TIMAT: Temporal Information Multi-Agent Transformer

Extended Abstract

Qitong Kang
Nankai University
Tianjin, China
2120220464@mail.nankai.edu.cn

Fuyong Wang
Nankai University
Tianjin, China
wangfy@nankai.edu.cn

Zhongxin Liu
Nankai University
Tianjin, China
lzhx@nankai.edu.cn

Zengqiang Chen
Nankai University
Tianjin, China
chenzq@nankai.edu.cn

ABSTRACT

In many specific tasks, training models with Multi-Agent Reinforcement Learning (MARL) to solve a task often leads to overfitting to the training environment. When dealing with multi-task, models specialized for a single task often fail to generalize, and retraining models often implies the consumption of computational resources. Therefore, it is necessary to establish a pre-trained model that can be quickly deployed in an online environment. Therefore, we propose temporal information multi-agent transformer (TIMAT) based on the transformer that extracts temporal information and models MARL as Sequence Models (SM). The advantage of this framework is that it can handle time information of arbitrary length and any number of agents regardless of the type, which greatly enhances the generalization ability of the model.

KEYWORDS

Reinforcement Learning, Transformer, Multiagent Systems

ACM Reference Format:
Qitong Kang, Fuyong Wang, Zhongxin Liu, and Zengqiang Chen. 2024. TIMAT: Temporal Information Multi-Agent Transformer: 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

MARL methods have demonstrated excellent performance in single tasks within simulated environment s[4, 6, 11]. However, bridging the gap between these methods and real-world applications still poses significant challenges. The lack of sufficient generalization ability in models when faced with variations in observations, states, and the number of agents across different tasks often necessitates retraining the model to adapt to new tasks [3, 5, 9].

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).

2 METHOD

The TIMAT framework is built using a transformer based on self-attention mechanisms [8]. As shown in the Figure 1, We combine the research results of sequence models with Multi-Agent Advantage Decomposition theorem [10], using Obs Encoder and Action Decoder as the universal parts of TIMAT to map the agent's observations into a sequence output of the agent's actions. Under such a framework, TIMAT can adapt to different observations inputs and different numbers of agents in various tasks, and it utilizes historical information, enhancing the model's generalization performance.

TIMAT consists of two parts: 1) offline TIMAT, and 2) online TIMAT. Compared to offline TIMAT, online TIMAT incorporates Critic Block and combines the global state s as an additional input with the output from Obs Encoder. This benefits us in quickly adapting to online tasks and more accurately evaluating state functions using MARL algorithms.

Obs block demonstrated in Figure 2 takes an observation sequence $o^{i_m}_{t-c:t}$ of arbitrary length c as input and uses self-attention mechanisms to represent the relevance of observations at different moments by computing the weight matrix softmax $\left(QK^T/\sqrt{D}\right)$ [8]. When this matrix is multiplied with V, it yields the weighted values for each observation moment. There's no need to compute values beyond the current moment, so only the last dimension of the output is extracted to gather past historical information.

During the offline training phase, offline TIMAT can only access information from the past c steps, and uses the cross-entropy loss function to map the observation sequence $\left\{o^{i_{1:n}}_1, \cdots o^{i_{1:n}}_t\right\}$, onto the actions A stored in dataset.

During the online training phase, the refined offline TIMMAT will be deployed as the pre-trained model to the online TIMAT. We use HAPPO based on the Actor-Critic method to train the online TIMAT because it, along with TIMAT, both employs multi-agent advantage decomposition theorem, ensuring the algorithm's monotonic improvement property when agents execute actions in a certain order [1, 2]. Each agent m first processes its own historical information $o^{i_m}_{t-c:t}$ through Obs Block. Under the effect of masked attention, Action Decoder uses the mixed observation $\left\{\tilde{o}^{i_1}_t, \cdots, \tilde{o}^{i_n}_t\right\}$ as queries to compute the relevance with the input

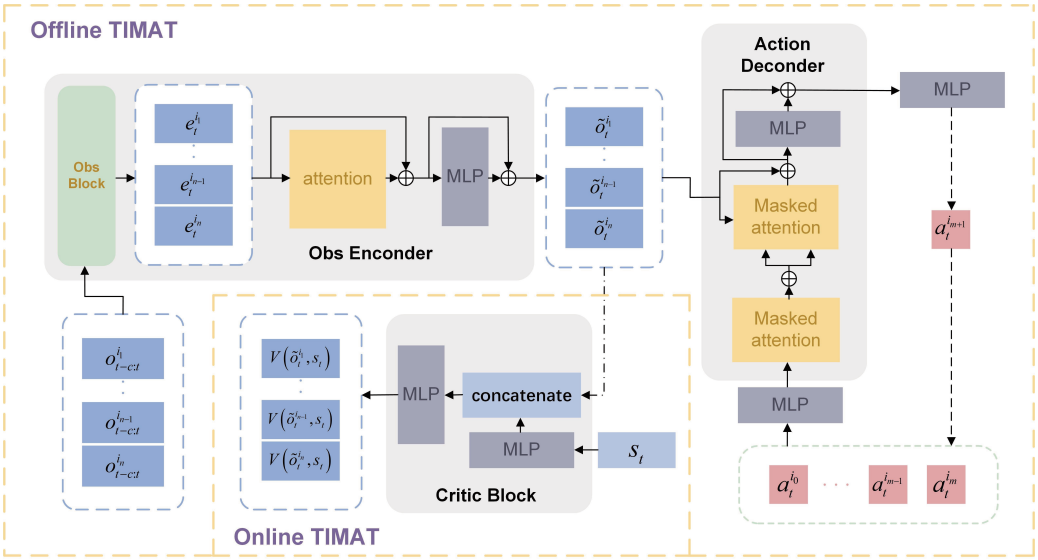

Figure 1: The architecture of TIMAT, taking observations of length c as input to obtain autoregressive output actions.

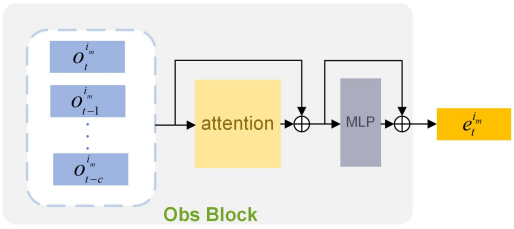

Figure 2: Observation Block:The processing procedure of observations for the agent m.

actions $\left\{a_t^{i_1}, \cdots, a_t^{i_{m-1}}\right\}$ and the agent m can only consider the actions of the previous $m-1$ agents when auto-regressively output the action $a_t^{i_m}$.

3 EXPERIMENTS

We focus on StarCraft Multi-Agent Challenge (SMAC) [7], a widely-used experimental environment in MARL that encompasses various types of tasks and constructs an offline dataset used in this environment from a well-trained MARL algorithm.

The experimental results show that the offline TIMAT framework demonstrates excellent generalization capabilities even when it is only used to model a handful of simple tasks. It exhibits a strong ability to generalize to unseen and challenging tasks, indicating that the model has improved the usage efficiency of offline datasets, which will accelerate the speed of online training.

Online TIMAT was compared with TIMAT without deploying a pre-trained model (no deploy) and two baseline methods (MAT and MAPPO). As shown in Figure 3, after a limited number of training steps, online TIMAT demonstrated a faster convergence rate and achieved a higher win rate across various task difficulties. Furthermore, we also observed that TIMAT, without pre-training,

Task		Reward
Easy	3m	15.2353 (±1.4925)
	8m	17.2078 (○) (±1.0413)
	2s3z	12.7336 (±1.3182)
	MMM	18.5335 (○) (±1.4665)
Hard	1c3s5z	11.3699 (±1.2569)
	10m vs 11m	10.0287 (±0.7369)
	3s5z	18.8785 (○) (±1.1215)
Super Hard	MMM2	6.1066 (±1.3908)
	6h vs 8z	8.2465 (±0.8413)
	3s5z vs 3s6z	10.5110 (±0.4728)

Table 1: Offline: The average reward corresponding to the model, with a maximum reward of 20. The symbol ○ represents the source task.

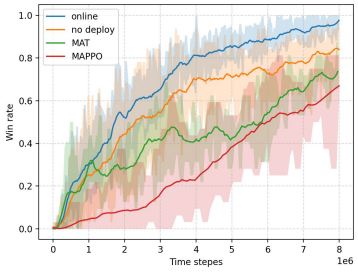

Figure 3: Performance comparisons on 5m vs 6m in SMAC.

outperformed other baseline algorithms in many tasks. This indicates that TIMAT possesses an efficient capability for temporal information processing.

REFERENCES

[1] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2021. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251* (2021).

[2] Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Haifeng Zhang, David Mguni, Jun Wang, Yaodong Yang, et al. 2021. Settling the variance of multi-agent policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 13458–13470.

[3] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[4] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[5] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*. PMLR, 2681–2690.

[6] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research* 21, 1 (2020), 7234–7284.

[7] 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).

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[9] Rundong Wang, Weixuan Wang, Xianhan Zeng, Liang Wang, Zhenjie Lian, Yiming Gao, Feiyu Liu, Siqin Li, Xianliang Wang, QIANG FU, et al. 2023. Multi-Agent Multi-Game Entity Transformer. (2023).

[10] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems* 35 (2022), 16509–16521.

[11] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems* 35 (2022), 24611–24624.