DDMA: Discrepancy-Driven Multi-agent Reinforcement Learning

Chao Li[1,4], Yujing Hu[2], Pinzhuo Tian[1,3], Shaokang Dong[1,4], and Yang Gao[1,4(✉)]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
chaoli1996@smail.nju.edu.cn, gaoy@nju.edu.cn
[2] NetEase Fuxi AI Lab, Hangzhou, China
[3] School of Computer Engineering and Science, Shanghai University, Shanghai, China
[4] Shenzhen Research Institute, Nanjing University, Shenzhen, China

Abstract. Multi-agent reinforcement learning algorithms depend on quantities of interactions with the environment and other agents to derive an approximately optimal policy. However, these algorithms may struggle in the complex interactive relationships between agents and tend to explore the whole observation space aimlessly, which results in high learning complexity. Motivated by the occasional and local interactions between multiple agents in most real-world scenarios, in this paper, we propose a general framework named Discrepancy-Driven Multi-Agent reinforcement learning (DDMA) to help overcome this limitation. In this framework, we first parse the semantic components of each agent's observation and introduce a proliferative network to directly initialize the multi-agent policy with the corresponding single-agent optimal policy, which bypasses the misalignment of observation spaces in different scenarios. Then we model the occasional interactions among agents based on the discrepancy between these two policies, and conduct more focused exploration on these areas where agents interact frequently. With the direct initialization and the focused multi-agent policy learning, our framework can help accelerate the learning process and promote the asymptotic performance significantly. Experimental results on a toy example and several classic benchmarks demonstrate that our framework can obtain superior performance compared to baseline methods.

Keywords: Multi agent · Reinforcement learning · Exploration

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has shown great potential in solving many real-world problems such as coordination of robot swarm [10] and autonomous car [1]. However, learning effective policies in such multi-agent systems remains challenging owing to the complex interactions among agents. These

S. Khanna et al. (Eds.): PRICAI 2022, LNCS 13631, pp. 91–105, 2022.
https://doi.org/10.1007/978-3-031-20868-3_7

dynamical interactions result in two major issues that prevent us from directly extending classic single-agent reinforcement learning algorithms to the multi-agent scenarios. One is the scalability problem, which means that the size of the joint observation-action space grows exponentially with the number of agents. The other is the non-stationarity problem, which means that the environment changes dynamically from the individual perspective of each agent owing to the simultaneous updates of all agents' policies.

Hence, existing MARL algorithms aim to solve these two problems to learn an optimal multi-agent policy. Recently the paradigm of Centralized Training with Decentralized Execution (CTDE) alleviates these two problems by accessing the global information during training and outputting the decentralized policies conditioned on the agents' local observation-action histories during execution. Such design strides a balance between the full centralization and the full decentralization. Specially, the multi-agent policy-based algorithms such as MADDPG [15], COMA [7], MAAC [11] train a centralized critic that takes as inputs the actions of all agents including the state information and then guide the optimization of decentralized policies with an actor-critic framework. Value-decomposition algorithms under this paradigm such as VDN [22], QMIX [19], QTRAN [21], QPLEX [25] learn a centralized state-action value function that can be factorized into the individual utility value functions conditioned on the local observation-action history of each agent, which needs to satisfy the IGM [21] constraint.

While the CTDE-based algorithms have achieved significant performance in some challenging scenarios such as StarCraft [20], their drawbacks are also apparent in the complex tasks. They usually require substantial interactions and trials in the tasks where agents are tightly coupled, which usually results in high learning complexity. In fact, the interactions between agents usually happen locally and occasionally in many real-world multi-agent systems, which means that each agent only needs to take some other agents into consideration and coordinates with them in some certain observations. In such situation, the algorithm should focus more on these interactive areas, instead of searching for the whole observation space aimlessly. Exploiting such properties existing in the interactions between agents can dramatically promote the learning efficiency of the CTDE-based algorithms.

However, modeling the interactions among agents accurately still remains intractable owing to the simultaneous updates of all agents' policies. Recent works [4,12] learn the reward dynamics of the multi-agent environment and distinguish the interactions through detecting the changes in the received immediate or long-term rewards. But these methods may fail owing to the inaccurate approximated reward model, which usually needs sufficient prior knowledge or complete search of the environment. In addition, some works [11,13] implicitly characterize the interactions between agents in real time through the attention mechanism. But such redundant networks and tedious optimizations may further increase the learning complexity of the algorithms. In this paper, we solve this problem by proposing several structural and learning novelties.

To simplify the multi-agent policy learning and reduce the high learning complexity of CTDE-based algorithms, we propose a general framework named Discrepancy-Driven Multi-Agent reinforcement learning (DDMA). In this framework, we first (1) initialize the multi-agent policy directly with a pre-learned single-agent optimal policy derived from an auxiliary single-agent scenario as previous works [4,12], then (2) recognize the dynamical interactions between agents through measuring the discrepancy between the updating multi-agent policy and the single-agent optimal policy. In detail, we propose an interaction detector to quantify this discrepancy. This detector can be used to recognize the interactive areas where current agent is more likely to be influenced by other agents. Accordingly, as shown in Fig. 1, we conduct more focused multi-agent policy learning in these interactive areas, instead of exploring and learning in the whole observation space aimlessly and costly. In addition, we parse the semantic components of the agents' observations and further introduce a proliferative network to achieve the direct initialization from the pre-learned single-agent optimal policy to the multi-agent policy, which can solve the terrible misalignment of observation spaces in different scenarios. Under the direct initialization and the focused multi-agent policy learning, DDMA can help accelerate the learning process and promote the asymptotic performance significantly.

In order to demonstrate the effectiveness of DDMA, we evaluate it in a toy example and three classic multi-agent scenarios under Multi-agent Particle Environment (MPE) [16]. Experimental results demonstrate that DDMA can achieve faster learning and better asymptotic performance compared to baseline methods. In addition, by ablation studies, we confirm that DDMA indeed promotes the learning efficiency with the direct initialization and the focused multi-agent policy learning.

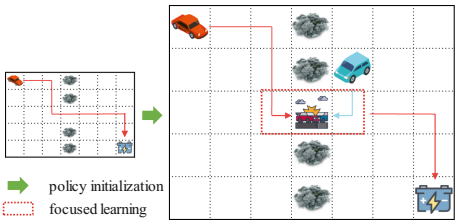


Fig. 1. Under the multi-agent policy initialized by the corresponding single-agent optimal policy, the agent (the red car) should conduct more focused multi-agent policy learning in the collision area (the red rectangle with dashed line).

In summary, the primary contributions of this paper are listed as follows: (1) Inspired by the pre-training mechanism, we introduce a proliferative network to directly initialize the multi-agent policy with the pre-learned single-agent optimal policy, which solves the misalignment of observation spaces in different scenarios. (2) In order to further improve the learning efficiency, we introduce an interaction detector to characterize the interactions between agents, and conduct more focused multi-agent policy learning in these interactive areas. (3) Experimental results highlight that DDMA can achieve superior performance compared to other algorithms. Ablation study further verifies the effectiveness of our algorithm.

2 Related Work

There exist many works devoting to reducing the high learning complexity rooted in MARL, such as transfer learning [3–5,12,24], curriculum learning [6,17,27] and game abstraction [13,26,29]. In this section, we briefly describe these methods and detail some related algorithms.

Transfer Learning. Transfer learning tries to improve the algorithm's learning efficiency through extracting and reusing some task-relevant knowledge. In MARL problem, some works focus on the knowledge transfer between agents, such as LeCTR [18], SEAC [2] and MAPTF [28]. Other works about the sparse-interaction problem learn to transfer knowledge from the corresponding single-agent scenario to the multi-agent scenario, where CQ-Learning [4] identifies the interactive areas through detecting changes in the received immediate rewards and conducts the multi-agent policy learning only in these areas selectively, as well as expanding the state representations. Then NSR [12] extends it to more difficult scenarios through detecting the changes in the long-term rewards. In this paper, we don't try to let agents choose whether to utilize the pre-learned knowledge or learn. Instead, our proliferative network structure shown in Sect. 4.1 can directly initialize the multi-agent policy with the pre-learned single-agent optimal policy, which makes us free from the negative transfer during learning.

Curriculum Learning. Curriculum learning considers first learning in the simple environments then conducting the final policy learning in the original multi-agent environment based on the pre-learned knowledge. In multi-agent curriculum reinforcement learning, DyMA-CL [27] introduces three kinds of knowledge transfer ways between the curriculum tasks with different number of agents. And it solves the misalignment of observation spaces in different tasks through aggregating information about others by GNN, which promises the direct initialization from pre-learned policy to the current policy. Although GAS [6] can implicitly create curriculum tasks through restricting available action space of each agent in different stages, the choices of action spaces in each curriculum stage matters and the final policy learned may be sub-optimal. Usually, curriculum learning and transfer learning are tightly coupled and are combined implicitly in related researches.

Game Abstraction. Game abstraction mainly attempts to reduce the scale of Markov Game, such as Mean Field MARL [29], G2ANet [13] and RODE [26]. Mean-field theory averages effects from neighboring agents to reduce the learning complexity of each agent, which enables efficient multi-agent policy learning in the large-scale scenarios. G2ANet models the interactive relationships between agents through a two-stage attention network structure. Such combination of hard and soft attention network can indicate the interactive strengths between all pair-wise agents and remove some humble correlations. RODE pre-defines several

roles based on the action-effect-based clustering, then conducts the hierarchical multi-agent policy learning.

3 Preliminary

In this section, we formalize the multi-agent task in our work and briefly review some concepts of reinforcement learning.

3.1 Partially Observable Stochastic Game

In this paper, we focus on Partially Observable Stochastic Game $< N, S, \boldsymbol{A}, R, P, O, \boldsymbol{Z}, \gamma >$, where S is the state space and $\boldsymbol{A} = \{A^1 \times A^2 \times \ldots \times A^n\}$ represents the joint action space of all agents. At each time step t, each agent $i \in N$ receives its partial observation $o_t^i \in Z^i$ according to the observation function $O(s_t, i)$ and selects action $a_t^i \in A^i$, forming a joint action $\boldsymbol{a_t}$. Then the environment will transit to the next state s_{t+1} according to the transition function $P(s_{t+1}|s_t, \boldsymbol{a_t})$, and provide each agent with its reward r_{t+1}^i according to the reward function $R(s_t, \boldsymbol{a_t}, i)$. Each agent i conditions its policy π^i on its own observation-action history τ^i. The value function of all agents' joint policy $\boldsymbol{\pi} = (\pi^1, \pi^2, ..., \pi^n)$ to agent i at state s is defined as: $V_{\boldsymbol{\pi}}^i(s) = E_{\boldsymbol{\pi}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}^i | s_t = s]$, where γ is the discount factor and each agent aims to maximize its own value function.

3.2 Reinforcement Learning

In single-agent RL, value-based algorithms define the state-action value function $Q(s, a) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$ and state value function $V(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s]$, which both can be updated according to the bellman equation. Then the optimal action can be derived by the greedy policy $a^* = argmax_a Q(s, a)$. In contrast, policy-based algorithms directly optimize the policy instead of evaluating the value function. Furthermore, actor-critic algorithms combine the former two, which guide the update of the actor according to the critic. In multi-agent RL, agents can utilize a MARL algorithm to learn approximately optimal policies to adapt to the multi-agent scenarios.

4 Method

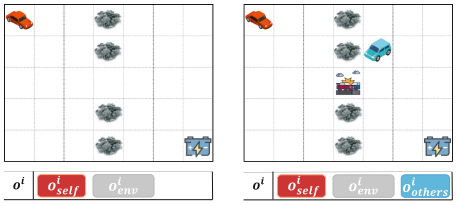
In this section, we introduce our framework DDMA to promote the multi-agent policy learning. We first describe how our method efficiently transfer the pre-learned knowledge. Then how we model the interactions among agents and conduct the focused multi-agent policy learning will be stated.

4.1 Initialization of the Multi-agent Policy

In contrast to the prior works that decide when transfer should occur [4,12], we propose to directly initialize the multi-agent policy with the corresponding single-agent optimal policy. The intuition behind this is that there exist similar situations between the single-agent scenario and the multi-agent scenario, as well as strong correlations between the policies in them. For example, when one company wants to train a group of UAV transport formations, it first put only one UAV into the target environment to learn some basic skills such as takeoff, landing and so on, which corresponds to the single-agent version of the target multi-agent scenario. After mastering these primary skills, each UAV can learn to coordinate with others and adapt to the multi-agent scenario more efficiently.

However, the misalignment of observation spaces in the single-agent scenario and the multi-agent scenario obstructs this direct initialization (we assume the same action space in these two scenarios). To overcome this limitation, we parse the semantic components of the agent's observation in the multi-agent scenario and accordingly introduce a novel proliferative policy network, which can simplify the structure of the multi-agent policy compared to GNN, attention networks [14,27] and avoid the tricky trade-off between the original policy learning and the extra policy distillation [12].

As shown in Fig. 2, the only difference between the multi-agent scenario and the single-agent scenario lies in the emergence of other agents. Accordingly, we parse the agent i's observation o^i in the multi-agent scenario into three semantic components: the agent's individual information o^i_{self}, the agent's cognition towards the environment such as the entities o^i_{env} and the cognition to all other agents o^i_{others} such as the relative distances. The former two components make up the agent's observation in the corresponding single-agent scenario. Here we assume the same action space in these two scenarios. As for the inconsistency of action spaces, we can also parse the semantic actions through mapping them to a latent space, and new actions can be approximated according to their neighboring latent representations.



(a) Components of the observation in single-agent scenario

(b) Components of the observation in multi-agent scenario

Fig. 2. The semantic components of the agent's observation in the single-agent scenario and the multi-agent scenario.

Based on this parse, we introduce the proliferative policy network, as shown in Fig. 3 4. Considering that the agent's observation o^i only contains $[o^i_{self}, o^i_{env}]$ in the single-agent scenario, the single-agent optimal policy takes as inputs these two parts. When the other agents emerge in the multi-agent scenario, o^i_{others} also emerges in the agent's observation, besides the $[o^i_{self}, o^i_{env}]$ existing in the single-agent scenario. Therefore, we decompose the actor in the multi-agent policy into two parts: (1) the main network that deals with $[o^i_{self}, o^i_{env}]$ and (2) the prolifer-

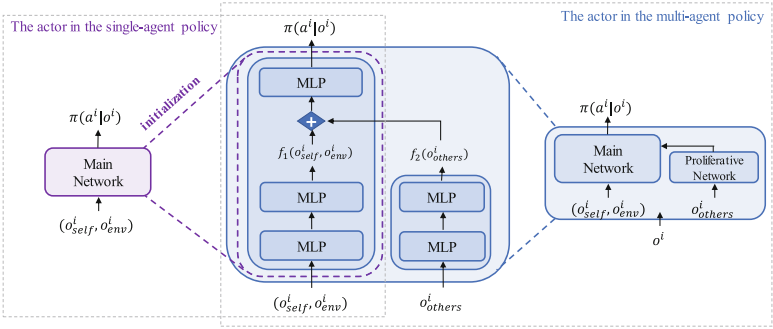


Fig. 3. The structure of the actor network. In the actor of the multi-agent policy, the main network deals with $[o^i_{self}, o^i_{env}]$ and the proliferative network embeds $[o^i_{others}]$. The main network can be initialized directly with the actor of the pre-learned single-agent optimal policy.

ative network that embeds $[o^i_{others}]$. Similarly, we also decompose the centralized critic into: (1) the main network that deals with $[o^i_{self}, o^i_{env}, a^i]$ and (2) the proliferative network that embeds $[o^{-i}_{self}, o^{-i}_{env}, a^{-i}]$, where $-i$ represents the other agents except agent i. Note that we remove o^i_{others} and o^{-i}_{others} from the centralized critic's inputs o^i, o^{-i} to derive a more compact global state representation.

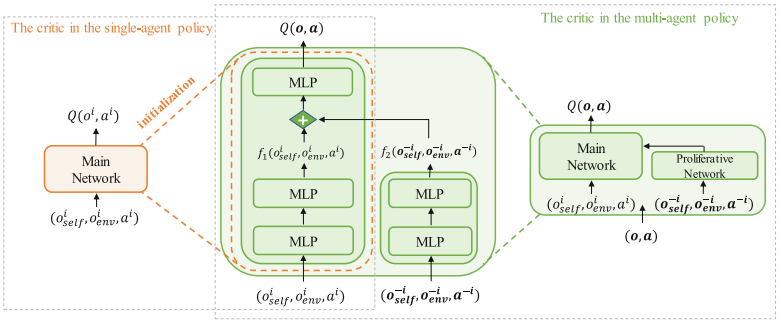


Fig. 4. The structure of the critic network. In the centralized critic of the multi-agent policy, the main network deals with $[o^i_{self}, o^i_{env}, a^i]$ and the proliferative network embeds $[o^{-i}_{self}, o^{-i}_{env}, a^{-i}]$. Similarly, the main network can be initialized directly with the critic of the pre-learned single-agent optimal policy.

Furthermore, we connect the proliferative network to the final layer of the main network. Specially, we use the element-wise summation of the outputs of the proliferative network and the outputs of the penultimate layer of the main network as the inputs of the final layer in the main network. Through such combination of these two networks, we can directly initialize the main network

in the multi-agent policy with the pre-learned single-agent optimal policy that is composed of only the same main network. Then the proliferative network can help update the whole policy through tracking other agents' information.

4.2 Focused Learning of the Multi-agent Policy

While each agent can benefit from this direct initialization owing to the lossless knowledge transfer, it may not coordinate with other agents well. In such situation, the agent should gradually adapt to the multi-agent scenario through updating its policy with a large number of interactive data. And current MARL algorithms usually force each agent to search for the whole observation space aimlessly and costly, which will result in high learning complexity.

Motivated by the occasional and local interactions between multiple agents in most real-world scenarios, we propose to model the interactions between agents and conduct more focused multi-agent policy learning in these interactive areas but a little in the other areas. Through focusing more on these interactive areas where rewards explicitly differ from the ones in the single-agent scenario, each agent can conduct steeper policy update in the corresponding observations to adapt to the multi-agent scenario, but only deviate slightly from the initialization of its policy in other areas that have similar rewards. Therefore, we can promote the learning efficiency through such adaptive and focused learning.

Specially, we characterize the dynamical interactions between agents through measuring the discrepancy between the multi-agent policy and the pre-learned single-agent optimal policy. The intuition behind this is that the initialized multi-agent policy will gradually diverge from its initialized version in some observations where agent following the pre-learned single-agent optimal policy may receive penalties owing to ignoring other agents. Based on this phenomenon, the discrepancy between the multi-agent policy and the pre-learned single-agent optimal policy will emerge in those areas where the interactions occur under all agents' current policies. Therefore, we use this discrepancy to evaluate whether the current agent is influenced by others under the current policies.

We first introduce how to measure the discrepancy between these two policies. DDMA trains a centralized critic that takes as inputs all agents' observations and actions when training, then updates the decentralized actor according to this critic, which follows the CTDE paradigm. Based on this centralized critic, the discrepancy between the multi-agent policy and the pre-learned single-agent optimal policy for agent i in observation o^i can be obtained according to:

$$D^i(o^i) = D_{KL}[\pi^i_{multi}(a^i|o^i, o^{-i}, a^{-i})||\pi^i_{single}(a^i|o^i)], \tag{1}$$

where:

$$\pi^i_{multi}(a^i|o^i, o^{-i}, a^{-i}) = \frac{e^{kQ^i_{multi}(o^i_{self}, o^i_{env}, a^i, o^{-i}_{self}, o^{-i}_{env}, a^{-i})}}{\sum_{a' \in A^i} e^{kQ^i_{multi}(o^i_{self}, o^i_{env}, a', o^{-i}_{self}, o^{-i}_{env}, a^{-i})}}, \tag{2}$$

$$\pi^i_{single}(a^i|o^i) = \frac{e^{kQ^i_{single}(o^i_{self}, o^i_{env}, a^i)}}{\sum_{a' \in A^i} e^{kQ^i_{single}(o^i_{self}, o^i_{env}, a')}}. \tag{3}$$

Note that we derive the Boltzmann policy according to the centralized Q-values instead of directly using the output of the actor network. This is because that the critic can characterize the long-term influence induced by the environmental rewards through bellman update compared to the actor which is updated under the supervision of the critic.

In order to achieve better generalization over the whole observation space, we then introduce an interaction detector for each agent to model this discrepancy, which takes each agent's observation as inputs and predicts whether the agent is influenced by other agents in the current observation. This detector is trained in the form of a binary classifier, whose loss can be written as:

$$\mathcal{L}_{detec} = CrossEntropy(p^i_{detec}(o^i), label(o^i)), \tag{4}$$

where $label(o^i)$ is set to 0 when $D^i(o^i) < \tau^i$ else 1, and $p^i_{detec}(o^i)$ refers to the output of this interaction detector. We set the threshold τ^i as some certain percentile of all $D^i(o^i)$ in the buffer (sixty in the code).

Finally, based on this detector model, we conduct more focused multi-agent policy learning in the interactive areas, instead of searching for the whole observation space aimlessly and costly. In detail, we equip each agent i with two exploration policies for environments with discrete and continuous action spaces respectively, which can induce the focused exploration in the areas with high interactive strengths, based on the predictions $p^i_{detec}(o^i)$.

The exploration policy for environments with continuous action spaces is defined as:

$$u^i_e(\tau^i) = u^i(\tau^i|\theta^i) + \alpha * |\max(0, p^i_{detec}(o^i) - 0.5)| * \mathcal{N}(0,1), \tag{5}$$

where α is the decay rate, u^i is the actor with parameters θ^i and $\mathcal{N}(0,1)$ represents the noise sampled from a standard normal distribution. We approximately calculate $\pi^i_{single}, \pi^i_{multi}$ through sampling several actions from the continuous action space. With this exploration policy, we add adaptive noise to the outputs of the actor to achieve more focused policy learning.

The exploration policy for environments with discrete action spaces is defined as:

$$\pi(u^i_e|\tau^i) = \begin{cases} 1 - \min(\epsilon + \frac{\max(0, p^i_{detec}(o^i) - 0.5)}{\sqrt{T}}, 1) & if\ u^i_e = u^{i,*} \\ \min(\epsilon + \frac{\max(0, p^i_{detec}(o^i) - 0.5)}{\sqrt{T}}, 1) & if\ u^i_e \neq u^{i,*}, \end{cases} \tag{6}$$

where T is the environment step, $u^{i,*}$ is the optimal action under the current policy and ϵ is the random probability.

Finally, DDMA drives agents to explore more in these interactive areas through these two exploration policies. Under this motivation, agents can conduct steeper policy update in the corresponding observations and adapt to the dynamical multi-agent scenario quickly.

4.3 Training

Now we briefly state the alternate updates of the interaction detector and the multi-agent policy. When learning the multi-agent policy, we freeze the detector and add extra exploration to the current action selection according to the exploration policies above. Similarly, when the detector is updated, the multi-agent policy learning is paused. Such optimization can guarantee stability and behaves like a Generative Adversarial Network (GAN) [9]. Interestingly, the interaction detector acts like a discriminator, which aims to find out the current most-likely interactive areas under all agents' current multi-agent policies. On the contrary, the multi-agent policy plays the role of the generator, which updates itself to change the interactive probabilities in all areas and misleads the delayed-updated detector. Owing to such adversarial relationship between these two components, DDMA can achieve the adaptive policy update in different areas according to the interaction detector and agents can adapt to the multi-agent scenario quickly based on such focused policy learning process.

5 Experiments

For the environment with discrete action space, We design Collision Corridor to illustrate our method comprehensively. As for the environment with continuous action space, we evaluate our method in three classic multi-agent scenarios under Multi-agent Particle Environment: Cooperative Navigation, Predator and Prey, and Individual Defense, where Individual Defense is constructed ourselves.

Below we will describe these environments briefly and show the superior performance of our method compared to other baselines. In addition, we also conduct some ablation studies to decide the effectiveness of each component.

5.1 Collision Corridor

We start with Collision Corridor to answer the following two questions: (1) Whether this algorithm can conduct more focused policy learning? (2) Can the interaction detector characterize the interactions between agents approximately?

Environments. As shown in Fig. 6 (a), each autonomous car starts in a corner and the goal is to arrive at the diagonal corner. A dangerous collision will occur when both cars try to pass through the corridor hidden in the obstacles, resulting in -10 penalty. Both cars will receive $+30$ reward if they all succeed in arriving at their target corners.

Baselines. In such situation with discrete actions, We compare DDMA with the state-of-the-art VDN and QMIX, as well as MAPG. We regard the original policy gradient algorithm [23] with a centralized critic as MAPG, and we achieve DDMA based on it.

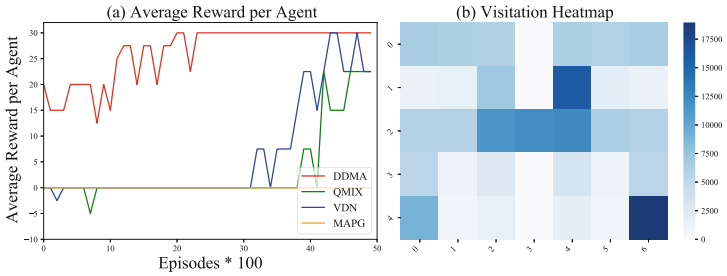


Fig. 5. (a) The experimental result in Collision Corridor. The result is averaged over four random seeds and we only show the average owing to the discrete reward setting. (b) The visitation heatmap shows the total visitation number of both cars in all areas during training.

Results. Figure 5 (a) exhibits the average rewards per agent of all algorithms. It's obvious that DDMA outperforms other baselines and converges faster, owing to the direct initialization and the focused multi-agent policy learning (owing to the less effort in pre-training the single-agent optimal policy, we don't consider it here). On the contrary, VDN and QMIX struggle to learn the policy through quantities of aimless trials. Even MAPG fails to solve this task, which further demonstrates the effectiveness of DDMA. Also, we present a heatmap to show the visitation number of both agents in all areas during training, which is shown in Fig. 5 (b). We observe that the agents always wander around the corridor to learn to coordinate with each other to avoid the collision, which demonstrates that the interaction detector indeed encourages the focused multi-agent policy learning in these interactive areas.

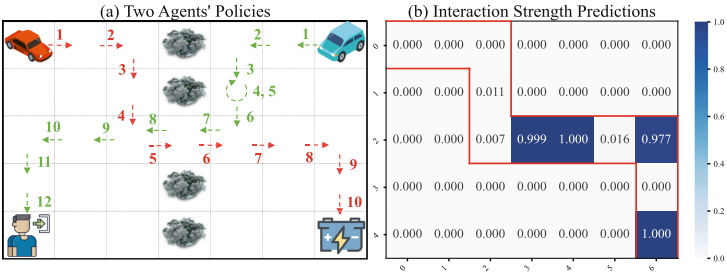


Fig. 6. (a) The final learned policies of two cars. The red line is the policy of car 1 and the green line is the policy of car 2. (b) The interaction strength predictions of car 1, where red lines represent the routes of car 1. (Color figure online)

In addition, we further analyze the effect of the interaction detector in some areas from the viewpoint of the upper left agent in Fig. 6 (b). Considering all agents' current policies shown in Fig. 6 (a), we notice that this agent tends to

recognize areas near the corridor and near its own target corner as the interactive areas. It is the collisions between agents in the corridor and the environmental setting that agents only receive the final reward after they both arrive at their goals respectively that produces such judgment, which verifies the effectiveness of the interaction detector in DDMA.

5.2 MPE Scenarios

After illustrating our method granularly in Collision Corridor, we further evaluate it in more complex scenarios under Multi-Agent Particle Environment.

Environments. Here we choose Cooperative Navigation, Predator and Prey and Individual Defense as the test scenarios.[1]

Baselines. In the situations with continuous action spaces, we choose MADDPG [15], MAAC [11], G2ANet [13] and Noisy-MADDPG [8] as the baselines.

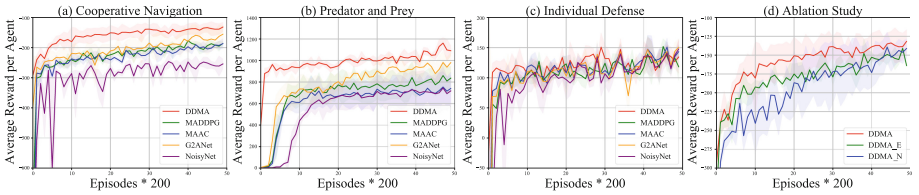


Fig. 7. (a), (b), (c) are the experimental results in Cooperative Navigation, Predator and Prey and Individual Defense respectively. (d) is the ablation experimental results in Cooperative Navigation. All results are averaged over four random seeds.

Results. Figure 7 (a) (b) (c) show the performance comparison against baselines. We can observe that DDMA outperforms baselines significantly in the former two scenarios. In detail, in Cooperative Navigation and Predator and Prey, DDMA begins with pretty performance owing to the initialization. What's more, DDMA learns faster and achieves significant performance, benefiting from the focused learning brought by the selective exploration. On the contrary, MAAC, G2ANet and other baselines still suffer from redundant and aimless policy learning, even getting trapped in the sub-optimal solutions.

In Individual Defense, DDMA behaves similarly as other baselines, which is owing to little interactions existing between agents. In this situation, each good agent pays more attention to approaching its own target landmark itself, ignoring the little collisions with other good agents owing to the low probabilities. Even the independent learner will adapt to this scenario well. What's more, this experimental results verify that DDMA essentially produces the interaction-driven multi-agent policy learning, and it will degrade into an independent learner when there is little interactions between agents.

[1] Please refer to https://github.com/chaobiubiu/DDMA for more details.

5.3 Ablation Study

To further demonstrate the effectiveness of each component in our method, we compare DDMA with extra baselines: (1) DDMA-E. We derive this baseline by removing the interaction detector's learning from vanilla DDMA, which can test the influence brought by this discriminator-like detector. (2) DDMA-N. This baseline only considers conducting the focused multi-agent policy learning through the interaction detector, regardless of the direct initialization from the single-agent optimal policy to the multi-agent policy, which can highlight the contribution from this fine initialization.

As shown in Fig. 7 (d), although the interaction detector drives much exploration, DDMA-N converges slowly owing to the lack of transferred knowledge. Similarly, benefiting from the superior initialization, DDMA-E can converge quickly but to a sub-optimal policy, which may result from the confused and aimless multi-agent policy learning. On the contrary, the vanilla DDMA can converge faster and achieve better performance due to the advantages existing in its all components.

In summary, these experiments demonstrate that DDMA indeed promotes the learning efficiency through the direct initialization and the focused policy learning. Drawing supports from these two factors mostly, DDMA can achieve superior performance in complex multi-agent tasks compared to other algorithms.

6 Conclusion

In this paper, we propose DDMA to alleviate the high learning complexity inherent in the multi-agent policy learning, through the direct initialization and the focused policy learning. Our approach has several benefits. It is simple and does not introduce any complex network structures. It is natural to extend the pre-learned single-agent optimal policy to further conduct the focused multi-agent policy learning instead of only initialization. Lastly, DDMA can be applied to any multi-agent situations with discrete or continuous action spaces.

Simplifying the tedious multi-agent policy learning matters mostly in MARL. We make a simple attempt to achieve it. We believe that MARL can be applied to some real scenarios by such way and we will continue to reduce the high learning complexity of MARL in future works.

Acknowledgements. This work is supported by Shenzhen Fundamental Research Program (No.2021Szvup056), Primary Research & Development Plan of Jiangsu Province (No. BE2021028), National Natural Science Foundation of China (No. 62192783), and Science and Technology Innovation 2030 New Generation Artificial Intelligence Major Project (No.2018AAA0100905).

References

1. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans. Industr. Inf. **9**(1), 427–438 (2012)
2. Christianos, F., Schäfer, L., Albrecht, S.: Shared experience actor-critic for multi-agent reinforcement learning, vol. 33, pp. 10707–10717 (2020)
3. Da Silva, F.L., Costa, A.H.R.: A survey on transfer learning for multiagent reinforcement learning systems. J. Artif. Intell. Res. **64**, 645–703 (2019)
4. De Hauwere, Y.M., Vrancx, P., Nowé, A.: Learning multi-agent state space representations. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 715–722 (2010)
5. Diuk, C., Cohen, A., Littman, M.L.: An object-oriented representation for efficient reinforcement learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 240–247 (2008)
6. Farquhar, G., Gustafson, L., Lin, Z., Whiteson, S., Usunier, N., Synnaeve, G.: Growing action spaces. In: International Conference on Machine Learning. PMLR, pp. 3040–3051 (2020)
7. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
8. Fortunato, M., et al.: Noisy networks for exploration. arXiv preprint arXiv:1706.10295 (2017)
9. Goodfellow, I.J., et al.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, pp. 2672–2680 (2014)
10. Huang, Y., Wu, S., Mu, Z., Long, X., Chu, S., Zhao, G.: A multi-agent reinforcement learning method for swarm robots in space collaborative exploration. In: 2020 6th International Conference on Control, Automation and Robotics (ICCAR), pp. 139–144. IEEE (2020)
11. Iqbal, S., Sha, F.: Actor-attention-critic for multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 2961–2970 (2019)
12. Liu, Y., Hu, Y., Gao, Y., Chen, Y., Fan, C.: Value function transfer for deep multi-agent reinforcement learning based on n-step returns. In: IJCAI, pp. 457–463 (2019)
13. Liu, Y., Wang, W., Hu, Y., Hao, J., Chen, X., Gao, Y.: Multi-agent game abstraction via graph attention neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 7211–7218 (2020)
14. Long, Q., Zhou, Z., Gupta, A., Fang, F., Wu, Y., Wang, X.: Evolutionary population curriculum for scaling multi-agent reinforcement learning. arXiv preprint arXiv:2003.10423 (2020)
15. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv:1706.02275 (2017)
16. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
17. Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M.E., Stone, P.: Curriculum learning for reinforcement learning domains: a framework and survey. J. Mach. Learn. Res. **21**, 1–50 (2020)

18. Omidshafiei, S., et al.: Learning to teach in cooperative multiagent reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6128–6136 (2019)
19. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 4295–4304 (2018)
20. Samvelyan, M., et al.: The starcraft multi-agent challenge. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2186–2188 (2019)
21. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 5887–5896 (2019)
22. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)
23. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems, pp. 1057–1063 (2000)
24. Vezhnevets, A., Wu, Y., Eckstein, M., Leblond, R., Leibo, J.Z.: Options as responses: grounding behavioural hierarchies in multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 9733–9742 (2020)
25. Wang, J., Ren, Z., Liu, T., Yu, Y., Zhang, C.: QPLEX: duplex dueling multi-agent Q-learning. arXiv preprint arXiv:2008.01062 (2020)
26. Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., Zhang, C.: RODE: learning roles to decompose multi-agent tasks. arXiv preprint arXiv:2010.01523 (2020)
27. Wang, W., et al.: From few to more: large-scale dynamic multiagent curriculum learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 7293–7300 (2020)
28. Yang, T., et al.: An efficient transfer learning framework for multiagent reinforcement learning, vol. 34 (2021)
29. Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., Wang, J.: Mean field multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 5571–5580 (2018)