Addressing Permutation Challenges in Multi-Agent Reinforcement Learning

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

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 ($\pi_i : \Omega \rightarrow \mathcal{U}$), 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 $\pi_i$ for actions $u_i$ given set-formatted observations $o_i$, encapsulating permutations.

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 $o_{1,i}$ is agent $a_i$’s state as seen from $a_i$, may exhibit internal swapping among $o_{1,i}$. The policy $\pi_i$ 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 $\pi_1$ for $a_1$ using local observation $o_1$ with possible permutations as seen by agent 1.](image)

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 $\pi_i : \Omega \rightarrow \mathcal{U}$ is PI if $\pi_i([o_{1,1},...,o_{1,m}]^\top) = \pi_i(g \times [o_{1,1},...,o_{1,m}]^\top), \forall g \in G$. A policy is PE if $\pi_i(g \times [o_{1,1},...,o_{1,m}]^\top) = g \times (\pi_i([o_{1,1},...,o_{1,m}]^\top))$, where $[o_{1,1},...,o_{1,m}]^\top \in \Omega$ [24]. A similar example is shown in Fig. 1 ($\pi_i$ is explained later). The permutation possibilities show 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.

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 \( \pi_t \) 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 \( o_{-i} \) is PE, we get a PE policy. So we changed only the input layer processing \( o_{-i} \) and the output layer for \( o_{-i} \) 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 \( t = 0 \), \( o_0 \). To keep \( o_t \) consistent with the order of \( o_0 \) in PE, it is necessary to realign the weights of the output layer with \( o_t \). Our approach can be summarized by the following points:

- For PI, our objective is to maintain the local observation \( o_{t-1} \) in the same arrangement as it was at the initial time-step \( o_0 \). For this we use the Attention module [16], but do not self-attend. We use the order of \( o_0 \), as \( Q \) to reorder each \( o_{t-1}^{j} ; j \neq i \), 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 \( o_{t-1} \equiv Q \) and each row \( w_k \) from weight matrix serves as both \( K \), \( V \) (Equation 1).

\[
\begin{align*}
    w'_k &= \text{softmax}(\frac{o_{t-1}^{j} w_k}{\sqrt{\text{dim}(w_k)}})w_k \\
\end{align*}
\]

For longer trajectories \( t >> 0 \), \( o_0 \) is not a good estimation for PI. To mitigate the problem, we use the PE strategy to aid variance. 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 \( o_0 \) 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 \( \omega_i \) 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.

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 \( \geq 0.6 \), \( \geq 0.8 \), and \( \geq 0.9 \) on StarCraft scenarios.

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.