Policy Representation Opponent Shaping via Contrastive Learning

Yuming Chen[1] and Yuanheng Zhu[1,2]

[1] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
chenyuming@ucass.edu.cn
yuanheng.zhu@ia.ac.cn

Abstract. To acquire results with higher social welfare in social dilemmas, agents need to maintain cooperation. Independent agents manage to navigate social dilemmas via opponent shaping. However, opponent shaping needs extra information of opponent. It is not always accessible in mixed tasks if agents are decentralized. To address this, We present PROS, which runs in a fully-independent setting and needs no extra information. PROS shapes the opponent with an extended policy that takes the opponent's dynamics as additional input. Instead of receiving policy from the opponent, we discriminate the policy representation via contrastive learning. In terms of experiments, PROS reaches the optimal Nash equilibrium in iterated prisoners' dilemma (IPD) and shows the same ability to maintain cooperation in Coin Game, a highly-dimensional version of IPD. The source code is available on *https://github.com/RandSF/Policy-Representation-Opponent-Shaping*.

Keywords: multi-agent systems · reinforcement learning · contrastive learning

1 Introduction

Besides fully-cooperative tasks [9] and zero-sum tasks [29], mixed cooperative-competitive tasks [3, 16] are gathering importance in multi-agent reinforcement learning (MARL), including self-driving cars [10], multi-robot control [12], etc. One major meaningful type of mixed task is the social dilemma, where there is a trade-off between the benefits for specific individual agents and higher social welfare, the average reward in MARL, for the system. To maintain high welfare, self-interested agents are expected to cooperate in social dilemmas. MARL agents are able to perform good policy in many mixed tasks with centralized training decentralized execution (CTDE) paradigm [16, 26, 28]. Such algorithms require a centralized aggregator who can obtain all information about the system. Learning of the aggregator becomes hard when the number of agents in the system goes larger. Is it able to learn cooperation spontaneously for decentralized independent agents?

Specifically, independent agents can address such social dilemmas via opponent shaping [5, 13, 27]. Opponent shaping takes the opponent's policy as a function of its own policy and updates its own policy by shaping the learning process of the opponent. In iterated prisoners' dilemma [1], where two agents play the prisoners' dilemma for infinite times, learning with opponent learning awareness (LOLA) [5] is the first autonomous learning agent to discover *tit-for-tat* (TFT) [1], i.e. performing cooperation first and copying the action opponent performed last time. Model-free opponent shaping (M-FOS) [17] learns a meta policy to defeat another algorithm via meta-game and succeeds to shape LOLA in IPD.

However, LOLA requires white-box access to the opponent's policy and assumes the opponent is a naive learner. Another weakness of LOLA is the low sample efficiency. M-FOS fails to learn a meta policy in large-scale environments, and alternatively, it learns a conditional vector to replace the meta policy from the global information of the system. Such information is not always accessible to independent agents.

In this paper, we introduce policy representation opponent shaping (PROS) to resolve the above problems. Without obtaining the opponent's action, PROS represents the opponent's dynamic change via contrastive learning to implement opponent shaping. Our contributions are two-fold as followed:

- We propose a framework that formulates the learning process in multi-agent environments as a meta-game. Agents can implement opponent shaping by learning an opponent-dynamics conditional policy in the game.
- We derive PROS, which uses a contrastive learning algorithm to learn the opponent-dynamics representation under a fully-independent setting, i.e. agents are trained decentralized and have no access to information about the opponent. With a proper learning objective, PROS learns faster than other algorithms with latent representation.

We show the feasibility to shape the opponent when the opponent's policy is unknown. Then we explain why the dynamics of the opponent can be discriminated with no extra information, and present a contrastive learning module to achieve such discrimination. In terms of experiments, We evaluate PROS in IPD, iterated matching pennies (IMP), and coin game. The result shows that PROS leads to equilibria with higher social welfare compared with baselines.

2 Relative Work

Opponent Modeling. To navigate the social dilemma as an independent agent, a number of works focus on opponent modeling, which reconstructs some aspect of the policy of the opponent. One straightforward idea is modeling the opponent's policy and using the predicted action [18]. The family of learning and influencing latent intent(LILI) [10, 20, 25] learns a latent policy of the other agents by high-level representation in order to influence the long-term behaviour of partner [10]. Comparing opponent shaping and PROS, such work pays more attention to cooperative environments and requires agents to learn how to exploit or co-adapt with the other agents.

Opponent Shaping: By influencing the future actions of the opponent, opponent shaping exploits the opponent whose policy is known. The family of LOLA updates the policy with the awareness that its own policy influences the opponent's learning process. Most of the work [13, 23, 27] focus on the convergence or consistency in self-play and assume white-box access to the opponent's model. To relax this assumption, M-FOS [17] learns in a meta-game, where each meta-step is an episode of the underlying[3] game. Agents take inner policy as a meta-action from meta-policy. Directly sampling inner policy from meta-policy is intractable in high-dimensional games and it has to replace meta-policy with cross-episode conditional vectors without theoretical guarantee. PROS uses latent representation instead of meta-policy, thus it learns in the same way in games on any scale.

Contrastive Learning. Technically, PROS uses contrastive learning to learn the representation of policy. There is a large body of research on applying contrastive learning to reinforcement learning, like pixel representation learning [11,15] and the exploration [7,21]. Contrastive learning is also used in meta-learning [14] and policy representation [6]., which are fields relative to our work. However, PROS is under a fully decentralized training process, that is, PROS uses no information from the opponent.

3 Background

3.1 Learning Process as an Repeated Game

In this section, we describe the learning process of agents and formulate the process to be an repeated game.

We consider several players learn to optimise their own individual objectives in a game $\mathcal{G} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \Omega, \mathcal{O}, \mathcal{P}, r, \gamma \rangle$, where $\mathcal{I} = \{1, 2, \dots, n\}$ is the set of n players. $\mathcal{S}$ denotes the state space. $\mathcal{A} = \times_{i \in \mathcal{I}} \mathcal{A}^i$ is the action space. Observation space $\Omega = \times_{i \in \mathcal{I}} \Omega^i$ and observation function $\mathcal{O} : \mathcal{S} \times \mathcal{A} \times \Omega \mapsto [0, 1]$ decide the observation of players. $r = \times_{i \in \mathcal{I}} r^i$ represents the reward function. $\mathcal{P}$ is the transition function of the state.

$\mathcal{G}$ can be regarded as a partially observable Markov decision process (POMDP). An agent learns in the POMDP to obtain an optimal policy $\pi^i : \mathcal{O}^i \mapsto \mathcal{A}^i$ that maximises the objective, for simplicity, we set it to be accumulated discounted rewards, i.e. discounted return $R^i(\pi^i, \pi^{-i}) := \sum_{t=0}^{T} \gamma^t r^i(s_t, a_t)$, where actions are sampled from policy $\pi^i(o_t^i)$, $\pi^{-i}(o_t^{-i})$ respectively. R^i is a mapping from joint policy space to the reals. An RL agent updates its parameterized policy π_ϕ^i based on the discounted return, which can be formalized to be

$$\pi_{k+1}^i = \pi_k^i + \alpha^i \nabla_\phi R^i(\pi^i | \pi_k^{-i})|_{\pi_k^i} \tag{1}$$

[3] In this paper, we use 'underlying' or 'inner' to refer to the game agents actually playing, and 'meta' or 'outer' to refer to the meta-game we introduce here.

where α^i is the learning rate and ϕ is the policy parameters of agent i. We will omit parameter ϕ at the notation of policy if the policy π^i_k refers to those in a certain epoch k. It should be noted that policies π in Equation (1) are not always regarded as a function of ϕ.

The return R^i implicitly depends on the game $\mathcal{G}$ besides policies. Thus the underlying game can be regarded as a function mapping joint policy to returns. We can formulate the learning process from initial policies to Nash equilibrium as a repeated game, which is the *meta-game*. In each round, agents perform a policy, get returns and update their policies.At the round k of the meta-game, agents interact in the underlying game $\mathcal{G}$ with policy π^i_k, π^{-i}_k and get discounted inner return $R^i_k = R^i(\pi^i_k, \pi^{-i}_k),\ R^{-i}_k = R^{-i}(\pi^{-i}_k, \pi^i_k)$.

A naive learner's learning process can be seen as learning to maxising R^i, regardless of the opponent, that is, the opponent policy π^{-i} is *not* the function of ϕ. For a naive learner, Equation (1) becomes

$$\pi^i_{k+1} = \pi^i_k + \alpha^i \nabla_\phi R^i(\pi^i(\phi)|\pi^{-i}_k)|_{\pi^i_k}$$

where $\pi^i(\phi)$ means π^i is parameterized by ϕ.

LOLA-like learners [5,13,23,27] have awareness of the opponent's policy and have white-box access to it. That is, LOLA-like learners know how the opponent updates, i.e., the opponent policy π^{-i} is accessible and $\pi^{-i}_{k+1} - \pi^{-i}_k$ (not π^{-i}) is determined by ϕ. Thus Equation (1) becomes

$$\begin{aligned}\pi^i_{k+1} &= \pi^i_k + \alpha^i \nabla_\phi R^i(\pi^i(\phi), \pi^{-i})|_{(\pi^i_k, \pi^{-i}_{k+1})} \\ &= \pi^i_k + \alpha^i \nabla_\phi \left[\nabla_1 R^i(\pi^i(\phi), \pi^{-i}) + \left(\mathit{\Delta}\pi^{-i}_k(\phi)\right)^\top \nabla_2 R^i(\pi^i(\phi), \pi^{-i})\right]_{(\pi^i_k, \pi^{-i}_k)}\end{aligned}$$

according to the Taylor expansion, where ∇_1 refers to the gradient w.r.t. the first variant and ∇_2 refers to the gradient w.r.t. the second variant. $\mathit{\Delta}\pi^{-i}_k$ is $\pi^{-i}_{k+1} - \pi^{-i}_k$.

One improvement of LOLA is M-FOS, which learns the optimal meta-policy via PPO [22] or Genetic Algorithm. An M-FOS learner maintains a meta-policy $\tilde{\pi}$ and directly samples policy π from it. M-FOS learner learns the meta-policy by optimising the discounted accumulation of returns

$$\tilde{\pi}^i_{k+1} = \tilde{\pi}^i_k + \alpha^i \nabla_\pi \sum_{k=0}^{K} \eta^k R^i(\pi^i_k, \pi^{-i}_k) \tag{2}$$

where η is the meta-discount factor set and K is the estimation of rounds to reach the equilibrium. Both of them are hyperparameters.

Since policy π^i is directly sampled before each interaction round, original M-FOS can only work in tabular environments where state and action spaces are not large. In large-scale environments, M-FOS uses a conditional vector c to avoid directly generating policy π^i or accessing opponent policy π^{-i}. In round k, conditional vector c_k is encoded from the trajectory of the last round

$$c_k = \mathcal{E}(\tau_{k-1}, c_{k-1}), \qquad \tau_{k-1} := \{(o^i_t, o^{-i}_t, a^i_t, a^{-i}_t, r^i_t, r^{-i}_t)\}$$

where $\mathcal{E}$ is the vector encoder. Action is sampled from policy π_k^i, i.e. $a_t^i \sim \pi_k^i(o_t^i, c_k^i)$.

Both of the vector encoder $\mathcal{E}$ and policy π_ϕ are updated according to Equation (2) every T rounds.

4 Method

M-FOS uses different implementations in tabular environments and highly dimensional environments since π^{-i} is infinitely dimensional in large-scale environments. To overcome such weakness, PROS learns an opponent-dynamics conditional policy $\pi_\phi^i(o_t^i, z_t)$ to implement opponent shaping. The opponent dynamic z is inferred from local trajectories $\tau^i = \{(o_t^i, a_t^i, r_t^i)\}$ via contrastive learning. Such a policy-representing module is agnostic to RL algorithms, thus it can work as a plug-in in RL algorithms. The network structure is illustrated in Fig. 1.

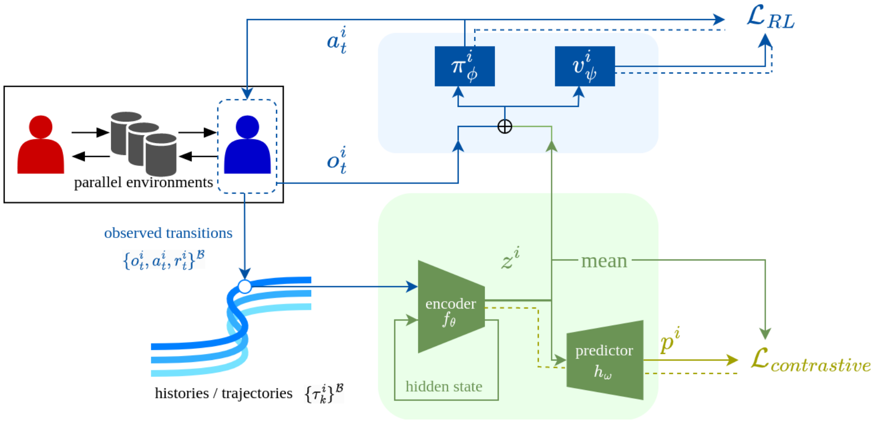

Fig. 1: Training of policy representation encoder. The blue area is the RL module (we use PPO in this paper), and the green area is the Dynamics encode module. The learning of opponent-dynamics encoder and policy are decoupled in terms of gradients, which is denoted as the dotted line.

4.1 Learning via Opponent-Dynamics Conditional Policy

PROS learns a latent z from local trajectory to have *meta-learning awareness* instead of learning the meta-game directly. The policy makes dicisions based on observation o^i and latent z.

Different latent z refer to different opponent's policies. However, we do not require the invertibility of z. There is no guarantee that π^{-i} can be reconstructed from z. It is a significant difference between PROS and works involving opponent modeling.

Similar to many algorithms in meta-learning, PROS optimises the accumulation of discounted inner returns of multiple episodes to estimate both the current policy's performance and the current learning step's performance.

$$\sum_{k=0}^{K} \gamma^k R^i(\pi^i_{j+k} | \pi^{-i}_{j+k}) \tag{3}$$

where K is the memory capacity of the agent and γ plays the same role as η in M-FOS.

Algorithm 1: PROS-PPO

```
Initialize encoder parameters $\theta$, predictor parameters $\omega$, actor parameters $\phi$,
  and critic parameters $\psi$;
Initialize policy representation vector $z$;
while true do
    for $k = 0$ to $K$ do
        reset buffer $\mathcal{D}$;
        reset parallel environment;
        for $t = 0$ to $T$ do
            sample action $a^i_t \sim \pi_\phi(o^i_t, z)$;
            apply the action to get new observation $o^i_{t+1}$ and reward $r^i_t$;
            encode current dynamics $z \leftarrow f_\theta(o^i_t, a^i_t, r^i_t, z)$;
            update parameters $\theta$ and $\omega$ with $\mathcal{L}_{contrastive}(z)$;
            // the loss is computed according to Equation (5)
            store data $(o^i_t, a^i_t, r^i_t)$ to buffer $\mathcal{D}$;
        end
    end
    update parameters $\phi$ and $\psi$ with data in $\mathcal{D}$ according to PPO;
    // the returns of adjoint episodes accumulate according to
       Equation (3)
end
```

4.2 Policy Representation via Contrastive Learning

In terms of inferring the opponent's dynamics in a fully-independent setting, the policy of opponent is unknown and agent needs to learn the opponent's policy from its local trajectory, i.e. the observation, action, and reward sequences $\{(o^i_t, a^i_t, r^i_t)\}^T_{t=0}$, where T is the length of the trajectory. To solve this problem, we use contrastive learning to learn the dynamic change caused by the opponent's policy. Rather than a CPC [19] training style used in CURL [11] to obtain the ability to predict the future of dynamics, we expect that the encoder is able to distinguish the dynamics itself, thus we train it in the way of policy representation [6].

Dynamics Discrimination. In an independent setting, agent does not know in advance any prior knowledge about the opponent. That is, agent has no assumption on the opponent's algorithm Γ^{-i}, and no access to the grounded-true trajectory node $(o_t^{-i}, a_t^{-i}, r_t^{-i})$ of the opponent.

If the action of the opponent a^{-i} is unknown, it is impossible to reconstruct the transition probability $P(s'|a^i, a^{-i}, s)$ of inner environments. However, discriminating the dynamics of the inner game, instead of directly reconstructing policy of the opponent, is theoretically feasible. With a fixed opponent policy, the mixed dynamics

$$P_{\pi^{-i}}(s'|a^i, s) := \sum_{a^{-i}} P(s'|a^i, a^{-i}, s)\pi^{-i}(a^{-i}|o^{-i}) \tag{4}$$

is invariant. And we assume that the opponent's policy satisfies the below property.

Property: There are not two different policies π_1^{-i}, π_2^{-i} satisfying the condition that for all state-action tuple (s', a^i, s), $P_{\pi_1^{-i}}(s'|a^i, s) = P_{\pi_2^{-i}}(s'|a^i, s)$.

We say two policies are different if there is some action a^{-i} such that $\pi_1^{-i}(a^{-i}|o^{-i}) \neq \pi_2^{-i}(a^{-i}|o^{-i})$. If for all state s, the observation o^i is different, we can discriminate the mixed dynamics with trajectories $\{\tau_k^i | \tau_k^i = (o_t^i, a_t^i, r_t^i)_{t=1}^{T_k}\}_{k=1}^K$. Such discrimination is the policy representation z. It can be captured by an auto-regressive encoder which takes history node (o_t^i, a_t^i, r_t^i) and the last output z_{t-1} as input.

Negative Sample. Another problem is contrastive learning requires negative samples. In terms of policy representaion [6], negative sample is a trajectory generated with a different opponent's policy, which is not always available in MARL as most environments are symmetric games and the homogeneous agents play the same policy repeatedly at the stable fixed point of the game.

We use SimSiam [4], a contrastive learning algorithm without negative samples, as the policy encoder. Following the notation in [4], we use f_θ to denote the encoder and h_ω the predictor. They are parameterized by θ and ϕ respectively and train as Fig. 1 shows.

Data Augmentation. Data augmentation remarkably improves the performance of the model. It generates new samples from one sample without causing significant semantic alterations. In policy representation, the semantics is the mixed dynamics, thus trajectories from the same policy can be regarded as augmented data. To further relax the assumption on the opponent, we do not require the opponent to perform the sample policy in one episode. Instead, PROS learns the representation at every time step. The feasibility is shown by Hjelm et al. [8].

The contrastive loss is

$$\mathcal{L} = \frac{1}{|\mathcal{B}|}\sum_{b=0}^{|\mathcal{B}|} \frac{h_\omega(f_\theta(x_b)) \cdot \perp(\frac{1}{|\mathcal{B}|}\sum_{d=0}^{|\mathcal{B}|} f_\theta(x_d)^\top)}{\|h_\omega(f_\theta(x_b))\| \cdot \perp\|\frac{1}{|\mathcal{B}|}\sum_{d=0}^{|\mathcal{B}|} f_\theta(x_d)\|} \tag{5}$$

where $\mathcal{B}$ refers to the batch of data, which are sampled from synchronous parallel environments at each step, $x_b = (o_t^i, a_t^i, r_t^i)_b$ is the data from a parallel environment, and $\perp$ is the stop gradient operator. The training process is shown as Algorithm 1.

5 Experiment

In this section, we perform experiments to evaluate the ability of PROS to maintain cooperation in mixed cooperative-competitive environments with different scales.

5.1 Experiment Setting

We compare PROS with four baseline algorithms:

- **PPO** [22]. We use iPPO rather than MAPPO [26] to optimise the policy of a self-interested agent without awareness of opponent.
- **M-FOS** [17]. M-FOS represents agents with opponent learning awareness without white-box access to the opponent's policy. It uses PPO as the RL backbone. To evaluate its scalability, We use M-FOS with conditional vector in all experiments.
- **LILI** [25]. LILI learns the latent policy of the other agents, which is similar to PROS but focuses on cooperative environments. It uses SAC as the RL backbone.

There are some works relative to PROS but we do not use them as baselines for some reason. LOLA-DiCE [5] requires approximating the updating of opponent's policy, which involves additional interactions with the imagined policy of opponent between episodes playing with the true opponent. This leads to LOLA-DiCE needing a simulator of the game. FURTHER [10], which covers techniques used in opponent shaping and latent representation, is proposed from the aspect of Active Markov Game, unlike all other works.

All the actor and critic networks are MLPs since all the following environments are fully observable. Networks of different algorithms have the same width and depth.

	C	D
C	(-1,-1)	(-3, 0)
D	(0, -3)	(-2,-2)

(a) Prisoners' dilemma

	H	T
H	(+1,-1)	(-1,+1)
T	(-1,+1)	(+1,-1)

(b) Matching pennies

Table 1: Payoff matrix of two iterated games

5.2 Iterated Matrix Games

Following prior work [5] on opponent shaping, we first study one-step-memory iterated prisoners' dilemma (IPD) and iterated matching pennies (IMP). The properties of the game are well studied [1].

In iterated matrix games, players play the same normal-form game repeatedly and maximise the discounted cumulative reward. The unique Nash equilibrium in IMP is uniformly selecting action. However, the folk theorem shows that there are infinite Nash equilibria in IPD, which are different in welfare. Table 1 shows the reward at each step.

5.3 Coin Game

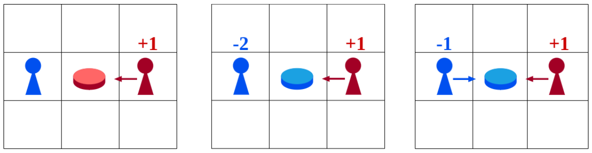

Fig. 2: Three situations of collecting a coin

We investigate the scalability and efficiency of PROS in coin game, first proposed by Lerer and Peysakhovich [12]. The coin game is a Markovian version of prisoners' dilemma, where two agents, colored red and blue respectively, are tasked with collecting coins in a 3x3 grid. Coins are either red or blue and spawn with the other color after being picked up. One Agent gains a $+1$ reward once it picks up a coin of any color. When agents pick up a coin of different color, the other agent gains a -2 reward. In coin game, a fully cooperative policy is always picking up coins of the same color and a selfish policy is greedily picking up all coins.

5.4 Results

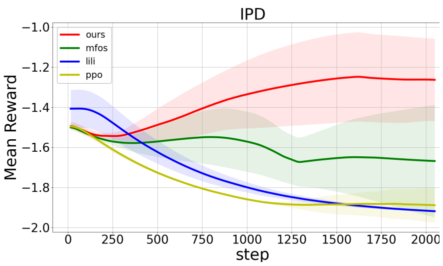

Fig. 3: The mean reward of both agents during training

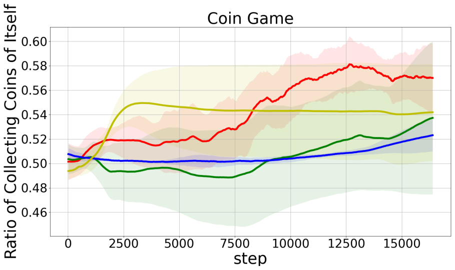

Fig. 4: The empirical percentage of picking up coins of the same colours.

Iterated Prisoners' Dilemma. In IPD, each algorithm plays with each other. We visualize the mean of both agents' rewards at each training step to measure the ability to learn to cooperate in an environment mainly guided by competing. Specifically, different mean rewards refer to different equilibria.

Fig. 3 shows that only PROS achieves long-term cooperation. M-FOS with conditional vector is able to learn to cooperate but underperforms the original version that is available in such a tabular environment (the mean reward of the original version is -1.01 as shown in [17]).

Iterated Matching Pennies. With the same learning rate, PROS has a minimum reward deviation from zero compared with two other on-policy algorithms, PPO and M-FOS. Such deviation characterizes the distance to the Nash equilibrium. LILI has the best result as it is off-policy with higher sample efficiency than others. Fig. 5 shows that PROS has the best result and the lowest standard error about the mean (SEM) among on-policy algorithms.

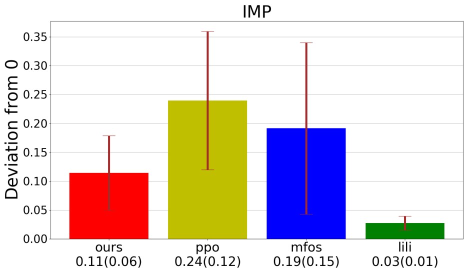

Fig. 5: Reward deviation from zero. Error bars show the SEM. The values of mean deviation is shown at the bottom of figure, with value of SEM in the brace.

Coin Game. The ratio of coins with the same color in all collected coins can be seen as the probability of cooperation during training. Fig. 4 shows that the cooperation probability of PROS, M-FOS, and LILI all increase over training, but PROS learns faster with a lower standard deviation. PPO seems to outperform LILI and M-FOS but the performance of PPO highly depends on the initial state, making it unstable. The ratios of all algorithms do not reach 1. The same problem has been reported in other works [5,27]. It is possibly caused by a large amount of redundancy in the neural network parameters [5].

6 Conclusion and Future Work

We have proposed PROS that applied opponent shaping [5] in cooperative-competitive mixed tasks with fully-independent settings. Although PROS does not require assumptions or extra information about opponents such as other opponent shaping algorithms, it cooperates better than baselines in mixed tasks like IPD. To implement opponent shaping in a fully-independent setting, we show that the opponent's dynamics can be learnt without information about the opponent's observations, actions, and rewards. With a contrastive learning module, PROS learns the representation of opponent's dynamics.

In the future, we could investigate the learning ability of PROS as decentralized agents in continuous decision tasks such as MPE [16], Multi-Agent MuJoCo [24] and Robotic Controlling [2], or use the representation to enhance the ability of agents trained in a CTDE paradigm.

Acknowledgements This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030400; in part by the National Natural Science Foundation of China under Grant 62293541; in part by the International Partnership Program of Chinese Academy of Sciences under Grant 104GJHZ2022013GC; and in part by the Youth Innovation Promotion Association CAS.

References

1. Axelrod, R., Hamilton, W.D.: The evolution of cooperation. science **211**(4489), 1390–1396 (1981)
2. Chai, J., Chen, W., Zhu, Y., Yao, Z.X., Zhao, D.: A hierarchical deep reinforcement learning framework for 6-DOF UCAV air-to-air combat. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2023)
3. Chai, J., Li, W., Zhu, Y., Zhao, D., Ma, Z., Sun, K., Ding, J.: UNMAS: Multiagent reinforcement learning for unshaped cooperative scenarios. IEEE Transactions on Neural Networks and Learning Systems (2021)
4. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
5. Foerster, J., Chen, R.Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., Mordatch, I.: Learning with opponent-learning awareness. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (2018)
6. Grover, A., Al-Shedivat, M., Gupta, J., Burda, Y., Edwards, H.: Learning policy representations in multiagent systems. In: International conference on machine learning. pp. 1802–1811. PMLR (2018)
7. Guo, Z., Thakoor, S., Pîslar, M., Avila Pires, B., Altché, F., Tallec, C., Saade, A., Calandriello, D., Grill, J.B., Tang, Y., et al.: Byol-explore: Exploration by bootstrapped prediction. Advances in neural information processing systems **35**, 31855–31870 (2022)
8. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: International Conference on Learning Representations (2019)
9. Hu, G., Zhu, Y., Zhao, D., Zhao, M., Hao, J.: Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning. IEEE Transactions on Neural Networks and Learning Systems **34**(8), 3966–3978 (2023)
10. Kim, D.K., Riemer, M., Liu, M., Foerster, J., Everett, M., Sun, C., Tesauro, G., How, J.P.: Influencing long-term behavior in multiagent reinforcement learning. Advances in Neural Information Processing Systems **35**, 18808–18821 (2022)

11. Laskin, M., Srinivas, A., Abbeel, P.: Curl: Contrastive unsupervised representations for reinforcement learning. In: International Conference on Machine Learning. pp. 5639–5650. PMLR (2020)
12. Lerer, A., Peysakhovich, A.: Maintaining cooperation in complex social dilemmas using deep reinforcement learning. arXiv preprint arXiv:1707.01068 (2017)
13. Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T., Whiteson, S.: Stable opponent shaping in differentiable games. In: International Conference on Learning Representations (2019)
14. Li, L., Yang, R., Luo, D.: FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In: International Conference on Learning Representations (2021)
15. Liu, M., Li, L., Hao, S., Zhu, Y., Zhao, D.: Soft contrastive learning with Q-irrelevance abstraction for reinforcement learning. IEEE Transactions on Cognitive and Developmental Systems (2022)
16. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems **30** (2017)
17. Lu, C., Willi, T., De Witt, C.A.S., Foerster, J.: Model-free opponent shaping. In: International Conference on Machine Learning. pp. 14398–14411. PMLR (2022)
18. Mealing, R., Shapiro, J.L.: Opponent modeling by expectation–maximization and sequence prediction in simplified poker. IEEE Transactions on Computational Intelligence and AI in Games **9**(1), 11–24 (2015)
19. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
20. Parekh, S., Losey, D.P.: Learning latent representations to co-adapt to humans. Autonomous Robots pp. 1–26 (2023)
21. Richemond, P.H., Grill, J.B., Altché, F., Tallec, C., Strub, F., Brock, A., Smith, S., De, S., Pascanu, R., Piot, B., et al.: Byol works even without batch statistics. arXiv preprint arXiv:2010.10241 (2020)
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
23. Willi, T., Letcher, A.H., Treutlein, J., Foerster, J.: Cola: Consistent learning with opponent-learning awareness. In: International Conference on Machine Learning. pp. 23804–23831. PMLR (2022)
24. de Witt, C.S., Peng, B., Kamienny, P., Torr, P.H.S., Böhmer, W., Whiteson, S.: Deep multi-agent reinforcement learning for decentralized continuous cooperative control. CoRR (2020)
25. Xie, A., Losey, D., Tolsma, R., Finn, C., Sadigh, D.: Learning latent representations to influence multi-agent interaction. In: Conference on robot learning. PMLR (2021)
26. Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., Wu, Y.: The surprising effectiveness of ppo in cooperative multi-agent games. Advances in Neural Information Processing Systems **35**, 24611–24624 (2022)
27. Zhao, S., Lu, C., Grosse, R.B., Foerster, J.: Proximal learning with opponent-learning awareness. Advances in Neural Information Processing Systems **35** (2022)
28. Zhu, Y., Li, W., Zhao, M., Hao, J., Zhao, D.: Empirical policy optimization for n-player Markov games. IEEE Transactions on Cybernetics (2022)
29. Zhu, Y., Zhao, D.: Online minimax Q network learning for two-player zero-sum markov games. IEEE Transactions on Neural Networks and Learning Systems **33**(3), 1228–1241 (2020)