Safe Multi-Agent Reinforcement Learning via Dynamic Shielding

Yunbo Qiu, Yue Jin, Lebin Yu, Jian Wang, Xudong Zhang
Department of Electronic Engineering, Tsinghua University, Beijing, China
{qyb18, jiny16, yulb19}@mails.tsinghua.edu.cn, {jian-wang, zhangxd}@tsinghua.edu.cn

***Abstract*—Improving the safety of policies trained by multi-agent reinforcement learning (MARL) is an essential problem for practical utilization. Traditional methods for safe MARL either fail to improve safety during training process, or require strong prior knowledge about the specific task, such as human intervention, expert policy, and state transition model. However, in practical applications, the safety during training process is also important, and strong prior knowledge of the task is generally inaccessible. In this paper, we propose a novel algorithm Dynamic Shielding for MARL (DS-MARL), which utilizes simple prior knowledge including agents' own motion model to provide a dynamic shield for MARL. DS-MARL aims to improve not only the safety of final policy, but also the safety of training process, without strong prior knowledge. Experimental results show that DS-MARL promotes the safety of both training process and final policy, and also increases success rate of final policy.**

***Index Terms*—safe reinforcement learning, multi-agent system, shielding**

I. Introduction

Multi-agent reinforcement learning (MARL) has achieved great success in many multi-agent applications, such as unmanned aerial vehicles [1], autonomous vehicles [2], and autonomous underwater vehicle [3]. However, traditional MARL algorithms [4], [5] only focus on maximizing agents' expected return, neglecting the safety of their policies, which may cause severe issues in practical applications.

There are three main approaches for safe reinforcement learning (RL): constrained RL methods, safety layer methods, and shielding methods. Constrained RL methods [6], [7] for single-agent cases define a cost function about safety to set constraints for training loss, and thus convert safe RL to constrained RL. Constrained RL methods are also extended to multi-agent cases [8], [9]. However, constrained RL methods can not provide sufficient safety during training process.

Safety layer methods [10], [11] rectify the original action output to the nearest safe action by adding a safety layer to policy functions, and thus provide a safety guarantee. However, prior knowledge about state transition model is required for safe actions but is hard to obtain in most applications.

Shielding methods observe every action generated by an agent's policy, and intervene when the action is considered unsafe by replacing the original action with a safe action. Traditional shielding methods involve judgment of safety and intervention with alternative safe action, which are realized by a human expert [12] or an expert prior policy [13], whereas both are hard to obtain in many applications. Besides, [14] utilizes cost-based advantage functions to provide shielding, which cannot provide sufficient safety guarantee, especially in training process, as shown in Section III. In addition, some shielding methods [15]–[18] use prior knowledge to accurately judge the safety condition after the execution of a certain action. However, the strong prior knowledge is also hard to obtain in many complex applications. For example, in a flocking navigation task [19] shown in Fig. 1(a), each agent has its own policy and can only sense obstacles by rangefinders. Therefore, in such a complex environment, the agent can not accurately judge unsafe conditions including the occurrence of collision with other agents or with obstacles.

In this paper, we aim to improve not only the safety of the resulting MARL policies after training but also those during training. In addition, we expect our algorithm does not require such strong prior knowledge such as state transition model, and thus can be applied in a wider range. We propose a novel algorithm Dynamic Shielding for MARL (DS-MARL), which provides a dynamic shield for MARL using simple safe metrics and a backup policy from agents' own motion model, rather than strong prior knowledge including state transition model or accurate judgment of safety conditions. Judgment of safety and alternation of action are both accomplished by the shield. The shield is dynamically adjusted according to the unsafety rate during training process, which controls the unsafety rate at a low value and gradually reduces the intervention rate as agents' policies become safer and better. We divide the training process into a termination phase and an intervention phase to train better original policies before shielding and provide safer shielding. The main contributions of this paper are listed as follows:

- A novel algorithm DS-MARL is proposed to provide a dynamic shield to improve the safety of MARL in both training and execution without strong prior knowledge.
- We conduct experiments to verify the effectiveness of DS-MARL in increasing safety rate in both training and execution and success rate of trained policies. Besides, we show the flexibility in the choice of fundamental MARL algorithm where DS-MARL builds.

II. Method

We use MARL to solve the Markov Games modeled similarly to [4]. At each time step, the global state is denoted as $\boldsymbol{s}$. Each agent i chooses an action a_i according to its local observation o_i. The joint action $\boldsymbol{a}$ and the joint observation $\boldsymbol{o}$ are the combinations of a_i and o_i of every agent i, respectively.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00222

Agents interact with the environment with $\boldsymbol{a}$ and then each agent i receives a new local observation o_i' and a reward r_i. Each agent i maintains a policy $\pi_i(o_i)$ to maximize the expectation of its return: $R_i = \sum_{t=0}^{\infty} \gamma^t r_{i,t}$, where γ is a discount factor.

A. Formulation for Safety

To improve the safety of MARL algorithms, DS-MARL uses **safety metrics** to judge **unsafety categories**, and a **backup policy** to intervene if necessary.

Unsafety Category. To improve the safety of MARL algorithms in a particular task, first of all, it should be clearly defined what conditions are unsafe for agents in the task. Each unsafety category m includes a state set: $U_m = \{\boldsymbol{s}|$ unsafety m occurs in $\boldsymbol{s}\}$.

For easier understanding, we exemplify unsafety categories $\{U_m\}$ in the flocking control task shown as Fig. 1(a). There are two unsafety categories U_{col} and U_{cross}, representing the state sets where collisions between obstacles and agents and collisions among agents occur, respectively.

Safety metric. For each unsafety category U_m, a safety metric $I_m(o_i, a_i)$ is utilized to roughly measure the unsafety resulting from the action a_i taken by agent i. $I_m(o_i, a_i)$ is a function of local observation and action of agent i, and can be calculated easily with simple prior knowledge including agents' own motion model. By simple prior knowledge, we mean that I_m is not required to be a precise metric to judge safety, but a rough metric as a reference, unlike strong prior knowledge such as state transition model.

For example, in the flocking navigation task, the safety metric for U_{col} about collisions with obstacles is designed as $I_{col}(o_i, a_i) = min_k(d_{obs}(o_i, k) - Pr(u_k, move))$ as Fig. 1(b), where $d_{obs}(o_i, k)$ is the sensed obstacle distance by rangefinder k, $move$ is the displacement of the agent i if a_i is taken, and $Pr(u_k, move)$ represents the projection of the displacement onto the direction of rangefinder k. I_{col} roughly estimates the nearest distance with obstacles if action a_i is taken, without detailed information on obstacles. The safety metric for U_{cross} about collisions with other agents is designed as $I_{cross}(o_i, a_i) = min_j(d_{ag}(o_i, j) - Pr(u_j, move))$ as illustrated in Fig. 1(c), where $d_{ag}(o_i, j)$ is the distance between agent i and j, and $Pr(u_j, move)$ represents the projection of the displacement of agent i onto the line connecting agent i and j. I_{cross} roughly estimates the nearest distance with other agents if action a_i is taken, without consideration of other agents' movement.

Backup Policy. A backup policy π_{back} can generate an alternative action if the original action a_i is considered unsafe. Note that π_{back} only functions as a relatively safe action, rather than an absolutely safe action as in [14] based on strong prior knowledge. Therefore, it is impossible to guarantee absolute safety.

In the example of flocking navigation task, the backup policy π_{back} is a combination of repulsive forces based on artificial potential field approach [20] to attempt to avoid collisions. Repulsive force to avoid U_{col} is $F_{col} = \sum_k(\frac{1}{d_{obs}(o_{i,t},k)} - \frac{1}{d_{col}})\frac{1}{d_{obs}(o_{i,t},k)^2}$, if $d_{obs}(o_{i,t}, k) < d_{col}$, and repulsive force to avoid U_{cross} is $F_{cross} = \sum_j(\frac{1}{d_{ag}(o_{i,t},j)} - \frac{1}{d_{cross}})\frac{1}{d_{ag}(o_{i,t},j)^2}$, if $d_{ag}(o_{i,t}, j) < d_{cross}$, where d_{col} and d_{cross} are thresholds.

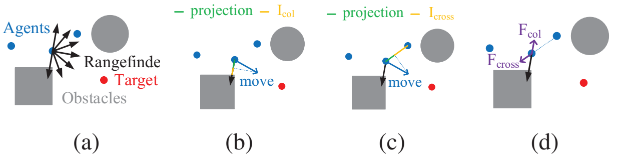

Fig. 1. Simplified diagram of navigation flocking task. (a)The environment of navigation flocking task. Three agents are required to navigate to the target and maintain the flock without collisions. Observation of each agent consists of relative positions with the target and other agents, obstacle distance detected by seven rangefinders, and agent's own velocity. Action of each agent is the applied force. (b)Safety metric I_{col}. (c)Safety metric I_{cross}. (d)F_{col} and F_{cross} for backup policy.

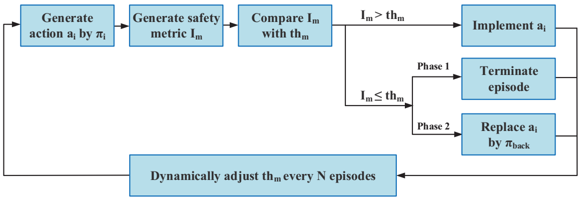
Generate action a_i by π_i
Generate safety metric I_m
Compare I_m with th_m
$I_m > th_m$
Implement a_i
Phase 1
Terminate episode
$I_m \leq th_m$
Phase 2
Replace a_i by π_{back}
Dynamically adjust th_m every N episodes

Fig. 2. Simplified training process of DS-MARL.

B. Dynamic Shielding for MARL

In this paper, we propose a novel algorithm Dynamic Shielding for MARL (DS-MARL) to improve safety of MARL during both training and execution. Besides, DS-MARL only requires simple safety metrics and a backup policy from agents' own motion model, rather than strong prior knowledge about tasks, such as state transition model. The training process of DS-MARL consists of a termination phase and an intervention phase, as shown in Fig. 2.

In the termination phase, after each agent i chooses an action a_i according to its policy $\pi_i(o_i)$, the safety metric $I_m(o_i, a_i)$ is generated for each unsafety category U_m, to be compared with a corresponding threshold th_m. If $I_m > th_m$, then the action a_i is considered safe, and the action a_i can be actually implemented. Otherwise, the action a_i is considered unsafe by current threshold th_m, and then the task in this episode will be terminated to prevent potential unsafety. Note that the measurement of safety is not required to be absolutely accurate, and thus the judgment of safety is not precise as described in Subsection II.A. In addition, the initial value of th_m is expected to be high, and thus agents are almost impossible to encounter unsafety with the initial th_m, at the expense of termination of some tasks where agents are actually safe.

By the comparison of safety metric I_m and corresponding threshold th_m, DS-MARL restricts agents' action space under a certain observation, and thus enhances safety of agents. With this shield, agents with randomly initialized policies are supposed to learn preliminary behaviors about how to succeed in the task. However, a large part of action space is forbidden by the shield with the initial th_m, which limits the policy performance. Therefore, with the progress of training and improvement of agents' policies, the protection of the shield is expected to be loosened and then agents are allowed to take more actions under certain observation.

To control the procedure of loosening the shield, we calculate $N_{unsafe,m}$ to count the occurrence of unsafety belonging to each category U_m in every N episodes. If the unsafety conditions in U_m occur rarely, agents are allowed to take actions in a larger action space under a smaller threshold to further elevate their performance. If the unsafety conditions in U_m occur frequently, the training agent policies are not good enough to act safely with the increased action space, and then the action space is decreased under a bigger threshold to protect agents. The detailed adjustment of threshold th_m is:

$$th_m = \begin{cases} th_m - \delta, & \text{if } \dfrac{N_{unsafe,m}}{N} \leq Th_{phase1,lower}, \\ th_m + \delta, & \text{if } \dfrac{N_{unsafe,m}}{N} > Th_{phase1,upper}, \\ th_m, & \text{otherwise}, \end{cases} \tag{1}$$

where δ, $Th_{phase1,lower}$ and $Th_{phase1,upper}$ are three hyperparameters. $Th_{phase1,upper}$ is set to be larger than $Th_{phase1,lower}$ to avoid frequent adjustment of thresholds and allow agents to gradually improve their performance when the action space is increased.

In the second phase, known as the intervention phase, safety metric $I_m(o_i, a_i)$ is also generated after each agent i chooses its action. However, if $I_m \leq th_m$, which means the action is considered unsafe by the current threshold th_m, the task will not terminate as in the termination phase. Instead, the agent will use the alternative action generated by the backup policy π_{back}, while others whose actions are considered safe still implement their original actions. The reason why the backup policy is only used in the second phase of training is that randomly initialized agents without preliminary training are likely to depend on the intervention of the backup policy, which hinders agents from learning their own policies. The shield controlled by th_m is dynamically adjusted in the same way as (1), except that hyperparameters $Th_{phase1,lower}$ and $Th_{phase1,upper}$ in (1) are replaced by $Th_{phase2,lower}$ and $Th_{phase2,upper}$ that are lower than their corresponding ones to further improve safety.

III. Experiments

A. Main Experiments

To verify our algorithm DS-MARL, we conduct experiments in the flocking navigation task adapted from [19]. We build our algorithm DS-MARL on a representative MARL algorithm MADDPG [4] to form DS-MADDPG. DS-MADDPG is contrasted with two baseline algorithms: MADDPG and SAILR-MADDPG. SAILR-MADDPG is designed for multi-agent cases based on MADDPG, which utilizes cost-based advantage functions [14] to provide shielding for safety. SAILR-MADDPG only requires a backup policy rather than strong prior knowledge as in [12], [13], [15]–[18], and thus we choose it as a baseline algorithm. Initial value of th_{col} and th_{cross} are 0.8 and 0.5. Hyperparameters δ, N, $Th_{phase1,lower}$, $Th_{phase1,upper}$, $Th_{phase2,lower}$, and $Th_{phase2,upper}$ are 0.05, 500, 0.03, 0.06, 0.015, and 0.03, respectively. Episodes of termination phase and intervention phase are both 50000. Each algorithm is run in 3 seeds.

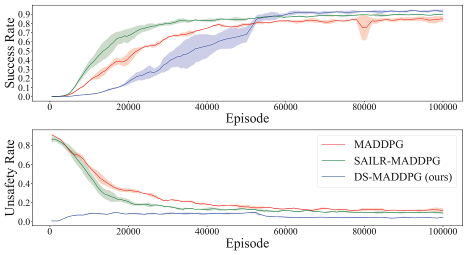

Fig. 3. Convergence curves of success rate and unsafety rate of main experiments. The first half of DS-MADDPG is termination phase, while the rest is intervention phase.

TABLE I
Statistical Results

Algorithm	**Final Success Rate**	**Final Unsafety Rate**	**Total Unsafety Rate**
MADDPG	0.849	0.122	0.246
SAILR-MADDPG	0.902	0.091	0.198
DS-MADDPG (ours)	0.935	0.043	0.064
only phase 1	0.853	0.088	0.084
only phase 2	0.006	0.375	0.201
MATD3	0.781	0.206	0.361
SAILR-MATD3	0.858	0.139	0.243
DS-MATD3 (ours)	0.911	0.070	0.079

We plot convergence curves of success rate and unsafety rate during training process in Fig. 3. It shows that after training, success rate of DS-MADDPG is higher than MADDPG and SAILR-MADDPG, and unsafety rate of DS-MADDPG is lower than baseline algorithms. Besides, unsafety rate of DS-MADDPG is always low during training and total unsafety rate of DS-MADDPG is largely reduced in contrast to MADDPG and SAILR-MADDPG. In conclusion, DS-MADDPG decreases unsafety rate during both training and execution process, and increases success rate. Detailed statistical results of all the experiments are shown in Table I.

B. Ablation Experiments

To validate the effectiveness of each phase in DS-MARL, we design two ablation algorithms that only involve the termination phase and the intervention phase, respectively. Convergence curves of success rate and unsafety rate are plotted in Fig. 4. The performance of the ablation algorithm with only the intervention phase is largely inferior to DS-

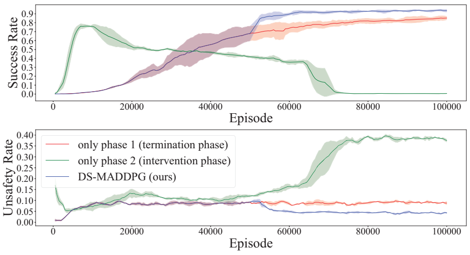

Fig. 4. Convergence curves of success rate and unsafety rate of ablation experiments.

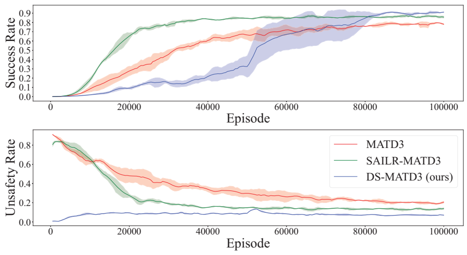

Fig. 5. Convergence curves of success rate and unsafety rate of experiments with MATD3 as the fundamental MARL.

MADDPG, since randomly initialized agents are likely to depend on the backup policy and fail to learn their own policies, as mentioned in Section II. DS-MARL is better than the ablation algorithm with only the termination phase, with the help of the backup policy to provide a better shield. The ablation experiments confirm the effectiveness of the two phases.

C. Experiments with Another Fundamental MARL

We build our DS-MARL on MADDPG in previous experiments, whereas DS-MARL can be built on any MARL algorithm. In this subsection, we take MATD3 [5] as the fundamental MARL algorithm. SAILR-MATD3 built on MATD3 is designed as another baseline algorithm. Convergence curves of success rate and unsafety rate are plotted in Fig. 5. Similar to main experiments, DS-MATD3 also increases success rate and decreases unsafety rate, and DS-MATD3 maintains a low unsafety rate during training in comparison with baseline algorithms.

IV. Conclusion

We propose a novel algorithm DS-MARL to provide safety for MARL algorithms with a dynamic shield consisting of two phases. DS-MARL provides safety not only during execution but also during training. Besides, DS-MARL does not require strong prior knowledge, such as human experts, expert prior policies, or state transition model. Experimental results validate that DS-MARL can improve safety during both training and execution, and promote success rate.

References

[1] Y. Qiu, Y. Jin, L. Yu, J. Wang, and X. Zhang, “Promoting cooperation in multi-agent reinforcement learning via mutual help,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[2] S. Bhalla, S. Ganapathi Subramanian, and M. Crowley, “Deep multi agent reinforcement learning for autonomous driving,” in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 67–78.

[3] M. Rahmati, M. Nadeem, V. Sadhu, and D. Pompili, “Uw-marl: Multi-agent reinforcement learning for underwater adaptive sampling using autonomous vehicles,” in *Proceedings of the International Conference on Underwater Networks & Systems*, 2019, pp. 1–5.

[4] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6382–6393.

[5] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, “Reducing overestimation bias in multi-agent domains using double centralized critics,” in *NeurIPS Workshop on Deep RL*, 2019.

[6] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[7] A. Stooke, J. Achiam, and P. Abbeel, “Responsive safety in reinforcement learning by pid lagrangian methods,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 9133–9143.

[8] S. Lu, K. Zhang, T. Chen, T. Başar, and L. Horesh, “Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 8767–8775.

[9] C. Liu, N. Geng, V. Aggarwal, T. Lan, Y. Yang, and M. Xu, “Cmix: Deep multi-agent reinforcement learning with peak and average constraints,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 157–173.

[10] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, “Safe exploration in continuous action spaces,” *arXiv preprint arXiv:1801.08757*, 2018.

[11] Z. Sheebaelhamd, K. Zisis, A. Nisioti, D. Gkouletsos, D. Pavllo, and J. Kohler, “Safe deep reinforcement learning for multi-agent systems with continuous action spaces,” in *ICML Workshop on Reinforcement Learning for Real Life*, 2021.

[12] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, “Trial without error: Towards safe reinforcement learning via human intervention,” in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2067–2069.

[13] F. Wang, B. Zhou, K. Chen, T. Fan, X. Zhang, J. Li, H. Tian, and J. Pan, “Intervention aided reinforcement learning for safe and practical policy optimization in navigation,” in *Conference on Robot Learning*. PMLR, 2018, pp. 410–421.

[14] N. C. Wagener, B. Boots, and C.-A. Cheng, “Safe reinforcement learning using advantage-based intervention,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 630–10 640.

[15] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[16] T.-Y. Yang, T. Zhang, L. Luu, S. Ha, J. Tan, and W. Yu, “Safe reinforcement learning for legged locomotion,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2454–2461.

[17] O. Bastani, S. Li, and A. Xu, “Safe reinforcement learning via statistical model predictive shielding.” in *Robotics: Science and Systems*, 2021.

[18] I. ElSayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng, “Safe multi-agent reinforcement learning via shielding,” in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 2021, pp. 483–491.

[19] Y. Qiu, Y. Jin, L. Yu, J. Wang, Y. Wang, and X. Zhang, “Improving sample efficiency of multi-agent reinforcement learning with non-expert policy for flocking control,” *IEEE Internet of Things Journal*, 2023.

[20] M. G. Park, J. H. Jeon, and M. C. Lee, “Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing,” in *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, vol. 3. IEEE, 2001, pp. 1530–1535.