BMRL: Backward Multi-Agent Reinforcement Learning with Permutation-Free Neural Networks

Anonymous Author(s)
Submission Id: «EasyChair submission id»

ABSTRACT

Multi-agent reinforcement learning (MARL) often faces the sparse reward and curse of dimensionality problem, which is much more severe than traditional reinforcement learning. To address these issues, in this paper, we propose a backward curriculum multi-agent reinforcement learning (BMRL) framework that can efficiently learn in environments where rewards are sparse and have a large state space. In the proposed framework, agents learn rewards through backward curriculum learning and enhance the agents' generalization ability through forward learning. We also introduce the multi-agent set transformer (MAST), which can work efficiently in the proposed framework. MAST utilizes the properties of permutation equivariant and permutation invariant to effectively reduce the search space in backward curriculum learning and forward learning When the initial state is static or changes dynamically. As a result, agents can learn effective policies faster than existing algorithms in environments where multiple agents must cooperate or where rewards are sparse.

KEYWORDS

Multi-Agent Reinforcement Learning, Backward Curriculum Learning, Permutation Invariance and Equivariance, StarCraft Multi-Agent Challenge, Google Research Football

ACM Reference Format:
Anonymous Author(s). 2025. BMRL: Backward Multi-Agent Reinforcement Learning with Permutation-Free Neural Networks. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

1 INTRODUCTION

Using artificial neural networks (ANNs) for reinforcement learning (RL) has led to rapid progress in game domains such as Atari [20, 21], Go [29], and StarCraft2 [33], where they have outperformed human champions. Recently, RL has moved from the restrictive environment of specific domains and has been combined with fundamental real-world problems, such as natural language processing (NLP) [3, 25] and robotics [10, 36]. However, the limitation of RL is that it can only train a single agent, which limits its use in various real-world problems that require cooperation. Cooperative multi-agent reinforcement learning (MARL), proposed to solve this problem, allows two or more agents to interact and learn to achieve a common goal. Cooperative MARL can be applied to complex environments where RL has been inaccessible, such as dynamic resource allocation [23] or traffic control [1].

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), A. El Fallah Seghrouchni, Y. Vorobeychik, S. Das, A. Nowe (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

However, MARL suffers more strongly from problems that make RL difficult. For example, sparse rewards make exploration difficult, preventing agents from obtaining meaningful rewards early in learning due to rare goal rewards. This issue, combined with the credit assignment problem [6], which requires determining which agents were meaningfully involved in the reward, dramatically increases the required exploration. Failure to solve this leads to the lazy-agent problem [30], where specific agents fail to act meaningfully, preventing them from learning an efficient joint policy. Curse of dimensionality [19], the exponentially growing search space as the number of agents increases or longer planning is done, is also very difficult. This phenomenon makes it difficult for the algorithm to balance exploration and exploitation appropriately. Collecting large amounts of meaningful data to learn effective policies consumes many resources and time, leading to the sample inefficiency problem.

To address these issues, we propose a backward curriculum multi-agent reinforcement learning (BMRL) framework. This framework utilizes backward curriculum learning to solve for sparse rewards so that even long-term episodes can be learned effectively. Our framework also uses a multi-agent set transformer (MAST). This algorithm has permutation-equivariant (PE) and permutation-invariant (PI) properties based on the Set Transformer [14] to reduce the search space of agents to improve sample efficiency and to effectively cope with dynamic changes in the initial state distribution that occur in backward curriculum learning. In our experiments, we evaluate the performance of the proposed framework and algorithm using StarCraft Multi-Agent Challenge (SMAC) [27] and Google Research Football (GRF) [13], which are popular MARL benchmarks. The experimental results show that our proposed algorithm significantly outperforms the learning speed of existing models. In particular, in the 11 vs. 11 game mode in GRF, which is one of the most difficult in the MARL benchmarks, We show that the proposed framework overcomes the sparse reward problem in soccer, allowing it to quickly reach a level where 11v11 hard probabilistic built-in AI can be utilized as a training target.

The main contributions of the paper are as follows:

- We suggest a framework that solves the sparse reward problem in sparse reward environments using backward curriculum learning.
- We propose a PE/PI multi-agent embedding using a Set Transformer to evaluate the merits of PE/PI properties in situations where the initial state distribution constantly shifts due to backward curriculum learning.
- We demonstrate that our proposed algorithm and framework can achieve superior learning performance on benchmarks such as SMAC and GRF.

2 RELATED WORK

2.1 Multi-Agent Reinforcement Learning

Many MARL algorithms have used a centralized training decentralized execution (CTDE) [18, 24] framework to allow multiple agents to maximize a common objective. CTDE is a method that utilizes extra signals such as actions and state information and each agent's observations during training to estimate a centralized value. MAPPO [35] directly applied PPO [28], one of the most famous RL algorithms, to MARL and outperformed QMIX [26], an existing robust baseline, by using a single critic value and shared observation and parameter sharing, where agents share policy and value function parameters. Therefore, many MARL algorithms use state information for centralized learning to estimate joint values, but this approach is difficult to use in environments where extra information is unavailable.

MAT [34] succeeded in inducing the action space of the algorithm to have linear complexity by inducing the MARL problem as a sequence modeling problem without taking advantage of the extra information. This algorithm has shown learning capabilities on benchmarks such as multi-agent MuJoCo [4], SMAC, and GRF comparable to or better than algorithms such as MAPPO and HAPPO [12]. However, because actions are processed sequentially, there is a drawback the training time slows down as the number of agents increases. It also does not guarantee PE because action sampling can vary depending on how the order of the agent's actions is determined.

2.2 Permutation-Invariant and Equivariant Modeling for Multi-Agent Systems

In MARL, most algorithms face sample inefficiency caused by the curse of dimensionality, as the state-action space grows exponentially with the number of agents [15]. To address this issue, previous studies [7, 8] have focused on applying both the PI property [11, 17] and PE property to the model, where the PI property allows permutations of observations to correspond one-to-one with a single state value, while the PE property facilitates a one-to-one correspondence between these permutations and action permutations. However, while these methods have exploited the PE/PI property to mitigate the curse of dimensionality and improve sample efficiency by reducing the state-action space, they have primarily focused on explaining the strengths of PE/PI in forward learning with static initial state distributions. In contrast, our proposed set transformer-based method demonstrates how the PE/PI property can be effectively leveraged in more complex backward curriculum learning situations where the initial state distribution changes dynamically.

2.3 Backward Curriculum Learning for Sparse Reward Environment

While advances in RL algorithms have enabled agents to find the optimal policy through the network's internal evolution, they still struggle to learn in environments where rewards are sparse or require long-term planning through extensive exploration. This tendency is mainly due to the excessive exploration required to obtain sparse reward signals early in learning. To solve this problem,

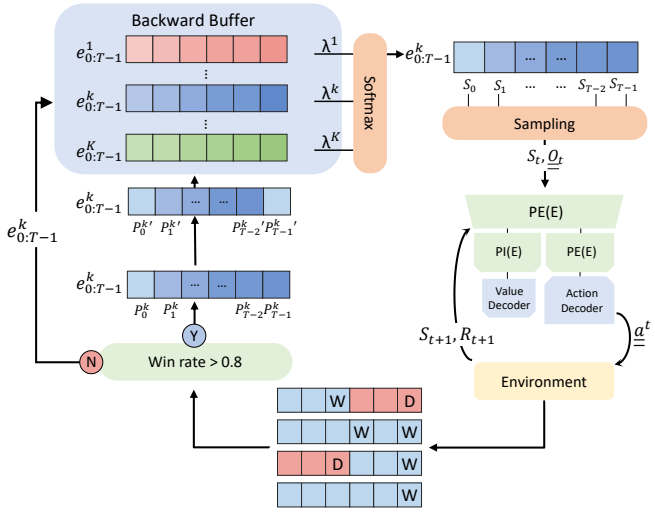

Figure 1: Overview of the proposed framework.

backward curriculum learning has been widely studied to enable learning in sparse reward environments with algorithms alone by gradually increasing the difficulty of learning rather than learning in complex environments from the beginning. Backward curriculum learning has shown that agents can successfully learn in RL environments previously inaccessible to them by learning from states close to the goal state in sparse reward environments [5, 9]. RFCL [31] solved the exploration problem in the early stages of learning by combining backward curriculum and forward curriculum using a small number of demo data simultaneously, leading to generalized performance. However, backward curriculum learning in MARL environments has been relatively under-researched.

In MARL, TiZero [16] achieved in learning high-level policies for an 11vs11 full soccer game by gradually increasing the learning difficulty through curriculum learning. However, randomly initializing player positions within a particular area to generate a curriculum, which results in irregular permutations of the initial player position observations, leads to excessive entropy in the initial state distribution. Excessively high entropy in the initial state distribution can cause the agent to have difficulty in curriculum learning, and this problem is particularly critical for agents that do not satisfy the PE/PI property. To address these issues, our framework introduces a different learning method that uses a small number of trajectory data to limit the entropy of the initial state distribution.

3 BACKGROUND

3.1 Problem Formulation

We formulate the MARL problem as a modified decentralized partially observable Markov decision process (Dec-POMDP) [2, 24]. Our modified Dec-POMDP is defined as 8-tuple:

$$\mathcal{M} = (\mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, P, \rho, R, \gamma),$$

where $\mathcal{N} \triangleq \{1, \ldots, n\}$ is a set of agents and $\mathcal{S}$ is a set of states of the environment. Joint observation space $\mathcal{O} \triangleq \prod_{i \in \mathcal{N}} \mathcal{O}^i$ is a Cartesian product of local observation spaces $\mathcal{O}^i$. Joint action space $\mathcal{A} \triangleq$

$\prod_{i \in N} \mathcal{A}^i$ is a Cartesian product of the agents' local action spaces $\mathcal{A}^i$. Transition probability function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the transition probability from s to s' given the a. $\rho : \mathcal{S} \to \mathbb{R}$ is a initial state distribution. Shared reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents a reward function model, and γ is the discount factor. At time step $t \in \mathbb{N}$, agents observe joint observations $o_t \in O$ and takes an action $a_t \in \mathcal{A}$ using the policy $\pi_\theta(a_t \mid o_t)$ parameterized by θ.

3.2 Permutation Invariant / Permutation Equivariance

In MARL, the state and action spaces increase exponentially with the number of agents. PE and PI models can efficiently reduce them by ignoring many duplicated combinations. Before we utilize a PE and PI property of the Set Transformer, each property needs to be defined.

Definition 3.1. A bijective function $\phi : N \to N$ is a permutation of set N. A permutation set Φ_N is a set of permutations of set N.

Definition 3.2. A vector permutation $\vec{\phi} : A^N \to A^N$ is a vector-valued function with a permutation of the index set,

$$\vec{\phi}(x) = \left(x_{\phi(0)}, x_{\phi(1)}, \ldots, x_{\phi(N-1)}\right).$$

A vector permutation set $\vec{\Phi_N}$ is a set of vector permutations with N indices.

Definition 3.3. A permutation equivariant function $f : A^n \to A^n$ is a function satisfying $\forall \vec{\phi} \in \vec{\Phi_n}, x \in A^n : f(\vec{\phi}(x)) = \vec{\phi}(f(x))$.

Definition 3.4. A permutation invariant function $f : A^n \to B$ is a function satisfying $\exists y \in B : \forall \vec{\phi} \in \vec{\Phi_n}, x \in A^n : f(\vec{\phi}(x)) = y$.

With the definitions, we can describe the properties of the model L with PE or PI. (1) and (2) are definitions of PE and PI, respectively.

$$\forall \vec{\phi} \in \vec{\Phi_N} : L\left(\vec{\phi}\,(a)\right) = \vec{\phi}\,(L\,(a)) \tag{1}$$

$$\exists c : \forall \vec{\phi} \in \vec{\Phi_N} : L\left(\vec{\phi}\,(a)\right) = c \tag{2}$$

3.3 Backward Curriculum Learning

Backward curriculum learning is used to accelerate many RL problems. In most cases, it is given as the initial state distribution.

Definition 3.5. A curriculum C is a sequence of Dec-POMDPs $\{C_i\}$ that only differ in ρ_is, where ρ_i is the initial state distribution of C_i, converging to real environment initial state distribution ρ_R as $i \to \infty$.

The main object of curriculum learning is to ease the space or action complexity at the initial phase of RL. It can be implemented with heuristics, experts' samples, or generative state models. We utilize a sample of simulation data to make backward curriculum learning.

Definition 3.6. Let $\mu(\rho_x, \rho_y) \in \mathbb{R}$ be a predefined difficulty metric function and ρ_G be a goal state distribution. A backward curriculum learning $\overleftarrow{C}$ is a curriculum learning that satisfies:

$$\forall i > j : \mathbb{E}_{x_i, x_G \sim \rho_i, \rho_G}\left[\mu(x_i, x_G)\right] \geq \mathbb{E}_{x_i, x_G \sim \rho_j, \rho_G}\left[\mu(x_j, x_G)\right].$$

3.4 Attention Module

Consider a MARL environment with n agents. There are n observation embeddings, forming a set consisting of n queries, each represented by an embedding vector of dimension $d_q : Q \in \mathbb{R}^{n \times d_q}$. And to transform this query, we need to create n_v key-value pairs $K \in \mathbb{R}^{n_v \times d_q}, V \in \mathbb{R}^{n_v \times d_v}$ as well. This allows us to define an attention function $\text{Attention}(Q, K, V)$ as follows [32]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V$$

The output of the Attention function $O \in \mathbb{R}^{n \times d_v}$ is a weighted sum of V, with values corresponding to larger keys having higher weights as they are more closely related to the query.

We can induce a multi-head attention function from a single attention function. First, project Q, K, V onto h different d_q^M, d_q^M, d_v^M-dimension vectors, respectively. Then, apply $\text{Attention}(\cdot, \cdot, \cdot; w_j)$ to each h projection and concatenate each output. Finally, concatenate each of the output O_j and perform a linear transformation, resulting in the following output:

$$\begin{aligned} \text{Multihead}\,(O, K, V) &= \text{Concat}\,(O_1, \ldots, O_h)\, W^O \\ \text{where}\;\; O_j &= \text{Attention}(QW_j^Q, KW_j^K, VW_j^V), \end{aligned}$$

where the projections are parameter matrices $W_j^Q, W_j^K \in \mathbb{R}^{d_q \times d_q^M}$, $W_j^V \in \mathbb{R}^{d_v \times d_v^M}$,$W^O \in \mathbb{R}^{hd_v^M \times d}$.

4 MAIN CONTRIBUTIONS

4.1 Model Architecture

We introduce the multi-agent set transformer (MAST). Consider a MARL environment with n agents training in an d_a-dimensional action space and d_o-dimensional observation space. Assuming a single $o_t \in \mathbb{R}^{n \times d_o}$ is sampled, MAST is a multi-head attention-based neural network that takes as input the joint observation o_t of the agents and outputs a PE joint action $a_t \in \mathbb{R}^{n \times d_a}$ and a PI joint state value $v_t \in \mathbb{R}^{n \times 1}$. MAST consists of two encoders that generate PE/PI embeddings and two decoders that output a_t and v_t, with the two encoder-decoder pairs acting as actor and critic, respectively. The two encoders share the embedding layer and one self-attention block (SAB) to map o_t to a common feature space. See Figure 2 for the whole structure of MAST. The output obtained by inputting o_t into the shared embedding layer is $e_t \in \mathbb{R}^{n \times d}$, where d is the hidden dimension of the embedding layer. $rFF(\cdot)$ is any row-wise feedforward layer, namely, a parameter-shared network among agents. For generalization, define arbitrary matrices $X, Y \in \mathbb{R}^{n \times d}$.

$$\begin{aligned} H &= \text{LN}\,(X, \text{Multihead}\,(X, Y, Y)) \\ \text{MAB}\,(X, Y) &= \text{LN}\,(H + rFF(H)) \\ \text{SAB}\,(X) &= \text{MAB}\,(X, X)\,, \end{aligned}$$

where LN is the Layer normalization layer.

4.1.1 ***Permutation Equivariant Actor.*** To ensure that the Permutation Equivariant Actor (PEA) can generate a fixed action permutation for the input, the SAB must be utilized to maintain the PE property. For this, we group these shared layers together and

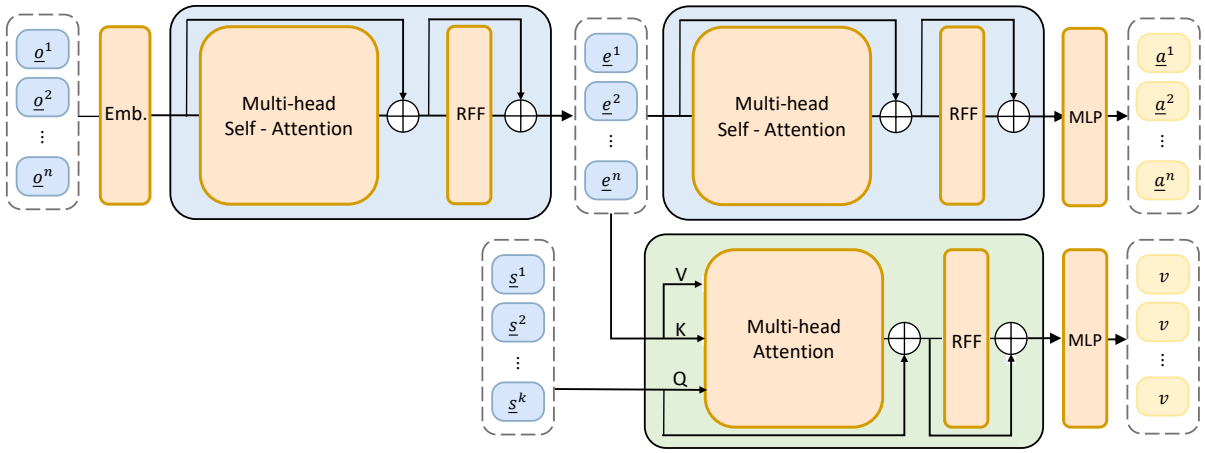

Figure 2: Architecture of the MAST. MAST takes the agents' joint observation o_t as input and output feature embedding Z using a shared embedding block (SEB) block. This embedding has the PE property and is used in the SAB and PMA blocks. An Z input to the SAB block is decoded as a_t while preserving the PE property. On the other hand, the Z input to the PMA block is converted to a PI embedding through S and attention operations. This PI embedding is then converted to v_t by the decoder, where all components of v_t have the same value.

define them as a Shared Embedding Block (SEB). The output of the SEB is defined as $Z \in \mathbb{R}^{n \times d}$ Let e_t pass through the two SAB one after the other and MLP layer and output a_t. It can be expressed mathematically as follows:

$$Z = \text{SEB}(o_t) = \text{SAB} \circ \text{Embed}(o_t) \in \mathbb{R}^{n \times d}$$
$$\text{PEA}(o_t) = \text{MLP} \circ \text{SAB} \circ \text{SEB}(o_t) \in \mathbb{R}^{n \times d_a}.$$

4.1.2 ***Permutation Invariant Critic.*** We utilize multi-head attention and a learnable parameter, a set of k seed vectors $S \in \mathbb{R}^{k \times d}$. The structure of pooling multihead attention (PMA) with k seed vectors and PIC, which aggregates features using Z output from SEB, can be defined as follows:

$$\text{PMA}_k(Z) = \text{MAB}(S, Z) \in \mathbb{R}^{k \times d}$$
$$\text{Flatten}(x) = (x_0^T, x_1^T, \ldots, x_{k-1}^T) \in \mathbb{R}^{1 \times kd}$$
$$\text{Repeat}(x, n) = (x, x, \ldots, x) \in \mathbb{R}^{n \times 1}$$
$$\text{PIC}(Z) = \text{Repeat} \circ \text{MLP} \circ \text{Flatten} \circ \text{PMA}_k(Z) \in \mathbb{R}^{n \times 1}.$$

In this case, the n outputs from the PIC will all have the same value. If more than two seed vectors are utilized, k vectors can be flattened. Using this architecture, we can compute the joint advantage function $s_t \in \mathcal{S}$, we can define a joint advantage function $A_t \triangleq A(a_t \mid s_t)$, which we can use to optimize the algorithm by maximizing the following clip objective:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid o_t)}{\pi_{\theta_{old}}(a_t \mid o_t)}$$
$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta) A_t, \text{clip}\left(r_t(\theta), 1 \pm \epsilon\right) A_t\right)\right].$$

4.2 Backward Curriculum Learning

Our purpose is to propose a practical framework for accelerating learning using backward curriculum learning in sparse reward environments where agents are unlikely to witness rewards even after extensive exploration or in environments that require long-term planning to design policies. The overall structure of BMRL can be seen in Figure 1. The learning in the framework is a combination of backward curriculum learning and forward learning. First, in backward curriculum learning, the agents initially have an initial state distribution ρ_1 close to the goal state. However, as the curriculum level C_i increases according to 3.6, the agents have a configuration ρ_i that is closer to ρ_R. After the backward curriculum learning is complete, the forward learning has a fixed ρ_R distribution ρ. It can effectively reach the goal state by utilizing the agents that have explored the goal state in backward curriculum learning.

There are two ways to generate ρ_i that are close to the goal state: randomly generating stochastically possible states or utilizing expert demonstration data. However, randomly generating probabilistic states is almost impossible to obtain natural data, such as episodic data generated by real agents, and rapidly changing ρ without sufficient conviction can cause inefficient learning. This is particularly devastating for model architectures that do not have PE/PI properties because if the agent's information changes rapidly from training to training, the number of possible input permutations increases proportionally to the number of agents that must be accommodated. We will, therefore, initialize ρ using n_d trajectories demonstrated by built-in AI. Using expert data ensures that we get trajectories that are similar to the data generated by agents, and it allows the agent to learn efficiently because it has a limited ρ compared to randomly initializing ρ.

We draw each trajectory τ_l with probability $p(\tau_l) = \text{softmax}(-d)_l$ to process backward curriculum learning, where $d_l = \|x_l - x_G\|_2$ with x_l and x_G being the location of ball $l \in \{0, \ldots, n_d - 1\}$ and the

location of target goal location, respectively. Then, we sampled a curriculum-dependent initial state $s_{i,0}$ with the truncated geometric distribution:

$$p(k) = \begin{cases} q(1-q)^k & \text{where } 0 \le k < t_i \\ 1 - \sum_{k'=0}^{t_i-1} p(k') & \text{if } k = t_i \\ 0 & \text{otherwise.} \end{cases}$$

The state $s_{i,0} = \tau_{l,t_i-k}$ is sampled from $p(k)$. Note that t_i is a time step of C_i determined by the curriculum level. As the curriculum learning progresses, ρ_i approaches ρ_R, and backward curriculum learning can be terminated without reaching ρ_R if the environmental conditions are satisfied. For the soccer domain, the episode trajectory can be farther away from the goal state in $s_{i,0}$ than the initial state ρ_R. Therefore, if d_i becomes farther than the distance from ρ_R to the goal state, the corresponding $p(\tau_i)$ switches from backward curriculum learning to ordinary forward learning. Therefore, the proportion of forward learning increases as the learning progresses.

4.3 Reward Shaping

Reward shaping was proposed to focus RL agents on exploring where they can reach the goal state [22]. The idea is to update the sparse reward of a state-action pair in R based on how suitable it is for reaching the goal state. This accelerates RL by increasing the density of rewards observed by the agents. The same principles can be applied to MARL environments.

Unlike SMAC, which has a high reward density by allowing players to earn a reward for each attack on the opponent, GRF provides a default reward based on score: 1 for scoring a goal and -1 for conceding a goal. This is the most intuitive reward that can be given in the soccer domain, but it typically requires much exploration for the agent to observe the score reward. To solve this problem and make training efficient, we need to design reward signals other than the score reward that the agents can frequently encounter to make it easier to observe the goal state. The following shows the types of rewards that agents can observe in the GRF benchmark.

- **Possession Rewards**: +0.02 for gaining possession of the ball, −0.01 for losing possession of the ball.
- **Score Rewards**: +5.0 for scoring, −1.0 for conceding a goal.
- **Out-of-bound Rewards**: Number of agents out of bounds during a match × 0.02.
- **Pass Rewards**: +0.2 for successful passes forward, +0.05 for passes backward.
- **Yellow Rewards**: +0.05 × (number of opponent cards − number of your team's cards) when your team or the opponent receives a yellow card.
- **Ball Position Rewards**: 0.002 (or −0.002) when inside the opponent's (or our own) penalty box, 0.001 (or −0.001) when close to the opponent's (or our own) goal, and 0.0 when the ball is close to the half line.

5 EXPERIMENTS

We utilize the SMAC and GRF benchmarks to validate the performance of MAST and BMRL. SMAC is an environment for verifying that different kinds of agents can cooperate to achieve a shared reward. GRF is an environment where multiple agents must successfully cooperate to reach a goal state in a sparse reward environment. In addition to the scenarios natively supported by GRF, we designed and experimented with new academy scenarios. and applied the same hyperparameters of the original algorithm presented in the original paper to reproduce its performance as much as possible.

To accurately assess the potential of our algorithm, we use MAT and MAPPO as a baseline, which are proven algorithms with high reliability and high performance. Since GRF Academy has a higher frequency of draws than SMAC, we evaluated it based on score (if +1 for a goal scored, −1 for a goal conceded) rather than win rate to evaluate this as well. All experimental values are based on averaging with Gaussian smoothing, and standard deviations were calculated and depicted as shaded regions around curves.

Table 1: Performance of evaluation win rate on all the SMAC benchmarks.

Map	MAST	MAT	MAPPO	Difficulty	Steps
8m	**100.0%**	97.7%	95.1%	7	1e6
1c3s5z	**99.9%**	93.6%	73.1%	7	1e6
MMM	**98.5%**	98.1%	92.9%	7	1e6
25m	**100.0%**	99.6%	80.8%	7	2e6
2c vs 64zg	**97.9%**	81.6%	95.0%	7	2e6
3s5z	95.3%	**97.1%**	63.6%	7	3e6
8m vs 9m	**97.7%**	96.9%	72.8%	7	5e6
10m vs 11m	**98.6%**	**98.6%**	85.6%	7	5e6
MMM2	**98.0%**	96.5%	73.2%	7	1e7

Table 2: Performance of evaluation goal score on the GRF scenarios.

Scenario	MAST	MAT	MAPPO	Steps
pass and shoot with keeper	**0.97**	0.91	0.95	2e6
3vs1 with keeper	**0.95**	0.93	0.95	3e6
5vs2 with keeper (ours)	0.97	0.98	**0.98**	1.5e6
2vs3 with keeper (ours)	**0.96**	0.95	0.90	3e6

5.1 BMRL Performance on GRF Benchmark

To demonstrate the potential of BMRL, we test the performance of our framework on GRF's 11 vs. 11 hard stochastic scenario (11 vs. 11 full game), which is among the most challenging of the MARL benchmarks. The 11 vs. 11 full game has powerful sample inefficiency due to its long episode length, sparse rewards, and many agents. Furthermore, the shared reward makes it impossible to properly evaluate off-the-ball movement, requiring a vast amount of exploration to create a meaningful policy. These difficulties make it impossible for current algorithmic improvements to consistently explore the goal state using forward learning alone in the 11 vs. 11 full game.

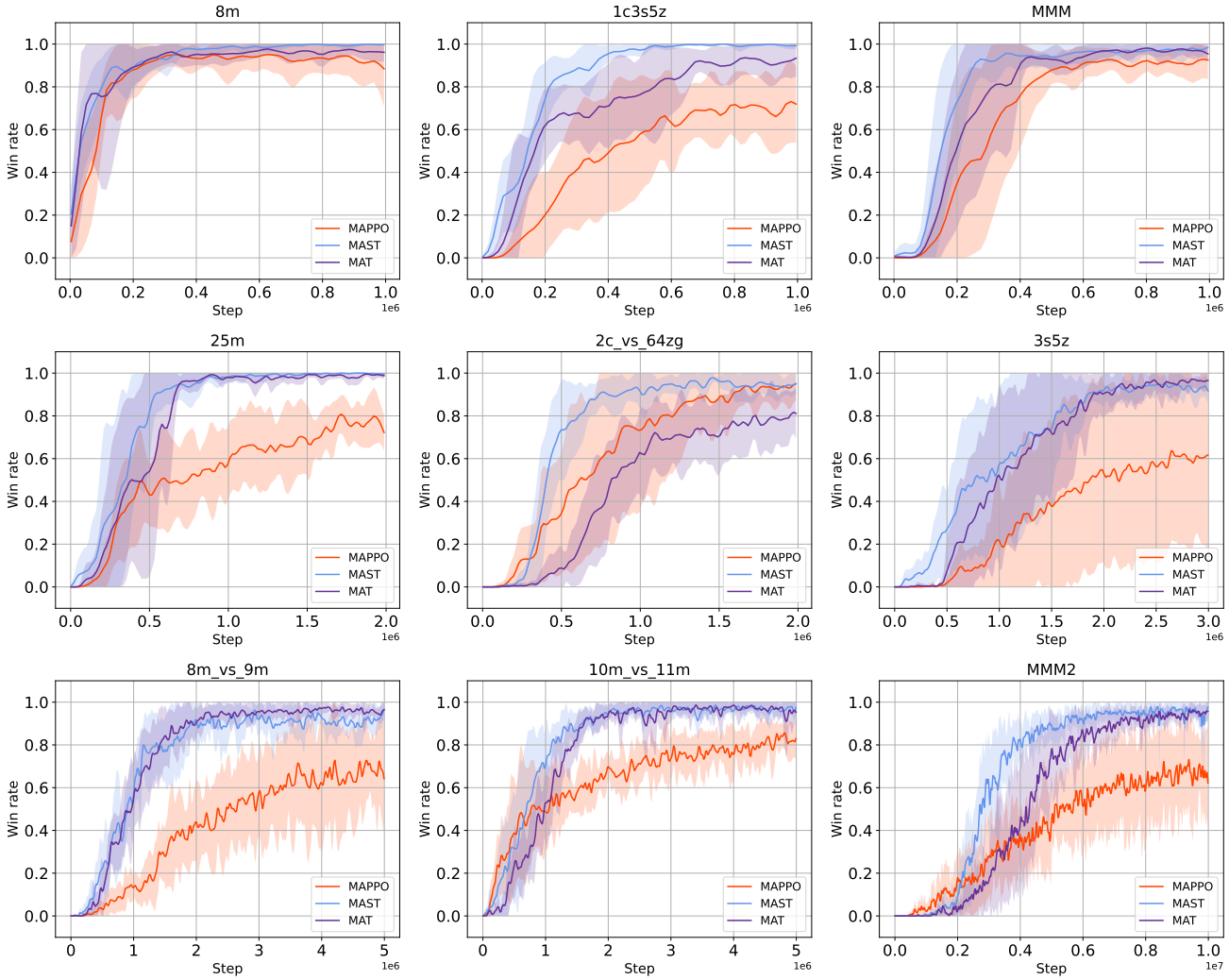

Figure 3: Performance comparison on SMAC benchmark introduced in Table 1.

Since BMRL requires expert demonstration data for backward curriculum learning, we need to create a suitable dataset and provide it to the framework. We created a single dataset for our experiments. The dataset we obtained consists of 3 episodes, each of which was created by a built-in AI. All algorithms that utilize backward curriculum learning will utilize this dataset. Also, while using backward curriculum learning, the evaluation was conducted with 11 vs. 11 full games to allow for comparisons with algorithms that do not use backward curriculum learning.

To objectively evaluate the performance of BMRL in 11 vs. 11 full game, we use different baselines in Table 3. BMRL-MAT is a framework that replaces MAST with MAT in the BMRL framework, and BMRL-MAP is a framework that replaces MAST with MAPPO. BMRL-DRS uses MAST to perform backward curriculum learning but without reward shaping.

Table 3: Variants of algorithms tested on GRF 11 vs. 11 full game scenario.

Algorithm	Use MAST	Reward Shape	Use Backward
BMRL	✓	✓	✓
BMRL-MAT	✗	✓	✓
BMRL-MAP	✗	✓	✓
BMRL-DRS	✓	✗	✓

To illustrate the strength of MAST in learning dynamic initial state distributions in BMRL, we first compare BMRL with BMRL-MAT and BMRL-MAP. As shown in Figure 5, BMRL with MAST has a much faster win rate and dispersion than BMRL-MAT and BMRL-MAP without MAST. This implies that MAST has an advantage over MAT and MAPPO in learning the initial state distribution. Also very encouraging is the evaluation of max rewards, which shows

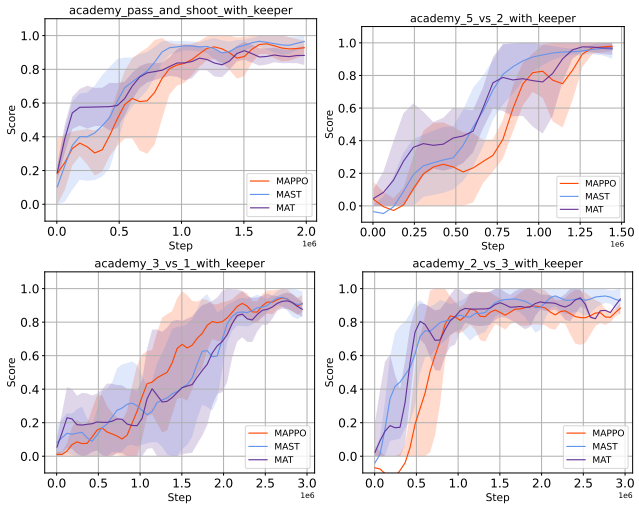

Figure 4: Performance comparison on four GRF scenarios introduced in Table 2.

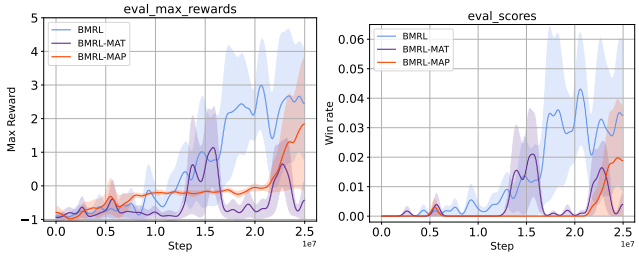

Figure 5: Performance comparison between BMRL, BMRL-MAT, and BMRL-MAP in GRF 11 vs. 11 scenario.

that BMRL can reach the goal state even during forward learning in the 11 vs. 11 full game by consistently achieving a particular max reward. This means it can actively explore the goal state in 11 vs. 11 full game, whereas previously forward learning alone could not actively explore the goal state in 11 vs. 11 full game.

5.2 MAST Performance on SMAC and GRF Benchamrks

According to Table 1, MAST achieved superior performance compared to MAT and MAPPO in most scenarios. In particular, MAST outperformed MAT and MAPPO in the environment where heterogeneous agents such as 1c3s5z, MMM, and MMM2 understand and interact. As shown in Figure 3, it finds the optimal policy much faster than traditional algorithms in environments where heterogeneous agents such as 1c3s5z, MMM, and MMM2 need to understand and interact with each other's roles. The fact that the milestones in these environments are faster than those of other algorithms before the win rate rises sharply supports the theoretical basis of MAST, designed to reduce the search space as much as possible. Furthermore, even when many homogeneous agents need to learn with a shared policy, such as 25m and 10m vs. 11m, MAST outperforms MAT, which is more potent at handling sequential actions, demonstrating that MAST can efficiently handle large numbers of agents. In Table 2, they also demonstrated superior learning ability, achieving the highest scores in three out of four scenarios. These results indicate that MAST can work well with static initial state distributions.

5.3 Ablation Study

In this section, we conduct ablation studies to analyze the importance of different components. For an objective performance analysis of the framework, we ablate some components of MAST and BMRL to verify that each block plays an important role.

Table 4: MAST variants with different ablated components.

Property	MAST	MAST-I	MAST-E	MAST-N	MAST-G
PE	✓	✓	✗	✗	✓
PI	✓	✗	✓	✗	✓

5.3.1 Effect of set attention blocks. To prove that each component of the proposed algorithm MAST is essential, we conduct experiments with three algorithms that exclude each block with the PE/PI property. Table 4 summarizes the characteristics of each algorithm. We replaced the PMA block with an SAB to remove the PI property and changed the SAB to an MLP to remove the PE property. The hyperparameters for all algorithms were set to be the same with MAST. W

Table 5: Ablation study of the MAST modules in SMAC and GRF scenarios.

Scenario	MAST	MAST-I	MAST-E	MAST-N	MAST-G
1c3s5z	**99.9%**	98.3%	97.8%	38.4%	99.4%
MMM	**98.5%**	96.6%	97.5%	32.2%	97.9%
2c vs 64zg	**97.9%**	91.0%	94.9%	88.3%	90.2%
3s5z	**95.3%**	82.5%	82.1%	33.6%	90.6%
3vs1 with keeper	**0.95**	0.86	0.41	0.38	0.93
5vs2 with keeper	0.97	**0.99**	0.45	0.22	**0.99**

The training results in Table 5 show that missing either the PMA or SAB block prevents the algorithm from converging reliably in most environments. In particular, for MAST-N with all PE/PI properties removed, the milestone at which the learning win rate increases rapidly drops sharply compared to the other ablation algorithms. These results support the rationale behind MAST, which is to reduce the search space as much as possible.

We use PMA blocks to assign PI property to MAST, but it has also been actively studied to assign PI property using GNNs. Therefore, we compare MAST-G and MAST with the PI property assigned using a graph convolution network (GCN) in which all agents are connected. Table 5 shows that PMA blocks converge more reliably than GCN blocks. This indicates that PMA blocks can extract state values more appropriately. However, MAST-G improves over MAST-I, which does not implement the PI property in most scenarios. This

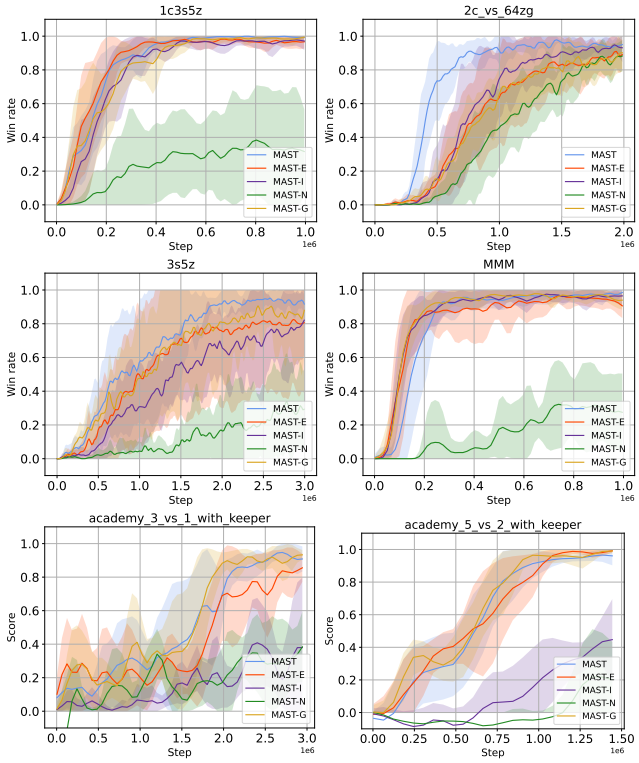

Figure 6: Performance comparison on MAST ablation study.

means that the PI property leads to performance improvements by mapping joint observations o_t to one state value v_t, regardless of the network architecture.

5.3.2 Effect of reward shaping in GRF. Before claiming the objective performance of BMRL, we must demonstrate the effect of the reward shaping we proposed in Section 4.3 on performance, proving that reward shaping is essential for directly improving the framework's performance. Therefore, we argue for the necessity of reward shaping by comparing the scoring performance and average reward in evaluation between BMRL-DRS, which removes all artificial reward shaping we proposed and BMRL. For BMRL-DRS, only the ±1 score reward is provided, which is the default reward provided by GRF. The evaluation was done using a built-in AI stochastic in 11 vs. 11 stochastic challenging scenarios.

Figure 7 shows that BMRL-DRS without reward shaping has a very low density of rewards in the environment besides the score reward, which causes it to struggle to learn compared to BMRL with various types of additional rewards. This shows that the reward-density change due to reward shaping positively impacts MAST's ability to optimize the policy.

Furthermore, to show that BMRL does not rely solely on reward shaping, we will demonstrate that BMRL-DRS without reward shaping can learn a distribution progressively closer to ρ_R as C_i increases from the goal state distribution ρ_1 through backward curriculum learning. To do this, we can use the L2 norm to define the difference between $s_{j,0}$ and ρ_G as a function of the ball's position, which is the most common criterion for determining the goal state. Thus, we can

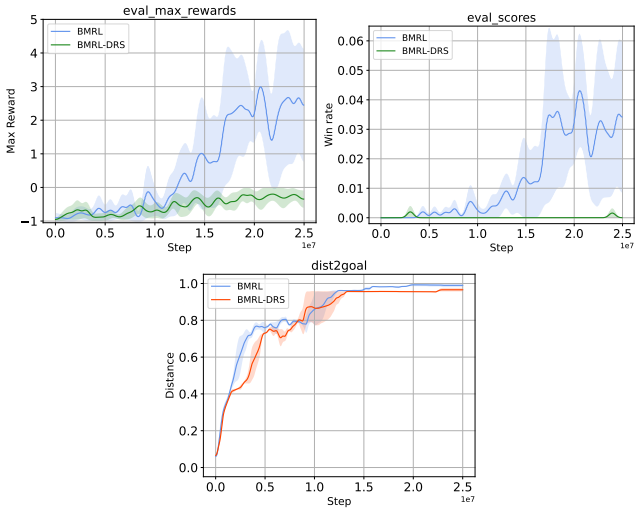

Figure 7: Performance comparison between DMRL-DRS and DMRL on GRF 11 vs. 11 full game scenario.

show that BMRL-DRS can learn the distribution in the neighborhood of ρ_R using backward curriculum learning by observing the gradual change of the ball's distance to the goal at $s_{j,0}$ like BMRL. The distance between the ball and the goal at ρ_R is 1. The dist2goal in Figure 7 shows that BMRL-DRS can learn distributions near ρ_R.

6 CONCLUSIONS

MARL should be able to learn long-term planning for environments with severe sparse reward or curse of dimensionality problems. In this paper, we propose a backward multi-agent reinforcement learning (BMRL) framework to solve these problems. BMRL allows the MAST algorithm with PE/PI property using Set Transformer to reduce the search space and solve the sparse reward problem in situations where the initial state distribution changes dynamically through backward curriculum learning. Using this, BMRL and MAST outperform the existing methods on SMAC and GRF benchmarks. In the future, we will build on this research to make backward curriculum learning more generalizable and applicable in various environments.

REFERENCES

[1] Ana L. C. Bazzan. 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Auton. Agents Multi Agent Syst.* 18, 3 (2009), 342–375.

[2] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research* 27, 4 (2002), 819–840.

[3] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4299–4307.

[4] Christian Schröder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip H. S. Torr, Wendelin Böhmer, and Shimon Whiteson. 2020. Deep Multi-Agent Reinforcement Learning for Decentralized Continuous Cooperative Control. *CoRR* abs/2003.06709 (2020).

[5] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse Curriculum Generation for Reinforcement Learning. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA,*

November 13-15, 2017, Proceedings (Proceedings of Machine Learning Research, Vol. 78). PMLR, 482–495.

[6] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2974–2982.

[7] Jianye Hao, Xiaotian Hao, Hangyu Mao, Weixun Wang, Yaodong Yang, Dong Li, Yan Zheng, and Zhen Wang. 2023. Boosting Multiagent Reinforcement Learning via Permutation Invariant and Permutation Equivariant Networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

[8] Somnath Hazra, Pallab Dasgupta, and Soumyajit Dey. 2024. Addressing Permutation Challenges in Multi-Agent Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2303–2305.

[9] Boris Ivanovic, James Harrison, Apoorva Sharma, Mo Chen, and Marco Pavone. 2019. BaRC: Backward Reachability Curriculum for Robotic Reinforcement Learning. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*. IEEE, 15–21.

[10] Jens Kober, J. Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *Int. J. Robotics Res.* 32, 11 (2013), 1238–1274.

[11] Ryan Kortvelesy, Steven D. Morad, and Amanda Prorok. 2023. Permutation-Invariant Set Autoencoders with Fixed-Size Embeddings for Multi-Agent Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). ACM, 1661–1669.

[12] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[13] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google Research Football: A Novel Reinforcement Learning Environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 4501–4510. https://doi.org/10.1609/aaai.v34i04.5878

[14] 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.

[15] Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E. Taylor, Wenyuan Tao, and Zhen Wang. 2022. PMIC: Improving Multi-Agent Reinforcement Learning with Progressive Mutual Information Collaboration. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 12979–12997.

[16] Fanqi Lin, Shiyu Huang, Tim Pearce, Wenze Chen, and Wei-Wei Tu. 2023. TiZero: Mastering Multi-Agent Football with Curriculum Learning and Self-Play. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). ACM, 67–76.

[17] Iou-Jen Liu, Raymond A. Yeh, and Alexander G. Schwing. 2019. PIC: Permutation Invariant Critic for Multi-Agent Deep Reinforcement Learning. In *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings (Proceedings of Machine Learning Research, Vol. 100)*, Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (Eds.). PMLR, 590–602.

[18] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *CoRR* abs/1706.02275 (2017).

[19] Kunal Menda, Yi-Chun Chen, Justin Grana, James W. Bono, Brendan D. Tracey, Mykel J. Kochenderfer, and David H. Wolpert. 2019. Deep Reinforcement Learning for Event-Driven Multi-Agent Decision Processes. *IEEE Trans. Intell. Transp. Syst.* 20, 4 (2019), 1259–1268.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* abs/1312.5602 (2013).

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.* 518, 7540 (2015), 529–533.

[22] Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, Ivan Bratko and Saso Dzeroski (Eds.). Morgan Kaufmann, 278–287.

[23] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. 2018. Credit Assignment For Collective Multiagent RL With Global Rewards. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 8113–8124.

[24] Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. 2008. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *J. Artif. Intell. Res.* 32 (2008), 289–353.

[25] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).

[26] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. arXiv e-prints, page. *arXiv preprint arXiv:1803.11485* (2018).

[27] Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 2186–2188.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017).

[29] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR* abs/1712.01815 (2017). arXiv:1712.01815

[30] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2017. Value-Decomposition Networks For Cooperative Multi-Agent Learning. *CoRR* abs/1706.05296 (2017).

[31] Stone Tao, Arth Shukla, Tse-kai Chan, and Hao Su. 2024. Reverse Forward Curriculum Learning for Extreme Sample and Demonstration Efficiency in RL. (2024).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008.

[33] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Çaglar Gülçehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat.* 575, 7782 (2019), 350–354.

[34] Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[35] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre M. Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[36] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia,*

December 1-4, 2020. IEEE, 737–744.