MA-MIX: Value Function Decomposition for Cooperative Multiagent Reinforcement Learning Based on Multi-Head Attention Mechanism

Extended Abstract

Yu Niu
Department of Computer Science,
Inner Mongolia University,
Hohhot, China
32109170@mail.imu.edu.cn

Hengxu Zhao
Department of Computer Science,
Inner Mongolia University,
Hohhot, China
1835935173@qq.com

Lei Yu*
Department of Computer Science,
Inner Mongolia University,
Hohhot, China
yuleiimu@163.com

ABSTRACT

Multi-Agent Deep Reinforcement Learning (MADRL) is a research field that combines deep learning and multi-agent reinforcement learning. In complex tasks, a single agent often finds it difficult to complete the task independently, thus requiring cooperation and communication between agents. However, communication between agents remains a key issue in multi-agent cooperative reinforcement learning. To address this issue, we propose a new method called Multi-Head Attention Mixing Network (MA-MIX), which aims to solve key challenges in multi-agent systems. MA-MIX is based on the multi-head attention mechanism and innovatively applied to agent networks, effectively solving the problem of information exchange and cooperation in multi-agent systems. We compared MA-MIX with traditional QMIX algorithms and other baseline algorithms. The experimental results show that MA-MIX has superior performance under the StarCraft Multi-Agent Challenge (SMAC) environment.

KEYWORDS

Multi-Agent Reinforcement Learning; Multi-Head Attention Mechanism; SMAC

ACM Reference Format:
Yu Niu, Hengxu Zhao, and Lei Yu. 2024. MA-MIX: Value Function Decomposition for Cooperative Multiagent Reinforcement Learning Based on Multi-Head Attention Mechanism: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Multi-Agent Reinforcement Learning (MARL)[6, 17] is a learning method that coordinates agents to cooperate in completing a common goal in complex tasks involving multiple agents. In traditional reinforcement learning, agents are often viewed as independent decision-makers, ignoring the interactions between agents. However, multi-agent deep reinforcement learning has great potential in solving many real-world problems, such as speech recognition[8, 13], coordination of robot swarms[5], autonomous driving[1, 4], group decision-making[9], natural language processing[21], and intelligent control[2]. Due to the actual communication constraints and the huge joint action space brought about by the increase in the number of agents, existing MARL algorithms usually use one or more centralized functions to learn the contribution of the actions chosen by the agents to the team goal, and the centralized function optimizes the parameters of the agents according to the global team reward. This method is called the Centralized Training and Decentralized Execution (CTDE) paradigm[3, 10]. For example, Value Decomposition Networks VDN[15], QMIX[11], QTRAN[14], Qatten[20] and QPLEX[18] are the main methods of this CTDE MARL. They have outstanding performance in many MARL tasks, such as StarCraft Multi-Agent Challenge (SMAC)[12].

However, most existing methods usually only consider the effect of a single agent on the entire multi-agent system, ignoring the mutual influence between agents. To address this issue, we propose a method based on the multi-head attention mechanism, MA-MIX. It allows each agent to selectively pay attention to the actions of other agents through the attention mechanism, in order to learn the optimal strategy by choosing actions that are more favorable to itself and other agents. We evaluated MA-MIX in the StarCraft Multi-Agent Challenge (SMAC) scenario and showed that MA-MIX outperforms QMIX and other baselines in various scenarios.

2 METHOD

In this section, we propose a novel network structure based on value decomposition, called MA-MIX. It consists of three main network structures: (1) an improved agent network based on the multi-head attention mechanism, (2) a mixing network, and (3) a weight network.

Figure 1(a) shows the structure of the mixing network and the weight network. The mixing network is a regular feed-forward neural network that takes the Q_i output from the agent network as input, performs a complex non-linear mixing, and produces the final Q_{tot}. The role of the weight network is to transform the global state S_t and the hidden state h_t^a of each agent into vectors of the same dimension, and then concatenate them to form the input. This process allows the weight network to consider both global and local information, enhancing the expressive power of the weight network. Each weight network first processes the input data using a fully connected linear layer, then applies a ReLU activation function,

*Corresponding author.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

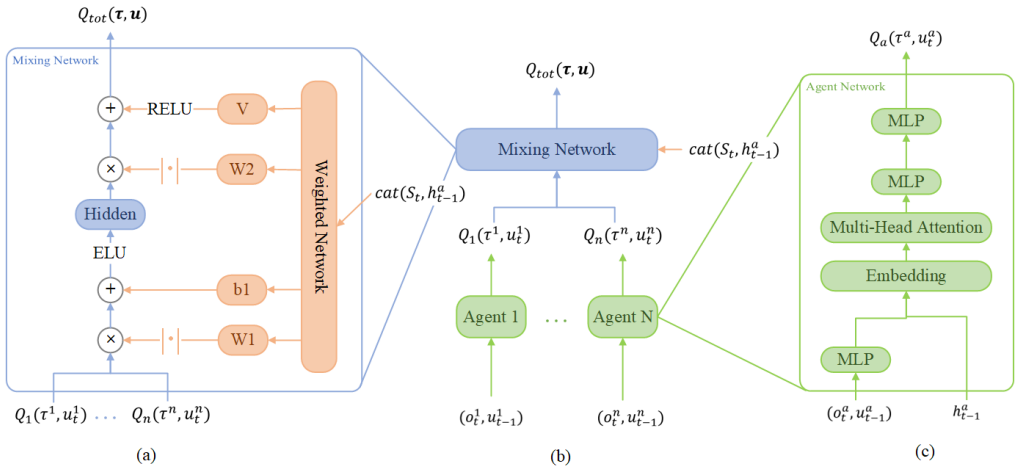

Figure 1: (a) Mixing network structure, shown in blue. Orange is the weighted network that generates the weights and biases of the mixing network. (b) The overall structure of MA-MIX. (c) Agent network structure, shown in green.

and finally uses another fully connected linear layer to calculate the weight value W of the agent. Finally, these weight values are used to mix each agent's local Q-value in the form of equation (1), generating the final joint action Q-value.

$$Q_{tot} = \left(Q_i^{(1\times n)}\mathbf{W_1}^{(n\times e)} + \mathbf{b_1}^{(1\times e)}\right)\mathbf{W_2}^{(e\times 1)} + V^{(1\times 1)} \tag{1}$$

Figure 1(c) presents the network structure of the agent. This network adopts an innovative approach, introducing the multi-head attention mechanism into the estimation of the agent's value function, significantly enhancing performance. Unlike traditional DRQNs, the multi-head attention mechanism has stronger expressive power, sequence modeling ability, and information aggregation ability, which shows excellent performance improvement when dealing with complex tasks. First, the input data (o_t^a, u_{t-1}^a) is processed through a fully connected layer and reshaping operation, then concatenated with the previous hidden state h_{t-1}^a. Next, the concatenated result is passed into our defined multi-head attention layer for attention calculation. Finally, the output goes through a multi-layer fully connected layer for feature extraction and transformation, generating the agent's action value function Q_a.

We train MA-MIX end-to-end. In order to find the optimal joint action-value function $Q^*(s, \boldsymbol{u}) = r(s, \boldsymbol{u}) + \gamma \mathbb{E}_{s'}[\max_{\boldsymbol{u}'} Q^*(s', \boldsymbol{u}')]$, we use the Adam optimizer and Q-Learning[19] with a deep neural network parameterized by θ[16] in MA-MIX to minimize the following loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\tau,u,r,\tau')\in D}[\left(r + \gamma V(\tau'; \theta^-) - Q(\tau, u; \theta)\right)^2]. \tag{2}$$

3 EXPERIMENTAL AND RESULTS

3.1 Comparison

Table 1 show the results of all methods in six tasks in SMAC.First, we tested MA-MIX in MMM and 2s3z simple scenarios. Encouragingly, we observed that most algorithms can achieve very high win rates in these two scenarios.However, the performance of QTran was not satisfactory, possibly because the actual relaxation might hinder the accuracy of its updates[7].Next, we tested MA-MIX in

Table 1: The average test win rate on SMAC maps. We trained for 2M time steps on each map

Maps	MA-MIX	QTran	Qatten	QMIX	QPlex
MMM	**100%**	31.2%	96.8%	100%	100%
2s3z	**99.08%**	84.3%	94.3%	99.45%	94.75%
5m_vs_6m	**77.80%**	5%	32%	61.70%	51.80%
8m_vs_9m	**100%**	0%	78%	90%	84%
MMM2	**95%**	0%	59.3%	60%	0%
corridor	**19.75%**	0%	0%	0%	0%

5mvs6m and 8mvs9m difficult scenarios. These scenarios represent different challenges, and existing methods fail to achieve consistent performance, while MA-MIX outperforms the rest of the algorithms significantly. Finally, we tested MA-MIX in MMM2 and corridor super-hard scenarios. Due to the increased difficulty and complexity, all methods performed poorly. We observed that in the corridor, while only the MA-MIX algorithm reached a win rate of about 20% after 2 million steps of training. In the MMM2 map, MA-MIX achieved a win rate of 70%. This result demonstrates the superiority of MA-MIX over QMIX and QPlex in handling super-hard SMAC scenarios.

4 CONCLUSION

This paper introduces a value decomposition hybrid network MA-MIX for cooperative MARL tasks. The core idea of MA-MIX is to enable each agent to pay attention to the information of other agents and cooperate and communicate with them in the process of generating Q-values. Through the multi-head attention mechanism, agents can learn more comprehensive and accurate strategies, thus significantly improving the cooperation performance and the overall performance of the system.

REFERENCES

[1] Y. Cao, W. Yu, W. Ren, and G. Chen. 2013. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Transactions on Industrial Informatics* 9, 1 (2013), 427–438.

[2] Ignacio Carlucho, Mariano De Paula, and Gerardo G Acosta. 2020. An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots. *ISA transactions* 102 (2020), 280–294.

[3] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[4] Yeping Hu, Alireza Nakhaei, Masayoshi Tomizuka, and Kikuo Fujimura. 2019. Interaction-aware decision making with adaptive strategies under merging scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 151–158.

[5] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. 2017. Guided Deep Reinforcement Learning for Swarm Systems. (2017).

[6] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*. PMLR, 3040–3049.

[7] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. Maven: Multi-agent variational exploration. *Advances in neural information processing systems* 32 (2019).

[8] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. 2016. Deep reinforcement learning: an overview. In *Proceedings of SAI Intelligent Systems Conference*, Vol. 2. Springer, 426–440.

[9] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. 2020. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics* 50, 9 (2020), 3826–3839.

[10] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research* 32 (2008), 289–353.

[11] Tabish Rashid, Mikayel Samvelyan, Chris Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. 4295–4304.

[12] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[13] Yih-Liang Shen, Chao-Yuan Huang, Syu-Siang Wang, Yu Tsao, Hsin-Min Wang, and Tai-Shih Chi. 2019. Reinforcement learning based speech enhancement for robust speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6750–6754.

[14] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

[15] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Conference on Autonomous Agents and Multiagent Systems*, Vol. 3. 2085–2087.

[16] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[17] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[18] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*. Virtual Event, Austria.

[19] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8 (1992), 279–292.

[20] Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. 2020. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939* (2020).

[21] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine* 13, 3 (2018), 55–75.