CONDITIONAL POLICY SIMILARITY: AN OVERLOOKED FACTOR IN ZERO-SHOT COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reinforcement learning in cooperative tasks usually 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. Zero-shot coordination performance is traditionally measured by cross-play, where individually trained agents are required to play with each other. However, cross-play scores vary a lot for different combinations of agents, making it not reliable enough to only use a model's average cross-play scores with several models to evaluate its zero-shot coordination performance. We think the reason for this phenomenon may be that the cross-play score is highly related to the similarity between an agent's training partner and testing partner, and this similarity varies widely. Therefore, we define the Conditional Policy Similarity between an agent's Training partner and Testing partner (CPSTT) and conduct abundant experiments to confirm a strong linear correlation between CPSTT and cross-play scores. Based on it, we propose a new criterion to evaluate zero-shot coordination performance: a model is considered better if it has higher cross-play scores compared to another model given the same CPSTT. Furthermore, 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 CPSTT value. We apply our scheme to four multi-agent reinforcement learning frameworks and their zero-shot coordination performance is improved whether measured by the traditional criterion or ours.

1 INTRODUCTION

In recent years, multi-agent reinforcement learning has been attracting increasing attention for its broad applications in cooperative tasks such as robot navigation (Han et al., 2020), traffic light control (Calvo & Dusparic, 2018) and fleet management (Lin et al., 2018). To tackle the instability of multi-agent settings, numerous researchers design specialized training frameworks and get satisfying outcomes (Rashid et al., 2018; Wang et al., 2020). These algorithms usually follow the setting of **self-play** (Tesauro, 1994), where a fixed group of agents is trained and tested together. Self-play-based training makes it easy for agents to learn cooperative behaviors, however, self-play-trained agents might overfit training partners and cannot cooperate with unseen agents well.

In order to deeply study the generalization performance of cooperative agents, Hu et al. (2020) put forward the concept of zero-shot coordination, where agents must coordinate with unfamiliar partners they have not seen before. Since then, researchers have been proposing methods to solve the problem, such as breaking symmetries (Hu et al., 2020; Treutlein et al., 2021) and learning conventions (Shih et al., 2021). 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, we notice 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. Therefore, cross-play scores of one kind of model with a few kinds of models cannot perfectly represent its zero-shot coordination performance.

To deal with this problem, we first give an intuitive explanation for why cross-play scores vary: 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. To explore the relationship between it and cross-play scores, we conduct numerous experiments in a cooperative card game Hanabi (Bard et al., 2020). In specific, we train 40 agents of four types in two-agent settings and get 40x40=1600 different combinations of agents. Their cross-play results reveal a strong linear correlation between CPSTT and scores: the Pearson correlation coefficient r_p between them can reach up to 0.94[1]. That is to say, cross-play scores increase almost linearly with CPSTT.

Given this linear correlation, it seems inadequate to only rely on cross-play scores to measure zero-shot coordination performance. For example, if type-A models have high cross-play scores with type-C models while type-B models have low cross-play scores with type-C models, it does not necessarily mean that the zero-shot coordination performance of type-B models is worse compared to type-A models. The reason may be that CPSTT between type-A models and type-C models is higher. Therefore, we propose a new criterion to evaluate zero-shot coordination, that is if one model achieves higher cross-play scores than another one given an arbitrary CPSTT, then it is considered better. Note that given enough cross-play results, the relationship between cross-play scores and CPSTT can be approximated by a linear fit function, whose parameters can help reflect the proposed criterion: If the linear fit function of type-A models is above that of type-B models, type-A models have better zero-shot coordination performance.

Furthermore, based on CPSTT, we propose a light-weighted scheme, Similarity-Based Robust Training (SBRT), to improve the zero-shot coordination performance of agents trained with self-play. It randomly disturbs the training partners' policies during training according to a pre-defined CPSTT value, in this way the overfitting of the agent to specific training partners is alleviated. We apply SBRT to four multi-agent reinforcement learning frameworks, IQL (Tan, 1993), VDN (Sunehag et al., 2018), AUX and SAD (Hu & Foerster, 2019), and experiments confirm that their zero-shot coordination performance is improved whether measured by traditional methods or ours.

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) We notice that the traditional criterion to evaluate zero-shot coordination performance (evaluating only with the average cross-play scores) is not reliable enough since cross-play scores of different combinations of models vary a lot. Based on CPSTT, we propose a new criterion to help evaluate and analyze zero-shot coordination performance more comprehensively, that is if one model achieves higher cross-play scores than another one given an arbitrary CPSTT, then it is considered better. This criterion can better tell whether the agent can zero-shot coordinate well with an unseen partner.

(3) To improve the zero-shot coordination performance of agents trained with self-play, we propose a light-weighted scheme SBRT and conduct extensive experiments to confirm its effectiveness. This successful attempt holds promise for combining robust reinforcement learning with zero-shot coordination.

2 Related work

In recent years, multi-agent reinforcement learning has been widely used in multi-agent tasks and achieved good results (Mahajan et al., 2019; Kuba et al., 2021). Unlike single-agent reinforcement learning, 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). To solve this problem, researchers let a fixed set of agents train and test together, which is called self-play (Tesauro, 1994). Several techniques, such as centralized training and decentralized execution (Lowe et al., 2017; Wang et al., 2020) as well as value decomposition (Rashid et al., 2018;

[1]We give the definition of r_p in the Appendix. For reference, $r_p = 1$ represents an unrealistically perfect correlation.

Son et al., 2019), are employed to enhance understanding and cooperation between agents, but unexpectedly makes the agents overfit their training partners: Experiments in a cooperative card game Hanabi (Bard et al., 2020) show that self-play agents can achieve super-human performance, but their cooperation performance drops severely when paired with unfamiliar partners.

To study this problem in depth, Hu et al. (2020) define zero-shot coordination, which requires agents to coordinate with unseen partners. Some researchers conduct extensive experiments (Leibo et al., 2021) and find some intriguing phenomena, such as rule-based AI teammates cooperating better with humans than learning-based AI teammates (Siu et al., 2021). However, none of them point out the importance of CPSTT in zero-shot coordination.

Researchers have made several attempts to improve zero-shot coordination performance. Hu et al. (2020) propose "other-play" (OP) to break symmetries in self-play, which is further augmented with tie-breaking (Treutlein et al., 2021). Following the idea of avoiding specific conventions among self-play agents, new methods have been proposed in recent years (Cui et al., 2021; Hu et al., 2021), which tend to get high intra-algorithm cross-play scores, but performance on inter-algorithm cross-play is not guranteed (Lucas & Allen, 2022). Another kind of solution is to let agents learn more knowledge about the task rather than the consensus with specific partners. For example, Shih et al. (2021) propose a learning framework that forces the agents to learn rule-dependent representation and convention-dependent representation separately. Some others adopt population-based training and try to make agents work well with a diverse group of agents (Lupu et al., 2021; Strouse et al., 2021; Zhao et al., 2021). Our SBRT scheme is different from the above.

We note that zero-shot coordination is similar to robust reinforcement learning in some ways. A typical setting is model misspecification (Mankowitz et al., 2019; Tessler et al., 2019), where the environment dynamics during training are different from those during testing. If partners are treated as a part of the environment (Peysakhovich & Lerer, 2018), zero-shot coordination can be considered as a special model misspecification. One way to approach this problem is to actively perturb the environment dynamics while training (Mankowitz et al., 2019), inspired by which we propose SBRT.

3 PRELIMINARIES

3.1 ZERO-SHOT COORDINATION AND CROSS-PLAY

Following Hu et al. (2020), our scheme is based on a Dec-POMDP (Nair et al., 2003). The environment is partially observable with N agents in it. At time t, with the global environment state being $s_t \in \mathbb{S}$, 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 $\tau_t^i = \{o_0^i, a_0^i, ..., o_t^i\}$ 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 $r_t = R(s_t, \boldsymbol{a}_t)$ and updates the global state $s_{t+1} \sim T(s'|s_t, \boldsymbol{a}_t)$, where $\boldsymbol{a}_t = [a_t^1, ..., a_t^N]$ is the joint action. The training goal is to maximize the expected return $J(\pi^1, ..., \pi^N) = \mathbb{E}_{\pi^1,...,\pi^N}[\sum_t \gamma^t r_t]$.

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:

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

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. To evaluate agents' performance in this setting, researchers commonly conduct cross-play, where individually trained agents are put together to cooperate. 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:

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

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 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: $M_o = (M_1 + M_2 + ... + M_K)/K$, where $M_1, M_2, ..., M_K$ are other multi-agent reinforcement learning frameworks (Lucas & 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.

3.2 TESTBED: HANABI

We conduct our experiments in 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. 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. Players act in turns, and there are three types of operations: hinting, discarding a card, and playing a card. Hinting is costly, but it tells a partner the location of all cards of a certain color or a certain rank in its hand. If a card is played at the wrong time, everyone loses a life token. The game ends when the deck is emptied or three life tokens are lost.

Since this task requires agents to reason about the beliefs and intentions of partners and self-play agents are quite familiar with their training parters, they can achieve super-human performance (Hu & 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 Eqa.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 Hanabi (Hu et al., 2020; Treutlein et al., 2021; Lucas & Allen, 2022; Hu et al., 2021).

4 THE VARIETY OF CROSS-PLAY SCORES

Zero-shot coordination performance is commonly measured by intra-algorithm cross-play scores or inter-algorithm cross-play scores. However, we conduct experiments and find that cross-play scores vary a lot, and it may not be enough to make evaluation simply based on the average scores. In specific, we test four multi-agent reinforcement learning frameworks, IQL (Tan, 1993), VDN (Sunehag et al., 2018), AUX and SAD (Hu & Foerster, 2019) [2] in two-player Hanabi. We train 10 models of each framework with different seeds and pair these 40 models to get 1600 combinations. Detailed cross-play scores are present in Table 1. We also present self-play scores in the table for ease of comparison.

It can be seen that all these four models have good and close self-play scores. However, their cross-play scores vary a lot. For example, when zero-shot coordinating with VDN models, IQL and SAD models behave similarly (10.33$\pm$0.65 vs 10.19$\pm$0.65), however, when zero-shot coordinating with AUX models, IQL models are better than SAD models (15.04$\pm$0.54 vs 12.75$\pm$0.49). Based on these facts, we think only using cross-play scores to evaluate zero-shot coordination performance is not reliable enough. Besides, we try to figure out why cross-play scores of different combinations of agents vary a lot.

5 CONDITIONAL POLICY SIMILARITY

Why do cross-play scores of different combinations of agents vary a lot, even for the combinations of agents trained by the same framework? 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 give a formal definition of this similarity (which we name as CPSTT), propose a way to estimate it,

[2] Our code is modified based on the opensource codebase of Hu et al. (2021).

Table 1: Self-play and cross-play scores of four kinds of baseline models

	Self-play	Cross-play			
		with IQL	with VDN	with AUX	with SAD
IQL	23.07±0.08	13.55±0.99	10.33±0.65	15.04±0.54	12.50±0.66
VDN	22.98±0.09	10.34±0.65	7.70±0.86	10.46±0.56	10.19±0.65
AUX	23.57±0.06	15.04±0.54	10.46±0.56	21.55±0.11	12.75±0.49
SAD	23.20±0.06	12.50±0.66	10.19±0.65	12.75±0.49	14.12±0.86

and present experimental results that indicate the strong linear correlation between it and cross-play scores.

5.1 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:*

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

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 steps. Then, π_1 makes n decisions $\{\pi_1(\tau_1), \pi_1(\tau_2), ..., \pi_1(\tau_n)\}$ based on n action-observation trajectories: $\{\tau_1, \tau_2, ..., \tau_n\}$. Let π_2 acts based on these n trajectories, and then the estimate for $S_\pi(\pi_1, \pi_2)$ becomes:

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

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, CPSTT$= S_\pi(\pi_o, \pi_p)$.

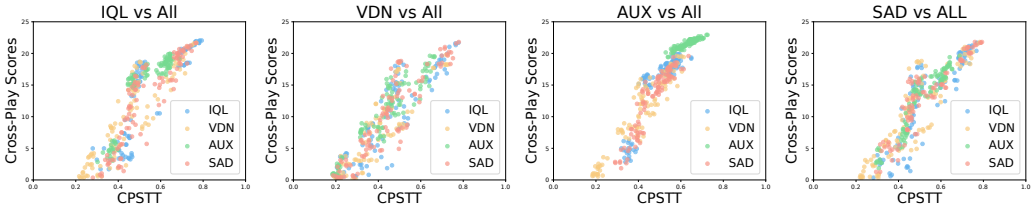
5.2 Explore the relationship between CPSTT and cross-play scores

We conduct abundant experiments to see whether CPSTT and cross-play scores are correlated. In specific, we experiment on the four types of models: IQL, VDN, AUX 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 et al., 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 40 models to get 1600 combinations. For each combination, we record the average scores of 10000 games and estimate $S_\pi(\pi_o, \pi)$ according to Eqa.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 0.943 (AUX). 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 ZSC as well as propose new ways to increase cross-play scores.

5.3 Evaluate zero-shot coordination performance based on CPSTT

Since cross-play scores vary a lot for different combinations of agents, it is not enough to simply measure a kind of model's zero-shot coordination performance by the average scores. It only reflects



r_p=0.874 y=41.9x-8.81 | r_p=0.904 y=37.2x-6.43 | r_p=0.943 y=48.3x-9.16 | r_p=0.904 y=40.20x-8.38

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. We also give the expression of the linear fit function approximating the relationship of cross-play scores (denoted as y) and CPSTT (denoted as x).

how this kind of model cooperates with a specific group of agents (i.e. the testing partners), but does not tell whether the agents can cooperate well with unseen agents, which heavily depends on CPSTT. Therefore, we propose a new criterion to evaluate zero-shot coordination performance: Suppose we need to compare the performance of two models π_1, π_2 given their training partners π_{1p}, π_{2p} and testing partner set P_{M_o}. We firstly calculate the cross-play score for π_i ($i = 1, 2$) given a CPSTT (i.e. $S_{\pi_i}(\pi_o, \pi_{ip})$):

$$J_s^{\pi_i}(x) = \mathbb{E}[J(\pi_i, \pi_o)|S_{\pi_i}(\pi_o, \pi_{ip}) = x \; and \; \pi_o \in P_{M_o}] \tag{5}$$

Then if $J_s^{\pi_1}(x) > J_s^{\pi_2}(x) \quad \forall x \in [0, 1]$, π_1 is considered better. In experiments, P_{M_o} contains finite models and it is infeasible to calculate $J_s^{\pi_i}(x)$ for an arbitrary x. However, given enough cross-play results, it can be approximated by a linear fit function, which can be obtained using ordinary least squares. For example, we present the linear fit function of each model in Fig. 1, and clearly AUX has the best zero-shot coordination performance.

Since cross-play scores are highly correlated with CPSTT and our criterion focus on the cross-play score of an agent given an arbitrary CPSTT, we believe this criterion can more comprehensively reflects zero-shot coordination performance.

Table 2: Coefficients of variation of estimated values

Cross-Play Scores : Std/Mean			
	IQL-2	VDN-2	AUX-2
IQL-1	0.004	0.015	0.005
VDN-1	0.009	0.009	0.018
AUX-1	0.002	0.030	0.002
Estimates of CPSTT : Std/Mean			
IQL-1	0.001	0.003	0.002
VDN-1	0.001	0.002	0.002
AUX-1	0.002	0.005	0.002

5.4 Investigative Experiments

5.4.1 Is our estimation of conditional policy similarities accurate enough?

In our experiments, for a combination of agents, the cross-play scores are obtained by averaging the results of 10000 games, and CPSTT is estimated with Eqa. 4 and data provided by 10000 games. To figure out whether our estimation is precise enough, we test 9 different combinations of 6 models (IQL-1,IQL-2,VDN-1,VDN-2,AUX-1,AUX-2). For each combination, we run the test 100 times (each test representing 10000 games) and get 100 sets of estimated values. Then we calculate the Std/Mean (i.e. coefficients of variation) of the estimated values to see if our estimation is accurate enough (Mean can be treated as the true value and Std can be treated as MSE of the estimation) and present them in Table 2. It can be seen that the standard deviation is quite small compared to the mean both for cross-play scores and CPSTT , which confirms the accuracy of our estimation.

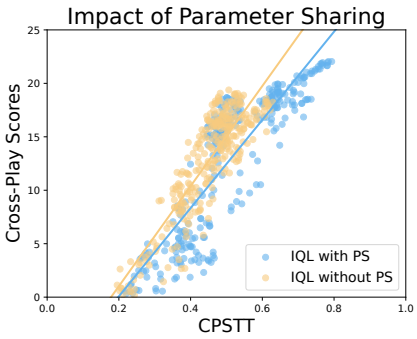
5.4.2 What if parameter sharing is not used?

In the above experiments, parameter sharing is enabled for self-play models to accelerate training, which is a practical and general technique. In this case, CPSTT is exactly the conditional policy

similarity between itself and the zero-shot coordination partner. What if parameter sharing is not used? Will the strong correlation between cross-play scores and conditional policy similarity still exist? To answer this question, we train 10 IQL agents without parameter sharing, make them cross-play with three kinds of models, (IQL, VDN and AUX) and exhibit the results in Fig. 2. Besides, cross-play results of IQL agents trained with parameter sharing are exhibited together for comparison. The results confirm the fact that cross-play scores are still strongly correlated with conditional policy similarity when the models are trained without parameter sharing. It is also worth noticing that agents trained without parameter sharing tend to have better zero-shot coordination performance. This may be because parameter sharing facilitates the formation of special conventions among training partners, exacerbating overfitting.

6 Improve 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.



(a)

r_p=0.872 (without parameter sharing)

Figure 2: Cross-play results of IQL models (trained with and without parameter sharing) with three kinds of base models. It can be seen that the linear correlation between CPSTT and cross-play scores is still strong in non-parameter-sharing settings.

6.1 Similarity-Based Robust Training

Inspired by robust reinforcement learning, 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

$$\pi^*, \pi_p^* = \arg\max_{\pi,\pi_p} J(\pi, \pi_a) \;\; s.t. CPSTT_\pi(\pi_p, \pi_a) = \alpha \tag{6}$$

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: $a_p = \pi_p(\tau)$,

$$\pi_a(\tau) = \begin{cases} a_p, & with\ probability\ \alpha \\ a_{alt} \neq a_p, & with\ probability\ 1-\alpha \end{cases} \tag{7}$$

We test three ways of choosing a_{alt}:

Worst alternative action: This choice requires π_a to satisfy

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

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

$$\pi_a = \arg\max_{\pi_a} J(\pi, \pi_a), \quad s.t. CPSTT_\pi(\pi_p, \pi_a) = \alpha \tag{9}$$

Table 3: Averaged Cross-Play Scores of OP and SBRT Models

Model Type	Intra-Alg	Inter-Alg	Model Type	Intra-Alg	Inter-Alg
IQL	13.55±0.99	12.39±0.21	AUX	**21.55±0.11**	13.05±0.19
IQL+OP	11.93±1.10	11.35±0.22	AUX+OP	21.52±0.15	**13.27±0.20**
IQL+SBRT	**15.72±0.70**	**13.25±0.20**	AUX+SBRT	21.05±0.10	12.14±0.18
VDN	7.70±0.86	9.97±0.20	SAD	14.12±0.86	11.32±0.20
VDN+OP	6.11±0.81	8.65±0.22	SAD+OP	11.35±0.93	9.25±0.19
VDN+SBRT	**11.38±0.92**	**12.14±0.20**	SAD+SBRT	**14.70±0.77**	**13.19±0.19**

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. Experiments show that this solution has the best overall performance (see Appendix. B for experiment details).

6.2 Evaluation of SBRT

In this subsection, we combine our scheme with IQL, VDN, AUX and SAD to see whether SBRT improves the zero-shot coordination performance of common self-play agents. We also include OP (Hu et al., 2020) in the comparison, which breaks symmetries to increase cross-play scores. We choose random alternative action as a_{alt} for SBRT and set $\alpha_r = 0.8$. Besides, we set $N_{st} = 400, N_{rt} = 100$ to make the total training epochs of SBRT models equal to that of baseline and OP models.

We test the cross-play performance of them paired with all models to get a more comprehensive analysis and visualize detailed results in Fig. 3. The linear fit functions of SBRT models are higher than that of OP models and baseline models, especially for VDN and SAD, so the zero-shot coordination performance of SBRT models is better from the view of our proposed criterion. To further prove the effectiveness of our proposed scheme, we use traditional ways to evaluate the models, presenting average intra-algorithm and inter-algorithm cross-play scores in Table 3. It can be seen that SBRT effectively increases the cross-play scores of IQL, VDN and SAD. However, it slightly reduces the cross-play scores of AUX, which is caused by SBRT lowering CPSTT of AUX models. In general, our scheme improves zero-shot coordination performances of self-play models whether measured by the traditional criterion or ours.

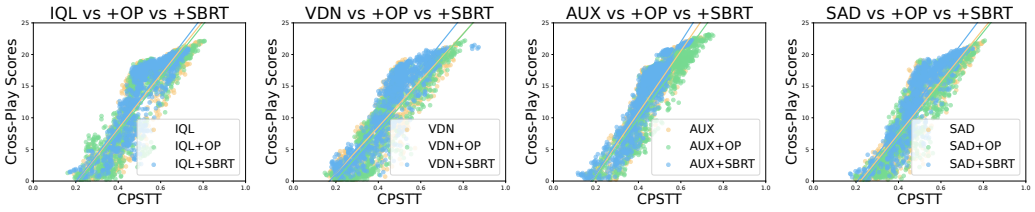


Figure 3: Detailed cross-play results of baseline models as well as OP and SBRT models. Judging from the linear fit function that approximating the relationship between cross-play scores and CPSTT, SBRT models have the best zero-shot coordination performance.

6.3 How α_r affects the performance of SBRT?

In our experiments, we set $\alpha_r = 0.8$ for SBRT, whose value represents the strength of policy disturbance while training. Note that $\alpha_r = 1$ means no disturbance at all, thus when $\alpha_r > 0.8$, SBRT will have less impact. What if we set it to a smaller value, i.e $\alpha_r = 0.6$? We visualize how α_r affects the performance of IQL+SBRT, VDN+SBRT and SAD+SBRT in Fig. 4. It can be seen that SBRT models with $\alpha_r = 0.6$ have lower cross-play scores as well as lower conditional policy

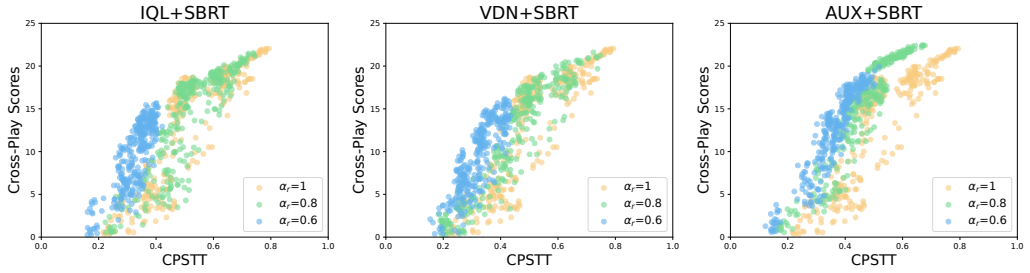


Figure 4: Detailed cross-play results of SBRT models trained with different α_r. Note that $\alpha_r = 1$ indicates baseline models.

similarities, which is because small α_r indicates strong disturbance on training agents' policies, and too strong disturbance makes training difficult.

7 Conclusion and Future Work

In this paper, we focus on zero-shot coordination, where agents are required to cooperate with unseen partners.Researchers commonly rely on average cross-play scores with a group of testing partners to evaluate agents' performance in this setting, however, we notice that cross-play scores of different combinations of agents vary a lot. As a consequence, the average cross-play scores heavily depend on the choice of testing partners, making the criterion not reliable enough. To handle this problem, we firstly 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 0.943. Based on this correlation, we propose a new criterion to evaluate zero-shot coordination, that is if one model achieves higher cross-play scores than another one given an arbitrary CPSTT, then it is considered better. Furthermore, the role of conditional policy similarity goes beyond assisting in the assessment. Its strong correlation with cross-play scores also indicates new ways to improve the scores. We propose a light-weighted scheme SBRT, that aims to improve the zero-shot coordination performance of agents, whose effectiveness is confirmed by adequate experiments.

Current work on zero-shot coordination mainly focuses on making agents learn intrinsic rules of cooperative games or avoiding special conventions between training partners. We hope the discovery of the strong correlation between cross-play scores and CPSTT along with our SBRT scheme inspired by robust reinforcement learning provide a new perspective for research in this area.

Our work can be extended in several ways. Firstly, we define conditional policy similarity in scenarios with discrete actions, and the definition may be extended to scenarios with continuous actions. Secondly, the definition of CPSTT can be extended to population-based training, where each agent has several training partners. Thirdly, SBRT can be further improved by wisely disturbing certain actions as some robust reinforcement learning frameworks do.

References

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Jeancarlo Arguello Calvo and Ivana Dusparic. Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In *AICS*, pp. 2–13, 2018.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 2020.

Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Foerster. K-level reasoning for zero-shot coordination in hanabi. *Advances in Neural Information Processing Systems*, 34:8215–8228, 2021.

Ruihua Han, Shengduo Chen, and Qi Hao. Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 448–454. IEEE, 2020.

Hengyuan Hu and Jakob N Foerster. Simplified action decoder for deep multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In *International Conference on Machine Learning*, pp. 4369–4379. PMLR, 2021.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2021.

Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*, pp. 6187–6199. PMLR, 2021.

Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. 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*, pp. 1774–1783, 2018.

Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30:6379–6390, 2017.

Keane Lucas and Ross E Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 853–861, 2022.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pp. 7204–7213. PMLR, 2021.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

Daniel J Mankowitz, Nir Levine, Rae Jeong, Abbas Abdolmaleki, Jost Tobias Springenberg, Yuanyuan Shi, Jackie Kay, Todd Hester, Timothy Mann, and Martin Riedmiller. Robust reinforcement learning for continuous control with model misspecification. In *International Conference on Learning Representations*, 2019.

Ranjit Nair, Milind Tambe, Makoto Yokoo, David Pynadath, and Stacy Marsella. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI*, volume 3, pp. 705–711. Citeseer, 2003.

Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8522–8528. IEEE, 2019.

Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2043–2044, 2018.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Andy Shih, Arjun Sawhney, Jovana Kondic, Stefano Ermon, and Dorsa Sadigh. On the critical role of conventions in adaptive human-ai collaboration. *arXiv preprint arXiv:2104.02871*, 2021.

Ho Chit Siu, Jaime Peña, Edenna Chen, Yutai Zhou, Victor Lopez, Kyle Palko, Kimberlee Chang, and Ross Allen. Evaluation of human-ai teams for learned and rule-based agents in hanabi. *Advances in Neural Information Processing Systems*, 34:16183–16195, 2021.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515, 2021.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 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*, pp. 2085–2087, 2018.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224. PMLR, 2019.

Johannes Treutlein, Michael Dennis, Caspar Oesterheld, and Jakob Foerster. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, pp. 10413–10423. PMLR, 2021.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning*, pp. 9876–9886. PMLR, 2020.

Rui Zhao, Jinming Song, Hu Haifeng, Yang Gao, Yi Wu, Zhongqian Sun, and Yang Wei. Maximum entropy population based training for zero-shot human-ai coordination. *arXiv preprint arXiv:2112.11701*, 2021.

A PEARSON CORRELATION COEFFICIENT AND LINEAR FIT FUNCTION

The linear correlation between two factors is commonly measured by the Pearson correlation coefficient (r_p), which is the covariance of the two factors divided by the product of their standard deviations. Given a set of values of two factors X and Y: $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, r_p can be calculated as:

$$r_p = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \tag{10}$$

r_p ranges between $[-1, 1]$, and the closer its absolute value is to 1, the stronger the linear correlation between the two factors is. In this case, the relationship between X and Y can be approximated with a linear fit function $y = kx + b$, whose parameters are calculated using ordinary least squares:

$$k, b = \arg\min_{k,b} \sum_{i=1}^{n} (y_i - kx_i - b)^2 \tag{11}$$

B DETAILS: HOW DIFFERENT a_{alt} CHOICES AFFECT SBRT?

We try three variants of SBRT, where SBRT(A) means best alternative action, SBRT(B) means worst alternative action and SBRT(C) means random alternative action. We apply them to IQL, VDN and AUX and execute cross-play. Results are detailed in Fig. 5. In general, SBRT(C) can improve the zero-shot coordination performance of self-play agents the most. Sometimes solution A and B improve cross-play scores more when CPSTT is high but is not so good when CPSTT is low. We speculate that solution C maximizes the diversity of π_a so that the main agent can better zero-shot coordinate with more agents.

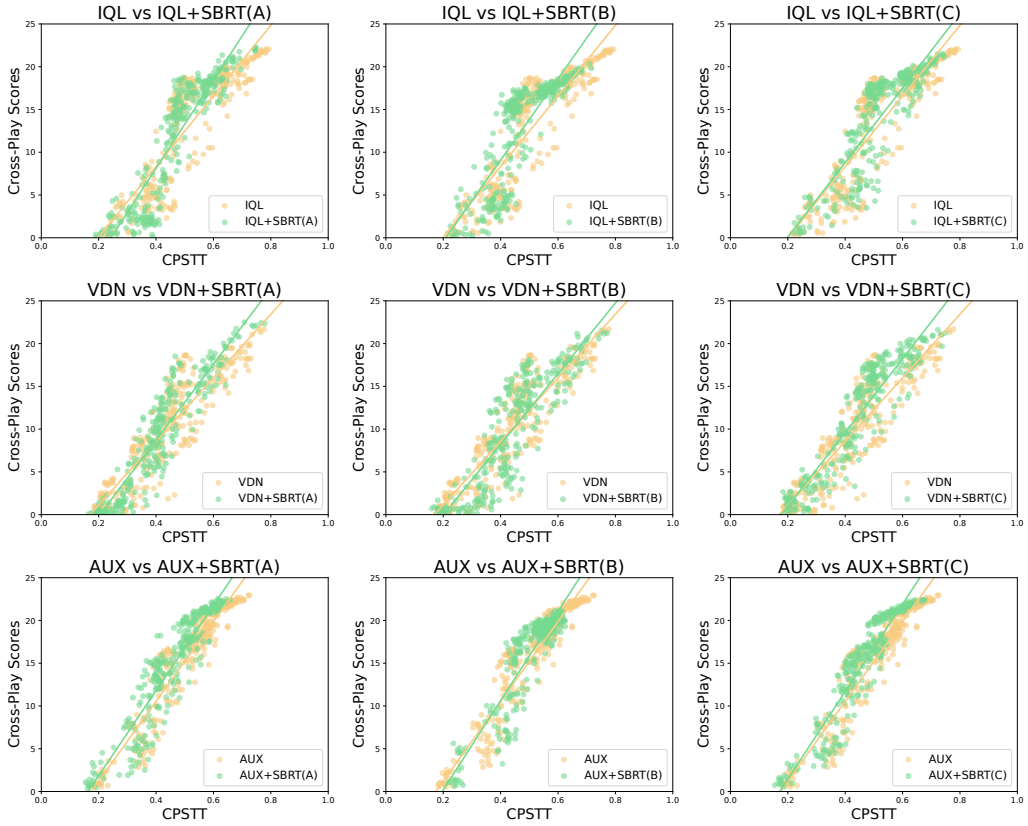


Figure 5: Detailed cross-play results of different types of SBRT models.