal{H}$.

Theorem 2 provides a horizon to establish $\ddot{\text{iMA-IRL}}$ objective function with regularizer $\psi$.

$$\ddot{\text{iMA-IRL}}(\psi)(\pi_E) = \arg \max_{\pi} -\psi(\pi) + \sum_{i=1}^{N} (\mathbb{E}^\pi_{E_i, Y}[r_i])$$

$$- \sum_{i=1}^{N} (\max_\pi \sum_{j=1}^{N} (\beta H_i(\pi_i) + \mathbb{E}^\pi_{E_i, Y}[r_i]),)$$

where $H_i(\pi_i) = \mathbb{E}_{\pi_i, \pi_{E \neq i}}[\sum_{j=1}^{N} \pi_i(a|s)]]$ is the discounted causal entropy for policy $\pi_i$ when other agents follow $\pi_{E \neq i}$, and $\beta$ 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 $\ddot{\text{iMA-IRL}}(\psi)(\pi_E) = \ddot{\text{MAIRL}}(\psi)(\pi_E)$; furthermore, if $N = 1$, $\beta = 1$ then $\ddot{\text{iMA-IRL}}(\psi)(\pi_E) = \text{IRL}(\psi)(\pi_E)$.

### C. Inconcurrent Multi-Agent Occupancy Measure Matching

We first define the inconcurrent occupancy measure in Markov games:

**Definition** For an agent $i \in [N]$ with a policy $\pi_i \in \Pi$ define its inconcurrent occupancy measure $\rho^e_{\pi_i} : S \times \mathcal{A} \cup \{\phi\} \rightarrow \mathbb{R}$ as $\rho^e_{\pi_i}(s, a)$.

$$\rho^e_{\pi_i}(s, a) = \left\{ \begin{array}{ll} \mathbb{E}_{\pi_i, \pi_{E \neq i}}[(1 - \zeta(i)) + \sum_{t=1}^{\infty} \gamma^t Pr(s_t = s|\pi_i, \pi_{E \neq i})Y(i|h_{t-1})], & \text{if } a \in \mathcal{A}_i, \\
\mathbb{E}_{\pi_i, \pi_{E \neq i}}[(1 - \zeta(i)) + \sum_{t=1}^{\infty} \gamma^t Pr(s_t = s|\pi_i, \pi_{E \neq i})(1 - Y(i|h_{t-1}))], & \text{if } a \in \{\phi\}. \end{array} \right.$$
where \( \pi_i, E_{-i} \) denotes \( \pi_i \) for agent \( i \), and \( \pi_{E_{-i}} \) for other agents.

In practice, we are only able to calculate \( \rho Е_{\pi} \) and \( \rho Е_{\pi_i} \). As following MAGAIL [25], we match the occupancy measure between \( \rho Е_{\pi} \) and \( \rho Е_{\pi_i} \) rather than \( \rho Е_{\pi} \) and \( \rho Е_{\pi_i,\pi_{E_{-i}}} \).

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 \( \psi_i \) in Proposition 1 (in Appx VI-B) contributes to the corresponding generative adversarial model where each agent \( i \) has a generator \( \pi_{\theta_i} \) and a discriminator, \( D_{\psi_i} \). 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:

\[
\min_{\theta} \max_{w} \mathbb{E}_{\pi_{\theta_i}, Y} \left[ \log D_{\psi_i}(s, a_i) \right] + \mathbb{E}_{\pi_{\theta_i}, Y} \left[ \log (1 - D_{\psi_i}(s, a_i)) \right].
\]

In practice, the input of \( \pi_{\theta_i} \) is \( Z \), the demonstration data from \( N \) expert agents in the same environment, where the demonstration data \( Z = \{(s_t, a_t)\}_{t=0}^{T} \) collected by sampling \( s_0 \sim \eta, I_0 \sim \zeta, I_{t+1} \sim Y, a \sim \pi^*(\cdot|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a) \). The assumptions include knowledge of \( N, \gamma, S, A \). Transition \( P \), initial state distribution \( \eta \), agent distribution \( \zeta \), 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 \( \pi_{\theta_i} \), we follow MAGAIL [25] to apply Multi-agent Actor-Critic with Kronecker-factors (MACK) and use the advantage function with baseline \( V_{\nu} \) 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 \( Y \) 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 \( Y \) 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, \( Y \) holds for some agent \( i \in [N] \) at time step \( t \), the agent \( i \) makes 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 iMA-IL with both stochastic and deterministic player function structures under cooperative games. We compared our iMA-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 9 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], then trained with iMA-IL under deterministic cooperative navigation, and stochastic cooperative navigation games. Fig. 3 shows the normalized rewards, when learning policies with BC, MAGAIL and iMA-IL, respectively.

When there is only a small amount of expert demonstrations, the normalized rewards of BC and iMA-IL increase, especially, when less demonstration data are used, i.e.,
less than 400 demonstrations. After a sufficient amount of demonstrations are used, i.e., more than 400, iMA-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 \(Y\), iMA-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)), iMA-IL performs consistently better than MAGAIL and BC. However, in the deterministic cooperative navigation game (Fig. 3(a)), with 200 demonstration, iMA-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 iMA-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, iMA-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 (iMA-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 iMA-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


A. Time Difference Learning

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

$$
\hat{v}_t(s^{(t)}; \pi, r, h_{t-1}) = E_t \left[ Y(i|h_{t-1})r(s^{(t)}, a^{(t)}) + \gamma \sum_{i \in \mathcal{I}} Pr(I_i|h_{t-1}) \sum_{s^{(t+1)} \in S} P(s^{(t+1)}|s^{(t)}, a^{(t)})\hat{v}_t(s^{(t+1)}) \right],
$$

$t \in \mathbb{N}^+, \forall s^{(t)} \in S, h_{t-1} \in \mathcal{H}$.

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

$$
\hat{q}_i^{(t)}(\{s^{(j)}, a^{(j)}\}_{j=0}^{t-1}, s^{(t)}, a^{(t)}; \pi, r, h_{t-1}) = \sum_{j=0}^{t-1} \gamma^j r_i(s^{(j)}, a^{(j)})I_{i,j} + \gamma E_t[Y(i|h_{t-1})r(s^{(t)}, a^{(t)}) + \gamma \sum_{i \in \mathcal{I}} Pr(I_i|h_{t-1}) \sum_{s^{(t+1)} \in S} P(s^{(t+1)}|s^{(t)}, a^{(t)})\hat{v}_t(s^{(t+1)}; \pi, r, h_{t-1})].
$$

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

$$
\hat{v}_i(s^{(0)}; \pi, r, q) \geq \hat{q}_i^{(t)}(\{s^{(j)}, a^{(j)}\}_{j=0}^{t-1}, s^{(t)}, a^{(t)}; \pi, r, h_{t-1}) \forall t \in \mathbb{N}^+, i \in [N], s^{(t)} \in S, a^{(t)} \in A_i, h_{t-1} \in \mathcal{H}.
$$

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

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

then

$$
\arg \min_{\pi} \sum_{i=1}^{N} \psi_i(r_i^\pi_{\pi_E} - r_i^\pi_E) = \arg \min_{\pi} \sum_{i=1}^{N} \psi_i(r_i^\pi_{\pi_E} - r_i^\pi_E) = \pi_E
$$