The Emergence of Individuality in Multi-Agent Reinforcement Learning

Jiechuan Jiang
Peking University
jiechuan.jiang@pku.edu.cn

Zongqing Lu
Peking University
zongqing.lu@pku.edu.cn

Abstract

Individuality is essential in human society, which induces the division of labor and thus improves the efficiency and productivity. Similarly, it should also be the key to multi-agent cooperation. Inspired by that individuality is of being an individual separate from others, we propose a simple yet efficient method for the emergence of individuality (EOI) in multi-agent reinforcement learning (MARL). EOI learns a probabilistic classifier that predicts a probability distribution over agents given their observation and gives each agent an intrinsic reward of being correctly predicted by the classifier. The intrinsic reward encourages the agents to visit their own familiar observations, and learning the classifier by such observations makes the intrinsic reward signals stronger and the agents more identifiable. To further enhance the intrinsic reward and promote the emergence of individuality, two regularizers are proposed to increase the discriminability of the classifier. We implement EOI on top of popular MARL algorithms. Empirically, we show that EOI significantly outperforms existing methods in a variety of multi-agent cooperative scenarios.

1 Introduction

Humans develop into distinct individuals due to both genes and environments [1]. Individuality induces the division of labor [2], which improves the productivity and efficiency of human society. Analogically, the emergence of individuality should also be essential for multi-agent cooperation.

Although multi-agent reinforcement learning (MARL) has been applied to multi-agent cooperation, it is widely observed that agents usually learn similar behaviors, especially when the agents are homogeneous with shared global reward and co-trained [3]. For example, in multi-camera multi-object tracking [4], where camera agents learn to cooperatively track multiple objects, the camera agents all tend to track the easy object for the cheap reward. However, such similar behaviors can easily make the learned policies fall into local optimum. If the agents can respectively track different objects, they are more likely to solve the task optimally. Many studies formulate such a problem as task allocation or role assignment [5, 6, 7, 8, 9]. However, they require that the agent roles are rule-based and the tasks are pre-defined, and thus are not general methods. Some studies intentionally pursue difference in agent policies by diversity [10, 11] or by emergent roles [12], however, the induced difference is not appropriately linked to the success of task. We argue that the key to solve this problem is individuality. The emergence of individuality can automatically drive agents to behave differently and take a variety of roles to complete tasks successfully.

Biologically, the emergence of individuality is attributed to innate characteristics and experiences. However, as in practice RL agents are mostly homogeneous, we mainly focus on enabling agents to develop individuality through interactions with the environment during policy learning. Intuitively, in multi-agent environments where agents respectively explore and interact with the environment, individuality should emerge from what they experience. In this paper, we propose a novel method for the emergence of individuality (EOI) in MARL. EOI learns a probabilistic classifier that predicts a

Preprint. Under review.

probability distribution over agents given their observation and gives each agent an intrinsic reward of being correctly predicted probability by the classifier. Encouraged by the intrinsic reward, agents tend to visit their own familiar observations. Learning the probabilistic classifier by such observations makes the intrinsic reward signals stronger and in turn makes the agents more identifiable. In this closed loop with positive feedback, agent individuality emerges gradually. However, at early learning stage, the observations visited by different agents cannot be easily distinguished by the classifier, meaning the intrinsic reward signals are not strong enough to induce agent characteristics. Therefore, we propose two regularizers for learning the classifier to increase the discriminability, enhance the feedback, and thus promote the emergence of individuality.

EOI is compatible with centralized training and decentralized execution. We realize EOI on top of two popular MARL methods, MAAC [13] and QMIX [14]. For MAAC, as each agent has its own critic, it is convenient to shape the reward for each agent. For QMIX, we introduce an auxiliary gradient and update the individual value function by both minimizing the TD error of the joint action-value function and maximizing its cumulative intrinsic rewards. We evaluate EOI in three scenarios, *i.e.*, Pac-Men, Windy Maze, and Firefighters. It is empirically demonstrated that EOI significantly outperforms existing methods. By ablation studies, we confirm that the proposed regularizers indeed improve the emergence of individuality even if agents have the same innate characteristics.

2 Related Work

MARL. We consider the formulation of Decentralized Partially Observable Markov Decision Process (Dec-POMDP), where at each timestep t each agent i receives a local observation o_i^t, takes an action a_i^t, and gets a shared global reward r^t. Agents together aim to maximize the total expected return $\mathbb{E}\sum_{t=0}^{T}\gamma^t r^t$, where γ is a discount factor and T is the time horizon. Many methods have been proposed for Dec-POMDP, most of which adopt centralized learning and decentralized execution. Some methods [15, 16, 13] extend policy gradient into multi-agent cases. Value function factorization methods [17, 14, 18] decompose the joint value function into individual value functions. Communication methods [19, 20, 21] share information between agents for better cooperation.

Behavior Diversification. Many cooperative multi-agent applications require agents to take different behaviors to complete the task successfully. Behavior diversification can be handcrafted or emerge through agents' learning. Handcrafted diversification is widely studied as task allocation or role assignment. Heuristics [5, 6, 7, 8] assign specific tasks or pre-defined roles to each agent based on goal, capability, visibility, or by search. M^{3}RL [9] learns a manager to assign suitable sub-tasks to rule-based workers with different preferences and skills. These methods require that the sub-tasks and roles are pre-defined, and the worker agents are rule-based. However, in general, the task cannot be easily decomposed even with domain knowledge and workers are learning agents.

The emergent diversification is first studied for single agent to learn reusable diverse skills in complex and transferable tasks. Maximum entropy methods [22, 23, 24] explore many possible ways to achieve a goal by maximizing the entropy of the policy. DIAYN [25] learns diverse skills without any reward signal by maximizing the mutual information between states and skill embeddings as well as entropy. In multi-agent learning, SVO [3] introduces diversity into heterogeneous agents for more generalized and high-performing policies in social dilemmas. Some methods are proposed for behavior diversification in multi-agent cooperation. ROMA [12] learns a role encoder to generate the role embedding, and learns a role decoder to generate the neural network parameters. However, there is no mechanism that guarantees the role decoder can generate different parameters, taking as input different role embeddings. Learning low-level skills for each agent using DIAYN is considered in [10, 11], where agents' diverse low-level skills are coordinated by the high-level policy. However, the independently trained skills limit the cooperation, and diversity is not considered in the high-level policy. Therefore, none of these works appropriately link behavior diversification to the success of cooperative task.

3 Method

Individuality is of being an individual separate from others. Motivated by this, we propose EOI, where agents are intrinsically rewarded in terms of being correctly predicted by a probabilistic classifier that is learned based on agents' observations. If the classifier learns to accurately distinguish agents,

agents should behave distinctly and thus individuality emerges. Two regularizers are introduced for learning the classifier to enhance intrinsic reward signal and promote individuality. Unlike existing work on behavior diversification, EOI directly correlates individuality with the task by intrinsic reward, and thus individuality emerges naturally during agents' learning. EOI can be applied to Dec-POMDP tasks and trained along with centralized training and decentralized execution algorithms. We design practical techniques to implement EOI on top of two popular MARL methods, MAAC and QMIX. We also propose an unbiased gradient estimation for learning from other agents' experiences to accelerate learning. This appears in Appendix due to the page limit.

3.1 Intrinsic Reward

As illustrated in Figure 1, two agents learn a task of collecting two items, where item b is the easier one. Collecting each item, they will get a global reward of $+1$. Sequentially collecting the two items by one agent or together is sub-optimal, sometimes impossible when the task has a limited horizon. The optimal solution is that the two agents go collecting different items simultaneously. It is easy for both agents to learn to collect item b due to the cheap reward. However, after that, item a becomes even harder to be explored, and the learned policies easily fall at local optimum. Nevertheless, the emergence of individuality can easily address the problem, *e.g.*, agents prefer to collect different items.

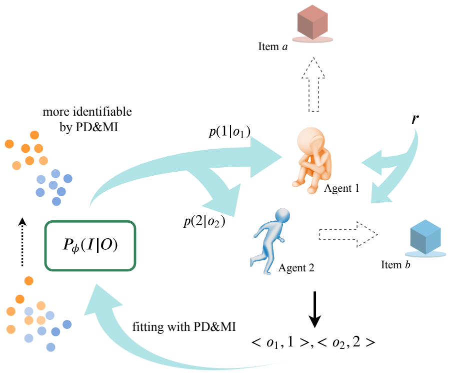


Figure 1: EOI

To enable agents to develop individuality, EOI learns a probabilistic classifier $P(I|O)$ to predict a probability distribution over agents given on their observation, and each agent takes the correctly predicted probability as the intrinsic reward at each timestep. Thus, the reward function for agent i is modified as

$$r + \alpha p(i|o_i),$$

where r is the global environmental reward, $p(i|o_i)$ is the predicted probability of agent i given its observation o_i, and α is a tuning parameter to weight the intrinsic reward. With the reward shaping, EOI works as follows. If there is initial difference between agent policies in terms of visited observations, the difference is captured by $P(I|O)$ as it is fitted using agents' experiences. The difference is then fed back to each agent as an intrinsic reward. As agents maximize the expected return, the difference in agents' policies is exacerbated together with optimizing the environmental return. Therefore, the learning process is a closed loop with positive feedback. As agents progressively behave more identifiably, the classifier can distinguish agents more accurately, and thus individuality emerges gradually.

The classifier $P_\phi(I|O)$ is parameterized by a neural network ϕ and learned in a supervised way. At each timestep, we take each agent i's observation o_i as input and the agent index i as the label and store the pair $< o_i, i >$ into a buffer $\mathcal{B}$. ϕ is updated by minimizing the cross-entropy loss (CE), which is computed based on the uniformly sampled batches from $\mathcal{B}$.

3.2 Regularizers of $P_\phi(I|O)$

In the previous section, we assume that there is some difference between agents' policies. However, in general, the difference between initial policies is small (even no differences if agents' policies are initially by the same network weights), and the policies will quickly learn similar behaviors due to the cheap reward. Therefore, the intrinsic rewards are nearly the same for each agent, which means no feedback in the closed loop. To generate the feedback in the closed loop, the observation needs to be identifiable and thus the agent can be distinguished in terms of observations by $P_\phi(I|O)$. To address this, we propose two regularizers: positive distance (PD) and mutual information (MI) for learning $P_\phi(I|O)$.

Positive Distance. The positive distance is inspired from the triplet loss [26] in face recognition, which is proposed to learn identifiable embeddings. Since o_i^t and its previous observations $\{o_i^{t-\Delta t}, o_i^{t-\Delta t+1}, \cdots, o_i^{t-1}\}$ are distributed on the trajectory generated by agent i, the previous observations in the Δt-length window could be seen as the positives of o_i^t. To make the probability distribution on the anchor o_i^t close to that on the positives, we sample an observation $o_i^{t'}$ from $\{o_i^{t-\Delta t}, o_i^{t-\Delta t+1}, \cdots, o_i^{t-1}\}$ and minimize the cross-entropy loss

$$\mathrm{CE}\left(p_\phi(\cdot|o_i^t), p(\cdot|o_i^{t'})\right).$$

The positive distance minimizes the intra-distance between the observations with the same "identity", which hence enlarges the margin between different "identities". As a result, the observations become more identifiable. Since the positives are naturally defined on the trajectory, the identifiability generated by the positive distance is actually induced by the agent policy. Defining the negatives is hard but we find that just using the positive distance works well in practice.

Mutual Information. If the observations are more identifiable, it is easier to infer the agent that visits the given observation most, which indicates the higher mutual information between the agent index and observation. Therefore, to further increase the discriminability of the classifier, we maximize their mutual information,

$$\begin{aligned}\mathrm{MI}(I;O) &= \mathcal{H}(I) - \mathcal{H}(I|O) \\ &= \mathcal{H}(I) - \mathbb{E}_{o\sim p(o)}\left[\sum_i -p(i|o)\log p(i|o)\right].\end{aligned}$$

Since we store $< o_i, i >$ of every agent in $\mathcal{B}$, the number of samples for each agent is equal. Fitting $P_\phi(I|O)$ using batches from $\mathcal{B}$ ensures $\mathcal{H}(I)$ is a constant. To maximize $\mathrm{MI}(I;O)$ is to minimize $\mathcal{H}(I|O)$. Therefore, equivalently, we sample batches from $\mathcal{B}$ and minimize

$$\mathrm{CE}\left(p_\phi(\cdot|o_i^t), p_\phi(\cdot|o_i^t)\right).$$

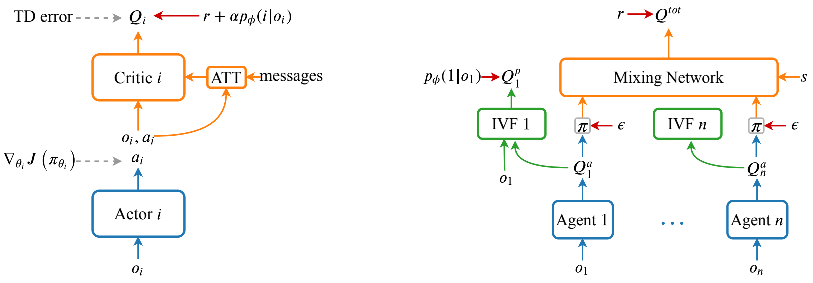
Therefore, the optimization objective of $P_\phi(I|O)$ is to minimize

$$\mathrm{CE}\left(p_\phi(\cdot|o_i^t), \text{one\_hot}(i)\right) + \beta_1 \mathrm{CE}\left(p_\phi(\cdot|o_i^t), p(\cdot|o_i^{t'})\right) + \beta_2 \mathrm{CE}\left(p_\phi(\cdot|o_i^t), p_\phi(\cdot|o_i^t)\right),$$

where β_1 and β_2 are hyperparameters. The regularizers increase the discriminability of $P_\phi(I|O)$, make the intrinsic reward signals stronger to stimulate the agents to be more distinguishable, and eventually promote the emergence of individuality. In this sense, $P_\phi(I|O)$ not only is the posterior statistics, but also serves as the inductive bias of agents' learning. The learning process of EOI is illustrated in Figure 1.

3.3 Implementation with MAAC and QMIX

Existing methods for reward shaping in MARL focus on independent learning agents *e.g.*, [3, 27, 28]. How to shape the reward in Dec-POMDP for centralized training has not been deeply studied. Since the intrinsic reward can be exactly assigned to the specific agent, individually maximizing the intrinsic reward is more efficient than jointly maximizing the sum of all agents' intrinsic rewards [29]. Adopting this idea, we respectively present the implementation with MAAC and QMIX for realizing EOI. MAAC is an off-policy actor-critic algorithm, where each agent learns its own critic, thus it is convenient to directly give the shaped reward $r + \alpha p_\phi(i|o_i)$ to the critic of each agent i, without



(a) EOI+MAAC

(b) EOI+QMIX

Figure 2: Illustration of EOI with MAAC and QMIX.

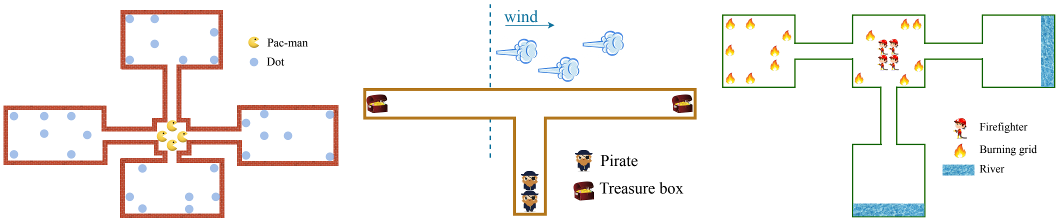


Figure 3: Illustration of experimental scenarios: Pac-men (*left*), Windy Maze (*mid*), Firefighters (*right*).

modifying other components, as illustrated in Figure 2a. The TD error of the critic and the policy gradient are the same as in [13].

In QMIX, each agent i has an individual action-value function Q_i^a. All the individual action-value functions are monotonically mixed into a joint action-value Q^{tot} by a mixing network. Each agent selects the action with the highest individual value, but the individual value has neither actual meaning nor constraints [14]. Therefore, we can safely introduce an auxiliary gradient of the intrinsic reward to the individual action-value function Q_i^a. Each agent i learns an intrinsic value function (IVF) Q_i^p, which takes as input the observation o_i and the individual action-value vector $Q_i^a(o_i)$ and approximates $\mathbb{E}\sum_{t=0}^{T}\gamma^t p(i|o_i^t)$ by minimizing the TD error,

$$\mathbb{E}_{<o_i,Q_i^a(o_i),o_i'>\sim\mathcal{D}}\left[\left(Q_i^p\left(o_i,Q_i^a(o_i)\right)-y\right)^2\right], \text{ where } y=p_\phi(i|o_i)+\gamma\bar{Q}_i^p(o_i',\bar{Q}_i^a(o_i')).$$

$\bar{Q}_i^a$ and $\bar{Q}_i^p$ are the target value functions and $\mathcal{D}$ is the replay buffer. In order to improve both the global reward and intrinsic reward, we update Q_i^a, parameterized by θ_i, towards maximizing $\mathbb{E}\left[Q_i^p\left(o_i,Q_i^a(o_i;\theta_i)\right)\right]$ along with minimizing the TD error of Q^{tot} (denoted as δ^{tot}), as illustrated in Figure 2b. Since the intrinsic value function is differentiable with respect to the individual action-value vector $Q_i^a(o_i;\theta_i)$, we can establish the connection between Q_i^p and Q_i^a by the chain rule, and the gradient of θ_i is,

$$\nabla_{\theta_i}J(\theta_i)=\frac{\partial\delta^{tot}}{\partial\theta_i}-\alpha\frac{\partial Q_i^p(o_i,Q_i^a(o_i;\theta_i))}{\partial Q_i^a(o_i;\theta_i)}\cdot\frac{\partial Q_i^a(o_i;\theta_i)}{\partial\theta_i}.$$

MAAC and QMIX are off-policy algorithms, and the environmental rewards are stored in the replay buffer. However, the intrinsic rewards are recomputed in the sampled batches before each update, since $P_\phi(I|O)$ is co-evolving with the learning agents, the previous intrinsic reward is outdated. The joint learning process of the classifier and agent policies can be mathematically formulated as a bi-level optimization, which is detailed in Appendix.

4 Experiments

For the experiments, we design three scenarios: Pac-Men, Windy Maze, and Firefighters, which are illustrated in Figure 3. In the evaluation, we first verify the effectiveness of EOI and the two regularizers on both MAAC and QMIX by ablation studies. Then, we compare EOI against EDTI [27] to verify the advantages of individuality with sparse reward, and against ROMA [12] and [10, 11] (denoted as HC and HC-simul respectively) to investigate the advantages of individuality over emergent roles and diversity. The details about the experimental settings of each scenario and

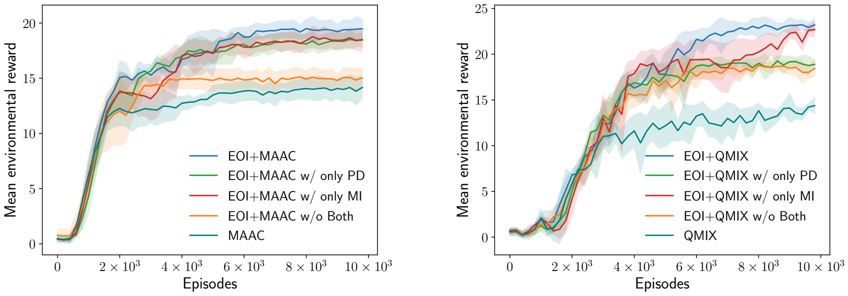


Figure 4: Learning curves in Pac-Men: EOI+MAAC (*left*) and EOI+QMIX (*right*).

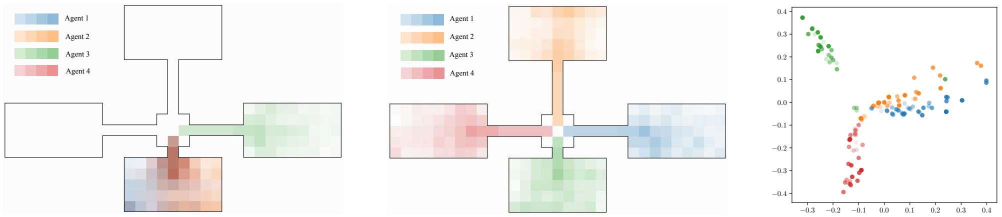


Figure 5: Distributions of agents' positions of QMIX (*left*) and EOI+QMIX (*mid*), and kernel PCA of agents' observations of EOI+QMIX (*right*) in Pac-Men. The darker color means the higher value.

the hyperparameters are available in Appendix. All the experimental results are presented using mean and standard deviation.

4.1 Pac-Men

In the scenario, there are four pac-men (agents) initialized at the maze center, and some dots initialized randomly in the four rooms, illustrated in Figure 3 (*left*). Each agent has a local observation that contains a square view with 5 × 5 grids centered at the agent itself. At each timestep, each agent can move to one of four neighboring grids or eat a dot. The agents only get a global reward, *i.e.*, the total eaten dots, at the final timestep. We trained all the models for five runs with different random seeds.

As depicted Figure 4, MAAC and QMIX get the lowest mean environmental reward respectively, since some agents learn to go to the same room and compete for the dots. This can be verified by the position distribution of QMIX agents in Figure 5 (*left*), where three agents move to the same room. MAAC agents behave similarly. At the early stage of training, it is easy for the agents to explore the bottom room and eat dots there to improve the reward. Once the agents learn to get the cheap reward, it is hard to explore other rooms, so the agents learn similar behaviors and fall at the local optimum.

Driven by the intrinsic reward without both regularizers, EOI obtains better performance than MAAC and QMIX. But the improvement is not significant since the observations of different agents cannot be easily distinguished when there is little difference between initial policies. The regularizers of PD and MI can increase the discriminability of $P_\phi(I|O)$, providing stronger intrinsic signals. Guided by $P_\phi(I|O)$ with PD or MI, the agents go to different rooms and eat more dots. MI theoretically increases the discriminability even the initial policies have no differences, while PD makes the observations distinguishable according to policies. Combining the advantages of the two regularizers leads to higher and steadier performance, as shown in Figure 4. With both two regularizers, the agents respectively go to the four rooms and achieve the highest reward, which is indicated in Figure 5 (*mid*). We also visualize the observations of different agents by kernel PCA, as illustrated in Figure 5 (*right*), where darker color means higher correctly predicted probability of $P_\phi(I|O)$. We can see $P_\phi(I|O)$ can easily distinguish agents given their observations.

To further investigate the effect of the regularizers, we show the learning curves of intrinsic reward and environmental reward of EOI with or without regularizers in Figure 6. EOI with the regularizers converges to higher intrinsic reward than that without regularizers, meaning agents behave more distinctly. With regularizers, the rapid increase of intrinsic reward occurs before that of environmental reward, which indicates the regularizers make $P_\phi(I|O)$ also serve as the inductive bias for the emergence of individuality.

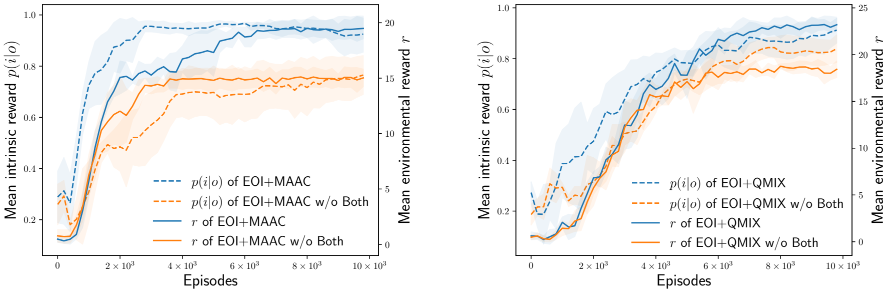


Figure 6: Learning curves of intrinsic reward and environmental reward of EOI+MAAC (*left*) and EOI+QMIX (*right*) in Pac-Men.

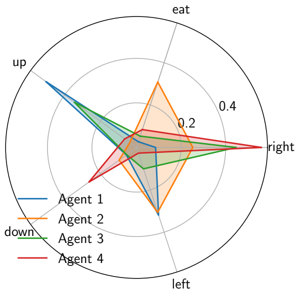


Figure 7: Action distributions of the initial action-value functions in QMIX.

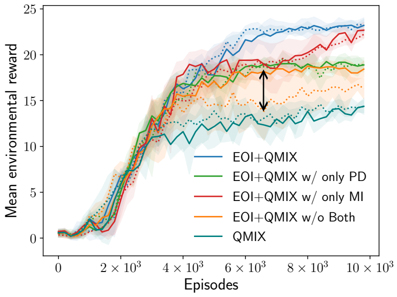


Figure 8: Learning curves with EOI+QMIX. The dotted lines are the version with the same initial action-value functions.

We also investigate the influence of the difference between initial policies. The action distributions of the four initial action-value functions in QMIX are illustrated in Figure 7, where we find there is a large difference between them. The inherent observation makes agents distinguishable initially, which is the main reason EOI without the regularizers works well. We then initiate the four action-value functions with the same network weights and re-run the experiments. The performance of EOI without the regularizers drops considerably, while EOI with the regularizers has almost no difference, as illustrated in Figure 8. Therefore, we can conclude that PD and MI also make the learning more robust to the initial policies. In other words, even with the same innate characteristics, agent individuality still emerges.

4.2 Windy Maze

In the scenario, there are two pirates (agents) initialized at the bottom of the T-shaped maze, and two treasure boxes initialized at the two ends, as illustrated in Figure 3 (*mid*). Each agent has a local observation that contains a square view with 5×5 grids and could move to one of four neighboring grids or open the box. There is a wind running right from the dotted line. Shifted by the wind, forty percent of the time, the agent will move to the right grid whichever action it takes. The agents could only get a global reward, *i.e.*, the number of the opened boxes, at the final timestep. We trained all the models for ten runs with different random seeds.

Figure 9 shows the learning curves of EOI+MAAC and EOI+QMIX. Under the effect of the wind, it is easy to go to the right end and open the box, even if the agent acts randomly. Limited by the time horizon, it is impossible that the agent first opens the right box and then goes to the left end for the other box. MAAC and QMIX only achieve the mean reward of 1. Due to the wind and the small state space, the trajectories of the two agents are similar, thus EOI without regularizers provides little help, where the intrinsic reward is not strong enough to induce individuality. With regularizers, the observations on the right path and the left path can be discriminated gradually. In the learning process, we find that the agents will first both learn to open the right box. Then one of them will change its policy and go left for higher intrinsic reward, and eventually the agents develop the distinct policies and get a higher mean reward.

In this scenario, the left box is hard to explore, thus we also compare EOI with EDTI [27], a multi-agent exploration method. EDTI gives the agent an intrinsic motivation, considering both curiosity and influence. It rewards the agent to explore the rarely visited states and to maximize the influence on the expected returns of other agents. The optimization of the intrinsic motivation is integrated

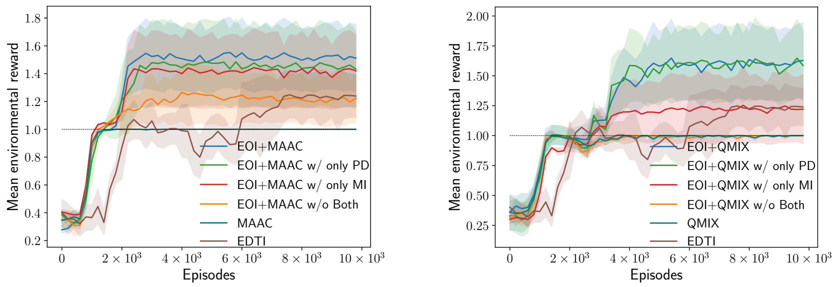


Figure 9: Learning curves in Windy Maze: EOI+MAAC (*left*) and EOI+QMIX (*right*).

with PPO [30]. However, EDTI does not perform well in this scenario. When the agent policies are similar, the two agents will be curious about similar states. When they both explore novel states, the environmental reward decreases, which draws the agents to go back to the right box again. So the curiosity is not stable, and we can see its fluctuation during training in Figure 9.

4.3 Firefighters

There are four areas in the forest, with some burning grids in two areas and two rivers in other areas. Four firefighters (agents) are initiated at the same burning area, illustrated in Figure 3 (*right*). The agent has a local observation that contains a square view with 5×5 grids and can move to one of four neighboring grids, spray water, or pump water. They share a water tank, which is initiated with four units water. Once the agent sprays water on the burning grid, it puts out the fire and consumes one unit water. Once the agent pumps water in the river, the water in the tank increases by 1 unit. The agents only get a global reward, *i.e.*, the number of the extinguished burning grids, at the final timestep. We trained all the models for five runs with different random seeds.

Figure 10 illustrate the learning curves. MAAC and QMIX fall into local optimum. The agents learn to put out the fire around the initial position for the cheap reward until the water is exhausted, and get the mean reward 4. They do not take different roles of pumping water or putting out the fire farther. Benefited from the emergent individuality, EOI+MAAC achieves the best performance with a clear division of labor, where two agents go to the river for pumping water, and two agents go to different areas for fighting with fire. EDTI could escape from the local optimum, but it converges to a lower mean reward than EOI. This is because encouraging curiosity and influence cannot help division of labor, and the intrinsic motivation of EDTI is too complex to suit the cases with more than two agents.

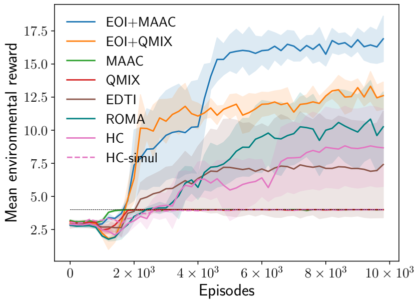


Figure 10: Learning curves in Firefighters.

ROMA [12], based on QMIX, uses a role encoder taking the local observation to generate role embedding, and decodes the role to generate the parameters of individual action-value function in QMIX. However, ROMA converges to a lower mean reward than EOI+QMIX and learns much slower. Due to the large parameter space, generating various parameters for the emergent roles is less efficient than encouraging diverse behaviors by reward shaping. Moveover, since the agents are initiated with the same observations, the role encoder will generate the similar role embeddings, which means similar agent behaviors at the beginning, bringing difficulty to the emergent roles.

HC [10] first provides each agent a set of diverse skills, which are learned independently by DIAYN (before 6×10^3 episodes), then learns a high-level policy to select the skill for each agent. Since the agents are trained independently without coordinating with others' skills, the skills could be diverse but might not be useful, *e.g.*, ignoring the pumping action. So it converges to a lower mean reward. We also train the primitive skills and high-level policy simultaneously, *i.e.*, HC-simul [11], and find it falls into local optimum. Before the diversity emerges in the skills, the high-level policy learns to choose the skill of putting out the nearby fire for the cheap reward, however other skills would not be chosen to learn diversity. That is to say, HC-simul only considers diversity in the low-level policies but ignores that in the high-level policy. EOI encourages individuality with the coordination provided by centralized training, considering both of the global reward and other agents' policies. Therefore, EOI is more effective than HC and HC-simul.

5 Conclusion

We have proposed EOI, a novel method for the emergence of individuality in MARL. EOI learns a probabilistic classifier that predicts a probability distribution over agents given their observation and gives each agent an intrinsic reward of being correctly predicted by the classifier. Two regularizers are introduced to increase the discriminability of the classifier. We realized EOI on top of two popular MARL methods and empirically demonstrated that EOI significantly outperforms existing methods in a variety of multi-agent cooperative tasks and the regularizers indeed improve the emergence of individuality even if the agents have the same innate characteristics.

Broader Impact

Not applicable.

References

[1] Julia Freund, Andreas M Brandmaier, Lars Lewejohann, Imke Kirste, Mareike Kritzler, Antonio Krüger, Norbert Sachser, Ulman Lindenberger, and Gerd Kempermann. Emergence of individuality in genetically identical mice. *Science*, 340(6133):756–759, 2013.

[2] Deborah M Gordon. The organization of work in social insect colonies. *Nature*, 380(6570):121–124, 1996.

[3] Kevin McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

[4] Wenqian Liu, Octavia Camps, and Mario Sznaier. Multi-camera multi-object tracking. *arXiv preprint arXiv:1709.07065*, 2017.

[5] Pedro V Sander, Denis Peleshchuk, and Barbara J Grosz. A scalable, distributed algorithm for efficient task allocation. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2002.

[6] Mehdi Dastani, Virginia Dignum, and Frank Dignum. Role-assignment in open agent societies. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.

[7] Mark Sims, Daniel Corkill, and Victor Lesser. Automated organization design for multi-agent systems. *Autonomous agents and multi-agent systems*, 16(2):151–185, 2008.

[8] Kathryn Sarah Macarthur, Ruben Stranders, Sarvapali Ramchurn, and Nicholas Jennings. A distributed anytime algorithm for dynamic task allocation in multi-agent systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

[9] Tianmin Shu and Yuandong Tian. M^{3}rl: Mind-aware multi-agent management reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[10] Youngwoon Lee, Jingyun Yang, and Joseph J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] Jiachen Yang, Igor Borovikov, and Hongyuan Zha. Hierarchical cooperative multi-agent reinforcement learning with skill discovery. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

[12] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020.

[13] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019.

[14] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

[15] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[16] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[17] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018.

[18] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019.

[19] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Individualized controlled continuous communication model for multiagent cooperative and competitive tasks. In *International Conference on Learning Representations (ICLR)*, 2019.

[20] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning (ICML)*, 2019.

[21] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[22] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning (ICML)*, 2017.

[23] Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. Composable deep reinforcement learning for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[24] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

[25] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations (ICLR)*, 2019.

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations (ICLR)*, 2020.

[28] Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. Liir: Learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[31] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Appendix

Mathematical Formulation

Let $\boldsymbol{\theta}$ and ϕ denote the parameters of the joint policies and the probabilistic classifier, respectively. Then, the whole learning process corresponds to the following bi-level optimization:

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}, \phi^*(\boldsymbol{\theta})) \\ s.t. \quad & \phi^*(\boldsymbol{\theta}) = \arg\min_{\phi'\in\Phi} \mathcal{L}(\phi', \boldsymbol{\theta}), \end{aligned}$$

where J is the RL objective with intrinsic reward, $\mathcal{L}$ is the loss function of the probabilistic classifier, and ϕ is an implicit function of $\boldsymbol{\theta}$. Therefore, to solve this optimization, we can iteratively update $\boldsymbol{\theta}$ by

$$\frac{\mathrm{d}J(\boldsymbol{\theta}, \phi^*(\boldsymbol{\theta}))}{\mathrm{d}\boldsymbol{\theta}} = \left.\frac{\partial J(\boldsymbol{\theta}, \phi)}{\partial \boldsymbol{\theta}}\right|_{\phi=\phi^*(\boldsymbol{\theta})} + \left.\frac{\mathrm{d}\phi^*(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\frac{\partial J(\boldsymbol{\theta}, \phi)}{\partial \phi}\right|_{\phi=\phi^*(\boldsymbol{\theta})}$$

where

$$\frac{\mathrm{d}\phi^*(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = -\left(\frac{\partial^2\mathcal{L}(\phi, \boldsymbol{\theta})}{\partial\phi\partial\phi^T}\right)^{-1}\left.\left(\frac{\partial^2\mathcal{L}(\phi, \boldsymbol{\theta})}{\partial\phi\partial\boldsymbol{\theta}^T}\right)\right|_{\phi=\phi^*(\boldsymbol{\theta})}$$

which is obtained by the implicit function theorem. In practice, the second-order term is neglected due to high computational complexity, without incurring significant performance drop, such as in meta-learning and GANs. Therefore, we can solve the bi-level optimization by the first-order approximation with iterative updates:

$$\begin{aligned} \phi_{k+1} &\approx \arg\min_{\phi} \mathcal{L}(\phi, \mathcal{B}_k) \\ \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \zeta_k \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \phi_{k+1}). \end{aligned}$$

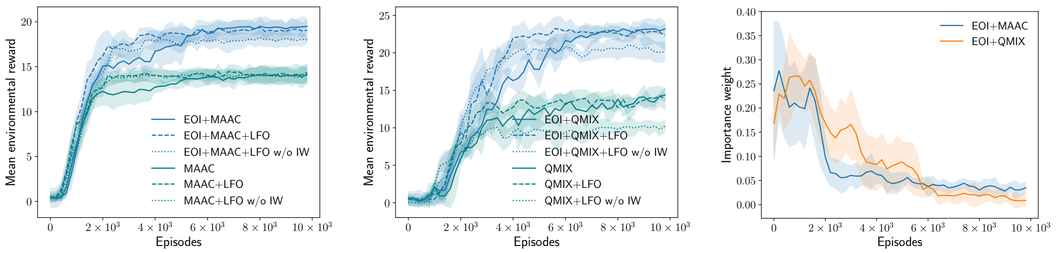
Learning from Experiences of Other Agents

When agents are exploring the environment, agents have various exploration patterns. The experiences of one agent can be helpful for the learning of other agents, *i.e.*, improving data efficiency and accelerating learning. Therefore, we propose an unbiased gradient estimate for learning from other agents' experiences. In centralized training, all agents' experience tuples $< O^t, A^t, r^t, O^{t+1}, s^t, s^{t+1} >$ are stored in the replay buffer, and sampled as batch Ω to compute the gradients jointly. For each agent i, the probability of sampled observation is $p(o|i)$. To use others' experiences, we randomly rearrange the agent index in a subset of Ω, denoted as ω. For example, in a two-agent case, two agents exchange o^t, a^t, and o^{t+1} to build a new training experience. In the experiences with shuffled index ω, the training experience of each agent i is sampled based on $p(o)$ rather than $p(o|i)$. By Bayes rule,

$$p(o|i) = \frac{p(i|o)p(o)}{p(i)}.$$

Since the number of observations for each agent is equal in the training set, $p(i)$ is a constant. Then we have

$$p(o|i) \propto p(i|o)p(o).$$



(a) EOI+MAAC

(b) EOI+QMIX

(c) importance weight

Figure 11: Learning curves of DOI with LFO on top of MAAC (a) and QMIX (b), and the curve of importance weight during the training (c) in Pac-Men.

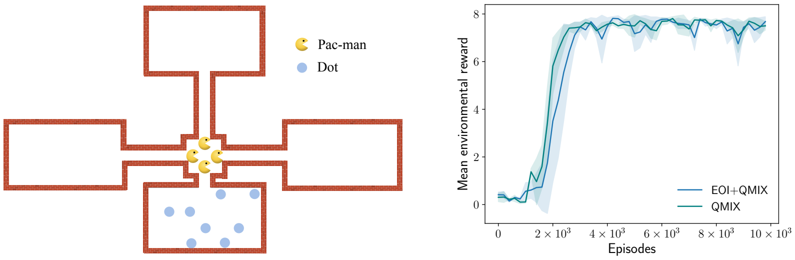


Figure 12: The easy version of Pac-Men (*left*) and learning curves (*right*).

We use $p_\phi(i|o)$ as importance weight (IW) for the gradient of individual action-value function/policy to make sure that the estimate of the expected gradients based on $p(o)$ is unbiased from that based on $p(o|i)$.

$$\mathbb{E}_{o\sim p(o|i)}\nabla_{\theta_i}J(\theta_i) \propto \mathbb{E}_{o\sim p(o)}p_\phi(i|o)\nabla_{\theta_i}J(\theta_i).$$

Figure 11a and 11b show the learning curves of EOI (with both PD and MI) with unbiased learning from others' experiences (LFO) and without importance weight respectively in Pac-Men. LFO greatly accelerates the learning of EOI since the agents have access to more experiences, and it converges to the similar performance with EOI since the gradient estimation is unbiased. Applying LFO to MAAC and QMIX also accelerates the convergence. Without important weight, LFO leads to lower mean reward, because the gradient estimate using others' experience is biased, which would harm the learning. To evaluate how much knowledge is learned from others, in Figure 11c we plot the curves of mean important weight on other agents' experiences $\mathbb{E}_{i,o}[\frac{1}{n-1}\sum_{j\neq i}p_\phi(i|o_j)]$, where i and j are the agent indexes before rearranging. At the beginning, as the agents are exploring the environment, the mean importance weight is high, which indicates that the agent learns a lot from others' experiences. In later phase, the mean importance weight becomes very low, because individuality has emerged and the experiences of agents are very different, the agent cannnot benefit from others' experiences. In the experiment, $|\omega|/|\Omega|$ is set as a constant 0.2. However, enlightened by the trend of the mean importance weight, we could also adjust $|\omega|/|\Omega|$ according to a pre-defined schedule, *e.g.*, time-decaying or adaptive methods related to the mean importance weight.

When Individuality Is Not Necessary

To investigate the performance of EOI in the scenarios where individuality is not necessary, we perform the experiments on an easy version of Pac-Men, where the dots are only distributed in the bottom room, illustrated in Figure 12 (*left*). To complete this task, the agents do not need to go to different rooms. As depicted in Figure 12 (*right*), EOI+QMIX converges to the same mean reward with the QMIX, which verifies that EOI does not conflict with the success of task where agent individuality is not necessary.

StarCraft II

To verify whether EOI could be applied in complex environments, we evaluate EOI on a micromanagement scenario of StarCraft II [31] `8m`, where 8 Marines face 8 enemy Marines. We trained all the models for three runs with different random seeds. Illustrated in Figure 13, EOI learns faster than ROMA and QMIX, showing the classifier $P_\phi(I|O)$ could be effectively trained and provide useful intrinsic signals in the highly dynamic environment with many agents. However, we found that the diversification in observations or trajectories is unnecessary in the optimal strategies in `8m`. Since we decay the exploration probability ϵ quickly, linearly from 1.0 to 0.05 over $10k$ timesteps, the exploration is limited. The intrinsic reward in EOI and the role embedding in ROMA could promote the exploration, that is why EOI and ROMA learn faster than QMIX in `8m`. Note that we represent the individual action-value function as MLP, not GRU, which means that the action value is generated only based on o_i^t, and the agents do not share the neural network parameters so that the result of QMIX is different from that in the original paper [14].

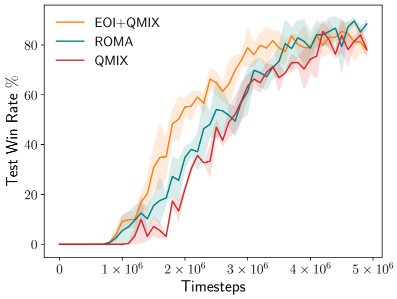


Figure 13: Learning curves in `8m`.

Similarity and Difference Between EOI and DIAYN

DIAYN is proposed to learn diverse skills with single-agent RL in the absence of any rewards. It trains the agent by maximizing the mutual information between skills (Z) and states (S), maximizing the policy entropy, and minimizing the mutual information between skills and actions (A) given the state. The optimization objective is

$$\begin{aligned}
&\mathrm{MI}(S;Z) + \mathcal{H}(A|S) - \mathrm{MI}(A;Z|S) \\
&= \mathcal{H}(Z) - \mathcal{H}(Z|S) + \mathcal{H}(A|S) - (\mathcal{H}(A|S) - \mathcal{H}(A|Z,S)) \\
&= \mathcal{H}(Z) - \mathcal{H}(Z|S) + \mathcal{H}(A|Z,S).
\end{aligned}$$

To maximize this objective, in practice DIAYN gives the learning agent an intrinsic reward $\log q(z|s) - \log p(z)$, where $q(z|s)$ approximates $p(z|s)$ and $p(z)$ is a prior distribution. EOI gives each agent an intrinsic reward of $p(i|o)$. Let agents correspond to skills and observations correspond to states, then EOI has a similar intrinsic reward with DIAYN. However, simply adopting the intrinsic reward in MARL, which can be seen as EOI without the regularizers, cannot provide strong intrinsic reward signals for the emergence of individuality. The two regularizers, which capture the unique characteristics of MARL, strength the reward signals and promote the emergence of individuality in MARL.

Hyperparameters

The hyperparameters of EOI and baselines in each scenario are summarized in Table 1. Since QMIX and MAAC are off-policy algorithms with replay buffer, we do not need to maintain the buffer $\mathcal{B}$ but build the training data from the replay buffer $\mathcal{D}$. For EDTI, ROMA, HC, and HC-simul, we use their default settings. In the experiments, the agents do not share the weights of neural network since parameter sharing causes similar agent behaviors.

Table 1: Hyperparameters

Hyperparameter	Pac-man	Windy Maze	Firefighters	8m
horizon (T)	30	15	20	120
discount (γ)		0.98		
batch size		128		64
replay buffer size		2×10^4		2×10^5
# MLP units		$(128, 128)$		$(64, 64)$
actor learning rate		1×10^{-3}		-
critic learning rate		1×10^{-4}		-
QMIX learning rate		1×10^{-4}		
MLP activation		ReLU		
optimizer		Adam		
ϕ learning rate		1×10^{-3}		
α in QMIX		0.05		
α in MAAC		0.2		-
β_1		0.04		0.01
β_2		0.1		0.01
Δt		4		