Inference of Intrinsic Rewards and Fairness in Multi-Agent Systems

Victor Villin
Department of Computer Science
Université de Neuchâtel
Neuchâtel, Switzerland
victor.villin@unine.ch

Christos Dimitrakakis
Department of Computer Science
Université de Neuchâtel
Neuchâtel, Switzerland
christos.dimitrakakis@unine.ch

Abstract

From altruism to antagonism, fairness plays a central role in social interactions. But can we truly understand how fair someone is, especially without explicit knowledge of their preferences? We cast this challenge as a multi-agent inverse reinforcement learning problem, explicitly structuring rewards to reflect how agents value the welfare of others. We introduce novel Bayesian strategies, reasoning about the optimality of demonstrations and characterisation of equilibria in general-sum Markov games. Our experiments, spanning randomised environments and a collaborative cooking task, reveal that coherent notions of fairness can be reliably inferred from demonstrations. Furthermore, when isolating fairness components, we obtain a disentangled understanding of agents preferences. Crucially, we unveil that by placing agents in different groups, we can force them to exhibit new facets of their reward structures, cutting through ambiguity to answer the central question: *who is being fair*?

1 Introduction

Fairness plays a pivotal role in decision-making in social interactions, influencing how individuals balance their own interests with the welfare of others [35, 15, 39]. A simple yet illustrative example is a parent playing chess with their child: rather than playing optimally, the parent might intentionally make suboptimal moves to allow the child to experience the joy of winning or the satisfaction of learning. Psychologically, this decision reflects a value system where utility is derived not only from winning but also from nurturing the child's growth and happiness.

From a game-theoretic perspective, individuals in social interactions can be viewed as rational agents aiming to maximise their rewards through strategic decision-making [44]. In the realm of multi-agent Reinforcement Learning (RL), reward functions are central to optimising agent policies, quantifying the utility of actions given the presence and actions of other agents, and steering agents towards specific behaviours [14].

Within a given environment, an agent can act in a selfish, adversarial, or altruistic manner. Without knowledge of the agent's reward function, our goal is to *infer* their motivation by placing them in situations where they must interact with other agents. We assume that each agent acts not only to maximise its intrinsic reward, but may wish to minimise or maximise rewards of other agents to some extent. Then the problem we need to solve is two-fold: Firstly, inferring each agent's intrinsic reward function, and secondly how they value their own reward vis-a-vis those of the other agents.

We choose to tackle this novel challenge through the lens of Multi-Agent Inverse Reinforcement Learning (MAIRL) [31], a promising tool for interpreting social interactions [9].

18th European Workshop on Reinforcement Learning (EWRL 2025).

Contributions. We propose practical algorithms for inferring the fairness of agents with unknown intrinsic rewards in a multi-agent system. Framing the task as a MAIRL problem, we model agent utilities as a linear combination of their intrinsic rewards and those of other agents. Although simultaneously identifying rewards and fairness might be hard, we show that by placing agents within diverse groups, we can obtain good estimates of the agents' fairness attitudes. We enumerate our main contributions below:

1. We formalise the fairness inference problem as an MAIRL problem, structuring rewards to reflect how an agent values the welfare of others (Section 4).
2. We investigate reward identifiability in MAIRL (Section 5), arguing that additional assumptions on the reward structure as well as entropy-regularisation are key elements for inference.
3. In Section 6, we present two Bayesian methods: one assumes that agents are Boltzmann-rational, while the other makes no assumptions on the policy structure [12].
4. In Section 7, we assess these methods on fairness and reward identification. We first use randomised Markov Games (MGs) as a general benchmark, before further experimenting on a collaborative cooking task [7]. Our experiments show that we can learn accurate fairness and intrinsic rewards with sufficient demonstrations and proper optimality modelling. We also show that collecting data from multiple agent groupings improves sample efficiency and reduces any remaining ambiguities about the agent's preferences.

2 Related Work

Multi-Agent Inverse Reinforcement Learning. Inverse RL (IRL) and MAIRL aim to infer reward functions that explain observed behaviours, assuming these are near-optimal [32, 31]. Defining optimality in multi-agent settings is significantly more complex than in single-agent scenarios. Although multi-agent problems can be decomposed into separate single-agent tasks [27, 17], many studies emphasise the importance of equilibrium concepts [27], such as Nash equilibria, for effective reward inference [38]. In entropy-regularised settings, equilibria are unique [46, 23, 29, 3, 30, 16, 20].

In our work, agents are assumed to have varying, hidden levels of fairness, rendering any problem general-sum. While MAIRL has been successfully applied in fully cooperative and zero-sum settings [23, 17, 29, 20, 46], its application to general-sum games is less straightforward. Nonetheless, effective reward recovery has been demonstrated under structural assumptions on the reward functions [38, 27, 16], including on real-world traffic data [30]. Despite this, MAIRL still faces challenges in identifying true rewards in general-sum environments, even with entropy regularisation [16]. One can mitigate this issue by varying the environment in which demonstrations are provided [6, 25, 40, 4, 33]. Similarly, changing an agent's policy in a multi-player game, can reveal new aspects of the other agents rewards [5]. *Our approach* takes inspiration from this, but instead of controlling agent policies, we deploy diverse agent combinations to disambiguate reward functions.

Bayesian IRL is a framework for inferring rewards through probabilistic reasoning [36]. Despite its success in single-agent scenarios [10, 41, 12], its application to MAIRL remains underexplored [28, 29]. Common drawbacks to existing MAIRL methods are strong assumption about behaviours, such as strict rationality or a specific amount of entropy regularisation [46, 27]. *We propose* a Bayesian modelling approach that only assumes ϵ-optimality of the policy, which we show performs significantly better than approaches which do make behavioural assumptions.

Fairness in MARL. Fairness is a natural concern in economics [24] and multi-agent systems [11, 21], and is typically enforced by encouraging agents to share welfare equitably or by applying constraints [34, 1, 2, 42].

The topic of inferring fairness has been studied in behavioural economics. [8] showed that subjects are concerned with increasing payoffs for everybody, based on a linear utility model, while [18, 19] proposed utility-based fairness concepts such as *envy*, *altruism* and *inequity aversion*, as well as behavioural concepts such as *reciprocity*. However, this work is generally limited to surveys and stylised games (cf.[19]). In MAIRL, some work considered the related topic of a theory of mind about other agents [45, 9]. However, the specific setting and techniques presented here are novel to this paper and to the best of our knowledge, this is the first work to use MAIRL methods to infer the fairness ideals of agents in a dynamic multi-agent game.

3 Preliminaries

A discounted n-player Rewardless Markov Game (RMG) can be formalised as a tuple $\mathcal{G} = \mathcal{G}(\cdot) = (\mathcal{S}, \boldsymbol{\mathcal{A}}, T, \gamma, \omega, \cdot)$, where $\mathcal{S}$ is a set of states, $\boldsymbol{\mathcal{A}} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is a set of discrete actions for each player, $T : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \Delta^{\mathcal{S}}$ is a transition function, $\gamma \in [0, 1[$ is a discount factor, and ω an initial state distribution. Throughout we suppose that, without a loss of generality, the n players have a common action space: $\boldsymbol{\mathcal{A}} = \mathcal{A}^m$. We also assume that permuting player indices at the start of games does not affect outcomes. Details on that last aspect are provided in the experiments section.

A policy $\pi : \mathcal{S} \to \Delta^{\mathcal{A}}$ is a probability distribution over a single agent's actions. A joint policy is given by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n) = (\pi_i, \boldsymbol{\pi}_{-i})$, where $\boldsymbol{\pi}_{-i}$ refers to the joint policy of all policies except policy i. We say we deploy a joint policy $\boldsymbol{\pi}$ on a game $\mathcal{G}$ when each player i follows policy π_i. We denote the set of policies by Π. Furthermore, we assume actions are independent, so that the probability of taking the joint action $\mathbf{a} \in \mathcal{A}^m$ at state s under the joint policy $\boldsymbol{\pi}$ is $\boldsymbol{\pi}(\mathbf{a} \mid s) = \prod_i \pi_i(a_i \mid s)$.

The reward function of an agent i, $R_i : \mathcal{S} \times A \to \mathbb{R}$, outputs a bounded real value given a state and a joint action. We write the joint reward function as $\mathbf{R} = (R_i)_i$. Plugging an arbitrary joint reward function $\mathbf{R}$ to an RMG $\mathcal{G}$ forms a general-sum Markov Game (MG) $\mathcal{G}(\mathbf{R})$. When deploying a joint policy $\boldsymbol{\pi}$ on a game $\mathcal{G}(\mathbf{R})$, we can define the Q-function of the i-th agent as:

$$Q^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R}),i}(s, \mathbf{a}) \coloneqq \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R})}[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, \mathbf{a}_t) \mid s_0 = s, a_0 = \mathbf{a}],$$

Furthermore, we can also define the value function as:

$$V^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R}),i}(s) \coloneqq \sum_{\mathbf{a}} \boldsymbol{\pi}(\mathbf{a} \mid s) Q^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R}),i}(s, \mathbf{a}), \qquad \bar{V}^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R}),i} \coloneqq \mathbb{E}_{s \sim q^{\boldsymbol{\pi}}}[V^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R}),i}(s)],$$

where $q^{\boldsymbol{\pi}}$ is the state distribution induced by $\boldsymbol{\pi}$. Throughout, we will simply state we are on some game $\mathcal{G}(\mathbf{R})$ and use the notations $Q^{\boldsymbol{\pi}}_i$ and $V^{\boldsymbol{\pi}}_i$ for brevity. In an entropy-regularised n-player MG, each agent i is regularised with some strictly positive constant β_i. The entropy-regularised value function of the i-th agent is defined as:

$$\tilde{V}^{\boldsymbol{\pi}}_i(s) \coloneqq \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{G}(\mathbf{R})}[\sum_{t=0}^{\infty} \gamma^t R_i(s, \mathbf{a}_t) - \beta_i \log \pi_i(a_i \mid s)].$$

The entropy-regularised $\tilde{Q}$-function is then defined analogously to Q, using $\tilde{V}$. For simplicity, we consider regularisation parameters equal for all agents, for any regularised MGs.

Optimality in MGs. There are multiple ways to define optimality in MGs, with the Nash equilibrium being the most common. A joint policy $\boldsymbol{\pi}^*$ is a Nash equilibrium if no agent can unilaterally improve their own expected return by deviating from their policy, assuming the policies of the other agents remain fixed. Formally, $\boldsymbol{\pi}^*$ is a Nash equilibrium of the game $\mathcal{G}(\mathbf{R})$ if, for any agent i and any state s,

$$V_i^{\boldsymbol{\pi}^*}(s) \geq V_i^{\{\pi_i\} \cup \boldsymbol{\pi}^*_{-i}}(s), \quad \forall \pi_i \in \Pi.$$

4 Problem formulation

We consider the problem of inferring both *fairness* and *intrinsic rewards* for agents drawn from a pool $\mathcal{P}$ of size $m \geq n$, when participating in a known game $\mathcal{G}$. Each agent $k \in \mathcal{P}$ is assumed to possess a *unknown* reward function R^*_k which they wish to maximise. This reward has two latent components: an *intrinsic reward function* $r^*_k : \mathcal{S} \times \mathcal{A}^m \to [r_{\min}, r_{\max}]$, capturing the agent's individual welfare, and a *fairness level* $\lambda^*_k \in [\lambda_{\min}, \lambda_{\max}]$, quantifying the extent to which it values the welfare of others.

We define $\mathcal{P}_n \coloneqq \{p \subseteq \mathcal{P} \mid |p| = n\}$ as the set of all possible *groupings* of n agents drawn from $\mathcal{P}$. Each grouping $p \in \mathcal{P}_n$ represents a set of agents that are matched to play the game $\mathcal{G}$. When an agent i gets paired, we use $R_{p,i}$ to indicate the reward function under grouping p.

When a grouping p plays the game, it deploys a joint policy $\boldsymbol{\pi}_p \in \Pi^n$. We denote by $\mathcal{D} \coloneqq (\Pi^n)^{|\mathcal{P}_n|}$ the set of all *policy deployments* over the groupings. An element $\boldsymbol{d} = (\boldsymbol{\pi}_p)_{p \in \mathcal{P}_n} \in \mathcal{D}$ specifies one possible way of assigning joint policies to every possible grouping in the population.

Example 4.1. *Consider a pool of 3 agents $\mathcal{P} = \{A, B, C\}$ and a 2-player game $\mathcal{G}$, then $\mathcal{P}_2 = \{\{A, B\}, \{A, C\}, \{B, C\}\}$. A deployment $\boldsymbol{d} \in \mathcal{D}$ consists of a joint policy for each grouping:*

$$\boldsymbol{d} = (\boldsymbol{\pi}_{\{A,B\}}, \boldsymbol{\pi}_{\{A,C\}}, \boldsymbol{\pi}_{\{B,C\}}),$$

where each $\boldsymbol{\pi}_{\{i,j\}} \in \Pi^2$ defines how the two agents i and j act when paired together.

We consider a setting in which we are given demonstrations in the form of a set of trajectories $\mathcal{X} = \bigcup_{p \in \mathcal{P}_n} \mathcal{X}_p = \{\tau_k\}_k$, where each $\mathcal{X}_p$ contains trajectories collected from games played by the grouping p. Each trajectory τ_k is a sequence of states visited and joint actions taken $\tau_k = (s_0^k, \mathbf{a}_0^k, s_1^k, \mathbf{a}_1^k, \dots)$. To enable inference over the underlying fairness structure and reward functions from these demonstrations, we first define a general class of reward models capable of capturing both agent-level preferences and fairness considerations.

Assumption 4.1 (Fairness-Structured Rewards)**.** *Let $p \in \mathcal{P}_n$ be a subset of n agents playing a game $\mathcal{G}$, and let $\mathbf{R}_p = (R_{p,i})_{i \in p}$ be the joint reward function induced by the grouping p. We assume that $\mathbf{R}_p$ is fairness-structured, meaning that for each agent $i \in p$,*

$$R_{p,i}(s, \mathbf{a}) = r_i(s, \mathbf{a}) + \frac{\lambda_i}{n-1} \sum_{k \in p,\, k \neq i} r_k(s, \mathbf{a}).$$

Under the assumption that rewards follow a fairness-based structure, our goal shifts from directly inferring the latent reward function set $\mathcal{R}^* = \{R_k^*\}_{k \in \mathcal{P}}$ to disentangling their two underlying components: the fairness levels $\mathcal{L}^* = \{\lambda_k^*\}_{k \in \mathcal{P}}$ and the intrinsic reward functions $\mathcal{I}^* = \{r_k^*\}_{k \in \mathcal{P}}$. From this point onward, the latent reward set is summarised by its intrinsic rewards and fairness levels, $\mathcal{R} = (\mathcal{L}, \mathcal{I})$, with joint rewards similarly represented as $\mathbf{R} = (\boldsymbol{\lambda}, \mathbf{r})$. Moreover, we let $\mathbb{P}(\cdot \mid \mathcal{X})$ denote the posterior over reward functions given demonstrations $\mathcal{X}$.

5 Identifiability of rewards in MAIRL

Before diving into reward inference, we discuss on the identifiability of rewards in MAIRL.

5.1 Uniqueness of equilibria

General-sum MGs may admit multiple equilibria, leading to ambiguity in the set of feasible rewards [16]. This non-uniqueness hinders reward inference, as distinct equilibria can correspond to radically different outcomes. A common remedy is to entropy-regularise the game, which guarantees a unique equilibrium. In such games, optimal policies correspond to a Quantal Response Equilibrium (QRE) [27, 46, 23, 26, 29, 3, 30, 16, 20], and take the form:

$$\pi_i^{\text{QRE}}(a_i \mid s) = \frac{\exp\left(\frac{1}{\beta} \sum_{\mathbf{a}_{-i} \in \mathcal{A}^{n-1}} \boldsymbol{\pi}_{-i}^{\text{QRE}}(s, \mathbf{a}_{-i}) \tilde{Q}_i^{\boldsymbol{\pi}^{\text{QRE}}}(s, a_i, \mathbf{a}_{-i})\right)}{\sum_{a_i' \in \mathcal{A}} \exp\left(\frac{1}{\beta} \sum_{\mathbf{a}_{-i} \in \mathcal{A}^{n-1}} \boldsymbol{\pi}_{-i}^{\text{QRE}}(s, \mathbf{a}_{-i}) \tilde{Q}_i^{\boldsymbol{\pi}^{\text{QRE}}}(s, a_i', \mathbf{a}_{-i})\right)}, \tag{1}$$

where β is the entropy regularisation coefficient. QRE policies act as softmax responses to expected Q-values over the policies of other agents. While this introduces entropy parameters into the inference process, it poses no fundamental obstacle: we can infer over them using a suitable prior of the parameter. Moreover, QRE policies vary smoothly with respect to rewards, enabling the use of gradient-based posterior sampling methods such as the Unadjusted Langevin Algorithm (ULA) [13].

5.2 Linearly separable rewards

Entropy regularisation guides the solution toward unique equilibria, but this alone does not guarantee identifiability, especially when reward functions have a strong dependence on the joint actions. Nonetheless, in the two-player zero-sum case, it was demonstrated that rewards are identifiable (up to a constant) if they are *linearly separable* [38, 16]. In our case, this corresponds to requiring that intrinsic rewards depend only on each agent's own action: $r_i(s, \mathbf{a}) = r_i(s, a_i)$. If we further assume that all agents share the same intrinsic reward and fairness levels are fixed at -1, we recover the identifiable zero-sum case from [16], making our model a natural generalisation. This suggests that identifiability may still hold for other fairness levels. However, we do not know agents' fairness, introducing additional degrees of freedom. This may again render the inference problem ill-posed.

5.3 Agent combinations and multiple equilibria

Leveraging *multiple agent combinations* could be crucial for resolving ambiguities in rewards. To illustrate our point, in a 2-player game with 3 agents, we can observe up to 3 different equilibria instead of just one. Those multiple equilibria can then be exploited for identification: if a candidate reward function explains one grouping's behaviour but not another's, it can be ruled out.

This idea, largely unexplored in MAIRL, aligns with recent findings in IRL and MAIRL, which show that varying demonstration environments helps reward identification [6, 40, 4]. Just as altering policies in multi-agent settings can shift transition dynamics [5], observing diverse groupings across equilibria constrains the space of compatible rewards.

6 Two Bayesian methods for inferring fairness and rewards.

All our inference algorithms use a Bayesian perspective, as it provides a principled framework for reasoning about reward uncertainty [36]. Given a prior distribution $\mathbb{P}(\cdot)$ over reward functions, the goal of Bayesian IRL is to recover the posterior distribution $\mathbb{P}(\cdot \mid \mathcal{X})$ conditioned on a set of demonstrations $\mathcal{X}$. Inference can then proceed by sampling from the posterior using Markov Chain Monte Carlo (MCMC) methods [36, 12, 41, 28, 29].

In this section, we introduce two Bayesian methods tailored to our multi-agent fairness setting. The first method directly constructs a posterior distribution over reward functions by evaluating the likelihood of observed demonstrations, assuming agents follow near-optimal responses to candidate rewards. To model behaviours, we extend the concept of Boltzmann-rational policies to the multi-agent setting, similarly to [20]. This approach serves as a baseline, building on a widely used Bayesian IRL approach [22, 5, 4]. The second approach follows a two-step procedure: first constructing a posterior over policies given demonstrations, and subsequently estimating a posterior over reward functions conditioned on the inferred policies [12].

6.1 Direct Reward Posterior (DRP)

One way to write the reward posterior over reward functions is to directly model the likelihood of the observed demonstrations under those rewards, i.e., $\mathbb{P}(\mathcal{X} \mid \mathcal{R}) = \prod_{\tau \in \mathcal{X}} \mathbb{P}(\tau \mid \mathcal{R})$. Since demonstrations are a partition over groupings, we can rearrange the product accordingly:

$$\mathbb{P}(\mathcal{X} \mid \mathcal{R}) = \prod_{p \in \mathcal{P}_n} \prod_{\tau \in \mathcal{X}_p} \mathbb{P}(\tau \mid \mathbf{R}_p).$$

This formulation is conceptually similar to that of [4], though their focus is on multiple environments (i.e., transition functions), whereas we consider different agent combinations.

Somehow, we need to specify the likelihood $\mathbb{P}(\tau \mid \mathbf{R})$, modelling how agents generate trajectories based on the underlying rewards. To get started, we suggest that demonstrated policies have a specific probabilistic structure for each reward function. This assumption allows us to encode the idea that reward functions should, to some degree, explain observed behaviours. As stated in Section 5, it is useful to consider regularised MGs, where equilibrium policies take the QRE form (1). Interestingly, the QRE form reduces to a Boltzmann-rational policy if the game is single-agent [20], a common choice in IRL. Taking the analogy further, the entropy parameter β in this context can naturally act as a sub-optimality coefficient: larger β implies more stochastic (less optimal) behaviour, and vice versa. We therefore assume all agents adopt QRE policies, and denote by $\boldsymbol{\pi}^{\text{QRE}}_{\mathbf{R}}(\cdot \mid \cdot; \beta)$ the QRE policy under joint reward $\mathbf{R}$ with entropy parameter β. Given a prior belief $\mathbb{P}(\beta)$ over β, we can marginalise it out, and obtain the trajectory likelihood:

$$\mathbb{P}(\tau \mid \mathbf{R}) = \prod_{(s,\mathbf{a}) \in \tau} \int_0^\infty \boldsymbol{\pi}_{\mathbf{R}}(\mathbf{a} \mid s; \beta) \mathrm{d}\mathbb{P}(\beta).$$

In practice, the integral can be approximated via a Riemann sum over a finite partition $(\beta_k)_k$ of $[0, \beta\max]$, where $\beta_{\max}$ is chosen sufficiently large. Substituting this expression back into the demonstration likelihood, this yields the full DRP:

$$\mathbb{P}(\mathcal{R} \mid \mathcal{X}) \propto \mathbb{P}(\mathcal{R}) \prod_{p \in \mathcal{P}_n} \prod_{\tau \in \mathcal{X}_p} \prod_{(s,\mathbf{a}) \in \tau} \int_0^\infty \boldsymbol{\pi}_{\mathbf{R}_p}(\mathbf{a} \mid s; \beta) \mathrm{d}\mathbb{P}(\beta). \tag{2}$$

As a result, this posterior can be explored using any standard posterior sampling techniques. For DRP, any prior can be picked for the rewards, while the prior for beta can typically be chosen exponential, putting more weights on near-optimality.

6.2 Policy-Oriented Reward Posterior (PORP)

In contrast to the DRP approach, where we modelled the likelihood of demonstrations given rewards, we now take a complementary view: we place a prior over policies, and infer a posterior over policies from the demonstrations. We hereby shift our focus to a prior on the *optimality* of the policies demonstrated [12]. Recalling that $\mathcal{D}$ denotes the set of possible joint policy deployments over all agent groupings, we express the posterior over rewards by marginalising over policy deployments:

$$\mathbb{P}(\mathcal{R} \mid \mathcal{X}) = \int_{\mathcal{D}} \mathbb{P}(\mathcal{R} \mid \boldsymbol{d}) \mathrm{d}\mathbb{P}(\boldsymbol{d} \mid \mathcal{X}), \tag{3}$$

assuming that rewards are conditionally independent of demonstrations given policies $\mathbb{P}(\mathcal{R} \mid \boldsymbol{d}, \mathcal{X}) = \mathbb{P}(\mathcal{R} \mid \boldsymbol{d})$. This reduces the need for us to make assumptions about the policy structure. Instead, it requires us to evaluate how likely a reward is, given a policy. Intuitively, a reward is more plausible if the policy is nearl-optimal under it. We formalise this via a *gap function* $\Delta_{\mathbf{R}} : \Pi \to \mathbb{R}^+$, which measures the suboptimality of policy $\boldsymbol{\pi}$ under joint reward $\mathbf{R}$. The likelihood of $\mathbf{R}$ given $\boldsymbol{\pi}$ is then:

$$\mathbb{P}(\mathbf{R} \mid \boldsymbol{\pi}) \propto \mathbb{P}(\mathbf{R}) \cdot e^{-c\Delta_{\mathbf{R}}(\boldsymbol{\pi})},$$

where c controls how strongly we believe demonstrations are optimal. Since each grouping induces a joint policy $\boldsymbol{\pi}_p$ with a corresponding reward $\mathbf{R}_p$, the rewards can be factorised across groupings: $\mathbb{P}(\mathcal{R} \mid \boldsymbol{d}) = \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}} \mathbb{P}(\mathbf{R}_p \mid \boldsymbol{\pi}_p)$. Plugging this into (3), we obtain:

$$\mathbb{P}(\mathcal{R} \mid \mathcal{X}) = \int_{\mathcal{D}} \frac{1}{Z_{\boldsymbol{d}}} \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}} \mathbb{P}(\mathbf{R}_p) \cdot e^{-c\Delta_{\mathbf{R}_p}(\boldsymbol{\pi}_p)} \mathrm{d}\mathbb{P}(\boldsymbol{d} \mid \mathcal{X}),$$

where $Z_{\boldsymbol{d}}$ is a normalisation constant dependent on the specific policy deployment $\boldsymbol{d}$. A practical difficulty here is that $Z_{\boldsymbol{d}}$ varies with each $\boldsymbol{d}$, and thus cannot be pulled outside of the integral. However, $Z_{\boldsymbol{d}}$ is approximately constant across policies, partly because the posterior over policy deployments $\mathbb{P}(\boldsymbol{d} \mid \mathcal{X})$ tends to be sharply peaked, especially when enough demonstration data is available.[1] This makes the following approximation reasonable:

$$\mathbb{P}(\mathcal{R} \mid \mathcal{X}) \overset{\propto}{\approx} \int_{\mathcal{D}} \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}} \mathbb{P}(\mathbf{R}_p) \cdot e^{-c\Delta_{\mathbf{R}_p}(\boldsymbol{\pi}_p)} \mathrm{d}\mathbb{P}(\boldsymbol{d} \mid \mathcal{X}). \tag{4}$$

This integral is easy to approximate, as long as a posterior over policies can be computed and sampled from, as we show in the next paragraph. We then only need to choose an appropriate gap function and reward prior, as we explain in the remainder of this section.

Two-step posterior sampling. We now outline a practical procedure to sample from the PORP (4). The method proceeds in two steps:

1. Given that the likelihood of a trajectory under a policy is simply the probability of the policy generating it, we have: $\mathbb{P}(\boldsymbol{\pi} \mid \tau) \propto \mathbb{P}(\boldsymbol{\pi}) \prod_{(s,\mathbf{a}) \in \tau} \boldsymbol{\pi}(\mathbf{a} \mid s)$. This allows us to factor the posterior over a full deployment of policies as:

$$\mathbb{P}(\boldsymbol{d} \mid \mathcal{X}) \propto \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}} \prod_{\tau \in \mathcal{X}_p} \mathbb{P}(\boldsymbol{\pi}_p) \prod_{(s,\mathbf{a}) \in \tau} \boldsymbol{\pi}_p(\mathbf{a} \mid s). \tag{5}$$

   In the first step, we draw N samples $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N$ from this posterior over policy deployments.

2. We can then approximate the reward posterior (4) using Monte Carlo integration:

$$\hat{\mathbb{P}}(\mathcal{R} \mid \mathcal{X}) \overset{\propto}{\approx} \sum_{k=1}^{N} \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}_k} \mathbb{P}(\mathbf{R}_p) \cdot e^{-c\Delta_{\mathbf{R}_p}(\boldsymbol{\pi}_p)}. \tag{6}$$

   We are then left with sampling from that approximate posterior to generate reward candidates.

This two-step approach enables efficient approximation of the posterior over joint rewards: first by inferring plausible policies from data, and then by weighting rewards according to how well these policies align with optimal behaviour under each reward hypothesis.

[1] An empirical study on the near constant nature of $Z_{\boldsymbol{d}}$ over $\boldsymbol{d}$ is provided in Appendix B

Gap functions. To evaluate proximity to equilibrium, we define two gap functions to use alongside the PORP. These functions are constructed such that a value of zero indicates an equilibrium, while larger values reflect deviations from optimality. The first is the Nash Imitation Gap (NIG) [37], recently standardised in the MAIRL setting [16]. It measures the maximum incentive any agent has to unilaterally deviate from the current policy (without entropy regularisation). A NIG of zero implies that no agent can benefit from deviating, thus identifying a Nash equilibrium.

Definition 6.1 (Nash Imitation Gap (NIG))**.** *Let $\mathcal{G}(\mathbf{R})$ be the underlying game and $\boldsymbol{\pi}$ some arbitrary joint policy. The NIG is defined as:*

$$\Delta^{V}(\boldsymbol{\pi}) \coloneqq \max_{i\in[n]} \max_{\pi_i \in \Pi} \bar{V}_i^{\{\pi_i\}\cup\boldsymbol{\pi}_{-i}} - \bar{V}_i^{\boldsymbol{\pi}} \, .$$

Unfortunately, the NIG can become intractable as the number players increases. Additionally, the NIG is overly sensitive to degenerate reward functions, such as constant reward functions, under which any policy is trivially optimal. To mitigate this, we recommend using a prior that penalises constant rewards, such as setting $\mathbb{P}(\mathbf{R}) = 0$ if $\mathrm{Var}(\mathbf{r}) < \varepsilon$ with $\varepsilon \in \mathbb{R}^{+}$. As an alternative, we propose the Policy Stability Gap (PSG), which addresses both computational efficiency and the handling of degenerate rewards by leveraging the known characterisation of optimal entropy-regularised policies.

Definition 6.2 (Policy Stability Gap (PSG))**.** *Let $\mathcal{G}(\mathbf{R})$ be the underlying game and $\boldsymbol{\pi}$ some arbitrary joint policy. Let $\mathcal{D}_{KL}$ be the Kullback-Leibler (KL) divergence. Furthermore, let the 'soft response' of the i-th agent to the joint policy $\boldsymbol{\pi}_{-i}$, under regularisation parameter β, be:*

$$\sigma^{\boldsymbol{\pi}_{-i}}(a_i \mid s; \beta) \coloneqq \frac{\exp\left(\frac{1}{\beta}\sum_{\mathbf{a}_{-i}\in\mathcal{A}^{n-1}} \boldsymbol{\pi}_{-i}(s,\mathbf{a}_{-i})\tilde{Q}_i^{\boldsymbol{\pi}}(s,a_i,\mathbf{a}_{-i})\right)}{\sum_{a_i'\in\mathcal{A}} \exp\left(\frac{1}{\beta}\sum_{\mathbf{a}_{-i}\in\mathcal{A}^{n-1}} \boldsymbol{\pi}_{-i}(s,\mathbf{a}_{-i})\tilde{Q}_i^{\boldsymbol{\pi}}(s,a_i',\mathbf{a}_{-i})\right)}. \tag{7}$$

We define the PSG as:

$$\Delta^{KL}(\boldsymbol{\pi},\beta) \coloneqq \max_{i\in[n]} \sum_{s\in\mathcal{S}} \mathcal{D}_{KL}\Big(\pi_i(\cdot \mid s) \;\Big|\Big|\; \sigma^{\boldsymbol{\pi}_{-i}}(\cdot \mid s;\beta)\Big).$$

The PSG quantifies how far a joint policy deviates from its corresponding soft response. If a policy is indeed a QRE under the reward candidate, then its KL divergence is zero, as it satisfies the fixed-point condition in (1). Moreover, constant reward functions yield uniform optimal policies under entropy regularisation, and therefore cannot achieve a zero PSG unless the policy is also uniform. PSG thus provides a natural regularisation against constant rewards. Importantly, the PSG is computationally efficient: it requires only the evaluation of $\tilde{Q}$ for the current joint policy. As with the direct reward posterior (Section 6.1), inference over the regularisation parameter β is necessary.

7 Experiments

We aim to see if we can disentangle fairness levels from intrinsic rewards and learn reliable estimates for each. In Section 7.1, we address this question using randomised MGs, evaluating both DRP and PORP on fairness estimation and reward accuracy, under limited number of demonstrations. We also perform an ablation study to quantify the impact of leveraging multiple agent groupings compared to a fixed single group. Next, in Section 7.2, we test our approach in a simplified collaborative cooking scenario inspired from the Overcooked game [7, 43, 1, 20], exploring the boundaries of inference in a more complex task.

To align with Section 5, we assume agents share the same intrinsic rewards ($r = r_1 = \cdots = r_m$), which are linearly separable and bounded in $[0, 1]$. We use an exponential prior for sub-optimality/entropy parameters and uniform priors for rewards, fairness, and policies unless stated otherwise. Demonstrations are generated with QRE policies using a hidden entropy parameter, and evenly distributed across groupings.

To assess the quality of learned estimates, we measure the mean absolute error over fairness and intrinsic rewards, rescaled such that an error of 1 matches the expected error of random guessing. For posterior evaluation, metrics are computed over sampled batches and averaged. Additional experimental results and details are provided in Appendices B and C.

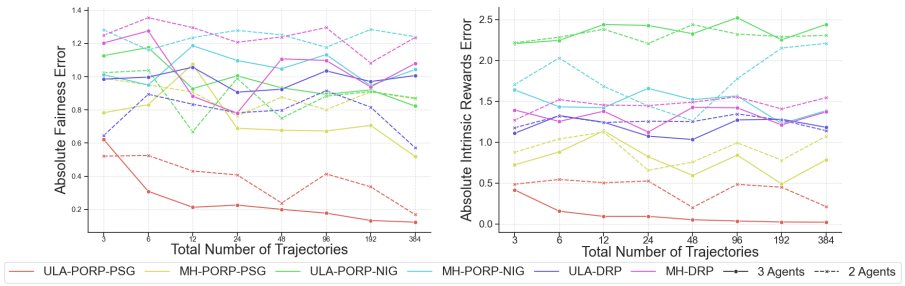


Figure 1: Average inference errors, over 10 seeds. Plots with error bars are provided in Appendix B.

7.1 Benchmarking on random MGs

In this experiment, we evaluate our methods with two posterior sampling strategies: Metropolis Hastings (MH) and ULA, alongside evaluations of the proposed gap functions (NIG and PSG).

Experimental setup. For each run, we generate a game with 5 states and 3 actions, sampling symmetric transition dynamics from a Dirichlet distribution to ensure invariance under player permutation: $T(s, a_1, a_2) = T(s, a_2, a_1)$. We limit the fairness space to $[-1, 1]$ and fix the intrinsic rewards to $+1$ for a single (s, a) pair and 0 elsewhere. A pool of 3 agents is then constructed by sampling uniformly fairness levels. Demonstrations consist of 100-timestep trajectories generated from all groupings. When using the NIG, we apply a prior that penalises constant reward functions, as discussed in Section 6.2. To assess the impact of multiple groupings, we conduct an ablation study with 2-agent pools, using a single fixed grouping for all demonstrations. Fairness errors are computed only over the first 2 agents, even for 3-agent runs, for consistent comparison.

Results. Figure 1 shows that ULA-PORP-PSG is the only method to consistently disentangle fairness from intrinsic rewards, recovering near ground-truth estimates even with limited demonstrations. Notably, it relies solely on the optimality prior, without any informative reward or policy prior, demonstrating that fairness-structured rewards can be inferred almost entirely from behaviour. MH-PORP-PSG also shows solid performance, though ULA appears more effective in sampling from the PORP. The ablation study in Figure 1 further confirms our hypothesis on the importance of groupings, with clear benefits observed for ULA-PORP-PSG.

7.2 Fairness inference in a collaborative cooking task.

In this experiment, we consider a task where two players act as chefs, working together to prepare and deliver tomato soups. Players must collect tomatoes, cook them, carry dishes and deliver soups. The kitchen layout (see Figure 2a) requires object interaction in the correct sequence and tight coordination. While players cannot move through each other, they start at the same position to make the game invariant to player permutations. Our setup includes 390 unique states and 5 actions (4 directions and 1 interaction), making Bayesian reward inference challenging. Given its strong performance in random MGs, we focus exclusively on ULA-PORP-PSG for this experiment.

Experimental setup. A player perceives a reward of one point each time they successfully deliver a soup. By representing the game state by the tuple $s = (s_g, s_1, s_2)$, where s_g contains global features, and s_1, s_2 describe the states of each player, we confirm that the ground-truth intrinsic reward function satisfies a linearly separable property. Indeed, deliveries only depend on one agent's action, and rewards are symmetric in player features: $r^1((s_g, s_1, s_2), a) = r^2((s_g, s_2, s_1), a)$, where r^i represents intrinsic rewards when the agent takes the role of player i.

A pool of 3 agents with fixed, hidden fairness values of 0, 0.5 and 1 is initially constructed. We then let agents demonstrate over trajectories of length 1000, and employ ULA-PORP-PSG to infer rewards. We limit the fairness space to $[0, 1]$. A prior belief over intrinsic rewards $\mathbb{P}(r) \propto \sum_{(s,a)} r(s, a)$ is chosen to encourage sparsity. Note that this prior still demands the method to identify the correct (s, a) pairs associated with successful soup deliveries.

Results. Figure 2 presents the inferred fairness and rewards as the number of demonstrations increases. Early on, ULA-PORP-PSG yields uncertain fairness estimates (Figure 2b), but quickly converges to

(a) Kitchen layout. (b) Fairness posteriors. (c) Intrinsic rewards error.

Figure 2: Experiments with ULA-PORP-PSG on the collaborative cooking task over 10 seeds. (a) The collaborative cooking task. (b) Fairness posteriors of a single run for an increasing number of trajectories. (c) Average intrinsic reward error over (s, a) pairs, segmented by whether the pair was visited and whether it corresponds to a successful delivery. Standard error is given by error bars.

the correct values as data accumulates. In Figure 2c, we plot errors for both all state-action pairs and those actually visited. The method struggles to distinguish delivery states under few demonstrations, but becomes accurate with more than 100 trajectories. With enough samples, the true reward function could perhaps be fully recovered. Hence, we confirm that fairness-structured rewards are identifiable from demonstrations alone, given only weak prior information.

8 Conclusion

We introduced the MAIRL problem of disentangling fairness from intrinsic preferences. We developed Bayesian inference techniques tailored to handle multiple agent groupings. Our experiments show that, with a suitable optimality prior and enough demonstrations, both fairness levels and intrinsic rewards can be reliably recovered, even in complex settings. Moreover, we find that learning from diverse agent combinations significantly improves identifiability of the true underlying rewards.

Limitations. It remains unclear why some sampling methods underperform. Understanding this gap could lead to more robust inference. The data needed could be high for expensive demonstrations.

Future work. One exciting direction is the development of an online inference mechanism that actively selects promising agent combinations, rather than observing them equally frequently. This could significantly cut the amount of demonstrations needed, hereby solving one limiting factor of our work. Another intriguing extension involves relaxing the assumption of fixed agent preferences. Allowing agents to express varying underlying preferences would mirror real-world dynamics more closely.

References

[1] John P. Agapiou, Alexander Sasha Vezhnevets, Edgar A. Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, DJ Strouse, Michael B. Johanson, Sukhdeep Singh, Julia Haas, Igor Mordatch, Dean Mobbs, and Joel Z. Leibo. Melting pot 2.0, 2023.

[2] Jasmine Jerry Aloor, Siddharth Nagar Nayak, Sydney Dolan, and Hamsa Balakrishnan. Cooperation and fairness in multi-agent reinforcement learning. *Journal on Autonomous Transportation Systems*, 2(2):1–25, 2024.

[3] Sage Bergerson. Multi-agent inverse reinforcement learning: Suboptimal demonstrations and alternative solution concepts. *arXiv preprint arXiv:2109.01178*, 2021.

[4] Thomas Kleine Buening, Victor Villin, and Christos Dimitrakakis. Environment design for inverse reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[5] Thomas Kleine Büning, Anne-Marie George, and Christos Dimitrakakis. Interactive inverse reinforcement learning for cooperative games. In *International Conference on Machine Learning*, pages 2393–2413. PMLR, 2022.

[6] Haoyang Cao, Samuel Cohen, and Lukasz Szpruch. Identifiability in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12362–12373, 2021.

[7] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

[8] Gary Charness and Matthew Rabin. Understanding social preferences with simple tests. *Quarterly journal of Economics*, pages 817–869, 2002.

[9] Yusi Chen, Angela Radulescu, and Herbert Zheng Wu. Unveiling the latent dynamics in social cognition with multi-agent inverse reinforcement learning. *bioRxiv*, pages 2024–10, 2024.

[10] Jaedeug Choi and Kee-Eung Kim. Map inference for bayesian inverse reinforcement learning. *Advances in neural information processing systems*, 24, 2011.

[11] Steven De Jong, Karl Tuyls, and Katja Verbeeck. Fairness in multi-agent systems. *The Knowledge Engineering Review*, 23(2):153–180, 2008.

[12] Christos Dimitrakakis and Constantin A Rothkopf. Bayesian multitask inverse reinforcement learning. In *European workshop on reinforcement learning*, pages 273–284. Springer, 2011.

[13] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. 2019.

[14] Jonas Eschmann. Reward function design in reinforcement learning. *Reinforcement learning algorithms: Analysis and Applications*, pages 25–33, 2021.

[15] Ernst Fehr and Klaus M Schmidt. A theory of fairness, competition, and cooperation. *The quarterly journal of economics*, 114(3):817–868, 1999.

[16] Till Freihaut and Giorgia Ramponi. On feasible rewards in multi-agent inverse reinforcement learning, 2025.

[17] Justin Fu, Andrea Tacchetti, Julien Perolat, and Yoram Bachrach. Evaluating strategic structures in multi-agent inverse reinforcement learning. *Journal of Artificial Intelligence Research*, 71:925–951, 2021.

[18] Sule Guney and Ben Newell. Fairness overrides reputation: the importance of fairness considerations in altruistic cooperation. *Frontiers in Human Neuroscience*, 7, 2013.

[19] Werner Güth, Rolf Schmittberger, and Bernd Schwarze. An experimental analysis of ultimatum bargaining. *Journal of economic behavior & organization*, 3(4):367–388, 1982.

[20] Nathaniel Haynam, Adam Khoja, Dhruv Kumar, Vivek Myers, and Erdem Bıyık. Multi-agent inverse q-learning from demonstrations. *arXiv preprint arXiv:2503.04679*, 2025.

[21] Hadi Hosseini. The fairness fair: Bringing human perception into collective decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22624–22631, 2024.

[22] Hong Jun Jeon, Smitha Milli, and Anca Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. *Advances in Neural Information Processing Systems*, 33:4415–4426, 2020.

[23] Wonseok Jeon, Paul Barde, Derek Nowrouzezahrai, and Joelle Pineau. Scalable multi-agent inverse reinforcement learning via actor-attention-critic. *arXiv preprint arXiv:2002.10525*, 2020.

[24] Daniel Kahneman, Jack L. Knetsch, and Richard H. Thaler. Fairness and the assumptions of economics. *The Journal of Business*, 59(4):S285–S300, 1986.

[25] Kuno Kim, Shivam Garg, Kirankumar Shiragur, and Stefano Ermon. Reward identification in inverse reinforcement learning. In *International Conference on Machine Learning*, pages 5496–5505. PMLR, 2021.

[26] Stefanos Leonardos, Georgios Piliouras, and Kelly Spendlove. Exploration-exploitation in multi-agent competition: convergence with bounded rationality. *Advances in Neural Information Processing Systems*, 34:26318–26331, 2021.

[27] Xiaomin Lin, Stephen C Adams, and Peter A Beling. Multi-agent inverse reinforcement learning for certain general-sum stochastic games. *Journal of Artificial Intelligence Research*, 66:473–502, 2019.

[28] Xiaomin Lin, Peter A Beling, and Randy Cogill. Multiagent inverse reinforcement learning for two-person zero-sum games. *IEEE Transactions on Games*, 10(1):56–68, 2017.

[29] Carlos Martin and A Sandholm. Bayesian multiagent inverse reinforcement learning for policy recommendation. In *AAAI Workshop on Reinforcement Learning in Games*, 2021.

[30] Negar Mehr, Mingyu Wang, Maulik Bhatt, and Mac Schwager. Maximum-entropy multi-agent dynamic games: Forward and inverse solutions. *IEEE transactions on robotics*, 39(3):1801–1815, 2023.

[31] Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Multi-agent inverse reinforcement learning. In *2010 ninth international conference on machine learning and applications*, pages 395–400. IEEE, 2010.

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

[33] Paul Nitschke, Lars Lien Ankile, Eura Nofshin, Siddharth Swaroop, Finale Doshi-Velez, and Weiwei Pan. Amber: An entropy maximizing environment design algorithm for inverse reinforcement learning. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024.

[34] Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2043–2044, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[35] Matthew Rabin et al. *Incorporating fairness into game theory and economics*. University of California at Berkeley, Department of Economics, 1992.

[36] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

[37] Giorgia Ramponi, Pavel Kolev, Olivier Pietquin, Niao He, Mathieu Laurière, and Matthieu Geist. On imitation in mean-field games. *Advances in Neural Information Processing Systems*, 36:40426–40437, 2023.

[38] Tummalapalli Sudhamsh Reddy, Vamsikrishna Gopikrishna, Gergely Zaruba, and Manfred Huber. Inverse reinforcement learning for decentralized non-cooperative multiagent systems. In *2012 ieee international conference on systems, man, and cybernetics (smc)*, pages 1930–1935. IEEE, 2012.

[39] Anka Reuel and Devin Ma. Fairness in reinforcement learning: A survey. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 1218–1230, 2024.

[40] Paul Rolland, Luca Viano, Norman Schürhoff, Boris Nikolov, and Volkan Cevher. Identifiability and generalizability from multiple experts in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 35:550–564, 2022.

[41] Constantin A Rothkopf and Christos Dimitrakakis. Preference elicitation and inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*, pages 34–48. Springer, 2011.

[42] Umer Siddique, Peilang Li, and Yongcan Cao. Towards fair and equitable policy learning in cooperative multi-agent reinforcement learning. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2024.

[43] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34:14502–14515, 2021.

[44] Jianrui Wang, Yitian Hong, Jiali Wang, Jiapeng Xu, Yang Tang, Qing-Long Han, and Jürgen Kurths. Cooperative and competitive multi-agent systems: From optimization to games. *IEEE/CAA Journal of Automatica Sinica*, 9(5):763–783, 2022.

[45] W. Yoshida, R. J. Dolan RJ, and K.J. Friston. Game theory of mind. *PLoS Computational Biology*, 4(12), 2008.

[46] Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, pages 7194–7201. PMLR, 2019.

A Details on reward inference.

A.1 Extraction of fairness levels and intrinsic rewards

To sample from the posterior of the fairness levels $\mathcal{L}$ or intrinsic rewards $\mathcal{I}$ independently, it is always possible to marginalise over the aspect we want to overlook:

$$\mathbb{P}(\mathcal{L} \mid \mathcal{X}) = \int_{([r_{\min}, r_{\max}]^{\mathcal{S}\times\mathcal{A}^m})^m} \mathbb{P}(\mathcal{R} \mid \mathcal{X}) \mathrm{d}\mathcal{I}, \quad \mathbb{P}(\mathcal{I} \mid \mathcal{X}) = \int_{[\lambda_{\min}, \lambda_{\max}]^m} \mathbb{P}(\mathcal{R} \mid \mathcal{X}) \mathrm{d}\mathcal{L}.$$

B Additional experimental results

Constant nature of $Z_{\boldsymbol{d}}$. We conducted an empirical analysis to study the invariance of the normalising factor $Z_{\boldsymbol{d}}$ across different policy deployments $\boldsymbol{d}$ for both the NIG and PSG gaps. To achieve this, we generated MGs of varying state and action space sizes, alongside a fixed pool of three agents, each equipped with a common intrinsic reward function r^* and fairness levels $\boldsymbol{\lambda}^*$.

First, we sampled 200 policies from the policy posterior, representing the initial step of the PORP procedure. For each policy deployment $\boldsymbol{d}$, we then estimated the corresponding value of $Z_{\boldsymbol{d}}$.

Given that the gap distributions are highly concentrated around the ground truth parameters $(\boldsymbol{\lambda}^*, r^*)$, we sampled $N = 2 \times 10^4$ fairness levels and intrinsic rewards from truncated normal distributions:

$$\boldsymbol{\lambda}^i \sim \mathcal{N}_{[\lambda_{\min}, \lambda_{\max}]}(\boldsymbol{\lambda}^*, 0.16), \quad r^i \sim \mathcal{N}_{[r_{\min}, r_{\max}]}(r^*, 0.16).$$

An empirical estimate of $Z_{\boldsymbol{d}}$ was then obtained through importance sampling:

$$\hat{Z}_{\boldsymbol{d}} = \frac{1}{N} \sum_{i=1}^{N} \prod_{\boldsymbol{\pi}_p \in \boldsymbol{d}} e^{\Delta_{(\boldsymbol{\lambda}^i, r^i)}(\boldsymbol{\pi}_p)} \cdot (r_{\max} - r_{\min})(\lambda_{\max} - \lambda_{\min})\, q(\boldsymbol{\lambda}^i, r^i),$$

where $q(\boldsymbol{\lambda}^i, r^i)$ is the density of the joint truncated normal distribution at $(\boldsymbol{\lambda}^i, r^i)$. We report the results for the configuration $r_{\max} = 1, r_{\min} = 0, \lambda_{\max} = 1$, and $\lambda_{\min} = 0$ in Figure 3. Our findings indicate that $Z_{\boldsymbol{d}}$ remains nearly constant across different policy deployments $\boldsymbol{d}$, with variations mostly within 20% for both gap functions.

Randomised MGs. We provide error bars (standard error) for every method evaluated in Figure 4.

Collaborative cooking. Figure 4 presents the fairness posteriors obtained from additional runs across varying numbers of demonstrations. We also provide a comparison of error rates for pool sizes of 2 and 3 agents in Table 1. These findings are consistent with our observations from the randomised MG experiments: leveraging multiple agent combinations significantly reduces ambiguity in identifying fairness and intrinsic rewards. Notably, with a pool size of 2, fairness remains practically unidentifiable even with many demonstrations, as average performance falls below the baseline of random guessing.

C Additional experimental details

Suboptimality/entropy parameter. To account for sub-optimality, we divided the interval $[0.04, 1]$ into 19 equal segments and appended an additional segment, $[1, 100]$. We applied an exponential prior, $\mathrm{Exp}(6)$, such that for each interval $[\beta_i, \beta_{i+1}]$, the belief measure is given by:

$$\mathbb{P}([\beta_i, \beta_{i+1}]) = e^{-6\cdot\beta_i} - e^{-6\cdot\beta_{i+1}}.$$

In the randomised MG experiments, the true agents policies used an entropy-parameter of 0.1. For the collaborative cooking experiments, we used a value of 0.05.

ULA. We employed an RMSProp based implementation of ULA, with parameters $\epsilon = 1 \times 10^{-8}, \beta = 0.99$.

Policy and reward parametrisation. We parameterise policies using a softmax representation. Specifically, the MH and ULA samplers operate over the policy parameter space $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, with the policy at state s and action a given by:

$$\pi_{\boldsymbol{\theta}}(s,a) = \frac{\exp(\theta_{s,a})}{\sum_{a'}\exp(\theta_{s,a'})}.$$

For reward modeling, we parameterise intrinsic reward components in $\mathbb{R}$, passing them through an offset sigmoid function for scaling:

$$r_{\boldsymbol{\phi}}(s,a) = \text{sigmoid}(\phi_{s,a}) \cdot (r_{\text{max}} - r_{\text{min}}) + r_{\text{min}},$$

where $\boldsymbol{\phi}$ represents the reward parameters. We also learn fairness for each agent i, λ_i, using similarly an offset sigmoid representation:

$$\lambda_{i,\mathbf{w}} = \text{sigmoid}(w_i) \cdot (\lambda_{\text{max}} - \lambda_{\text{min}}) + \lambda_{\text{min}},$$

where $\mathbf{w}$ are the fairness parameters.

NIG. To keep the NIG differentiable w.r.t. fairness and intrinsic rewards, we computed it under entropy-regularisation, using a small entropy parameter $\beta = 0.04$.

Markov game parameters. For all experiments, we used a discount factor of $\gamma = 0.9$. To generate transition probabilities in randomised Markov games, we used a Dirichlet distribution with parameter 1.2.

Collaborative cooking task. Our implementation of the cooking task is inspired from the Overcooked game, which is widely used as a benchmark for RL and IRL [7, 1, 20]. We provide a visualisation of the sequence of actions needed to deliver a soup in Figure 6. In our setup, game states are described by:

1. Player positions. Players cannot have the same position, except at the start where they overlap.
2. Player state: carrying nothing, a plate, a tomato or a soup.
3. Pot state: empty or ready.

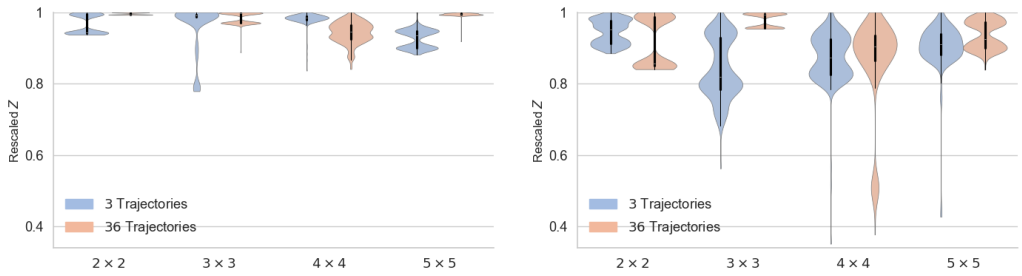
Notably, players do not have orientations, and have full observability of the game. A player has 5 actions:

1. Moving up, down, left and right.
2. Interact. The action's effect depends on the nearby tiles: take a plate, put tomatoes into the pot, etc.

We kept the game simple enough in order to be able to fit everything tractable when computing equilibria, or for it to fit into GPU RAM when performing Bayesian inference.

Implementation. We provide the code used for our experiments at https://github.com/53H3m/fair_mairl

Compute. For randomised Markov game experiments, we used an `Intel(R) Xeon(R) CPU E5-2683 v4` with 32GB of RAM. For the collaborative cooking task, we used Tensorflow's GPU acceleration to sample from the PORP, using an `NVIDIA GeForce RTX 4090` GPU with 24 GB of RAM. See Table 4 for computation times per method.



(a) NIG.

(b) PSG.

Figure 3: Rescaled normalising constant $Z_{\boldsymbol{d}}$ computed across random games of varying sizes ($|\mathcal{S}| \times |\mathcal{A}|$), gap functions, and 200 policy deployments sampled from the policy posterior step of the PORP, with a pool of 3 agents. Each value of $Z_{\boldsymbol{d}}$ is estimated using 2×10^4 reward samples and rescaled by the maximum value attained across all deployments.

Table 1: Mean and standard error of fairness and intrinsic reward errors given a total of 3072 trajectories on the collaborative cooking task, grouped by pool size. Results are averaged over 10 independent runs. All error values are rescaled such that a value of 1 corresponds to the expected error of random guessing. For fairness comparisons, errors are computed exclusively with respect to the two agents common to both pool sizes, ensuring consistency across evaluations. Relative fairness error is the sum of pairwise distance between agents fairness: $\sum_{i,j} \big||\lambda_i - \lambda_j| - |\lambda_i^* - \lambda_j^*|\big|$, capturing the fairness relationships between agents.

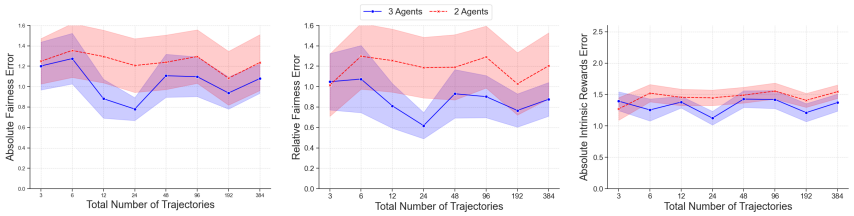
Pool Size	Absolute Fairness Error	Relative Fairness Error	Intrinsic Rewards Error
2	1.16 ± 0.06	0.62 ± 0.03	0.62 ± 0.03
3	$\mathbf{0.46 \pm 0.05}$	$\mathbf{0.25 \pm 0.03}$	$\mathbf{0.21 \pm 0.01}$

Table 2: PORP Hyperparameters for Random MGs and Collaborative Cooking experiments.

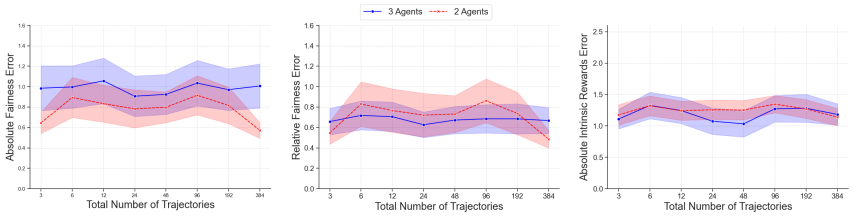
Hyperparameter	Random MGs	Collaborative Cooking
c (NIG)		25
c (PSG)		15
Sampling warmup fraction		0.5
ULA learning rate (policy)	0.05	0.1
ULA Number of samples (policy)		6000
ULA learning rate (reward)		0.2
ULA Number of samples (reward)	5000	9000
MH step size (policy)	$0.01 \leftrightarrow 1$	N/A
MH Number of samples (policy)	16000	N/A
MH step size (reward)	$0.01 \leftrightarrow 1$	N/A
MH Number of samples (reward)	5000	N/A

Table 3: DRP Hyperparameters for Random MGs and Collaborative Cooking experiments.

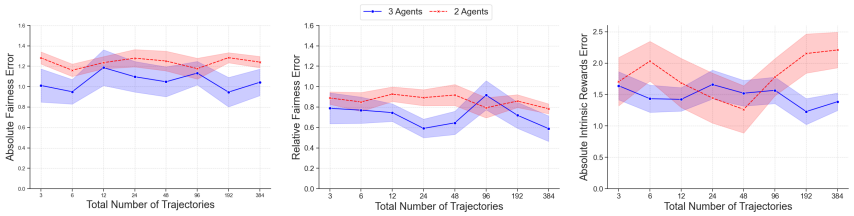
Hyperparameter	Random MGs	Collaborative Cooking
Sampling warmup fraction (reward)	0.5	N/A
ULA learning rate	0.03	N/A
ULA Number of samples	5000	N/A
MH step size	$0.01 \leftrightarrow 1$	N/A
MH Number of samples	5000	N/A



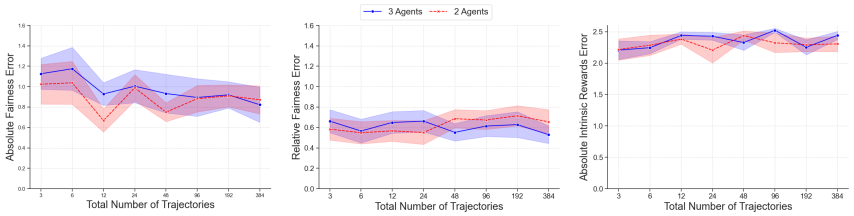
(a) MH-DRP.



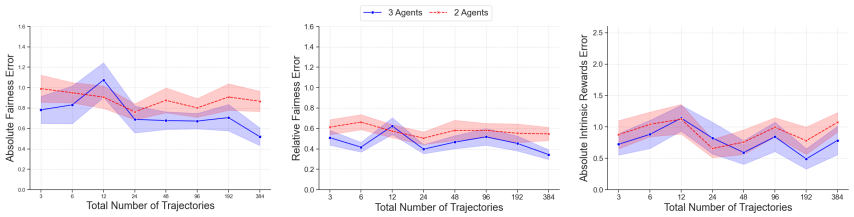
(b) ULA-DRP.



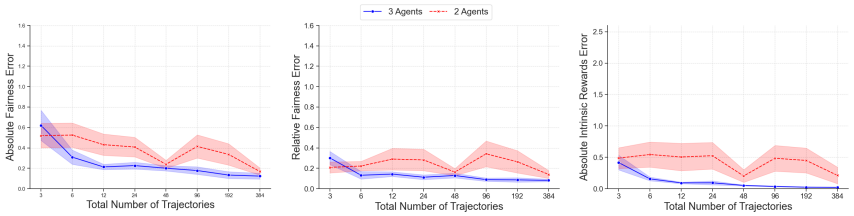
(c) MH-PORP-NIG.



(d) ULA-PORP-NIG.



(e) MH-PORP-PSG.



(f) ULA-PORP-PSG.

Figure 4: Average inference errors over trajectories. Errors are rescaled such that an error of 1 corresponds to the expected error of random guessing. Shaded areas represent standard error, computed over 10 distinct seeds. Relative fairness error is the sum of pairwise distance between agents fairness: $\sum_{i,j} \left| |\lambda_i - \lambda_j| - |\lambda_i^* - \lambda_j^*| \right|$, capturing the fairness relationships between agents.

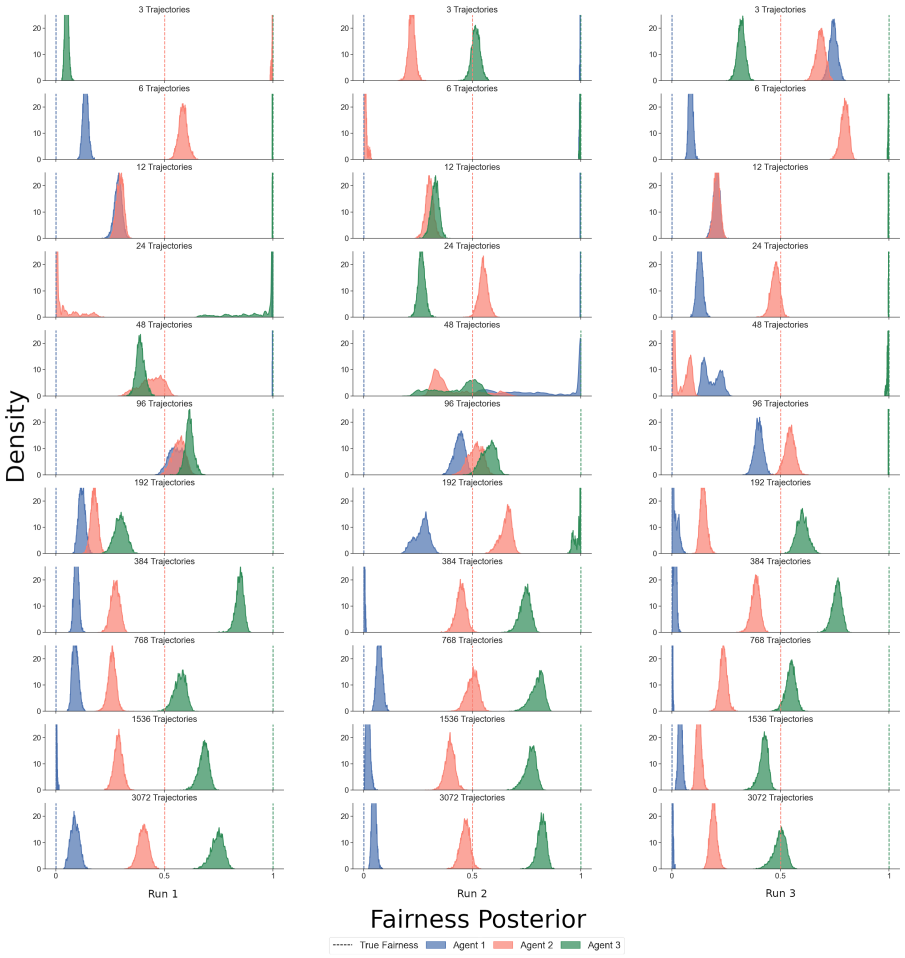
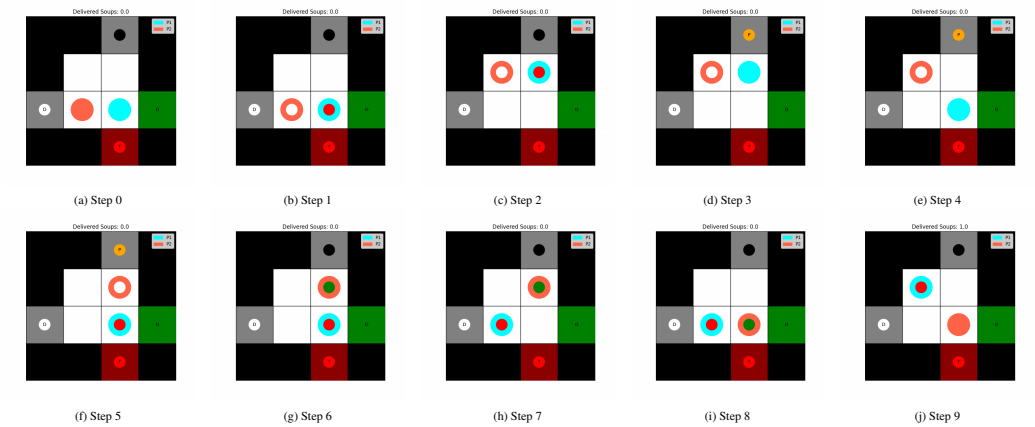


Figure 5: Fairness posteriors obtained given an increasing number of demonstrations, for 3 different seeds.

Table 4: Average computation time for all methods on randomised MGs with 5 states, 3 actions, and 384 trajectories, evaluated on an `Intel(R) Xeon(R) CPU E5-2683 v4`.

Method	**Computation Time (seconds)**
ULA-PORP-NIG	6896.5
MH-PORP-NIG	4711.0
ULA-DRP	1785.0
ULA-PORP-PSG	1098.0
MH-PORP-PSG	933.0
MH-DRP	732.5



(a) Step 0 (b) Step 1 (c) Step 2 (d) Step 3 (e) Step 4

(f) Step 5 (g) Step 6 (h) Step 7 (i) Step 8 (j) Step 9

Figure 6: Example trajectory extract in the collaborative cooking task. Blue and red circles represent the players. The red tile marks the tomato stand, while the white circle indicates the plate stand. The top black/yellow circle shows the pot, with black representing empty and yellow representing filled. Finally, the green tile designates the delivery location where players must deliver their soup. (a) players pick up a plate and a tomato. (d) Blue player puts its tomato into the pot: the soup is ready. (g) Red player fills its plate with soup. (j) Red player delivers the soup.