Addressing Permutation Challenges in Multi-Agent Reinforcement Learning

Extended Abstract

Somnath Hazra
IIT Kharagpur
Kharagpur, India
somnathhazra@kgpian.iitkgp.ac.in

Pallab Dasgupta
Synopsys
Santa Clara, USA
pallabd@synopsys.com

Soumyajit Dey
IIT Kharagpur
Kharagpur, India
soumya@cse.iitkgp.ac.in

ABSTRACT

In Reinforcement Learning, deep neural networks play a crucial role, especially in Multi-Agent Systems. Owing to information from multiple sources, the challenge lies in handling input permutations efficiently, causing sample inefficiency and delayed convergence. Traditional approaches treat each permutation source as individual nodes for inference. Our novel approach integrates an attention mechanism, allowing us to capture temporal dependencies and contextually align inputs. The attention mechanism enhances the alignment process, allowing for improved information processing. Empirical evaluations on SMAC environments demonstrate superior performance compared to baselines, achieving a higher win rate on 68% of test evaluations.

KEYWORDS

Multi-Agent Reinforcement Learning; Permutation Invariance; Permutation Equivariance; Attention

ACM Reference Format:
Somnath Hazra, Pallab Dasgupta, and Soumyajit Dey. 2024. Addressing Permutation Challenges in Multi-Agent Reinforcement Learning: 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

Reinforcement Learning, a cornerstone in controlling single-agent [1, 18, 21] and Multi-Agent Systems (MAS) [4, 10], faces new challenges in the expanding realm of Multi-Agent Reinforcement Learning (MARL). Notably, the issues of *Permutation Invariance (PI)* and *Permutation Equivariance (PE)* emerge as critical challenges. In MARL, agents perceive state information as unordered sets [15], posing a challenge for traditional Deep Neural Networks (DNNs) unequipped to handle such structures with permutations [27].

In a multi-agent setting; each agent, a_i, selects actions ($u_i \in \mathcal{U}$) from a local policy ($\pi_i : \Omega \to \mathcal{U}$), derived from its local observation ($o_i \in \Omega$) of the environment. Traditionally, spatial information is represented as sets without specific order, challenging traditional DNNs. Our objective is to learn policies π_i for actions u_i given set-formatted observations o_i, encapsulating permutations.

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).

Consider a system with three agents a_1, a_2, and a_3, each perceiving its own and others' states, introducing potential permutations. For instance, agent a_1 with local observation $o_1 = \{o_{1,1}, o_{1,2}, o_{1,3}\}$, where $o_{i,j}$ is agent a_j's state as seen from a_i, may exhibit internal swapping among $o_{1,i}$. The policy π_1 achieves Permutation Invariance if it consistently produces unchanged output [9], and Permutation Equivariance if its output aligns with the input permutation [12]. *Our objective is to learn policies that respect both PI and PE properties, accommodating observations as sets.*

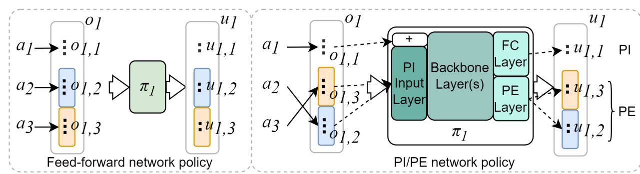

Figure 1: PI and PE policy π_1 for a_1 using local observation o_1 with possible permutations as seen by agent 1.

We have inputs of size m. Let G be the set of all permutation matrices of sizes $m \times m$, and $g \in G$. A policy $\pi_i : \Omega \to \mathcal{U}$, is PI if $\pi_i([o_{i,1}, ..., o_{i,m}]^\top) = \pi_i(g \times [o_{i,1}, ..., o_{i,m}]^\top), \forall g \in G$. A policy is PE if $\pi_i(g \times [o_{i,1}, ..., o_{i,m}]^\top) = g \times (\pi_i([o_{i,1}, ..., o_{i,m}]^\top))$, where $[o_{i,1}, ..., o_{i,m}]^\top \in \Omega$ [24]. A similar example is shown in Fig. 1 (π_i is explained later). The permutation possibilities shows exponential growth to increasing number of agents.

Existing approaches [5, 9, 22, 26], use Graph Nets [3] and Transformers [16], address permutation challenges, but often fail to treat inputs as independent sources before output summarization [25]. We propose an novel methodology that leverages the attention mechanism to effectively capture permutation patterns and align state information in MAS. Our contributions include the following.

(1) A methodology incorporating the attention mechanism, enhancing the model's capability to handle permutations in multi-agent systems by capturing permutation patterns.
(2) An approach to address disruptions in the auto-regressive input sequence caused by frequent permutations, ensuring a more practical algorithm for handling inconsistencies.
(3) Empirical evaluations on StarCraft [13] and GRF [6] environments demonstrating improved sample efficiency and convergence compared to existing methods.

1.1 Related Work

We build upon VDN [14] as our foundational algorithm for solving the Dec-POMDP problem [11]. Contemporary approaches to PI/PE such as data augmentation [23] encounters scalability issues as generating all possible permutations is infeasible in a continuous input space. In MAS, the dynamic and stochastic nature of the environment necessitates parsing and modeling the hidden semantics of the environment as perceived by local agents. Works such as [5, 8, 9, 19, 20] employ shared embedding layers to capture state semantics, albeit with limitations in representation capability [17]. To address this, recent works utilize hyper-networks [2] and self-attention from Transformers [7, 22, 26] to predict separate weights for each semantic component, enhancing representational capability. Utilizing the Markov property of the system further aids in interpolation and weight prediction for hyper-networks.

2 METHODOLOGY

The input to π_i are the local observations, o_i. Each o_i is composed of state information from other active agents, as perceived by agent i, o_{-i}, concatenated with its own information o_{+i}, i.e., $o_i = [o_{+i}, o_{-i}]$. In Figure 1, $o_{-i} = \{o_{1,2}, o_{1,3}\}$. By minimal modification [2], if the input layer which accepts o_{-i} is PI, we get a PI policy. Similarly, if the output layer providing u_{-i} is PE, we get a PE policy. So we changed only the input layer processing o_{-i} and the output layer for u_{-i} as shown in Figure 1. The remaining layers were kept similar to those used in a DNN policy.

The weights of each layer in a neural network are stored in a matrix, where the row dimension corresponds to the input vector and the column dimension corresponds to the output vector. In order to maintain consistency in the input to the deeper layers for PI, it is desired that over time the input to the neural network remains in the same form as it was at $t = 0$, o_i^0. To keep u_i^t consistent with the order of o_i^t in PE, it is necessary to realign the weights of the output layer with o_i^t. Our approach can be summarized by the following points.

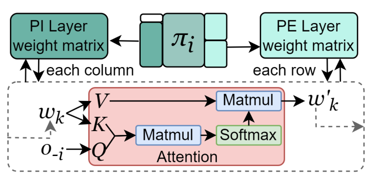

Figure 2: Overall methodology to design the PI/PE policy.

- For PI, our objective is to maintain the local observation o^t_{-i} in the same arrangement as it was at the initial time-step o^0_{-i}. For this we use the Attention module [16], but do not self-attend. We use the order of o^0_{-i}, as Q to reorder each $o^t_{i,j}$; $j \neq i$, used as K and V; through attention.
- For PE, we rearrange each row weights based on the outputs, u_{-i}, according to each $o_{i,j}$; $j \neq i$. Here $o^t_{-i} \equiv Q$ and each row w_k from weight matrix serves as both K, V (Equation 1).

$$w'_k = softmax\left(\frac{o^t_{-i} w_k^\top}{\sqrt{dim(w_k)}}\right) w_k \tag{1}$$

For longer trajectories ($t >> 0$), o_i^0 is not a good estimation for PI. To mitigate the problem, we use the PE strategy to aid invariance. Here, instead of reordering each row of the weight matrix, we reorder each column of our weight matrix. *In summary, we can say that, if the columns of the weight matrix are permuted according to the given input vector, we achieve PI; but if the rows are permuted according to the input vector, we achieve PE.* The complete methodology is outlined in Figure 2.

3 EXPERIMENTAL RESULTS

Our approach was evaluated on StarCraft [13] and Google Research Football [6] benchmarks. As baselines we used PIC [9], HPN [2], SET [26], DS [8], MEM [22], and ASN [19]. In the results, our approaches are: • *Permutation Agnostic System (PAS)*, where we used o_i^0 as the approximation for PI • *Permutation Equivariant System (PES)*, where we used the equivariance approach for PI and PE.

Both the benchmark environments can be modelled as Dec-POMDP [11]; consist of cooperating and competing agents; we control the cooperative agents. The observation space is continuous, where o_i consists of state information from all other agents in the environment (o_{-i}) apart from o_{+i}. The action space is discrete for both; where some actions may be directed towards an ally or enemy agent (PE) depending on the environment specifications.

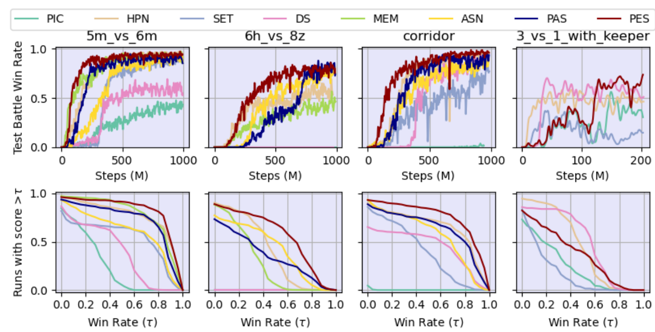

Figure 3: Evaluation results on StarCraft and GRF.

As shown in the results in Figure 3, our PES approach out-performs the baselines for most scenarios. The PAS approach did not give equally good results owing to state-estimation errors. The summary of the evaluations is shown in Figure 4, where we present the percentage of evaluations where the mean win rate was ≥ 0.6, ≥ 0.8, and ≥ 0.9 on StarCraft scenarios.

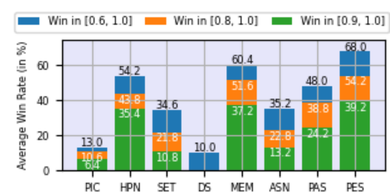

Figure 4: Comparison of average % of episodes where win rate≥ τ.

4 CONCLUSION

In this work, we proposed a novel approach to efficiently address invariance and equivariance problems by using the attention mechanism. Our method integrates PI and PE layers into conventional policy networks, overcoming permutation challenges and improves decision accuracy. Its efficacy was empirically evaluated on benchmark environments, where it outperformed existing methods and indicated promise for enhanced multi-agent system performance via efficient training and convergence. Looking forward, our work sets the stage for leveraging attention mechanisms in MARL for more complex challenges, and further exploration is needed for its applicability in diverse scenarios, promising advancements in MARL research.

REFERENCES

[1] Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* (2022), 1–49.

[2] HAO Jianye, Xiaotian Hao, Hangyu Mao, Weixun Wang, Yaodong Yang, Dong Li, Yan Zheng, and Zhen Wang. 2022. Boosting Multiagent Reinforcement Learning via Permutation Invariant and Permutation Equivariant Networks. In *The Eleventh International Conference on Learning Representations*.

[3] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[4] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2021), 4909–4926.

[5] Ryan Kortvelesy, Steven Morad, and Amanda Prorok. 2023. Permutation-Invariant Set Autoencoders with Fixed-Size Embeddings for Multi-Agent Learning. *arXiv preprint arXiv:2302.12826* (2023).

[6] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4501–4510.

[7] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*. PMLR, 3744–3753.

[8] Yan Li, Lingxiao Wang, Jiachen Yang, Ethan Wang, Zhaoran Wang, Tuo Zhao, and Hongyuan Zha. 2021. Permutation invariant policy optimization for mean-field multi-agent reinforcement learning: A principled approach. *arXiv preprint arXiv:2105.08268* (2021).

[9] Iou-Jen Liu, Raymond A Yeh, and Alexander G Schwing. 2020. PIC: permutation invariant critic for multi-agent deep reinforcement learning. In *Conference on Robot Learning*. PMLR, 590–602.

[10] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3133–3174.

[11] 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.

[12] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. 2017. Equivariance through parameter-sharing. In *International conference on machine learning*. PMLR, 2892–2901.

[13] 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).

[14] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296* (2017).

[15] Yujin Tang and David Ha. 2021. The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 22574–22587.

[16] 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).

[17] Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A Osborne. 2019. On the limitations of representing functions on sets. In *International Conference on Machine Learning*. PMLR, 6487–6494.

[18] Hao-nan Wang, Ning Liu, Yi-yun Zhang, Da-wei Feng, Feng Huang, Dong-sheng Li, and Yi-ming Zhang. 2020. Deep reinforcement learning: a survey. *Frontiers of Information Technology & Electronic Engineering* 21, 12 (2020), 1726–1744.

[19] Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. 2019. Action semantics network: Considering the effects of actions in multiagent systems. *arXiv preprint arXiv:1907.11461* (2019).

[20] Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. 2020. From few to more: Large-scale dynamic multiagent curriculum learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7293–7300.

[21] Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. 2022. Deep reinforcement learning: a survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[22] Yaodong Yang, Guangyong Chen, Weixun Wang, Xiaotian Hao, Jianye Hao, and Pheng-Ann Heng. 2022. Transformer-based working memory for multiagent reinforcement learning with action parsing. *Advances in Neural Information Processing Systems* 35 (2022), 34874–34886.

[23] Zhenhui Ye, Yining Chen, Guanghua Song, Bowei Yang, and Shen Fan. 2020. Experience augmentation: Boosting and accelerating off-policy multi-agent reinforcement learning. *arXiv preprint arXiv:2005.09453* (2020).

[24] Raymond A Yeh, Yuan-Ting Hu, Mark Hasegawa-Johnson, and Alexander Schwing. 2022. Equivariance discovery by learned parameter-sharing. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1527–1545.

[25] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems* 30 (2017).

[26] Fengzhuo Zhang, Boyi Liu, Kaixin Wang, Vincent Tan, Zhuoran Yang, and Zhaoran Wang. 2022. Relational reasoning via set transformers: Provable efficiency and applications to MARL. *Advances in Neural Information Processing Systems* 35 (2022), 35825–35838.

[27] Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. 2019. Deep set prediction networks. *Advances in Neural Information Processing Systems* 32 (2019).