

Improving Zero-Shot Coordination Performance Based on Policy Similarity

Lebin Yu, Yunbo Qiu, Quanming Yao, Xudong Zhang, Jian Wang

Department of Electronic Engineering, Tsinghua University
 {yulb19,qyb18}@mails.tsinghua.edu.cn, quanmingyao@gmail.com, {zhangxd, jianwang}@tsinghua.edu.cn

Abstract

Over these years, multi-agent reinforcement learning has achieved remarkable performance in multi-agent planning and scheduling tasks. It typically follows the self-play setting, where agents are trained by playing with a fixed group of agents. However, in the face of zero-shot coordination, where an agent must coordinate with unseen partners, self-play agents may fail. Several methods have been proposed to handle this problem, but they either take a lot of time or lack generalizability. In this paper, we firstly reveal an important phenomenon: the zero-shot coordination performance is strongly linearly correlated with the similarity between an agent’s training partner and testing partner. Inspired by it, we put forward a Similarity-Based Robust TraininG (SBRT) scheme that improves agents’ zero-shot coordination performance by disturbing their partners’ actions during training according to a pre-defined policy similarity value. To validate its effectiveness, we apply our scheme to three multi-agent reinforcement learning frameworks and achieve better performance compared with previous methods.

Introduction

In recent years, multi-agent reinforcement learning has been attracting increasing attention for its broad applications in automated planning and scheduling tasks such as robot navigation (Li et al. 2019), traffic control (Calvo and Dusparic 2018; Ghosh et al. 2021) and fleet management (Lin et al. 2018; Bøgh et al. 2022). Unlike single-agent tasks, where the agent only needs to interact with a stationary environment, multi-agent setting brings instability caused by agents’ continuous updating of their policies (Lowe et al. 2017; Mahajan et al. 2019). To solve this problem, researchers let a fixed set of agents train and test together, which is called self-play (Tesauro 1994). Self-play makes it easy for agents to quickly understand partners’ policies and adjust their own plans. However, self-play-trained agents usually cannot cooperate with unseen agents well during testing (Hu et al. 2020).

In order to deeply study the generalization performance of cooperative agents, researchers put forward the concept of zero-shot coordination (Hu et al. 2020), where agents must coordinate with unfamiliar partners that they have not seen

before. Since then, several methods have been proposed to solve the problem, which can be divided into two categories. The first is population-based training (Shih et al. 2021; Lucas and Allen 2022), where agents are trained with multiple and diverse partners. This kind of method allows agents to learn more generalized policies, but is time-consuming. The second is avoiding specific conventions among self-play agents by making agents reason about the task (Hu et al. 2021; Cui et al. 2021) or breaking symmetries (Hu et al. 2020; Treutlein et al. 2021). Agents trained by this kind of framework can cooperate well with unseen partners trained by the same framework, but perform poorly when cooperating with partners trained by other frameworks.

In these papers, cross-play is adopted to evaluate zero-shot coordination performance, where individually trained agents are required to play with each other. However, experiments show that the cooperative performance of different combinations of agents varies widely, even if they are trained by the same framework with different random number seeds. We give an intuitive explanation for this phenomenon: If the testing partner is similar to an agent’s training partner, the agent might cooperate with this unseen partner well. Since the similarity between agents varies widely, so does the cross-play score. To measure the similarity, we define Conditional Policy Similarity between an agent’s TraininG partner and Testing partner (CPSTT), which is the expected probability of an agent’s training partner and testing partner taking the same action when coordinating with the agent. In addition, we conduct numerous experiments in a cooperative card game Hanabi (Bard et al. 2020) and reveal a strong linear correlation between CPSTT and cross-play scores.

Inspired by this linear correlation, we propose a light-weighted scheme, Similarity-Based Robust TraininG (SBRT), to improve the zero-shot coordination performance of agents. It randomly disturbs the training partners’ policies during training according to a pre-defined CPSTT value, in which way the overfitting of the agent to specific training partners is alleviated. We apply SBRT to three multi-agent reinforcement learning frameworks, IQL (Tan 1993), VDN (Sunehag et al. 2018) and SAD (Hu and Foerster 2019), and experiments confirm that their zero-shot coordination performance is improved.

Our contributions are summarized below:

1. We define CPSTT and conduct extensive experiments to

reveal its strong linear correlation with cross-play scores: cross-play scores increase almost linearly with it. This finding can provide a new perspective on zero-shot coordination.

2. Inspired by the linear correlation, we propose a light-weighted scheme SBRT that randomly perturbs agents’ partners’ policies during training according to a pre-defined CPSTT value.
3. We apply SBRT to three multi-agent reinforcement learning frameworks and experimentally confirm the effectiveness of our methods.

Background

Decentralized Partially Observable Markov Decision Process

We use a decentralized partially observable Markov decision process (Dec-POMDP) (Nair et al. 2003) to model a multi-agent cooperation task. The environment is partially observable with N agents in it. At timestep t , with the global environment state being $s_t \in \mathbf{S}$, agent i gets its observation $o_t^i \sim O(o|i, s_t)$ and chooses an action a_t^i according to its action-observation trajectory $\tau_t^i = \{o_0^i, a_0^i, \dots, o_t^i\}$ as well as its policy π^i : $a_t^i \sim \pi^i(a^i|\tau_t^i)$. The environment then feeds back a shared reward $r_t = R(s_t, \vec{a}_t)$ and updates the global state $s_{t+1} \sim T(s'|s_t, \vec{a}_t)$, where $\vec{a}_t = [a_t^1, \dots, a_t^N]$ is the joint action. The training goal is to maximize the expected return $J(\pi^1, \dots, \pi^N) = \mathbb{E}_{\pi^1, \dots, \pi^N} [\sum_t \gamma^t r_t]$.

To accomplish the training goal, researchers have proposed many multi-agent reinforcement learning frameworks, including policy-gradient-based ones (Foerster et al. 2018; Kuba et al. 2021) and value-based ones (Rashid et al. 2018; Son et al. 2019). They typically make a fix group of agents play and learn together so that agents can quickly learn cooperative behaviors.

Zero-shot Coordination and Cross-play

Following the common setting in this area (Hu et al. 2020), we formulate zero-shot coordination in two-agent scenarios. Suppose an agent with policy π is trained along with a partner agent with policy π_p , the training goal is described as follows:

$$\pi^*, \pi_p^* = \arg \max_{\pi, \pi_p} J(\pi, \pi_p) \quad (1)$$

If π and π_p continue to cooperate during testing, they are likely to get high scores. However, zero-shot coordination requires agents to coordinate with unseen partners. A similar setting is ad-hoc teamwork (Rahman et al. 2021; Zand, Parker-Holder, and Roberts 2022), where agents must model unfamiliar partners’ policies and adjust their own strategies during interaction. In comparison, zero-shot coordination does not allow agents to change their policies when facing unseen partners.

To evaluate agents’ performance in this setting, researchers commonly conduct cross-play, where individually trained agents are put together to cooperate during testing. Besides, the zero-shot coordination performance of a training framework is measured by the average cross-play score of multiple agents trained with this framework. Below we

give a formulaic representation of this process. Suppose the policy trained by a training framework M has a distribution P_M , then in the case that the testing partner is trained by a training framework M_o , the zero-shot coordination performance of M can be expressed as:

$$Z(M) = \mathbb{E}_{\pi \sim P_M, \pi_o \sim P_{M_o}} [J(\pi, \pi_o)] \quad (2)$$

It is evident that $Z(M)$ heavily relies on M_o . Based on the different choices of M_o , cross-play can be divided into two categories:

(1) Intra-algorithm cross-play: $M_o = M$, which means the unseen partners are trained with the same training framework but different random number seeds (Cui et al. 2021). This kind of test is easily accessible and objective, but its disadvantage is also obvious: it does not indicate whether the agents can cooperate well with agents obtained by other frameworks.

(2) Inter-algorithm cross-play: $M_o = (M_1 + M_2 + \dots + M_K)$, where M_1, M_2, \dots, M_K are other multi-agent reinforcement learning frameworks (Lucas and Allen 2022). This seems to be more comprehensive, however, the K tested frameworks may not represent all feasible frameworks, so the results may be biased.

To evaluate the zero-shot coordination performance of agents more comprehensively, we conduct both intra-algorithm and inter-algorithm cross-play in the experiments.

Conditional Policy Similarity

Cross-play scores of different combinations of agents vary a lot, even for the combinations of agents trained by the same framework. To account for this phenomenon, we make an intuitive guess that an agent should coordinate with an unseen partner well if this partner is similar to the agent’s training partner. In this section, we define CPSTT to measure this similarity, propose a way to estimate it, and present experimental results that indicate the strong linear correlation between it and cross-play scores.

Definition and Estimation of Conditional Policy Similarity

Below we give our definition of conditional policy similarity:

Definition 1. *In a two-agent game, the conditional policy similarity between π_1 and π_2 conditioned on π is:*

$$S_\pi(\pi_1, \pi_2) = \mathbb{E}_{\tau \sim P_\tau(\pi, \pi_1)} [\pi_1(\tau) = \pi_2(\tau)] \quad (3)$$

where $P_\tau(\pi, \pi_1)$ denotes the distribution of action-observation trajectory generated by π and π_1 playing with each other.

Conditional policy similarity measures how similar π_2 is to π_1 from π ’s perspective, and can be estimated in a Monte Carlo approach: Let π and π_1 play the game several times, and assume there are total n timesteps. Then, π_1 makes n decisions $\{\pi_1(\tau_1), \pi_1(\tau_2), \dots, \pi_1(\tau_n)\}$ based on n action-observation trajectories: $\{\tau_1, \tau_2, \dots, \tau_n\}$. Let π_2 acts based on these n trajectories, and then the estimate for $S_\pi(\pi_1, \pi_2)$

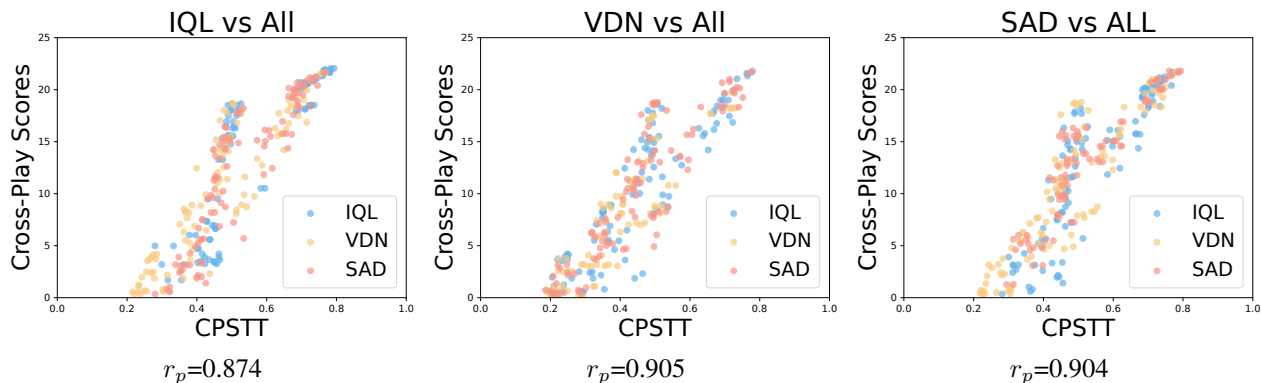


Figure 1: Each figure shows the detailed cross-play results of one kind of model with all kinds of models. r_p is Pearson correlation coefficient, a statistic describing the linear correlation between two factors ($r_p = 1$ represents a completely linear relationship). Generally speaking, the linear correlation between CPSTT and cross-play scores is strong.

becomes:

$$\bar{S}_\pi(\pi_1, \pi_2) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\pi_1(\tau_i)=\pi_2(\tau_i)} \quad (4)$$

In this paper, we focus on the conditional policy similarity between an agent’s training partner and testing partner (i.e. CPSTT) in zero-shot coordination. Given an agent with policy π , its training partner policy π_p and testing partner policy π_o , $\text{CPSTT} = S_\pi(\pi_o, \pi_p)$.

Testbed: Hanabi

We experiment on Hanabi (Bard et al. 2020), a cooperative card game often used to study zero-shot coordination. Each card has a color and a rank, and players are required to play cards in a legal order to complete five stacks of cards, one for each color. There are 5 colors and 5 ranks and the maximum score is 25. Hanabi has a typical cooperative reward scheme, where agents get rewarded for increasing team scores and punished for losing team life tokens. Note that each player can see the cards in everyone’s hand except its own, hence it must guess what their cards are based on others’ cues as well as provide valuable information for others.

Since this task requires agents to reason about the beliefs and intentions of partners and self-play agents are quite familiar with their training partners, they can achieve super-human performance (Hu and Foerster 2019). However, they easily learn special conventions to get high self-play scores. For example, one can hint ‘blue’ to let a partner play the first card from the left. This kind of trick helps quickly achieve self-play goals in (1) but does not help agents cooperate well with unfamiliar partners. As a result, zero-shot coordination is difficult in Hanabi, and a large part of the work in this field experiments on it (Hu et al. 2020; Treutlein et al. 2021; Lucas and Allen 2022; Hu et al. 2021).

Explore the Relationship Between CPSTT and Cross-play Scores

We conduct abundant experiments to see whether CPSTT and cross-play scores are correlated. In specific, we experiment on the three types of models: IQL, VDN and SAD.

They are typical self-play algorithms, and we apply parameter sharing to them to accelerate training, which is a widely-used technique (Christianos, Schäfer, and Albrecht 2020) for self-play in homogeneous multi-agent games. Therefore, the agent’s training partner is itself and CPSTT for these four kinds of models is $S_\pi(\pi_o, \pi)$. We train 10 models of each framework with different seeds and pair these 30 models to get 900 combinations. For each combination, we record the average scores of 10000 games and estimate $S_\pi(\pi_o, \pi)$ according to (4).

To visualize how conditional policy similarity affects cross-play scores, we exhibit the detailed cross-play results in Fig. 1. These figures show the same pattern: scores increase with similarities. This confirms our guess that if the testing partner is similar to an agent’s training partner, the cooperation tends to be good. Then how strong their linear correlation is exactly? We present the Pearson correlation coefficient (r_p) beneath each figure. The lowest is 0.874 (IQL), and the highest is 0.905 (SAD). For reference, $r_p = 1$ represents an unrealistically perfect correlation. Therefore, the linear correlation between CPSTT and cross-play scores is strong. Based on this fact, we can make a more comprehensive analysis of zero-shot coordination as well as propose new ways to increase cross-play scores.

Improving Cross-play Scores Based on Conditional Policy Similarity

Our experiments in the above section show that cross-play scores increase almost linearly with CPSTT. Therefore, cross-play scores can be increased in two ways based on this correlation. The first is to increase CPSTT. However, zero-shot coordination means cooperating with unseen partners, and it is almost impossible to increase the policy similarity between training partners and unknown agents. The second is more feasible, that is to improve cross-play scores while CPSTT is fixed, based on which we propose our scheme.

Similarity-Based Robust Training

Inspired by robust reinforcement learning (Mankowitz et al. 2018), we put forward a robust training scheme SBRT that can be applied to common self-play multi-agent reinforcement learning frameworks. The training objective of it is

$$\pi^*, \pi_p^* = \arg \max_{\pi, \pi_p} J(\pi, \pi_a) \quad s.t. CPSTT_{\pi}(\pi_p, \pi_a) = \alpha \quad (5)$$

In the first phase of training, we set $\alpha = 1$, which means the training is the same as self-play, allowing the agents to quickly optimize their policies. After training for N_{st} epochs, when the training is close to convergence, we set $\alpha = \alpha_r$ and perform robust training for N_{rt} epochs. α_r, N_{st} and N_{rt} are training hyperparameters. We implement π_a by disturbing the partner’s chosen action when it interacts with the environment. To be specific, given an action-observation trajectory τ and the action generated by π_p : $a_p = \pi_p(\tau)$,

$$\pi_a(\tau) = \begin{cases} a_p, & \text{with probability } \alpha \\ a_{alt} \neq a_p, & \text{with probability } 1 - \alpha \end{cases} \quad (6)$$

We test three ways of choosing a_{alt} :

Worst alternative action: This choice requires π_a to satisfy

$$\pi_a = \arg \min_{\pi_a} J(\pi, \pi_a), \quad s.t. CPSTT_{\pi}(\pi_p, \pi_a) = \alpha \quad (7)$$

This method follows the idea of adversarial training (Pan et al. 2019), where an adversary tries to minimize the main agent’s performance. Under a DQN-based framework with a Q-function $Q(\tau, a)$, $a_{alt} = \arg \min_a Q(\tau, a)$.

Best alternative action: This choice requires π_a to satisfy

$$\pi_a = \arg \max_{\pi_a} J(\pi, \pi_a), \quad s.t. CPSTT_{\pi}(\pi_p, \pi_a) = \alpha \quad (8)$$

The reason for this choice is that the policies of zero-shot coordination partners are usually not bad, hence this restriction might make π_a more like them. Under a DQN-based framework with a q function $Q(\tau, a)$, $a_p = \arg \max_a Q(\tau, a)$ and $a_{alt} = \arg \max_{a \neq a_p} Q(\tau, a)$.

Random alternative action: This solution simply picks a random feasible action as a_{alt} . It maximizes the exploration of possible unseen partner policies instead. We find in experiments that this is the best solution and choose it to implement SBRT in the main experiments.

Evaluation of SBRT

To see whether SBRT improves the zero-shot coordination performance of common self-play agents, we combine our scheme with three baseline multi-agent reinforcement learning frameworks: IQL, VDN and SAD. We also include OP (Hu et al. 2020) in the comparison, which breaks task-specific symmetries to increase cross-play scores. In this task, OP shuffles the color of the cards to prevent agents from forming special consensus (e.g. hinting ‘blue’ to let a partner play the first card from the left).

As for the training hyperparameters, $\alpha_r = 0.8$, $N_{st} = 400$ and $N_{rt} = 100$. Besides, we set the total training epochs of baseline and OP models to 500 to be consistent with SBRT. To increase the reliability of the results, we train

Model Type	Cross-Play Scores	
	Intra-Algorithm	Inter-Algorithm
IQL	13.55±0.99	12.39±0.21
IQL+OP	11.93±1.10	11.35±0.22
IQL+SBRT	15.72±0.70	13.25±0.20
VDN	7.70±0.86	9.97±0.20
VDN+OP	6.11±0.81	8.65±0.22
VDN+SBRT	11.38±0.92	12.14±0.20
SAD	14.12±0.86	11.32±0.20
SAD+OP	11.35±0.93	9.25±0.19
SAD+SBRT	14.70±0.77	13.19±0.19

Table 1: Mean and standard error of cross-play scores of baseline, OP and SBRT models

10 models with different random number seeds for each method.

We conduct inter-algorithm and intra-algorithm cross-play to measure the zero-shot coordination performance of the testing models. Intra-algorithm cross-play requires agents to cooperate with partners trained by the same framework. For example, IQL with IQL models, VDN+OP with VDN+OP models. Inter-algorithm cross-play requires agents to cooperate with partners trained by other frameworks. For example, for IQL, IQL+OP and IQL+SBRT models, their inter-algorithm cross-play partners are raw VDN and SAD models.

Results shown in Table 1 confirm that that SBRT effectively increases the inter-algorithm and intra-algorithm cross-play scores of IQL, VDN and SAD. An intuitive explanation for the validity of SBRT is that it diversifies the policies of agents training partners, thereby avoiding the formation of special consensus and allowing agents to perform better when cooperating with unseen testing partners.

Conclusion and Future Work

We focus on zero-shot coordination, where agents are required to cooperate with unseen partners. Several methods have been proposed to solve this problem, however, they either lack generalizability or take a lot of time. In this paper, we define conditional policy similarity and find it important in zero-shot coordination: cross-play scores are strongly correlated with CPSTT, and the Person correlation coefficient between them can be as high as 0.905. Based on this correlation, we propose a light-weighted scheme SBRT, that improves the zero-shot coordination performance of agents by perturbing the training partners’ policies according to a predefined CPSTT value. To confirm its effectiveness empirically, we apply it to three multi-agent reinforcement learning frameworks (IQL, VDN and SAD). Results show that SBRT successfully improves the inter-algorithm and intra-algorithm cross-play scores of these models. We hope the discovery of the strong correlation between cross-play scores and CPSTT along with our SBRT scheme provide a new perspective for research in this area.

References

- Bard, N.; Foerster, J. N.; Chandar, S.; Burch, N.; Lanctot, M.; Song, H. F.; Parisotto, E.; Dumoulin, V.; Moitra, S.; Hughes, E.; et al. 2020. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280: 103216.
- Bøgh, S.; Jensen, P. G.; Kristjansen, M.; Larsen, K. G.; and Nyman, U. 2022. Distributed Fleet Management in Noisy Environments via Model-Predictive Control. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, 565–573.
- Calvo, J. A.; and Dusparic, I. 2018. Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights Control. In *AICS*, 2–13.
- Christianos, F.; Schäfer, L.; and Albrecht, S. 2020. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10707–10717.
- Cui, B.; Hu, H.; Pineda, L.; and Foerster, J. 2021. K-level Reasoning for Zero-Shot Coordination in Hanabi. *Advances in Neural Information Processing Systems*, 34: 8215–8228.
- Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Ghosh, S.; Laguna, S.; Lim, S. H.; Wynter, L.; and Poonawala, H. 2021. A deep ensemble method for multi-agent reinforcement learning: A case study on air traffic control. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 468–476.
- Hu, H.; and Foerster, J. N. 2019. Simplified Action Decoder for Deep Multi-Agent Reinforcement Learning. In *International Conference on Learning Representations*.
- Hu, H.; Lerer, A.; Cui, B.; Pineda, L.; Brown, N.; and Foerster, J. 2021. Off-belief learning. In *International Conference on Machine Learning*, 4369–4379. PMLR.
- Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. 2020. “Other-Play” for Zero-Shot Coordination. In *International Conference on Machine Learning*, 4399–4410. PMLR.
- Kuba, J. G.; Chen, R.; Wen, M.; Wen, Y.; Sun, F.; Wang, J.; and Yang, Y. 2021. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *International Conference on Learning Representations*.
- Li, J.; Harabor, D.; Stuckey, P. J.; Felner, A.; Ma, H.; and Koenig, S. 2019. Disjoint splitting for multi-agent path finding with conflict-based search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, 279–283.
- Lin, K.; Zhao, R.; Xu, Z.; and Zhou, J. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1774–1783.
- Lowe, R.; WU, Y.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Advances in Neural Information Processing Systems*, 30: 6379–6390.
- Lucas, K.; and Allen, R. E. 2022. Any-Play: An Intrinsic Augmentation for Zero-Shot Coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 853–861.
- Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32.
- Mankowitz, D.; Mann, T.; Bacon, P.-L.; Precup, D.; and Mannor, S. 2018. Learning robust options. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJ-CAI*, volume 3, 705–711. Citeseer.
- Pan, X.; Seita, D.; Gao, Y.; and Canny, J. 2019. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, 8522–8528. IEEE.
- Rahman, M. A.; Hopner, N.; Christianos, F.; and Albrecht, S. V. 2021. Towards open ad hoc teamwork using graph-based policy learning. In *International Conference on Machine Learning*, 8776–8786. PMLR.
- Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4295–4304. PMLR.
- Shih, A.; Sawhney, A.; Kondic, J.; Ermon, S.; and Sadigh, D. 2021. On the critical role of conventions in adaptive human-AI collaboration. *arXiv preprint arXiv:2104.02871*.
- Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.
- Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2085–2087.
- Tan, M. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, 330–337.
- Tesauro, G. 1994. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2): 215–219.
- Treutlein, J.; Dennis, M.; Oesterheld, C.; and Foerster, J. 2021. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, 10413–10423. PMLR.
- Zand, J.; Parker-Holder, J.; and Roberts, S. J. 2022. On-the-fly Strategy Adaptation for ad-hoc Agent Coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 1771–1773.