Agent-Level Inverse Reinforcement Learning for Mean Field Games through Policy Matching

Anonymous Author(s)

Abstract: Inverse reinforcement learning (IRL) algorithms can be used to recover a reward function from expert demonstrations that can explain expert behaviours. While IRL shows promise in many multi-agent tasks, its application is often hindered by a large number of agents, due to the explosion of the joint state-action space. Mean field games (MFG) are natural models of large-scale multi-agent systems, which simplifies the model through reducing the interactions among agents to those between an individual agent and the average effect from the population. This paper thus addresses the task of IRL for MFG, which aims to recover the agent-level (ground-truth) reward function from expert demonstrations sampled from a mean field Nash equilibrium (MFNE). We put forward a novel framework called Agent-Level Mean Field IRL (AL-MFIRL). The central idea of AL-MFIRL is to express the policy in terms of rewards and tune a reward function by optimising this policy so as to match the expert policy. We show that the learned reward function by AL-MFIRL necessarily interprets the expert behaviours as a MFNE under standard assumptions. To derive a practical algorithm, we adopt the smooth Boltzmann policy to approximate the elicited policy and optimises this policy using gradient methods. We evaluate AL-MFIRL on five simulated environments. Experiment results demonstrate the outperformance of AL-MFIRL on reward recovery and robustness against the changing environment dynamics.

Keywords: Inverse Reinforcement Learning, Mean Field Games, Mean Field Nash Equilibrium

1 Introduction

Multi-agent reinforcement learning (MARL) provides a general paradigm for decision making under uncertainty in multi-agent systems (MAS), where agents operate autonomously in a shared environment hoping to maximise self-centred long-term rewards [1]. Accompanied by advancements of deep neural networks, MARL has been instrumental in many tasks such as simulating social dilemmas [2], playing real-time strategy games [3], and optimising multi-robot control systems [4]. The majority of successful applications, however, deal with MAS with a small number of agents [5]. Yet, many scenarios involve a much larger population of individuals, such as traffic networks that contain millions of vehicles [6], online games with tens of thousands of players [7], and online businesses that face a large customer body [8]. Applying MARL to such systems faces two main challenges:

The first challenge is scalability of population size. Since changes in the policy of an agent affect others, an agent faces a non-stationary environment. Hence, an agent must incorporate into its training the actions of all other agents. This prevents existing MARL algorithms from being applied to large-scale MAS due to the exponential expansion of the joint state-action space [9]. In many situations, though, the system can be assumed to consist of homogeneous agents, i.e., the agents are identical, indistinguishable, and interchangeable [10]. This assumption warrants the formulation of the system behaviours in the asymptotic limit when the number of agents approaches infinity. The theory of mean field games (MFG) aims to analyse large-scale MAS in the asymptotic limit [11]. Instead of explicitising agents’ joint state-action, MFG uses a single entity, termed mean field, to denote the aggregated behaviours of the population at large. The game trajectories are thus determined...
by the interactions between agent-level and population-level dynamics of the system: The former concerns a single representative agent, while the latter provides an overview of the population. This reduction to a dual-view interplay motivates the mean field Nash equilibrium (MFNE) where the desired policy of an agent achieves a best-response to the mean field that is in turn consistent with the policy. Critically, MFNE is shown to be the MFG counterpart to Nash equilibrium in the sense that it constitutes an approximate Nash equilibrium in the corresponding finite-agent game [12].

The second challenge is reward engineering in modeling MAS. The reward function is a pivotal factor for RL to elicit agents' desired behaviours [13]. As a system grows, handcrafted rewards are increasingly prone to inaccuracies that lead to unexpected behaviours [14]. *Imitation learning* (IL) makes it possible to avoid manual reward engineering [15]. In particular, inverse reinforcement learning (IRL), an important IL paradigm, takes as input sampled executions of "well-behaved" policies, i.e., "expert demonstrations", and tunes the reward function so that the elicited agent’s behaviours align with the expert policy [16]. By revealing the reward function, IRL is useful not only for deducing agents’ intentions but also for re-optimising policies when the environment dynamics changes [17]. Although some multi-agent IRL methods demonstrate effectiveness in several MAS tasks [18, 19, 20, 21, 22, 23], they are not salable to MAS with a large population size. To automatically learn a suitable reward function for large-scale MAS, it is thus important to consider IRL for MFG, which aims to recover the reward function behind observed MFNE behaviours.

Notably, studying IRL for MFG is new in the literature albeit a recent attempt relies on reformulating MFNE as an optimal solution to a Markov decision processes (MDP) [24], which embodies the "collective will" of agents in the MFG. Accordingly, performing IRL on this MDP infers the population’s aggregated societal reward defined on policies and mean fields rather than states and actions. Although this "centralised" setting allows us to directly apply existing IRL methods for MDP to MFG, several implicit flaws are brought. First, the reformulation holds only for the MFNE that is socially optimal, as it presupposes that agents are cooperative to optimise a common societal reward. Thus, the inference would be biased if expert demonstrations are sampled from an ordinary (non-socially-optimal) MFNE as multiple MFNE might coexist [25, 26, 27, 28]. Second, sampling mean fields and policies at the population level requires the access to each agent’s state-action configuration at the agent level, which might result in sample inefficiency.

**Contribution.** This paper addresses the above two challenges in MARL while resolving the issues in the existing work. By focusing on agent behaviours in MFG, we aim to resolve the first challenge (scalability). To resolve the second challenge (reward engineering), we introduce Agent-Level Mean-Field IRL (AL-MFIRL), a novel IRL framework for MFG that intends to recover the agent-level (ground-truth) reward function from sampled plays in MFNE. The resulting MFG model avoids handcrafted reward function and facilitates agent-level optimal control in large-scale MAS. Thanks to the agent-level inference, AL-MFIRL for the first time realises an unbiased inference for the ground-truth reward function of MFG, and avoids the issue of sample inefficiency in population-level inference. In designing the AL-MFIRL framework, we note that the conventional IRL paradigms cannot guarantee the consistency between the elicited policy and the mean field in the demonstrated MFNE (See Section 4). We thus propose a different approach: instead of directly tuning the reward, we first construct a function that expresses the elicited policy in terms of rewards and then optimise this policy so as to match the observed expert policy. We call this technique mean-field policy matching and show that at optimality the learned reward function necessarily interpretes expert demonstrations as a MFNE (See Theorem 1). To derive a practical algorithm through gradient methods, we implement AL-MFIRL by using the (differentiable) Boltzmann policy to approximate the elicited policy. We evaluate AL-MFIRL in five simulated MFG environments. Experimental results demonstrate that AL-MFIRL can accurately recover the underlying reward function even if demonstrations are sampled from an ordinary MFNE. Moreover, AL-MFIRL shows higher sample efficiency and higher robustness against changing environment dynamics, outperforming the aforementioned population-level inference method.

## 2 Preliminaries

### 2.1 Mean Field Games

Mean field games (MFG) approximate the interactions in large-scale MAS. To simplify the exposition, throughout, we focus on discrete-time MFG with finite players, finite state and action spaces and a finite time horizon [29]. First, consider an $N$-player game with state space $S$ and action space $A$. A joint state of $N$ agents is a tuple $(s^1, \ldots, s^N) \in S^N$ where $s^i \in S$ is the state of
the $i$th agent. As $N$ goes large, instead of modelling each agent individually, MFG models a representative agent and collapses the joint state into an empirical distribution, called a mean field, given by $\mu(s) \triangleq \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{s^i=s\}}$, where $\mathbb{1}$ denotes the indicator function. To describe the game dynamics, write $P(\mathcal{X})$ for the set of probability distributions over set $\mathcal{X}$. The transition function $p: \mathcal{S} \times \mathcal{A} \times P(\mathcal{S}) \to P(\mathcal{S})$ specifies how states evolve at the agent-level, i.e., an agent’s next state depends on its current state, action, and the current mean field. This function also induces a transition function at the population-level which maps a current mean field to the next mean field based on all agents’ current states and actions. Let $T \in \mathbb{R}$ denote a finite time horizon. A mean field flow (MF flow for short) thus consists of a sequence of $T+1$ mean fields $\mu \triangleq \{\mu_t\}_{t=0}^{T}$, where the initial value $\mu_0$ is given, and each $\mu_t$ is the empirical distribution obtained by applying the transition above to $\mu_{t-1}$. Let $s_0, a_0, \ldots, s_T, a_T$ denotes a trajectory of states and actions of a fixed agent. The running reward of an agent at each step is specified by the reward function $r: \mathcal{S} \times \mathcal{A} \times P(\mathcal{S}) \to \mathbb{R}$. The agent’s long-term reward is thus the sum $\sum_{t=0}^{T} \gamma^t r(s_t, a_t, \mu_t)$, where $\gamma \in (0, 1)$ is the discounted factor. Summarising the above, MFG is defined as the tuple $(\mathcal{S}, \mathcal{A}, p, \mu_0, r, \gamma)$.

A time-varying stochastic policy is $\pi \triangleq \{\pi_t\}_{t=0}^{T}$ where $\pi_t: \mathcal{S} \to P(\mathcal{A})$ is the per-step policy at step $t$, i.e., $\pi_t$ directs the agent to choose action $a_t \sim \pi_t(\cdot|s_t)$. Given MF flow $\mu$ and policy $\pi$, the expected return (cumulative rewards) of an agent during the whole course of a game is given by $J(\mu, \pi) \triangleq \mathbb{E}_{s_0 \sim \mu_0, a_0 \sim \pi_0, s_1 \sim p, \ldots, s_T \sim p} \sum_{t=0}^{T} \gamma^t r(s_t, a_t, \mu_t)$.

### 2.2 Mean Field Nash Equilibrium

An agent seeks an optimal policy so as to maximise the expected return. If a MF flow $\mu$ is fixed, we will derive an induced MDP with a non-stationary transition function. An optimal policy of the induced MDP is called a best response to the corresponding fixed MF flow. We denote the set of all best-response policies to a fixed MF flow $\mu$ by $\Psi(\mu) \triangleq \arg \max_{\pi} J(\mu, \pi)$. However, since all agents optimise their policies simultaneously, the MF flow would shift. The solution thus needs to consider how the policy at the agent level affects MF flow at the population level. Since all agents are identical and rational, at optimality everyone would follow the same policy. Under this assumption, the dynamics of MF flow is governed by the (discrete-time) McKean-Vlasov (MKV) equation [30]:

$$\mu_{t+1}(s^i) = \sum_{s \in \mathcal{S}} \mu_t(s) \sum_{a \in \mathcal{A}} \pi_t(a|s) p(s'|s, a, \mu_t).$$

Denote $\mu = \Phi(\pi)$ as the MF flow induced by a policy that fulfills MKV system. The conventional solution concept for MFG is the mean field Nash equilibrium (MFNE), where agents adopt the same policy that is a best response to the MF flow, and in turn, the MF flow is consistent with the policy.

**Definition 1** (Mean Field Nash Equilibrium). A pair of MF flow and policy $(\mu^*, \pi^*)$ constitutes a mean field Nash equilibrium if $\pi^* \in \Psi(\mu^*)$ and $\mu^* = \Phi(\pi^*)$.

Shown in [31, 32], a MFNE is guaranteed to exist under the standard assumptions that both the reward function and the transition function are continuous and bounded. Through defining any mapping $\hat{\Psi}: \mu \to \pi$ that identifies a policy in $\Psi(\mu)$, we get a composition $\Gamma = \hat{\Phi} \circ \hat{\Psi}$, the so-called MFNE operator. Repeating the MFNE operator, we derive the fixed point iteration for the MF flow. The standard assumption for the uniqueness of MFNE is the contractivity $^{1}$ of $\Gamma$ [35, 34]. Note that the contractivity does not hold in general [32], which results in the coexistence of multiple MFNE.

### 3 Inverse Reinforcement Learning for MFG

Since MFG simplifies the model of large-scale MAS, many RL methods to compute MFNE and its variants have recently emerged [34, 25, 32]. However, RL agents are notoriously prone to unexpected behaviours due to the reward mis-specification. Consequently, these burgeoning RL methods for MFG can result in unexpected behaviours if the reward function does not capture all important aspects of a task [13]. Hand-tuning rewards for RL agents is challenging as it requires human domain knowledge. In MFG, since the reward of an agent is coupled with the population via the mean field, manually designing reward functions becomes more difficult. In this section, we address the task of inverse reinforcement learning (IRL) for MFG that aims to avoid manual specification of the reward function in a MFG by inferring the ground-truth reward function from expert demonstrations.

$^{1}$The distance between two mean fields is commonly measured by $\ell_1$-Wasserstein distance in the MFG literature [33, 34, 32]. Thus, the contractivity, if holds, is under the $\ell_1$-Wasserstein distance.
3.1 Problem Statement

Suppose we have no access to the ground-truth reward function \( r(s, a, \mu) \) but have a set of expert trajectories \( \mathcal{D}_E = (\tau_1, \ldots, \tau_M) \) sampled from a total number of \( M \) game plays under an unknown MFNE \((\mu^E, \pi^E)\), where a game play refers to an execution of expert policy \( \pi^E \) over all \( N \) agents. Each \( \tau_j = ((s^j_t, a^j_t, s^j_{t+1}))_{t=0}^T \) is the state-action trajectory of the \( j \)th agent in the \( j \)th game play.

Following the convention in IRL [36, 37, 23], we assume that \( \mathcal{D}_E \) provides the entire supervision signals, i.e., we cannot further communicate with the expert for additional information. IRL for MFG asks for a reward function under which \((\mu^E, \pi^E)\) achieves a MFNE.

3.2 The AL-MFIRL Framework

We next present our proposed method for IRL for MFG, which we call Agent-Level Mean Field IRL (AL-MFIRL), as it infers the ground-truth reward function using the trajectories of individual agents. The agent-level inference makes AL-MFIRL avoid the issues of biased inference and sample inefficiency in population-level inference [24], which we technically explain in detail in Sec. 4. We next give a formal description of AL-MFIRL.

Following standard practice, we assume that the reward function \( r \) is parameterised by \( \omega \in \mathbb{R}^d \) thus writing it as \( r_\omega \). Also assume \((\mu^E, \pi^E)\) is induced by some unknown true parameter \( \omega^* \), i.e., \((\mu^E, \pi^E) = (\mu^{E, \omega^*}, \pi^{E, \omega^*})\). IRL for MFG thus reduces to a search process for \( \omega^* \). Let \( \pi^\omega \) denote the best-response policy to \( \mu^E \) under the reward function \( r_\omega \). The central idea of AL-MFIRL is to tune \( \omega \) by matching \( \pi^\omega \) and \( \pi^E \). This technique, which we name as mean-field policy matching, is the key to AL-MFIRL. Now, let us elaborate the procedures of mean-field policy matching.

Since the true MF flow \( \mu^E \) is unknown, we use an estimation \( \hat{\mu}^E \equiv \{\hat{\mu}^E_t\}_{t=0}^T \). Due to the homogeneity of agents, we derive \( \hat{\mu}^E \) by averaging the frequencies of occurrences of states of all agents:

\[
\hat{\mu}^E_t(s) = \frac{1}{M} \sum_{j=1}^M \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{(s^j_t, t \rightarrow s)}.
\]

Similarly, we estimate the expert policy \( \pi^E \) by \( \hat{\pi}^E \equiv \{\hat{\pi}^E_t\}_{t=0}^T \) such that

\[
\hat{\pi}^E_t(a|s) = \frac{\sum_{j=1}^M \sum_{i=1}^N \mathbb{1}_{(s^j_t, t \rightarrow s, a^j_t, t \rightarrow a)}}{\sum_{j=1}^M \sum_{i=1}^N \mathbb{1}_{(s^j_t, t \rightarrow s)}}.
\]

The mean-field policy matching employs two key ingredients: The first is the mapping \( \hat{\Psi} \) in the MFNE operator \( \Gamma = \Phi \circ \hat{\Psi} \) that identifies a best-response policy \( \pi^\omega \) to the estimated MF flow \( \hat{\mu}^E \) under the reward function \( r_\omega \). The second ingredient is a loss function \( \mathcal{L} \) that measures the difference between the elicited policy \( \pi^\omega \) and the estimated expert policy \( \hat{\pi}^E \), given the estimated MF flow \( \hat{\mu}^E \). We can then succinctly express our task as the optimisation problem below:

\[
\min_{\omega} \mathcal{L}(\pi^\omega; \hat{\pi}^E, \hat{\mu}^E) \quad \text{where} \quad \pi^\omega = \hat{\Psi}(r_\omega; \hat{\mu}^E).
\]

To implement the AL-MFIRL framework, it remains to instantiate the \( \hat{\Psi} \) and \( \mathcal{L} \) functions above.

### 3.2.1 Identifying Best-Response Policies as Greedy Policies

The desired \( \hat{\Psi} \) mapping relies on the action value function for MFG, which represents the expected return a state-action pair under a MF flow \( \mu \) and a policy \( \pi \). Formally, we define it as follows:

\[
Q(t, s, a, \mu) \triangleq r(s, a, \mu) + \mathbb{E}_{\pi^\mu \sim \pi^\mu} \left[ \sum_{t'=t+1}^T \gamma^{t'-t} r(s_t, a_t, \mu_t) \right].
\]

We can use backward induction to recursively compute the action value function: starting from the terminal step \( t = T \), \( Q(T, s, a, \mu) = r(s, a, \mu_T) \); then for \( 0 \leq t < T \) we recursively compute:

\[
Q(t, s, a, \mu) = r(s, a, \mu_t) + \gamma \mathbb{E}_{\pi^\mu \sim \pi^\mu} \left[ Q(t+1, s', a', \mu_{t+1}) \right].
\]

For a fixed MF flow \( \mu \), we say a policy \( \pi^\star \) is a greedy with respect to \( \mu \) if \( \pi^\star_t (| s ) \) picks an action

\[
a_t \in \arg \max_{a \in A} Q^\star(t, s, a, \mu)
\]
uniformly at random, where
\[
Q^*(t, s, a, \mu) = r(s, a, \mu_t) + \gamma E_{\omega \sim p} \left[ \max_{\omega' \in \mathcal{A}} Q^*(t + 1, s, a, \mu) \right]
\]  
(8)
denotes the optimal action value function. Since \( \pi^* \) maximises the action value function for each time step, it achieves a best response to the corresponding MF flow \( \mu \). This provides the intuition to define the mapping \( \pi^\omega = \hat{\Psi}(r_\omega; \mu^E) \) by letting \( \pi^\omega \) be greedy with respect to \( \mu^E \) under the reward function \( r_\omega \). More formally, we write \( \pi_t^\omega(a|s) = \arg \max_{a \in A} Q^\omega_t(t, s, a, \mu^E) \), where \( Q^\omega_t(t, s, a, \mu^E) \) denotes the optimal action value function under the reward function \( r_\omega \). By Eq. (8) and Eq. (7), we can recursively compute \( \pi^\omega \) and \( Q^\omega \) using backward induction.

### 3.2.2 Measuring the Difference over Policies

The loss function \( \mathcal{L} \) measures differences between the elicited behaviours and the observed behaviours, which is captured by the following notion: Two policies \( \pi \) and \( \pi' \) are said to be consistent with respect to MF flow \( \mu \) if \( \pi_t(\cdot | s) = \pi'_t(\cdot | s) \) whenever \( \mu_t(s) > 0 \). With this intuition, we define our loss function based on Kullback-Leibler (KL) divergence \( D_{KL} \), which is commonly used in imitation learning to discriminate between the learned and the observed expert behaviours [38]. It is easy to see that \( \mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) = 0 \) whenever \( \pi^\omega \) is consistent with \( \tilde{\pi}^E \) with respect to \( \mu^E \).

\[
\mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) \triangleq \sum_{t=0}^T \sum_{s \in S} \mu^E_t(s) D_{KL}(\tilde{\pi}^E_t(\cdot | s) \parallel \pi^\omega_t(\cdot | s)).
\]  
(9)

Intuitively, the importance of the difference over two policies with respect to \( t \) and \( s \) is weighted by the proportion of \( s \) specified by the estimated expert mean field \( \mu^E_t \).

The following standard assumption in the MFG literature [34] ensures that with \( \hat{\Psi} \) and \( \mathcal{L} \) instantiated above, at optimality Eq. (4) can produce the true parameter \( \omega^* \).

**Assumption 1.** In a MFNE (\( \mu^*, \pi^* \)), \( \pi^* \) is greedy with respect to \( \mu^* \).

Intuitively, Assumption 1 can be viewed as an inductive bias over the MFNE. Since a best-response policy is generally not unique, the expert policy might not choose actions with highest \( Q \) values uniformly at random (e.g., any policy is a best response under a constant reward function). As a result, in general we cannot capture an arbitrary best-response using the greedy policy. While, with Assumption 1, we can sidestep this issue and ensure that at optimality \( \pi^\omega \) coincides with \( \pi^* \).

Now, we are ready to present our main result, which shows that the optimal solution to Eq. (4) is an asymptotically consistent estimator of the true reward parameter \( \omega^* \).

**Theorem 1.** Let the demonstrated trajectories in \( \mathcal{D}_E = \{ (\tau^E_1, \ldots, \tau^E_M) \}_{M=1}^\infty \) be independent and identically distributed (i.i.d.) and sampled from a MFNE induced by an unknown parameterised reward function \( r_\omega(s, a, \mu) \). Under Assumption 1, with probability 1 as the number of samples \( M \to \infty \), the equation \( \mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) = 0 \) has a root \( \hat{\omega} \), such that \( \hat{\omega} = \omega^* \).

**Proof.** Under the i.i.d. assumption, we have that \( \hat{\pi}^E = \pi^E \) and \( \hat{\mu}^E = \mu^E \) with probability 1 as \( M \to \infty \), due to the law of large numbers. Having this, by Assumption 1, we further know that \( \hat{\omega} \) is a minimiser \( \hat{\omega} \in \{ \omega | \mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) = 0 \} \) if and only if \( \pi^\omega \) and \( \tilde{\pi}^E \) are consistent with respect to \( \mu^E \). Finally, due to the fact that \( \pi^E = \pi^{\omega^*} \), we have that \( \omega^* \) is a minimiser of \( \mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) \).

### 3.3 Practical AL-MFIRL using Gradient Methods

We are now left to optimise the objective \( \mathcal{L}(\pi^\omega; \tilde{\pi}^E, \mu^E) \) in Eq. (4) so as to approach the optimal reward parameter. We consider gradient methods, where \( \nabla \mathcal{L} \) \(^2\) is computed by

\[
\nabla \mathcal{L} = \sum_{t=0}^T \sum_{s \in S} \mu^E_t(s) \nabla D_{KL}(\tilde{\pi}^E_t(\cdot | s) \parallel \pi^\omega_t(\cdot | s)) = - \sum_{t=0}^T \sum_{s \in S} \mu^E_t(s) \sum_{a \in A} \frac{\hat{\pi}^E_t(a|s)}{\tilde{\pi}^E_t(a|s)} \nabla \pi^\omega_t(a|s).
\]  
(10)

One difficulty with such an approach is that the greedy policy \( \pi^\omega \) is non-differentiable [34]. We thus need a smooth mapping to approximate \( \hat{\Psi} \). To this end, we adopt the Boltzmann policy \(^3\)

---

\(^2\)To simplify notations, we omit \( \omega \) in \( \nabla \omega \) and write it as \( \nabla \).

\(^3\)Other smooth operators (e.g., Mellowmax [39]) may also be used to approximate the mapping \( \hat{\Psi} \).
prescribes that the elicited policy by the reward function adopted principle for earlier IRL methods is complementary view where the reward function is not given, and hence the need for IRL for MFG. While, all these works rely on the presence of reward functions. Our work takes a computing tic games to approximate Nash equilibria. Guo et al. [4] use mean field theory to approximate joint actions in large-population stochastic games to approximate Nash equilibria. Guo et al. [34] present a Q-learning-based algorithm for computing stationary MFNE. Subramanian and Mahajan [25] use RL to compute local MFNE (a relaxed version). While, all these works rely on the presence of reward functions. Our work takes a complementary view where the reward function is not given, and hence the need for IRL for MFG. 

**Algorithm 1** Agent-Level Mean Field Inverse Reinforcement Learning with Gradient Methods

1: **Input:** MFG with parameters \((S, A, p, \mu_0, \gamma)\) and demonstrations \(D_E = \{(s^1_j, \ldots, s^N_j)\}_{j=1}^M\).
2: **Initialization:** reward parameter \(\omega\).
3: Estimate the empirical expert MF flow \(\hat{\mu}^E\) and expert policy \(\pi^E\) according to Eq. (2) and Eq. (3).
4: for each epoch do
5:   for \(t = T, \ldots, 0\) do
6:     Calculate \(\nabla \hat{\pi}^E(t)(a|s)\) (or estimate \(\hat{\nabla} \pi^E(t)(a|s)\)) according to Eq. (12) and Eq. (13).
7:   end for
8:   Calculate \(\hat{\nabla} L\) (or empirical gradient \(\hat{\nabla} L\)) according to Eq. (10).
9:   Update \(\omega\) to decrease \(L\) according to \(\nabla^2 L\) (or \(\hat{\nabla} L\)).
10: end for
11: **Output:** Learned reward function \(r_\omega\).

### 4 Related Work and Discussions

**RL for MFG.** MFG were pioneered by Lasry and Lions [11] and Huang et al. [41]. Mathematically, the dynamics of the system is governed by two stochastic differential equations: one models the backward dynamics of a representative agent’s value functions and the other models the forward dynamics of mean fields. Learning in MFG has attracted great attentions and most methods are based on RL. Yang et al. [5] use mean field theory to approximate joint actions in large-population stochastic games to approximate Nash equilibria. Guo et al. [34] present a Q-learning-based algorithm for computing stationary MFNE. Subramanian and Mahajan [25] use RL to compute local MFNE (a relaxed version). While, all these works rely on the presence of reward functions. Our work takes a complementary view where the reward function is not given, and hence the need for IRL for MFG.

**IRL for MDP.** The problem of IRL is first introduced by Ng and Russell [17] on MDP. The widely-adopted principle for earlier IRL methods is reward expectation matching (REM) [42, 43], which prescribes that the elicited policy by the reward function \(r_\omega\) should produce the same expected
cumulative reward as that of the expert policy. In the context of MFG, this approach leaves open
the possibility that at optimality the elicited policy \( \pi^* \) is different from the \( \pi^E \), even though they
produce the same reward. This invalidates the approach for IRL on MFG, as \( \pi^* \) may induce a MF
flow that is different from \( \mu^E \), failing to guarantee the consistency requirement of MFNE. Note
that at that stage, IRL was ill-defined as multiple policy can be optimal to a reward function (i.e.,
the policy ambiguity). The maximum entropy (MaxEnt) IRL [43] framework solves this ambiguity
by assuming that the expert uses a near-optimal policy and tuning the reward function by finding
the trajectory distribution with the maximum entropy. However, MaxEnt IRL is incompatible with
MFNE in the sense that an agent in MFNE never takes sub-optimal actions. The incompatibilities of
these conventional IRL methods with MFG motivate the technique of mean-field policy mapping in
AL-MFIRL, where we tune the reward function by matching \( \pi^* \) and \( \pi^E \) (estimated by \( \hat{\pi}^E \)). Since
the expert policy \( \pi^E \) is fixed, AL-MFIRL avoids the policy ambiguity.

**IRL for MAS.** Recently, IRL has been extended to the multi-agent setting. Most works assume
specific reward structures, including fully cooperative games [20, 22], fully competitive games [21],
or either of the two [18, 19]. For general stochastic games, Yu et al. [23] present MA-AIRL, a multi-
ageant IRL method using adversarial learning. However, all these prior methods are not scalable to
games with a large population of agents. Notably, Šošić et al. [44] propose SwarmIRL that views
an MAS as a swarm system consisting of homogeneous agents, sharing the same idea of mean field
approximation. But it cannot handle non-stationary (time-varying) policies and non-linear reward
functions. Our work makes no modelling assumptions on policies and reward functions.

**IRL for MFG.** As discussed earlier, the most related work is [24] that proposes an IRL method for
MFG, which we call MFG-MDP IRL. It focuses on population’s behaviours by reformulating MFNE
as an optimal solution to a MDP, and applies MaxEnt IRL methods to this MDP. We technically
reiterate two issues associated with this setting: (1) Biased inference. Reformulating MFG as MDP
implies that a MFNE optimises population’s societal rewards, but a MFNE is not necessarily socially
optimal if multiple equilibria exist [25, 26, 27, 28]; (2) Sample inefficiency. MFG-MDP IRL can only
obtain a single policy-mean field sample from one sample of game play, whereas each game play
contributes \( N \) samples in our AL-MFIRL. We present detailed explanations and discussions about
MFG-MDP IRL in Appendix A. Since at the agent-level, the expert policy must be a best response
to the expert MF flow, our AL-MFIRL can recover the ground-truth reward functions without bias.

### 5 Experiments

**Tasks and Baseline.** We evaluate MFIRL on five simulated MFG tasks: investment in product
(VIRUS), Rock-Paper-Scissors [32] (RPS) and Left-Right [32] (LR), ordered in decreasing complexity.
Detailed descriptions and settings can be found in Appendix B. We test AL-MFIRL against MFG-
MDP IRL [24], as it is the only IRL method for MFG in the literature, as of the present.

**Performance Metrics.** A learned reward function is considered of good quality if its induced MFNE
is close to that induced by the ground-truth reward function. Thus, we evaluate a learned reward
function using the following two indicators that measure the gap between the two MFNE: (1) Expected
return. The expected return of the MFNE induced by a learned reward function. (2) Deviation
from expert policy (Dev. Policy). We use the loss function \( L \) to measure the distance over policies.

**Training Procedures.** In simulated environments, we have access to ground-truth reward functions
and environment dynamics, which allows us to numerically compute ground-truth MFNE by
repeating the MFNE operator \( \Gamma = \Phi \circ \Psi \) with \( \Psi \) defined as the greedy policy (see Sec. 3.2.1). We take 100 agents
and sample expert trajectories with 50 time steps, which is same as the number of time steps used in [37, 23]. We
set the inverse Boltzmann temperature \( \beta = 1 \). We use one-hot encoding to represent states and actions. For both
algorithms, we adopt the same neural network architecture as the reward model: two hidden layers of 64 leaky
rectified linear units (ReLU) each. Implementation details are given in Appendix C.

**Verification of the Bias in Population-Level Inference.** We verify our claim on the bias in MFG-MDP IRL on
MALWARE, as it is known to have non-socially-optimal MFNE [25]. We train two experts: following [35], the socially optimal MFNE is computed by using DDPG [47] to solve the MDP reduced from MFG (see Appendix A for details); the ordinary one is obtained by repeating MFNE operator. To indicate the bias, we calculate expected returns of the socially optimal MFNE induced by the learned reward function and the ground truth, respectively. Results are depicted in Fig. 1. MFG-MDP IRL shows expert-like performance if trajectories are sampled from a socially optimal MFNE but shows large deviations under an ordinary MFNE. This verifies the bias of population-level inference.

**Reward Recovery under Original Dynamics.** We first carry out tests with unchanged dynamics. Results are depicted in Fig. 2. On all tasks, AL-MFIRL is more sample efficient than MFG-MDP IRL, verifying our arguments on the sample inefficiency of MFG-MDP IRL. Meanwhile, AL-MFIRL achieves near-expert performance, while MFG-MDP IRL shows larger deviations on INVEST and MALWARE even though with a large number of samples, as the demonstrations might be sampled from an ordinary MFNE in these two tasks.

![Figure 2: Results for original environments. The solid line shows the median and the shaded area represents the standard deviation over 10 independent runs.](image)

**Table 1: Results for new environments. Mean and variance are taken across 10 independent runs.**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Algorithm</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Return</td>
<td>EXPERT</td>
<td>INVEST</td>
</tr>
<tr>
<td></td>
<td>AL-MFIRL</td>
<td>-35.051</td>
</tr>
<tr>
<td></td>
<td>MFG-MDP IRL</td>
<td>-35.542 ± 0.677</td>
</tr>
<tr>
<td></td>
<td>EXERT</td>
<td>-18.055</td>
</tr>
<tr>
<td></td>
<td>AL-MFIRL</td>
<td>-18.519 ± 0.245</td>
</tr>
<tr>
<td></td>
<td>MFG-MDP IRL</td>
<td>-19.151 ± 0.507</td>
</tr>
<tr>
<td></td>
<td>AL-MFIRL</td>
<td>-1.167</td>
</tr>
<tr>
<td></td>
<td>MFG-MDP IRL</td>
<td>-1.614 ± 0.042</td>
</tr>
<tr>
<td></td>
<td>AL-MFIRL</td>
<td>94.274</td>
</tr>
<tr>
<td></td>
<td>MFG-MDP IRL</td>
<td>93.578 ± 2.508</td>
</tr>
<tr>
<td></td>
<td>AL-MFIRL</td>
<td>-0.518</td>
</tr>
<tr>
<td></td>
<td>MFG-MDP IRL</td>
<td>-1.700 ± 0.078</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dev. Policy</th>
<th>AL-MFIRL</th>
<th>MALWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.305 ± 0.017</td>
<td>0.411 ± 0.025</td>
</tr>
<tr>
<td></td>
<td>1.303 ± 0.334</td>
<td>1.466 ± 1.322</td>
</tr>
<tr>
<td></td>
<td>1.544 ± 0.012</td>
<td>1.892 ± 0.237</td>
</tr>
<tr>
<td></td>
<td>7.000 ± 0.541</td>
<td>7.550 ± 0.841</td>
</tr>
<tr>
<td></td>
<td>0.683 ± 0.035</td>
<td>0.734 ± 0.373</td>
</tr>
<tr>
<td></td>
<td>0.035 ± 0.012</td>
<td>0.650 ± 0.042</td>
</tr>
</tbody>
</table>

**Re-optimisation under New Dynamics.** To investigate the robustness against changing environment dynamics, we then change the transition function (see Appendix B for details), recompute MFNE induced by the ground truth and the learned reward functions (trained with 10 game plays), and calculate two metrics again. Results are summarised in Tab. 1. Consistently, AL-MFIRL outperforms MFG-MDP IRL on all tasks. We owe the low robustness of MFG-MDP IRL to that the issue of biased inference can be exacerbated by the changing dynamics. In constrast, AL-MFIRL shows higher robustness as it is unbiased. To summarise, AL-MFIRL recovers ground-truth reward functions with high sample efficiency and high robustness, in line with our theoretical analysis.

6 Conclusion and Future Work

This paper amounts to an effort towards agent-level IRL for MFG. We propose the first framework, AL-MFIRL, that can recover the agent level (ground-truth) reward function for MFG with no bias. AL-MFIRL follows mean-field policy matching which ensures that the learned reward function is an asymptotically consistent estimate of the ground truth. To derive a practical algorithm, we implement AL-MFIRL by using the (differentiable) Boltzmann policy and gradient methods. Experimental results demonstrate that AL-MFIRL is able to recover a reward function behind the observed MFNE behaviours, as well as robust against the change of environment dynamics. A direction for future work is to design a practical AL-MFIRL algorithm that is efficient on tasks with high dimensional and/or continuous state-action spaces.
References


