EMOTE: An Explainable Architecture for Modelling the Other through Empathy

Manisha Senadeera[1] , **Thommen Karimpanal George** [1,2] , **Stephan Jacobs**[3] , **Sunil Gupta**[1] and **Santu Rana**[1]
[1]Applied Artificial Intelligence Institute, Deakin University
[2]School of Information Technology, Deakin University
[3]University of Queensland
{msenadeera, thommen.karimpanalgeorge, sunil.gupta, santu.rana}@deakin.edu.au, s.jacobs@uq.edu.au

Abstract

Empathy allows us to assume others are like us and have goals analogous to our own. This can also at times be applied to multi-agent games - e.g. Agent 1's attraction to green balls is analogous to Agent 2's attraction to red balls. Drawing inspiration from empathy, we propose EMOTE, a simple and explainable inverse reinforcement learning (IRL) approach designed to model another agent's action-value function and from it, infer a unique reward function. This is done by referencing the learning agent's own action value function, removing the need to maintain independent action-value estimates for the modelled agents whilst simultaneously addressing the ill-posed nature of IRL by inferring a unique reward function. We experiment on minigrid environments showing EMOTE: (a) produces more consistent reward estimates relative to other IRL baselines (b) is robust in scenarios with composite reward and action-value functions (c) produces human-interpretable states, helping to explain how the agent views other agents.

1 Introduction

Empathyis a process which helps us understand the feelings and goals of another, by using ourselves as a point of reference. Hoffman [1996] defines it as *"any process where the attended perception of the object's state generates a state in the subject that is more applicable to the object's state or situation than to the subject's own prior state or situation."* So when we ask: *How would we feel if we were in a similar situation? What objects or goals, though not exactly the same, do we feel similarly towards?*, an implicit assumption is the belief that who we are observing is analogous or similar to us, even if our goals or preferences are different. As a simple example: I love chocolate but dislike liquorice. If I were to see someone eating and enjoying liquorice, I would not immediately relate to their enjoyment. However, if I *imagined* they are instead eating chocolate, I would understand their joy and subsequently infer the levels of enjoyment as being similar to mine when eating chocolate. In this work, we aim to translate this intuition into an explainable approach for inverse reinforcement learning (IRL) for analogous agents. Specifically, we consider two agents to be analogous if for every feature that one agent values, there exists a corresponding feature that the second agent values to a similar degree. We refer to such similarly valued features as *analogous features* i.e. my love of chocolate is analogous to the other's love of liquorice. We note that analogous features may be identical (e.g., neutral features such as the chocolate/liquorice wrapper which both agents interact with in the same way) or different (chocolate or liquorice itself).

Inspired by empathy, we present EMOTE - an *Explainable architecture for Modelling the Other Through Empathy*. EMOTE allows a single learning agent to model one or more independent agents behaving under fixed policies (e.g. pretrained robots), the reward functions of which the learning agent is not privy to. A two-stage neural network architecture is used to infer the action-value function and corresponding reward function for each independent agent. Assuming an analogous relationship with the learning agent an *Imagination Network* learns an empathetic representation of the independent agent's state during the first stage. We interpret the resulting *empathetic state* to be the independent agent's state, as imagined by the learning agent (e.g. liquorice reimagined as chocolate) from an empathetic perspective. This empathetic state is fed into a second network, a copy of the learning agent's own action-value function, to observe what values the learning agent associates with this empathetic state. The resulting value is then assumed to be the value associated to the independent agent's state by the independent agent.

Designing the learning agent to reference its own action-value function leads to a more stable representation of the independent agent's action-values and rewards, as these quantities become grounded in the learning agent's value function (learned through interactions), alleviating the need to maintain independent action-value estimates for each independent agent and naturally ensures that the rewards of all agents lie in a similar scale, a feature that is beneficial in multi agent approaches based on composite reward and value functions [Noothigattu *et al.*, 2019; Alamdari *et al.*, 2021; Raileanu *et al.*, 2018; Senadeera *et al.*, 2022]. A key challenge of IRL is it's ill-posed nature [Ng *et al.*, 1999] (existence of many possible reward functions corresponding to an optimal behaviour). For example if you see an agent moving from point A to point B, one reward function to explain the behaviour is that the agent is drawn towards point

B. An equally possibly reward function however is that the agent is repelled from A. By referencing the learning agents value function and rewards, under the analogousness assumption, our method naturally rectifies this and is able to infer a unique reward function. Additionally, leveraging the learning agent's own functions allows the independent agent's empathetic state to be *human-interpretable*, permitting interrogation of the inferences made by the learning agent about the other agents' goals. Finally in contrast to most inverse reinforcement learning (IRL) methods, our approach works online with sampled transitions of the modelled agent, without the need to solve the MDP for each reward function estimate.

The benefits of EMOTE are: (1) inference of a unique solution to the IRL problem, (2) ensuring the independent agent's action-values and inferred rewards lie in a similar scale to that of the learning agent. (3) Ability to infer the reward function online with sampled transitions from the independent agent without needing iterative reward function estimates (4) Generation of empathetic states that are human-interpretable.

We first compare EMOTE to other IRL methods showing it produces more realistic inferred reward values. EMOTE's ability to handle composite action-value and reward functions algorithms is evaluated. Specifically, using assistive and adversarial multiagent scenarios, we show that EMOTE produces more consistent reward estimates of the independent agent that are robust to various settings and layouts. Further, we show the Imagination Network is capable of recovering interpretable empathetic states (i.e., finding analogous features), proving beneficial for explainability.

2 Related Literature

Modelling the other: Modelling other agents is a key component of multi-agent reinforcement learning (MARL). There exists a vast body of work ranging from inferring the other's policy [Foerster *et al.*, 2018; Wen *et al.*, 2019; Hu *et al.*, 2020; Shu and Tian, 2018], goals and beliefs [Raileanu *et al.*, 2018; Moreno *et al.*, 2021] and value functions [Zhao *et al.*, 2022; He *et al.*, 2016]. These works predominantly assume all agents are trained concurrently which differs from our intended setting where only a single agent is trained. Modelling agents who behave under a fixed pre-trained policy are typically tackled in Theory of Mind (ToM) [Rabinowitz *et al.*, 2018] and Inverse Reinforcement Learning (IRL) [Ng *et al.*, 2000] literature. A small subset combines the two problems to create environments in which a learning agent (to be trained) coexists with and models a pre-trained (independent) agent [Papoudakis *et al.*, 2021; Senadeera *et al.*, 2022], in line with the problem setting of our work. However, such works generally do not produce interpretable models or produce arbitrarily scaled action-value estimates, impeding the accurate inference of the agents' behaviour.

Inverse Reinforcement Learning: Within the multi-agent IRL literature the primary focus is inferring rewards of multiple agents. This can range from independent agents whose rewards are used to train a central controller [Natarajan *et al.*, 2010] or understanding an individual's reward function by observing the collective behaviour towards a task [Wu *et al.*, 2023; Reddy *et al.*, 2012; Lin *et al.*, 2017]. As our work assumes modelling each of the independent agents separately (producing an individual action-value function and reward) we compare our work to the single agent IRL literature. Limitations of past works include needing access to the modelled agent's policy [Ng *et al.*, 2000], the ability to iteratively solve the MDP [Ziebart *et al.*, 2008], access to other policy trajectories from the modelled agent [Boularias *et al.*, 2011] and full trajectories [Klein *et al.*, 2012]. The work closest to ours is that of the Cascaded Supervised IRL (CSI) algorithm [Klein *et al.*, 2013]. In this work, the authors use a supervised learning method to infer a proxy optimal action-value function and apply a rearrangement of the Bellman Equation to infer the reward function. In our work we augment the action-value function model in the CSI algorithm to be appropriate for our multi-agent setting, with the added benefit of explainability. Other notable works which don't require iteratively solving the MDP include Relative Entropy (RE) [Boularias *et al.*, 2011] which uses stochastic gradient descent to minimise the relative entropy between the trajectory distributions of an expert and another baseline policy. Another is Structured Classification for IRL (SCIRL) [Klein *et al.*, 2012] which, similar to CSI, uses a proxy to the action-value function but instead uses a linear parameterisation for this action-value function based on feature expectations. One of the fundamental problems in IRL is the existence of a large number of candidate reward functions corresponding to a given arbitrary behaviour, making it difficult to consistently infer a correct reward function. By referencing the learning agent's own action-value function, our approach infers a unique reward function, elegantly narrowing the space of candidate reward functions.

Modelling based on oneself: Some works model the other agent based on their own model. Raileanu *et al.,* [2018] trains the learning agent on all possible goals and uses this information to infer the hidden goal of the other agent. Inspired by empathy as well, Bussmann *et al.,* [2019] proposes imposing the learning agent's value function directly on the independent agent, using this to infer the other's intent. An obvious limitation is that this assumes the independent agent has the same values as the learner. Our work eschews this assumption, allowing for different and even opposing intentions.

Composite value and reward functions: In multi-agent scenarios with composite reward or value functions (e.g. summation of two or more reward/value estimates), appropriate scaling is important to ensure stable behaviours. Alamdari *et al.,* [2021] builds a joint function of learning agent and independent agent rewards (whose rewards are already known). A similar joint reward function is built by Senadeera *et al.,* [2022] although they use IRL to infer the rewards of the independent agent. As a result of the space of potential rewards that can emerge through IRL, Noothigattu *et al.,* [2019] mitigates the issues of a misalignment by scaling the independent agent's functions by a constant (the ratio of the $l1$ norms of the learning agent's reward vector and the IRL inferred rewards of the independent agent). This simple $l1$ norm based normalisation fails in complex scenarios, and is constrained to combining only two reward or action-value functions. Our architecture ensures that the range of the inferred action-value and reward functions will be comparable to those of the learning agent, obviating the need for any additional scaling.

3 Methodology

3.1 Problem Setting

We formulate our problem within the context of a Markov Decision Process (MDP) framework [Puterman, 2014] $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma >$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the space of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition function governing the probability of moving to the next state s', having taken an action a in the current state s. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow R$ defines the reward an agent receives for taking an action in the current state, and $\gamma \in (0, 1]$ is the discount factor.

To model the independent agent using EMOTE, we consider settings consisting of a learning agent, which we train, and one (or more) independent agent(s) (sharing the same action and state spaces) and behaving as per a fixed (time independent) policy, which may be deterministic or stochastic. The underlying reward function of the independent agent is unknown to the learning agent. Our EMOTE architecture consists of training the action-value function for the learning agent Q_{learn} using rewards R returned from the environment, as is done in standard RL implementations. Simultaneously we estimate the independent agent's action-value function $\hat{Q}_{indep}$ in a supervised manner (as explained later in Section 3.2) using its $< s_i, a_i, s_i^{'} >$ transitions which we assume to be accessible (similar to a real world setting involving robots who share sensory input information). Each agent is assumed to also have their own reward function R_{indep}, initially unknown to the learning agent, and later inferred as $\hat{R}_{indep}$ from the estimated action-value function $\hat{Q}_{indep}$.

3.2 The EMOTE Architecture

The components of the EMOTE architecture used to estimate the independent agent's value function $\hat{Q}_{indep}$ are shown in Figure 1. It consists of a two stage neural network architecture. The first of these, the Imagination Network ($M_{imagine}$) parameterised by $\theta_{imagine}$, takes as input state s_i, as observed by the independent agent, and outputs an empathetic state s_e representing the independent agent's state as *imagined* empathetically by the learning agent. We assume there exists a $\theta_{imagine}$ where:

$$\forall s_i, s_e = M_{imagine}(s_i; \theta_{imagine}) \quad (1)$$

such that the learning agent's greedy action in s_e matches the independent agent's action a_i in state s_i. Formally, we define the empathetic state in Definition 3.1:

Definition 3.1 (Empathetic State)**.** In a multiagent reinforcement learning scenario involving a learning agent with action-value function Q_{learn} and an independent agent (sharing the same state space $\mathcal{S}$ and action space $\mathcal{A}$) who behaves as per an arbitrary fixed (time independent) policy π_{fixed}, an empathetic state s_e is a state where:

$$\underset{a'}{argmax}\ Q_{learn}(s_e, a') = a_i$$

where $a_i \sim \pi_{fixed}(s_i)$. Consequently, under the analogousness assumption, the action-value function of the learning agent and independent agent are equivalent for actions a_i taken by the independent agent, such that:

$$Q_{learn}(s_e, a_i) \approx \hat{Q}_{indep}(s_i, a_i)$$

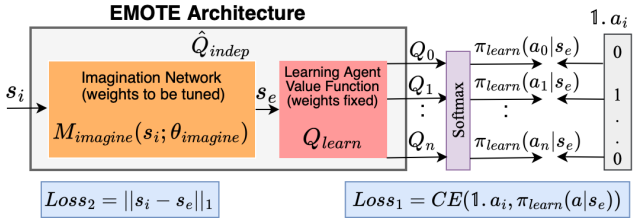


Figure 1: EMOTE architecture of the independent agent's value function $\hat{Q}_{indep}$ comprising a two stage neural network. The first is the Imagination Network $M_{imagine}$ which takes in the state s_i perceived by the independent agent and outputs an empathetic state s_e. The second is a copy of the learning agent's value function Q_{learn}. s_e is fed into Q_{learn} and associated Q-values are output. Only $M_{imagine}$ is trained via a loss function comprising the difference between s_i and s_e and the categorical cross entropy between the predicted softmax actions $\pi_{learn}(a|s)$ and the observed one hot encoded action of the independent agent $1 \cdot a_i$.

In order to estimate the independent agent's action values, we pass the obtained empathetic state s_e through a copy of the learning agent's own action-value function Q_{learn} (the second stage of the architecture). The intuition is that since s_e for the learning agent produces the same actions (behaviours) as the independent agent in state s_i, it is reasonable to assume that for a given action, how the independent agent values s_i is similar to how the learning agent values s_e. Having now obtained $\hat{Q}_{indep}$, we can infer the rewards of the independent agent $\hat{R}_{indep}$ (which is assumed to follow a fixed policy) as per the inverted Bellman equation shown in Equation 2:

$$\hat{R}_{indep}(s, a) = \hat{Q}_{indep}(s, a) - \gamma \max_{a'} \hat{Q}_{indep}(s', a') \quad (2)$$

where γ is the discount factor. As $\hat{Q}_{indep}$ is inferred from Q_{learn}, as Q_{learn} converges, $\hat{Q}_{indep}$ also converges, and consequently, $\hat{R}_{indep}$ also converges to a unique solution. This is due to the one to one relationship between a reward function and action-value function [Klein *et al.*, 2013]. Moreover, as $\hat{Q}_{indep}$ is grounded to values produced by Q_{learn} ($\approx \hat{Q}_{indep}$ as shown previously), we can expect $\hat{R}_{indep}$ to have a similar scale or magnitude as the rewards R that produced Q_{learn}.

Loss Terms While the weights of Q_{learn} are trained using DQN, we train $M_{imagine}$ using a loss term (Equation 3) comprising two parts. The first ($Loss_1$) is a categorical cross entropy (CE) loss which minimises the difference between the softmax predicted action from the Q_{learn} copy and the action (a_i) actually taken by the independent agent. This loss is to ensure the greedy action of Q_{learn}, for s_e, matches the independent agent's action (per Equation 1).

$$\mathcal{L}(\theta_{imagine}) = (1 - \delta) \underbrace{CE(1.a_i, \pi_{learn}(a|s_e))}_{Loss_1} + \delta \underbrace{\|s_i - s_e\|_1}_{Loss_2} \quad (3)$$

where

$$\pi_{learn}(a|s_e) = \frac{e^{Q_{learn}(s_e, a)}}{\sum_a e^{Q_{learn}(s_e, a)}} \quad (4)$$

and s_e is obtained as per Equation 1. The loss term $Loss_2$ focuses on state reconstruction, aiming to produce an empathetic state s_e matching the original state s_i. Together, the goal is to produce an empathetic state s_e through minimal changes to s_i, while still satisfying the behavioural similarities specified via $Loss_1$. Components used to construct the two loss terms are shown in Figure 1.

The hyperparameter $\delta \in [0, 1]$ balances the importance of reconstructing the empathetic state s_e to be similar to the original state s_i (interpretability) and the accuracy of the learning agent's predicted actions in s_e. We note that in practice, a copy of Q_{learn} for EMOTE is updated every few episodes (to maintain stability).

4 Experiments

We conducted two sets of experiments to assess EMOTE's performance, all of which were played in finite, episodic environments where actions are taken sequentially by each agent. The first is a 3 Agent Game environment to evaluate the effectiveness of EMOTE as an IRL algorithm. The second, conducted on four different environment settings, was to evaluate EMOTE's suitability to applications involving composite reward and action-value functions. In both experiments the primary goal of the learning agent (red arrow) is to collect all red pellets (done by stepping on top of them). The independent agents also try to collect their own pellets. All games end when the learning agent has either collected all of its pellets, the timer runs out or, in the adversarial games, the learning agent is harmed. Using the inferred rewards and action-value function of the independent agent, an augmented reward is used to influence the learning agent's behaviour within the game whilst maintaining the primary goal of collecting all of its pellets. Details of the reward functions for each game are shown in Figure 2. Supplementary materials available[1].

3-Agent Game Experiment This environment contains 2 independent agents alongside the learning agent. The learning agent can collect pellets of any colour, causing a reduction in the number of pellets available to other agents, but only receives rewards for collecting its own. The learning agent may choose to step on another's pellet to quickly reach one of its own. In order to reduce the other agent pellets consumed, we train the learning agent on an augmented reward considering the other two agent's preferences. The reward R' used to train the learning agent is defined below where $\hat{R}_{IA1}$ and $\hat{R}_{IA2}$ are the inferred rewards of the two independent agents and λ is a hyperparameter to balance the importance of the learning agent's own rewards and that of the other agents.

$$R^{'} = \lambda R + (1 - \lambda)(\hat{R}_{IA1} + \hat{R}_{IA2}) \tag{5}$$

This experiment was designed to (1) compare EMOTE to existing IRL methods (2) examine its performance modelling multiple independent agents. The baseline IRL algorithms selected were CSI [Klein *et al.*, 2013], SCIRL [Klein *et al.*, 2012] and RE [Boularias *et al.*, 2011] as they operated with trajectories from the fixed optimal policies of the independent agent. The RE algorithm required a comparative trajectory from a random policy which was prepared beforehand.

Sympathy Framework Experiment The second set of experiments was selected to demonstrate the effectiveness of EMOTE in composite reward and action-value function applications. Our work is constrained to settings that meet the criteria outlined in Section 3.1. To the best of our knowledge, works that fit these settings were [Senadeera *et al.*, 2022; Bussmann *et al.*, 2019; Noothigattu *et al.*, 2019; Papoudakis *et al.*, 2021]. We demonstrate integration of EMOTE on Senadeera *et al.,* [2022] Sympathy Framework only, as other works assumed the independent agent's behaviour was (a) random [Bussmann *et al.*, 2019], (b) involved a complex switching policy to train the learning agent involving imitation learning via human example trajectories [Noothigattu *et al.*, 2019] or (c) did not have access to the independent agent's trajectories during testing [Papoudakis *et al.*, 2021]. The Sympathy Framework was designed to elicit considerate behaviours in the learning agent towards the independent agent under both an assistive and adversarial setting. We set the environmental rewards for the learning agent to be the same across the assistive and adversarial environments to illustrate the ability to infer a consistent reward functions, even when the environment layout changes.

Assistive 1 and 2 In Assistive 1 the independent agent is locked behind a door and in Assistive 2 one of the independent agent's pellets is locked behind the door, which can only be opened by the learning agent stepping on a button positioned in front of it. Once pressed, the door remains open. Pellets are placed randomly in each episode. The objective of this game is to train a considerate learning agent who will open the door and assist the independent agent.

Adversarial 1 and 2 The independent agent can harm the learning agent (ending the game). If the learning agent steps on the button feature (switching the button status from 0 to 1 for a finite period of time) it can harm the independent agent, accumulating a positive reward. Harming occurs when the harming agent is within 1 square from the other agent. This framework aims to train the learning agent to avoid harming the independent agent despite receiving positive environment rewards for doing so. In Adversarial 2 the learning agent is only concerned with pellet collection and the independent agent is only concerned with harming the learning agent.

To demonstrate the versatility of EMOTE, we designed experiments that (1) illustrate various environment settings (two assistive and two adversarial), (2) demonstrate the ability to construct the empathetic state both as a feature-by-feature transformation, or a whole state (image) transformation, and (3) demonstrate how our approach can handle both visual and non-visual features that influence environment dynamics in a complex and non-linear fashion. Environments were designed using MarlGrid [Ndousse, 2020] based on MiniGrid [Chevalier-Boisvert *et al.*, 2018]. EMOTE is assessed by:

1. **Performance:** Whether EMOTE can adapt the learning agent's behaviour successfully to the task.
2. **Inferred Independent Agent Rewards:** Are the rewards inferred via IRL from the EMOTE architecture comparable to the learning agent's rewards?

[1] https://github.com/manishasena/EMOTE

3. **Explainability**: Do the analogous features and empathetic states explain the independent agent's behaviour?

4.1 Baselines

In each experiment, policies are learned via DQN [Mnih *et al.*, 2015]. A 5x5 vision field is imposed around each agent, representing visual state information. Among the following baselines, CSI, RE and SCIRL were only used in the 3 Agent Game experiment, and the Sympathy baseline only run in the Sympathy Framework experiment:

EMOTE baselines: We present two baselines trained on the EMOTE architecture - E-Feature and E-Image:

E-Feature: A feature-based Imagination Model where each state cell is represented as a feature. s_e is constructed with a feature-by-feature transformation.

E-Image: An Image-based Imagination Model, where the entire state is transformed to create s_e.

Selfish: A greedy Learning agent who does not model or consider the independent agent's preferences.

CSI: The CSI algorithm.

SCIRL: The SCIRL algorithm.

RE: The RE algorithm.

Sympathy: Agents trained as per [Senadeera *et al.*, 2022] using the CSI algorithm to behave in a manner that is considerate to the other agent, while still being able to achieve its own objectives.

Benchmark (3 Agent Game): Imagination Model is replaced with rule-based transformation, where independent agents' pellets are replaced with learning agent's pellet.

Benchmark (Sympathy Framework): Imagination Model is replaced with a transformation that swaps the agent's pellet colours and makes the button invisible in the observed state, while button status remains. This mimics the hypothesis that how much the learning agent values its pellets is how much the independent agent values its own pellets and the independent agent does not consider the button to be important (treats it the same as the floor), as it cannot press it.

4.2 Performance

Table 1a shows the Win rate and the learning agent's consumption of independent agent pellets in the 3 Agent system. Table 1b shows the Win Rate and Door Open rate for the two Assistive environments, and Harm Rate (learning agent harming independent agent) for each adversarial experiment.

In the 3 Agent Game, compared to the three IRL algorithms (CSI, SCIRL and RE), both EMOTE runs (*E-Feature* and *E-Image*) consistently produce high win rates (on average $>$0.93) and reduced consumption of the independent agent's pellets ($<$1.88 or 1.25 for the two independent agents, respectively), performing close to the benchmark performance. Though SCIRL and RE produced a slightly higher win rate, this was paired with higher consumption of independent agent's pellets. For the Sympathy Framework, in the Assistive environments, *E-Feature* and *E-Image* result in high win ($>$0.92) and door open rates ($>$0.83 on average), producing similar or better results to the *Sympathy* baseline, at times even outperforming *Benchmark*. In the adversarial environments, all harm rates were lower than the *Selfish* baseline. *E-Feature*, *E-Image* and *Benchmark* outperformed *Sympathy* with a lower harm rate ($<$0.24, compared to the 0.45 for *Sympathy* in Adversarial 1), and produced higher win rates ($>$0.4, compared to the highest average for *Sympathy* being 0.31). Compared to *Selfish* win rates of other baselines were either similar (Adversarial 1) or lower (Adversarial 2). Overall, EMOTE had more considerate behaviours than *Sympathy*.

4.3 Inferred Reward Values

Figure 2 shows the independent agent's inferred rewards ($\hat{R}_{indep}$) from the EMOTE and *Benchmark* action-value functions ($\hat{Q}_{indep}$), and contrasts them against those from CSI, SCIRL and RE (3 Agent Game) and *Sympathy* (from Sympathy Framework games). For reference the learning agent's environmental rewards for the same features are shown alongside the independent agent's $\hat{R}_{indep}$ to examine similarities in reward values and identification of analogous features.

3 Agent Game

One would expect the independent agents consuming their pellets (IA1/2 Pellet) to look similar to the learning agent's reward for consuming its own pellet (LA pellet). For *E-Feature* and *E-Image*, a notable positive value is inferred for IA 1 Pellet in Figure 2a ($\hat{R}_{IA1}$) and similarly for IA 2 Pellet in Figure 2b ($\hat{R}_{IA2}$). This inference was not consistently observed by SCIRL or RE methods. For CSI, a strong negative reward was associated with taking a step, whist the value inferred by *E-Feature* and *E-Image* were closer to the learning agent's step value. For all features, the *Benchmark* reward inferences matched closely in range to the learning agent.

Assistive

A similar relationship between LA Pellet and IA Pellet is expected in the Assistive games. In Figure 2c and Figure 2d (Assistive 1 and 2) *E-Feature*, *E-Image*, and the *Benchmark* baselines infer positive rewards for IA Pellet, although their magnitudes vary. The rewards inferred by the *Benchmarks* are closest to the learning agent's reward for LA pellet (+10). Under EMOTE and *Benchmark* baselines, the rewards inferred for Assistive 1 and 2 are similar. This demonstrates the architectures potential to learn a robust model with consistent reward values even when the environment layout changes. In contrast, *Sympathy* inferred different rewards for the two environments (strong positive for button press in Assistive 1, but slight negative in Assistive 2 (Figure 2a and b). Additionally despite high win rate in Assistive 2, *Sympathy* baseline did not result in as high of a door opening rate (Table 1b) as it wrongly inferred a negative reward for door opening.

Adversarial

For the adversarial games (Figure 2d and e) *E-Feature*, *E-Image* and *Benchmark* captured the strong negative reward the independent agent associates with being harmed similar to that the learning agent receives when killed. The *Sympathy* baseline was not able to infer a negative reward for the independent agent being harmed. It also failed to capture a strong enough positive reward for the independent agent consuming a pellet and inferred a strong negative reward for taking a step and pressing the button, leading to a low win rate and

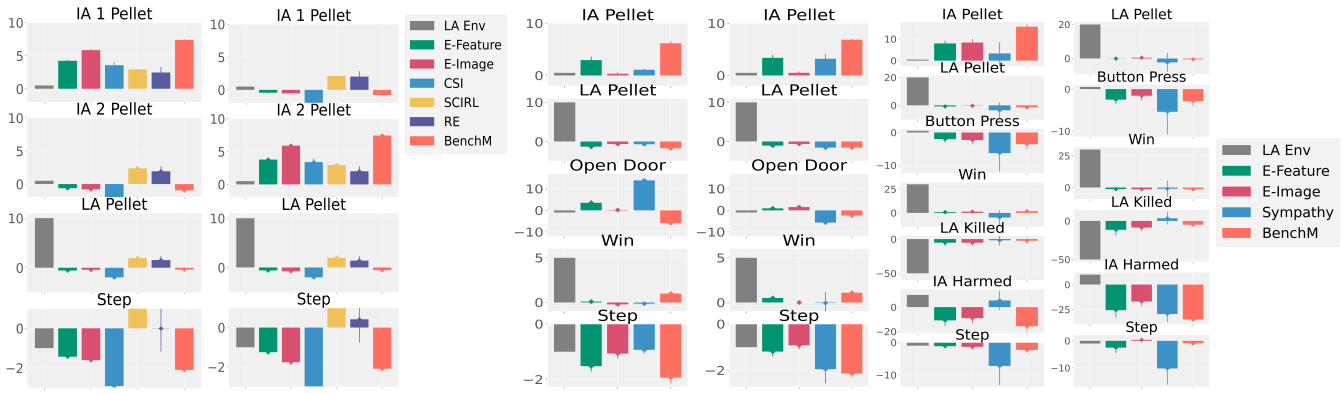
	Win	IA 1 Pellet	IA 2 Pellet
Selfish	0.95±0.04	2.38±0.38	1.66±0.33
E-Feature	0.93±0.06	1.88±0.35	1.25±0.31
E-Image	0.93±0.06	1.75±0.35	1.08±0.3
CSI	0.87±0.09	1.87±0.33	1.33±0.31
SCIRL	0.96±0.04	2.29±0.37	1.72±0.36
RE	0.94±0.05	1.9±0.35	1.26±0.31
BenchM	0.91±0.07	1.75±0.33	1.08±0.3

(a) 3 Agent Game

		Selfish	Sympathy	E-Feature	E-Image	Benchmark
Ass 1	Door	0.44±0.15	0.99±0.01	0.99±0.01	0.83±0.11	1±0
	Win	0.8±0.11	0.93±0.05	0.93±0.05	0.92±0.06	0.93±0.05
Ass. 2	Door	0.09±0.07	0.17±0.1	0.4±0.15	0.38±0.15	0.19±0.11
	Win	0.89±0.08	0.92±0.06	0.94±0.05	0.92±0.06	0.92±0.06
Adv. 1	Harm	0.5±0.15	0.45±0.15	0.24±0.12	0.17±0.1	0.1±0.07
	Win	0.46±0.15	0.26±0.13	0.4±0.15	0.41±0.15	0.34±0.14
Adv. 2	Harm	0.29±0.13	0.12±0.09	0.1±0.08	0.12±0.09	0.03±0.02
	Win	0.73±0.13	0.31±0.14	0.44±0.15	0.47±0.15	0.33±0.14

(b) Sympathy Framework Games

Table 1: Performance results. 3 Agent Game: EMOTE has overall high Win rate while lower consumption of Indep Agent pellets. Sympathy Framework: Assistive games show win rate (higher is better) and door-opening rate (higher is better). Adversarial games show win rate (higher is better) and harm rate (lower is better) from the learning agent toward the independent agent.



(a) $\hat{R}_{IA1}$ (b) $\hat{R}_{IA2}$ (c) Assistive 1 (d) Assistive 2 (e) Adversarial 1 (f) Adversarial 2

Figure 2: Indep Agent's (IA) $\hat{R}_{indep}$ (averaged over last 100 episodes). Learning Agent's (LA) rewards included for reference. (a)-(b): 3 Agent Game inferred rewards for Independent Agent 1 and 2, respectively. (c)-(f) are from the Sympathy Framework environments. *CSI, SCIRL, RE and Sympathy* rewards are scaled to have the same $l1$ norm as the LA's rewards.

	s_i E-Feature		s_i E-Image	
Game	0	1	0	1
Adv 1	0.76 ± 0.11	0.24 ± 0.11	0.79 ± 0.09	0.22 ± 0.10
Adv 2	0.85 ± 0.32	0.15 ± 0.32	0.88 ± 0.16	0.12 ± 0.16

Table 2: Adversarial: s_e button status for s_i button status

high harm rate. We expect the poor performance is due to the $l1$ norm scaling. This inappropriate scaling of the reward components in turn results in poor estimates of the sympathetic reward. In Adversarial 2, *Sympathy* inferred a strong negative for independent agent being harmed, leading to a reduced harm rate. Despite this *Sympathy* had lower win rates compared to EMOTE or *Benchmarks* (Table 1b).

4.4 Empathetic State

EMOTE produces interpretable empathetic states s_e explaining some of the Section 4.2 results. Figure 3 and 4 shows original state s_i for each game alongside empathetic state s_e (at the end of training) from the *E-Feature* and *E-Image*. The learning agent's pellet (LP) is fairly consistently transformed to the floor by s_e, indicating lack of importance the independent agent(s) places on it. Additionally, the colour of independent agent's pellets (IP) becomes that of the learning agent's pellets (magenta) explaining the inferred rewards for IA Pellet in Figure 2. This suggests the architecture interprets an analogous relationship between the independent agent and its pellets and the learning agent and its own pellets.

In s_e for Assistive 1 (Figure 4), the button either remains unchanged, changes to a door or an IP. As the independent agent cannot interact with it, s_e maps it to irrelevant features from the learning agent's perspective. In contrast, Assistive 2 transforms the button at times to LP. This is expected, as how the independent agent moves towards the door (in front of which the button is placed) is similar to how the learning agent moves towards its own pellet.

In Adversarial 1 and 2, the button usually disappears in s_e. This is expected as it is not important for the independent agent who can't influence it. However, the predicted button status in Table 2 shows for a button status of 0 in s_i (button is

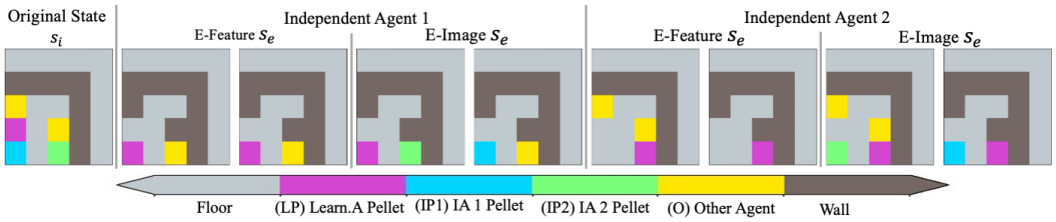


Figure 3: Empathetic states s_e produced for original state s_i (Col 1) as seen by Independent Agent 1 (Col 2 - 5) and Agent 2 (Col 6 - 9). Generated by E-Feature (Col 2-3,6-7) and E-Image (Col 4-5,8-9). Details on s_e construction provided in Supplementary.

Figure 4: Empathetic states s_e produced by E-Feature (Col 2 - 4) and E-Image (Col 5 - 7) for original state s_i (Col 1).

inactive and independent agent can harm the learning agent) the resulting status prediction in s_e is closer to 1, while a button status value of 1 in s_i (button is active and learning agent can harm the independent agent) produces a prediction in s_e closer to 0. This indicates EMOTE identifies the power dynamic change from the button (a non-visual feature).

5 Discussion and Constraints

In practise EMOTE can be applied to multiagent applications where analogousness holds e.g., football-playing agents where goal-scoring is analogous for the two competing sides/agents, or cleaning by two cooperating heterogeneous agents - one who vacuums while the other mops. We acknowledge limitations in situations with (1) Mismatched Complexities: If the Independent Agent (IA) has more complex/diverse goals than the learning agent (LA), the model of the IA will be rudimentary and uninformative, due to mapping back to the LA, failing to capture IA's true complexity, and (2) Unparalleled objectives: When there is little to no similarity in goals: e.g, if LA's objective is collecting items, while IA's goal is exploring the environment. Finding analogous features to produce an empathetic state can be difficult.

When applying our method, the hyperparameter $\delta \in [0, 1]$ (Equation 3) balances the two loss terms. As δ approaches 1, s_e becomes similar to original state s_i, causing the inferred rewards of the independent agent to be similar to that of the learning agent. In practice the bottleneck imposed by the second model (Q_{learn}) led to the finding that high δ values were beneficial. In particular, it allowed s_e to reproduce common features such as walls and floors, contributing to better performances. Another practical feature which improved performance was setting δ to 1 until an average error in $Loss_2$ was within a threshold ψ. This preconditioning has $M_{imagine}$ recreate s_i before adapting s_e to only make the necessary changes to mimic the independent agent. This was applied to 3 Agent game ($\psi = 5e - 4$). δ settings in Supplementary.

6 Conclusion

We presented the EMOTE architecture, enabling a learning agent to model the action-value function and rewards of an independent agent under the "analogous" assumption. A key benefit is unique and consistent reward inferences, owing to the learning agent's own action-values being referenced for the modelling. EMOTE was designed to generate interpretable empathetic states, useful for verifying the analogous relationships between features for the two agents. EMOTE was shown to be well suited to multiagent learning algorithms, particularly those utilising composite action-value or reward functions with EMOTE producing more consistent rewards (despite re-configurations of the environment).

References

[Alamdari *et al.*, 2021] Parand Alizadeh Alamdari, Toryn Q Klassen, Rodrigo Toro Icarte, and Sheila A McIlraith. Be considerate: Objectives, side effects, and deciding how to act. *arXiv preprint arXiv:2106.02617*, 2021.

[Boularias *et al.*, 2011] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 182–189. JMLR Workshop and Conference Proceedings, 2011.

[Bussmann *et al.*, 2019] Bart Bussmann, Jacqueline Heinerman, and Joel Lehman. Towards empathic deep q-learning. In Huáscar Espinoza, Han Yu, Xiaowei Huang, Freddy Lecue, Cynthia Chen, José Hernández-Orallo, Seán Ó hÉigeartaigh, and Richard Mallah, editors, *Artificial Intelligence Safety 2019*, CEUR Workshop Proceedings, pages 1–7. CEUR-WS.org, 8 2019.

[Chevalier-Boisvert *et al.*, 2018] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for gymnasium, 2018.

[Foerster *et al.*, 2018] Jakob Foerster, Richard Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *AAMAS 2018: Proceedings of the Seventeenth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2018.

[He *et al.*, 2016] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé, III. Opponent modeling in deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1804–1813, New York, New York, USA, 20–22 Jun 2016. PMLR.

[Hoffman, 1996] Martin L Hoffman. Empathy and moral development. *The annual report of educational psychology in Japan*, 35:157–162, 1996.

[Hu *et al.*, 2020] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pages 4399–4410. PMLR, 2020.

[Klein *et al.*, 2012] Edouard Klein, Matthieu Geist, Bilal Piot, and Olivier Pietquin. Inverse reinforcement learning through structured classification. In *NIPS*, 2012.

[Klein *et al.*, 2013] Edouard Klein, Bilal Piot, Matthieu Geist, and Olivier Pietquin. A cascaded supervised learning approach to inverse reinforcement learning. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 1–16, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Lin *et al.*, 2017] 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.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Moreno *et al.*, 2021] Pol Moreno, Edward Hughes, Kevin R McKee, Bernardo Avila Pires, and Théophane Weber. Neural recursive belief states in multi-agent reinforcement learning. *arXiv preprint arXiv:2102.02274*, 2021.

[Natarajan *et al.*, 2010] 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.

[Ndousse, 2020] Kamal Ndousse. Marlgrid. https://github.com/kandouss/marlgrid, 2020.

[Ng *et al.*, 1999] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

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

[Noothigattu *et al.*, 2019] Ritesh Noothigattu, Djallel Bouneffouf, Nicholas Mattei, Rachita Chandra, Piyush Madan, Kush R Varshney, Murray Campbell, Moninder Singh, and Francesca Rossi. Teaching ai agents ethical values using reinforcement learning and policy orchestration. *IBM Journal of Research and Development*, 63(4/5):2–1, 2019.

[Papoudakis *et al.*, 2021] Georgios Papoudakis, Filippos Christianos, and Stefano Albrecht. Agent modelling under partial observability for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:19210–19222, 2021.

[Puterman, 2014] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Rabinowitz *et al.*, 2018] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4218–4227. PMLR, 10–15 Jul 2018.

[Raileanu *et al.*, 2018] Roberta Raileanu, Emily L. Denton, Arthur D. Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *ICML*, 2018.

[Reddy *et al.*, 2012] 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.

[Senadeera *et al.*, 2022] Manisha Senadeera, Thommen George Karimpanal, Sunil Gupta, and Santu Rana. Sympathy-based reinforcement learning agents. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 1164–1172, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[Shu and Tian, 2018] Tianmin Shu and Yuandong Tian. M^3 rl: Mind-aware multi-agent management reinforcement learning. *arXiv preprint arXiv:1810.00147*, 2018.

[Wen *et al.*, 2019] Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. *arXiv preprint arXiv:1901.09207*, 2019.

[Wu *et al.*, 2023] Haochen Wu, Pedro Sequeira, and David V. Pynadath. Multiagent inverse reinforcement learning via theory of mind reasoning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 708–716, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[Zhao *et al.*, 2022] Jian Zhao, Mingyu Yang, Youpeng Zhao, Xunhan Hu, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. Mcmarl: Parameterizing value function via mixture of categorical distributions for multi-agent reinforcement learning, 2022.

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