Adaptive Learning of Centralized and Decentralized Rewards in Multi-agent Imitation Learning

Yasin M. Yousif[1][0000−0002−5282−7259] and Jörg P. Müller[1][0000−0001−7533−3852]

Technical University of Clausthal, Institute of Computer Science, Germany
https://www.ifi.tu-clausthal.de/

Abstract. Imitation learning aims at teaching agents to perform desired behaviors by observing expert demonstrations. This approach can be generalized to multi-agent environments, where it is possible to achieve a mutually beneficial policies equilibrium through the use of specialized rewards. One such reward structure can be implemented by using centralized rewards for cooperative agents and decentralized rewards for non-cooperative agents. However, in mixed multi-agent environments, that contain both cooperative and competitive agents, it is necessary to develop a more nuanced approach to reward allocation. In these situations, one part of the reward can be shared among all cooperative agents while another part targets each agent individually. To address this challenge, our work proposes a novel two-component reward model for each agent: a centralized shared component and a decentralized agent-specific component. We conducted several experiments using three different environments to evaluate the performance of our proposed model compared to its individual components. Our results showed that the combined model outperform all of its constituent models in mixed environments, effectively imitating the centralized reward for cooperative environments but exhibiting no improvement in competitive environments. Finally, we tested the transparency of our model by providing representative examples and examining the scalar weights assigned to the centralized and decentralized components within the combined model. [1]

Keywords: Multi-agent Environment · Inverse Reinforcement Learning · Imitation Learning

1 Introduction

Imitating a given behavior of an expert to automate complex tasks is a promising alternative to encoding explicit rules for performing these tasks, such as automated driving [3], modeling the motion of crowds in realistic simulations [15], or teaching a robot arm to move like a human [4]. The goal of imitation learning is to learn to imitate an expert policy as closely as possible, using only a set of trajectories generated by that policy. Inverse reinforcement learning, a subfield of

[1] code available at: https://github.com/engyasin/Adaptive_learning_4_MAIL

imitation learning, has the additional advantage of recovering the reward signal of that expert policy [6].

For many real-world applications, the models of active entities (agents) being learned through imitation are part of a Multi-Agent System (MAS); hence, behavior learning cannot be approached at the level of individual agents but must consider dynamic interactions in a multi-agent decision-making context. This motivates previous works [21,12,19] towards developing multi-agent models based on an equilibrium that should be reached in the optimal case, such as Nash equilibria among the policies of different agents. For cooperative environments, this equilibrium leads to a centralized reward structure among all agents [19], while for non-cooperative environments it results in a decentralized reward structure with independent rewards for each agent [21].

On one hand, the centralized reward is suitable when all agents care about a common objective to be fulfilled, such as players on a football team who are rewarded whenever any member scores a goal. On the other hand, decentralized rewards do not make assumptions about cooperation or competition but can also ignore these features of the behavior.

As a real world example, in traffic modeling, the behaviour of an agent (traffic participant) in a shared space [10] depends on the actions of other agents. So if a group of pedestrians are moving together, there are two forces, among others, that affect the behaviour: a force to give way to the other group members (cooperation force), and a force to take the shortest path to destination (competition force). This type of environments, which involves both cooperative and competitive behaviours, is known as mixed environments [17].

The most relevant previous works in the field of multi-agent imitation learning include Multi-Agent Generative Adversarial Imitation Learning (MAGAIL) [19], a multi-agent variant of Generative Adversarial Imitation Learning (GAIL) [8]. GAIL has been shown to outperform previous methods in imitation learning [8], such as behavior cloning [2], maximum entropy [22] and apprenticeship learning [1]. MAGAIL proposed three structures of the reward: centralized for cooperative games, decentralized for mixed games, and zero-sum rewards for competitive games. Later, Multi-Agent Inverse Reinforcement Learning (MAIRL) [21], which is based on Adversarial Inverse Reinforcement Learning (AIRL) [6] showed better results than MAGAIL. The improvement in MAIRL is due to the usage of a new equilibrium, Logistic Stochastic Best Response Equilibrium (LSBRE), which did not assume optimality of the expert trajectories.

However, neither MAGAIL nor MAIRL propose a customized reward structure for mixed environments, where a global and local rewards exist for every agent. Additionally, the need for such model is bigger in real world datasets where the true reward structure is unknown or hard to estimate, e.g., in navigation of autonomous vehicles or for pedestrian trajectory prediction [11]. This motivated the contribution of this paper: a novel reward model based on MAIRL [21], combining centralized with decentralized rewards.

Our main contributions of this work include:

- Improved performance for mixed environments using a combined reward model that outperforms decentralized and centralized models when imitating expert behavior.
- Enhanced transparency of the deep learning model by learning separate centralized and decentralized rewards, then combining them, which provides meaningful values for understanding behaviors in multi-agent environments.

2 Related Works

In this section, we survey state-of-the-art imitation learning methods for both single agents (Subsection 2.1) and multi-agent systems (Subsection 2.2).

2.1 Single-Agent Imitation Learning

In imitation learning, the goal is to efficiently learn a desired behavior for a specific task from an expert performing that task [16]. Behavior Cloning (BC) is its simplest form and relies on supervised learning from a dataset. However, BC faces challenges such as requiring large amounts of training data and drifting due to accumulated errors over multiple steps [3].

Another approach in imitation learning is inverse reinforcement learning (IRL), which learns the reward model from the dataset. IRL methods impose restrictions on the shape of the learned reward function, limiting their ability to learn complex rewards [22,1].

Recent imitation learning techniques include Generative Adversarial Imitation Learning (GAIL) [8], which learns both the policy and reward models iteratively. GAIL avoids issues in other IRL methods by matching state-action distributions directly between expert and trained policies, using a discriminator to classify expert pairs of states and actions from non-expert ones [8].

2.2 Multi-Agent Imitation Learning

MAGAIL [19] was based on the concept of Nash equilibrium, which assumes rational behavior from agents. MAGAIL proposed different structures for reward based on game types (cooperative, mixed, and competitive). For cooperative games, centralized rewards are shared among all cooperating agents; but in mixed and competitive games, decentralized rewards are more efficient [19].

In contrast, MAIRL [21] does not use Nash equilibrium, thus removing the condition of expert rationality is removed, so another equilibrium, namely the LSBRE, is used. However, unlike MAGAIL, this method did not test distinct reward structures targeted towards cooperative, competitive, or mixed games.

3 Methodology

3.1 Preliminaries

Markov Games The multi-agent system formulation in this work is based on Markov Games [13], which is an extension of Markov Decision Process (MDP)

[9] for MAS. A Markov game is defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathbf{r}, \eta)$, where $\mathcal{S}$ is the set of states and $\mathcal{A}$ is the set of each agent set of actions $(\mathcal{A}_1, \ldots, \mathcal{A}_k, \ldots, \mathcal{A}_N)$ for N interacting agents, where each agent i has $\mathcal{A}_i$ as its actions set. $\mathcal{P}$ is the set of transition probabilities for all the agents. Let further be η the set of initial state distribution probabilities, γ be the discount factor for the reward, and $\mathbf{r}$ the reward vector for the agents. The subscript of i for any value, for instance the reward, r_i, indicates the relevance to agent i, but a subscript of $-i$ denotes the value vector of all the agents except i.

The goal of Markov games is to find an optimal set of policies for each agent $\{\pi_1, ..., \pi_N\}$ in order to maximise its return, where $\pi_i(a|s)$ is the probability of taking an action a in a state s for an agent i, as a stochastic policy. The joint policy is defined as the product of all policies' probabilities of all the agents:

$$\boldsymbol{\pi}(a|s) = \prod_{i=1}^{N} \pi_i(a_i|s) \tag{1}$$

GAIL Inspired by Generative Adversarial Network (GAN) [7] in the supervised learning literature, the GAIL method was proposed [8] for imitation learning. It uses an iterative training strategy to train the reward model and the policy model adversely on the expert trajectories dataset at each training step. Equation 2 defines the training objective in GAIL:

$$\min_{\theta} \max_{\omega} \boldsymbol{E}_{\pi_E}[\log D_\omega(s,a)] + \boldsymbol{E}_{\pi_\theta}[\log(1 - D_\omega(s,a))] \tag{2}$$

Here, D_w is a discriminator that classifies expert from policy (state-action) pairs, and π_θ is the parameterized policy that aims at maximizing its score under D_w. θ and ω are the policy and the discriminator parameters, respectively.

AIRL Despite the improvement done in GAIL, it doesn't recover ground truth reward from the expert policy like what it is done in IRL methods. So AIRL [6] was introduced, which recover a shaped reward in addition to the policy. Its main difference from GAIL is in the definition of the discriminator network D_w, which is set to:

$$D_\omega(s,a) = \frac{\exp f_\omega(s,a)}{\exp f_\omega(s,a) + \pi(a|s)} \tag{3}$$

Where $f_\omega(s,a)$ is a model that represents the trained advantage function, and it is trained using the same objective as in GAIL's discriminator. The policy objective is similar to equation 2

MAGAIL To apply GAIL to the multi-agent case, MAGAIL defines three custom objectives, based on the centralized and decentralized discriminators shown in Fig.1 (a,b) and the zero-sum case. The training objective function for the centralized reward is defined as:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\omega_T}} \boldsymbol{E}_{\boldsymbol{\pi_E}}[\log D_{\omega_T}(\boldsymbol{s},\boldsymbol{a})] + \boldsymbol{E}_{\boldsymbol{\pi_\theta}}[\log(1 - D_{\omega_T}(\boldsymbol{s},\boldsymbol{a}))] \tag{4}$$

Here, $\boldsymbol{\pi_\theta}$ represents the agents' joint policy and the $\boldsymbol{\pi_E}$ is the expert joint policy. $\boldsymbol{s}, \boldsymbol{a}$ are the joint states and actions, respectively. $\boldsymbol{\theta}$ are the agents' joint

policy parameters and $\boldsymbol{\omega_T}$ is the centralized discriminator parameters. The decentralized case of objective function is defined as follows:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\omega}} \boldsymbol{E}_{\boldsymbol{\pi_E}}[\sum_{i=0}^{N} \log(D_{\omega_i}(s_i, a_i))] + \boldsymbol{E}_{\boldsymbol{\pi_\theta}}[\sum_{i=0}^{N} \log(1 - D_{\omega_i}(s_i, a_i))] \tag{5}$$

Here, D_{ω_i} is the discriminator model for agent i, and ω_i is its parameters, but if it is shared among all agents, we can simply use ω for the shared set of parameters. Lastly the competitive case has the objective function:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\omega_c}} \boldsymbol{E}_{\boldsymbol{\pi_E}, \pi_\theta}[\sum_{i=0}^{N} \log(D_{\omega_{ci}}(s_{iE}, a_{iE}, \boldsymbol{s}_{-i}, \boldsymbol{a}_{-i}))] + \\ \boldsymbol{E}_{\boldsymbol{\pi_E}, \pi_\theta}[\sum_{i=0}^{N} \log(1 - D_{\omega_{ci}}(s_i, a_i, \boldsymbol{s}_{-iE}, \boldsymbol{a}_{-iE}))] \end{aligned} \tag{6}$$

Where ω_{ci} is the discriminator parameters for the agent i. The expert part should represent an entire competing team of a two-teams competition games. Defining the competitive case as in equation 6, would require access to the expert policy in order to respond to new states induced by following the generator policy for the agent under training $D_{\omega_{ci}}(s_i, a_i, \boldsymbol{s}_{-iE}, \boldsymbol{a}_{-iE})$. Therefore this cannot be applied for real-world datasets, for which the expert policy is unknown.

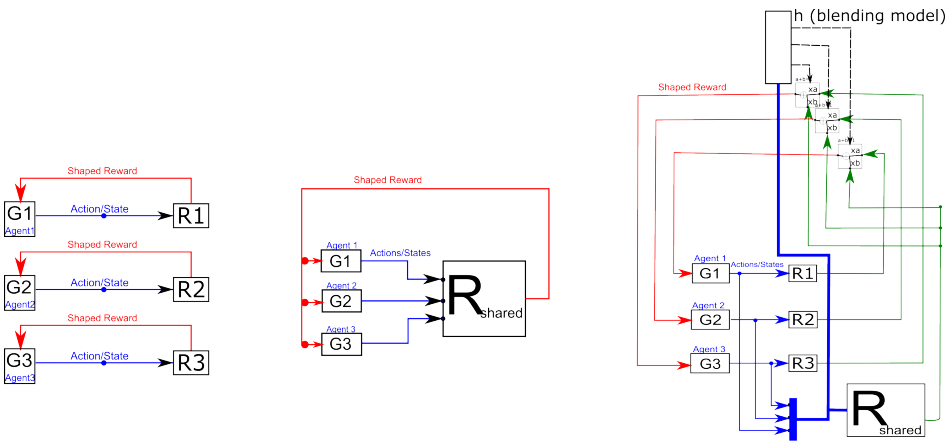
MAIRL [21] is considered a multi-agent extension of AIRL, and it introduces a new equilibrium (LSBRE). The final objectives is similar to MAGAIL in equation 5, but with different discriminator model as in equation 3. The advantage of using MAIRL is that the condition of expert optimality is removed due to the new equilibrium. As MAIRL showed better imitation results than MAGAIL, therefore it is used in the implementation in this work.

3.2 Proposed Formulation

MAIRL, as an Inverse Reinforcement Learning (IRL) method, has the advantage of recovering a shaped reward from the trajectories of the expert. However, in the case of fully cooperative agents, we can assume a shared reward returned by a shared discriminator, D_{ω_T} as in equation 4.

The decentralized rewards, in equation 5, would be suitable for other cases of non-cooperative environments. However, if the behaviour in the environment includes cooperation as well as competition, as in mixed environments, then a form of a combination between the two structures maybe optimal. In this work, a new model to find this combination coefficients for a linear formula of the two rewards, is introduced. This model predicts two numbers (α_i,β_i), for every agent i. So, if we have $R_{cooperative}$ and R_{i_single} as the learned rewards for centralized and decentralized cases respectively for agent i among N agents, then the model learns the ground truth reward based on the following formula R_{i_gt}:

$$R_{i\_gt} = \alpha_i \times R_{i\_single} + \beta_i \times R_{cooperative} \tag{7}$$



a) Decentralized Reward Model

b) Centralized Reward Model (Cooperative games)

c) Adaptive Blending of Centralized and Decentralized Rewards Model (Unknown games)

Fig. 1: The decentralized (a), centralized (b) and blended reward of both (c) models are shown respectively. Three agents are shown, where G is the generator/policy model and R is the reward model

For every $i \in \{0, ..., N\}$. Furthermore, the condition: $\alpha_i + \beta_i = 1$ is used to make the interval of R_{gt} values match that of R_{coop} and R_{single}. Additionally, these coefficients (α_i, β_i) are found by training on the same input as the reward model, i.e (state,action) pairs, and the function that calculates the coefficients is denoted by h_ρ where ρ is the function parameters' set:

$$[\alpha_1, ..., \alpha_N, \beta_1, ..., \beta_N] = h_\rho(\mathbf{s}, \mathbf{a}) \tag{8}$$

This will make the combination of the rewards a function for the current states and actions, which in turn will define the cooperative and non-cooperative signals balance. h_ρ is a linear neural layer in our implementation.

Algorithm 1 shows the steps for training the model. The corresponding graph is shown in Fig 1 part (c). The model of the function h_ρ will be trained with the objective of the single discriminator D_{ω_i}, i.e classifying expert from generated (state,action) pairs, where the new shaped reward $f_\omega(s, a)$ is trained. The policy will be trained then with the objective in equation 5.

Lastly, in this work, as in MAIRL [21], f_ω in equation 3 is estimating the advantage function; it is broken down to an estimated value function and a true reward function. However, the input reward signals to the combining function

h_ρ, are coming from these advantage functions f_ω, which represent the learned shaped reward in this work.

Algorithm 1 Multi-agent imitation learning with blended reward of centralized and decentralised components

Input: Shared policy initial parameters θ_i,
Decentralized shaped reward initial parameters ω_i,
Centralized shaped reward initial parameters ω_T,
Combining model initial parameters ρ,
Expert trajectories dataset $\mathcal{D} = \{(s_j, a_j)\}_{j=0}^{M}$
Output: Trained parameters of θ_i, ω_i, ω_T, ρ for the function h
for *epoch* in *epochs_num* **do**
    Train centralized shaped reward on the global states-actions pairs $(\mathbf{s}, \mathbf{a})$ according to binary loss of classification, where D_{ω_T} is given in Equation 3
    for $agent_i$ in *all_agents* **do**
        Train the decentralized shaped rewards D_{ω_i} according to Equation 5
    end for
    Train the linear combining function h defined in Equation 8 according to the binary loss of classification
    Train the shared policy model $\pi_i$with the final combined reward using Policy Proximal Optimization (PPO) [18]
end for

4 Evaluation

To test our combined reward structure, three experiments are conducted in three environments: cooperative with occasional competition situations, adversarial mixed environment (from [14]), and zero-sum football (from [5]).

We started each experiment with the same initialization of policy model, expert trajectories dataset, and reward model. We used a warm start method based on behavior cloning to initialize the policy model by training decentralized, centralized, and combined reward models. The evaluation criteria are euclidean distance between learned policy trajectories and expert trajectories, allowing a direct comparison without assumptions of expert optimality.

We measured distance using two metrics: average displacement error (ADE) for calculating distances between corresponding points of the same timesteps, and final displacement error (FDE) measuring the distance between last two points. Additionally, we analyzed the combined model coefficients in selected situations to understand the learned behavior.

4.1 Cooperative Spread Particles Environment

This environment consists of three cooperative agents, each trying to reach one of three landmarks without colliding. The goal is for all agents to distribute themselves evenly over the landmarks in the shortest possible paths while avoiding

collision. An expert rule-based model uniformly distributes agents on landmarks and avoids collision, as depicted in Fig. 3a.

4.2 Mixed Adversary Particles Environment

In this environment, three cooperative agents try to prevent a single competing agent from reaching a specific landmark among others, shown in Fig. 3b. At least one of the cooperating agents must reach their designated landmark while blocking the competing agent's progress. The expert consists of a rule-based model for cooperating agents and a machine learning model for the competing agent, trained on ground truth rewards. State definitions, actions, and reward functions are in line with [20], which is commonly used to evaluate multi-agent reinforcement learning methods like MAGAIL.

4.3 Zero-Sum Mini-Grid Football Environment

This environment represents a football game between two teams with 2 agents each, and involves passing the ball and receiving negative rewards for goals scored against one's team, as shown in Fig 4b. Each agent has a field of view of size 5×5 grid cells. The expert modelis trained with PPO [18] using ground truth rewards after 35 million training iterations, of football gameplay.

5 Results and Discussion

After an initial training phase using behaviour cloning as a warm-start for the AIRL generator model, the same initial set of model's parameters is used for three training rounds for the centralized, decentralized and blended rewards models. Three tests were run on each of the environments with different expert dataset sizes, and their results are shown in the following subsections.

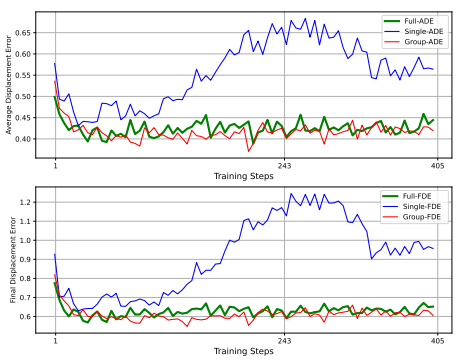
5.1 Cooperative Spread Particles Environment

Fig. 2a shows the ADE and FDE on the y-axis with the training steps on the x-axis where each full model step consists of 2000 training steps of the generator model, and 16 training steps of the shaped reward model.

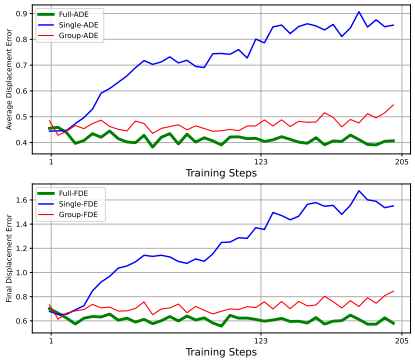
It's noted that the cooperative reward model perform better than the decentralized reward. The full combined reward model (in green) follows closely the centralized model (in red), and both models improve on the baseline of behaviour cloning, whose value is around 0.5 of ADE, as shown in Table 1.

When checking the learned reward for this environment, it rarely deviated from imitating the centralized reward, i.e always zero weights for the decentralized reward, however for some cases, it showed some interesting patterns. For example in Fig 3a, the decentralized weight is negative for agent A_3, where it is the only agent that follows its nearest available landmark to maximize the centralized reward, therefore the decentralized weight is minimized.

The learned rewards values are very small, but not exactly zeros, as in Fig 3a only two digits after comma are shown.



(a) ADE and FDE for the cooperative environment (dataset size = 10)



(b) ADE and FDE for the mixed environment (dataset size = 10)

Fig. 2: ADE and FDE for cooperative (a) and mixed (b) environments. Single ADE is decentralised, group ADE is centralized, and full ADE is the new model

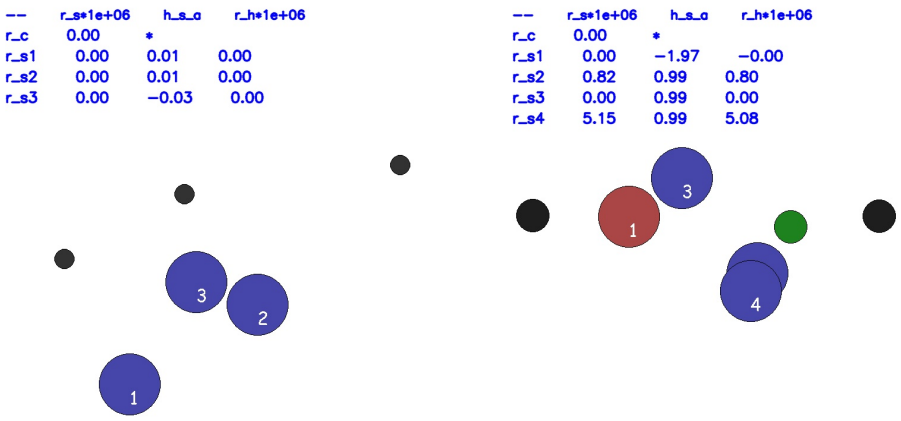
5.2 Mixed Adversary Particles Environment

For the mixed environment, it is shown from the combined reward performance (in green), that it clearly outperform the other rewards in Fig 2b, as well as the behaviour cloning baseline.

For the learned rewards and combining model weights, the following snapshot in Fig 3b shows that agents A_4, A_2 are maximizing their own reward by approaching the green dot, so they have higher decentralized reward than the rest. They also have a high decentralized weights of 0.99. Agent A_1 has a negative decentralized weight, which shows that it's affected by the centralized reward from all the agents, i.e. it follows the other agents, similar to the expert demons.

Table 1: ADE and FDE average values for the last 30% of the training steps and for three different sizes of datasets (10,80,60) on the two particles environments and (10,20,40) on the football environment. Behavior cloning and expert models are shown as well.

Model	**Cooperative**		**Adversary**		**Zero-Sum**	
	ADE	FDE	ADE	FDE	ADE	FDE
Expert Model	0.0	0.0	0.0	0.0	0.05	0.13
Behaviour Cloning	0.46	0.71	**0.43**	0.66	**1.23**	**2.71**
Centralized	0.49	**0.67**	0.48	0.74	2.0	2.80
Decentralized	0.56	0.96	0.60	0.99	1.97	2.80
Combined Reward	**0.44**	**0.67**	**0.43**	**0.65**	2.17	3.08



(a) A snapshot for the cooperative environment, with the learned decentralized (r_s), centralized (r_c), combined weights of the decentralized part (h_{s_a}), and final blended reward (r_h)

(b) A snapshot for the mixed environment, with the learned decentralized (r_s), centralized (r_c), combined weights of the decentralized part (h_{s_a}), and final blended reward (r_h)

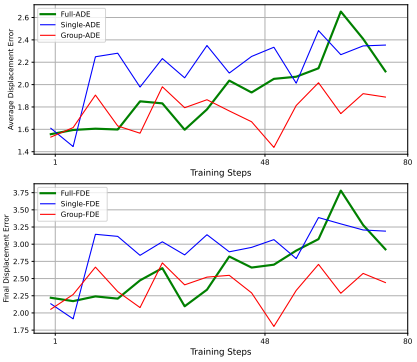
Fig. 3: Snapshots of both cooperative (a) and mixed (b) environments

5.3 Zero-Sum Mini-Grid Football Environment

In this environment, all reward models performed worse than the behavior cloning baseline due to its pure competitive nature (see Fig. 4a). For a dataset size of 10 (i.e., 10 trajectories), centralized rewards had lower errors; however, decentralized rewards had lower errors for larger datasets. Unlike mixed environments, combining weights were either all zeros or all ones in this case, imitating the centralized part or the decentralized part fully (shown in Fig 4b as h_{s_a}). This can be explained because of the relative complexity of this football environment compared to the plain particles environment. We can see that all inverse reinforcement learning methods did perform worse than the baseline. Given only a limited expert dataset trajectories, it is hard for these methods to generalize well. Further investigating of these cases is a possible direction of a future work.

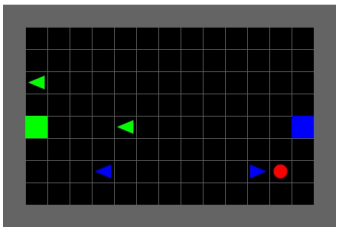
Table 1 shows the average errors for all the experiments done in this work, on the last 30% part of the training. In summary, the proposed structure showed better performance in the case of mixed adversary environment, with respect to the error criteria of the euclidean distances.

The advantage of using the combined reward model for transparency was examined by testing on some states, and showing how it blends both of the centralized and decentralized rewards for every state. The combined model also



(a) ADE and FDE for the zero-sum environment (dataset size = 10)

--	r_s*1e+04	h_s_a	r_h*1e+04
r_c	0.00	*	
r_s1	0.01	1.00	0.01
r_s2	0.02	1.00	0.02
r_s3	5.28	−3.00	−15.83
r_s4	1.29	1.00	1.29



(b) A snapshot for the zero-sum environment, with the learned decentralized (r_s), centralized (r_c), combined weights of the decentralized part (h_{s_a}), and final blended reward (r_h)

Fig. 4: ADE/FDE errors (a), and a snapshot (b) of the zero-sum environment

showed an ability to learn from the best model in the cooperative environment, but without major improvements.

6 Conclusion and Future Work

In this work, we proposed a novel Multi-Agent Imitation Learning (MAIL) reward model combining multi-agent centralized and single-agent decentralized models trained using MAIRL. Our evaluation in three environments showed superior performance compared to individual components. The enhanced transparency of our model provides meaningful values for understanding behaviors in multi-agent scenarios, influenced by the current state-action pair. Future work includes extending and evaluating our method on real traffic datasets where ground truth reward information isn't available, and investigating zero-sum game cases weak performance through testing with suitable reward structures alongside centralized and decentralized rewards.

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning. p. 1 (2004)
2. Bain, M., Sammut, C.: A framework for behavioural cloning. In: Machine Intelligence 15. pp. 103–129 (1995)
3. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018)

4. Fang, B., Jia, S., Guo, D., Xu, M., Wen, S., Sun, F.: Survey of imitation learning for robotic manipulation. International Journal of Intelligent Robotics and Applications **3**(4), 362–369 (2019)
5. Fickinger, A.: Multi-agent gridworld environment for openai gym. `https://github.com/ArnaudFickinger/gym-multigrid` (2020)
6. Fu, J., Luo, K., Levine, S.: Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248 (2017)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
8. Ho, J., Ermon, S.: Generative adversarial imitation learning. Advances in neural information processing systems **29** (2016)
9. Howard, R.A.: Dynamic programming and markov processes. John Wiley (1960)
10. Johora, F.T., Müller, J.P.: Modeling interactions of multimodal road users in shared spaces. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 3568–3574. IEEE (2018)
11. Johora, F.T., Yang, D., Müller, J.P., özgüner, ü.: On the generalizability of motion models for road users in heterogeneous shared traffic spaces. IEEE Transactions on Intelligent Transportation Systems **23**(12), 23084–23098 (2022). https://doi.org/10.1109/TITS.2022.3192138
12. Le, H.M., Yue, Y., Carr, P., Lucey, P.: Coordinated multi-agent imitation learning. In: International Conference on Machine Learning. pp. 1995–2003. PMLR (2017)
13. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Machine learning proceedings 1994, pp. 157–163. Elsevier (1994)
14. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems **30** (2017)
15. Martinez-Gil, F., Lozano, M., García-Fernández, I., Romero, P., Serra, D., Sebastián, R.: Using inverse reinforcement learning with real trajectories to get more trustworthy pedestrian simulations. Mathematics **8**(9), 1479 (2020)
16. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J.: An algorithmic perspective on imitation learning. Foundations and Trends® in Robotics **7**(1-2), 1–179 (2018). https://doi.org/10.1561/2300000053, `http://dx.doi.org/10.1561/2300000053`
17. Plaat, A.: Multi-Agent Reinforcement Learning, pp. 219–262. Springer Nature Singapore, Singapore (2022)
18. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
19. Song, J., Ren, H., Sadigh, D., Ermon, S.: Multi-agent generative adversarial imitation learning. Advances in neural information processing systems **31** (2018)
20. Terry, J.K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sulivan, R., Santos, L., Perez, R., Horsch, C., Dieffendahl, C., Williams, N.L., Lokesh, Y., Sullivan, R., Ravi, P.: Pettingzoo: Gym for multi-agent reinforcement learning. arXiv preprint arXiv:2009.14471 (2020)
21. Yu, L., Song, J., Ermon, S.: Multi-agent adversarial inverse reinforcement learning. In: International Conference on Machine Learning. pp. 7194–7201. PMLR (2019)
22. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: Aaai. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008)