Sequential Order Adjustment of Action Decisions for Multi-Agent Transformer (Student Abstract)

Shota Takayama, Katsuhide Fujita

Tokyo University of Agriculture and Technology
takayama@katfuji.lab.tuat.ac.jp, katfuji@cc.tuat.ac.jp

Abstract

Multi-agent reinforcement learning (MARL) trains multiple agents in shared environments. Recently, MARL models have significantly improved performance by leveraging sequential decision-making processes. Although these models can enhance performance, they do not explicitly consider the importance of the order in which agents make decisions. We propose AOAD-MAT, a novel model incorporating action decision sequence into learning. AOAD-MAT uses a Transformer-based actor-critic architecture to dynamically adjust agent action order. It introduces a subtask predicting the next agent to act, integrated into a PPO-based loss function. Experiments on StarCraft Multi-Agent Challenge and Multi-Agent MuJoCo benchmarks show AOAD-MAT outperforms existing models, demonstrating the effectiveness of adjusting agent order in MARL.

Introduction

Multi-agent reinforcement learning (MARL) is widely used to solve complex cooperative tasks in areas such as traffic control, robotics, and games(Albrecht, Christianos, and Schäfer 2024; Lowe et al. 2017; Jaques et al. 2019; Lazaridou, Peysakhovich, and Baroni 2017). However, MARL presents challenges not seen in single-agent settings, such as the non-stationarity of the environment and the exponential growth of the joint action space with more agents.

A significant breakthrough in MARL was achieved with the introduction of the Multi-Agent Transformer (MAT) (Wen et al. 2022), which leveraged the power of sequence modeling techniques to enhance MARL performance. By treating the multi-agent decision-making process as a sequence generation task, the MAT model could effectively capture inter-agent dependencies and improve overall team performance. Although these approaches have shown promising results, they have not explicitly considered the effectiveness of the order in which agents make decisions. The order of action decisions can significantly influence the overall performance and stability of MARL systems.

To address this issue, we propose an Agent Order of Action Decisions-MAT (AOAD-MAT), a novel Transformer-based multi-agent deep reinforcement learning model that explicitly incorporates and learns the optimal order of agent action decisions. The architecture of the proposed AOAD-MAT model is inspired by the successful MAT architecture and introduces a dedicated mechanism to predict and optimize the sequence in which agents should act.

AOAD-MAT: Agent Order of Action Decisions-MAT

Sequential Action Decision Order Prediction We propose a sequential action decision order prediction system that predicts the order in which agents take actions. The first agent is predetermined, and subsequent agents are predicted by a learner integrated into the decoder. This learner shares parameters with the action prediction component and operates as a branching subtask. Figure 1 shows this process, illustrating how the system predicts the next agent to act and determines the action decision order.

Architecture The encoder architecture and loss function are unchanged from those of the prior MAT method(Wen et al. 2022). For the decoder, we introduce a sequential action decision-order prediction system and design a loss function that includes subtasks. Since the decoder (actor network) performs multitask learning, including predicting the next agent to act, it is necessary to design a loss function that converges both action and next-action agent predictions. We define the state $s_t^m = (\hat{o}_t^{\hat{i}_{1:m}}, \hat{a}_t^{\hat{i}_{1:m-1}})$ to represent the observation and action history up to the m-th agent at time t. This state serves as the input to the m-th step of the decoder. By incorporating both the joint observation $\hat{o}_t^{\hat{i}_{1:m}}$ and joint action $\hat{a}_t^{\hat{i}_{1:m-1}}$, The actor network outputs two probability distributions to act the following roles:

- $\pi_a^m(\theta)$: Action prediction distribution. It approximates the action $a^{\hat{i}_m}$ of agent $\hat{i}_m$, which is the current agent making a decision.
- $\pi_i^m(\theta)$: Next agent prediction distribution. It approximates $\sigma(i_{m+1})$, which determines the next agent to act in the sequence.

In other words, $\pi_a(\theta)$ predicts the optimal action for the current agent, whereas $\pi_i(\theta)$ predicts which agent should act next according to the permutation σ.

When predicting the policy function in Proximal Policy Optimization (PPO)(Schulman et al. 2017), the ratio of

Benchmark	Task	Difficulty	AOAD-MAT	MAT-adjust	MAT	Steps
SMAC	5m_vs_6m	Hard	**100.0**(1.4)	**100.0**(1.4)	96.9(1.4)	1×10^8
	MMM2	Hard+	**100.0**(1.4)	**100.0**(1.4)	**100.0**(1.4)	5×10^7
	6h_vs_8z	Hard+	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	1.5×10^7
	3s5z vs 3a6z	Hard+	**100.0**(1.7)	96.9(3.4)	**100.0**(0.0)	5×10^7
MA-MuJoCo	HalfCheetah (6×1)	Expert	**13686**(1286)	12778(1090)	11970(256)	1×10^8

Table 1: Median and standard deviation of win rates in SMAC and mean and standard deviation of average episode rewards in MuJoCo and the number of execution steps in evaluation phase

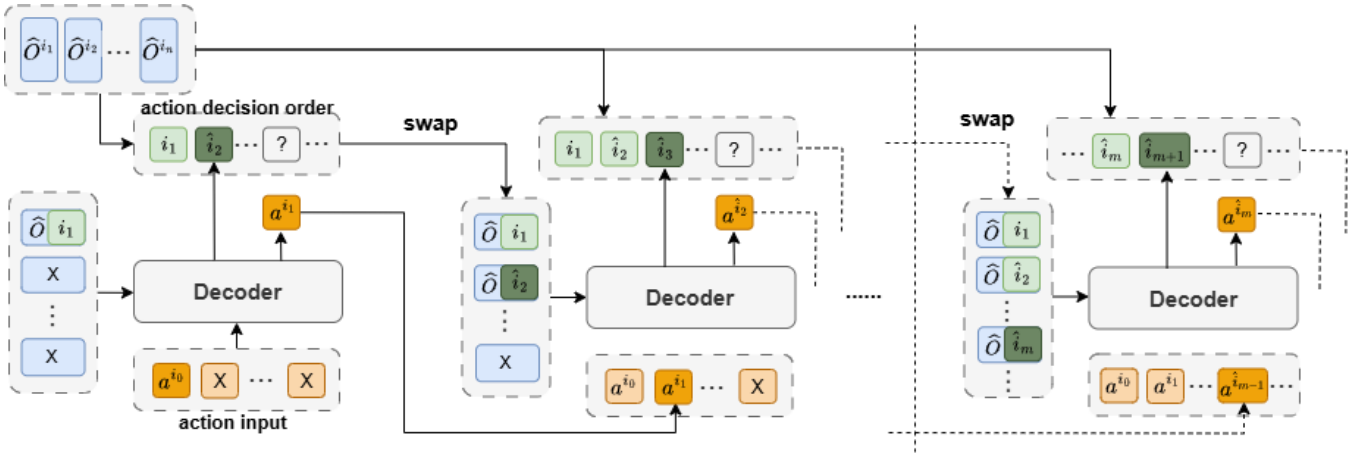

Figure 1: Sequential Action Decision Order Prediction: First, a dummy action a^{i_0} is input as the 0th action. At this point, it is necessary to predetermine i_1 the agent number that will act first. Then, output $\hat{i}_2$ the agent that will act next. Based on the updated agent action decision order, the latent representations of the observations $\hat{o}$, which are the output of the encoder, are swapped. Finally, the previous action output is added to the input and the same steps are repeated.

probability distributions before and after the update is used. Considering the ratio $r_a^m(\theta)$ of $\pi_a^m(\theta)$ and the ratio $r_i^m(\theta)$ of $\pi_i^m(\theta)$, the product of them is calculated to obtain the overall ratio $r^m(\theta)$, which is then used in the advantage function. This approach differs from traditional multitask learning, where a weighted sum of loss functions is typically used.

Experiment

Our experiments aim to evaluate the performance of the proposed AOAD-MAT in SMAC(Samvelyan et al. 2019) and MA-MuJoCo(Peng et al. 2021) with discrete and continuous action spaces, respectively. These experiments demonstrate the performance and the effectiveness of predicting the order of action decisions under different MARL scenarios.

As Baselines, we compare the proposed AOAD-MAT model with the traditional MAT, MAT-adjust with PPO clip and PPO-epoch tuned.

Experimental Result

We compare the proposed AOAD-MAT with MAT and MAT-adjust (with reduced PPO clip value) on four SMAC scenarios: 5m_vs_6m, 6h_vs_8z, MMM2, and 3s5z_vs_3s6z.

In the MA-MuJoCo tasks, we focus on the HalfCheetah (6×1) task, which requires controlling six joints of the HalfCheetah robot to maximize its forward velocity. Table1 demonstrates that the proposed AOAD-MAT achieves approximately 10% improvement in median reward compared to baselines with a substantial increase in the 95% confidence interval's upper bound. In other words, the proposed AOAD-MAT achieves higher peak performance than the baseline methods. Since there is no upper limit on the aspect of velocity being compared, it becomes clear that predicting the action decision order significantly affects performance. In addition, AOAD- MAT's adaptive action order becomes particularly effective in later stages of learning due to its ability to navigate the vast exploration space.

Conclusion

In this paper, we proposed the AOAD-MAT model that predicts the action decision order of agents to facilitate sequential learning. The proposed AOAD-MAT explicitly incorporated action decision sequences into its learning process, allowing the model to learn and predict the optimal order of agent actions based on Transformer-based architecture. We conducted several experiments on the SMAC and MA-MuJoCo benchmarks to evaluate the effectiveness of the proposed method. The experimental results showed that the proposed AOAD-MAT outperformed existing MAT and other baseline methods. We highlighted the effectiveness of adjusting the agent order of action decisions order in MARL.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 22H03641, and JST FOREST (Fusion Oriented REsearch for Disruptive Science and Technology) Grant Number JPMJFR216S.

References

Albrecht, S. V.; Christianos, F.; and Schäfer, L. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press.

Jaques, N.; Lazaridou, A.; Hughes, E.; Gulcehre, C.; Ortega, P.; Strouse, D.; Leibo, J. Z.; and De Freitas, N. 2019. Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 3040–3049. PMLR.

Lazaridou, A.; Peysakhovich, A.; and Baroni, M. 2017. Multi-Agent Cooperation and the Emergence of (Natural) Language. In *International Conference on Learning Representations*.

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

Peng, B.; Rashid, T.; Schroeder de Witt, C.; Kamienny, P.-A.; Torr, P.; Böhmer, W.; and Whiteson, S. 2021. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34: 12208–12221.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.-M.; Torr, P. H. S.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Wen, M.; Kuba, J.; Lin, R.; Zhang, W.; Wen, Y.; Wang, J.; and Yang, Y. 2022. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35: 16509–16521.