SCHOLASTIC-ACTOR-CRITIC FOR MULTI AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Actor-Critic framework of multi-agent reinforcement learning(MARL) is gathering more attention nowadays. Centralized training with decentralized execution allows the policies to use extra information to ease the training while enhancing overall performance. In such a framework, the quality of critic profoundly impacts the final average rewards. Thus we present a method, called Scholastic-Actor-Critic(SMAC), that involves a more powerful critic to maintain efficiency in ample knowledge acquisition. The headmaster critic is designed to group agents with proper size and proper timing, while other critics update simultaneously at the decision time. The learning rule includes additional terms account for the impact of other agents within a group. Our method receives higher payouts compared to other state-of-the-art methods and is robust against the explosion of dimension during training. We apply our method to the Coin Game, the Cooperative Treasure Collection(CTC) (Lerer & Peysakhovich, 2017) and a dynamic battle game, MAgent(Zheng et al., 2018). Experiment results are all satisfying.

1 INTRODUCTION

MARL(Multi-Agent Reinforcement Learning) is gathering more attention in deep learning researches. Artificial agents thus perform better to interact both with other agents and humans in complex partially competitive or sequential dilemma occasions. MARL is a big topic with fully cooperative settings, competitive settings and mixed settings. It is still challenging to make decisions with inadequate information in applications, such as playing games, advertising and self-driving cars.

The ability to maintain cooperation and competition in a variety of complicated situations is essential in MARL. Early works focus on improving policy or value constructing methods (Foerster et al., 2018b) (Silver et al., 2016) (Sukhbaatar et al., 2017)(Gupta et al., 2017), promoting more effectively opponent modeling methods (He et al., 2016)(Foerster et al., 2018a)(Metz et al., 2016)(Tesauro, 2004) and enhancing communication between opponents (Foerster et al., 2017) (Lerer & Peysakhovich, 2017) (Das et al., 2017) (Foerster et al., 2016) (Mordatch & Abbeel, 2018) (Sukhbaatar et al., 2016) (Lauer & Riedmiller, 2000) (Matignon et al., 2007) (Omidshafiei et al., 2017).

In cooperative-and-competitive settings, Iterated Prisoners' Dilemma is a traditional problem, in which selfish actions usually lead to an overall bad result. At this time, cooperation maximizes social welfare, which leads to an average best outcome. In this setting, the measurement is the total of rewards of all agents, while randomly initialized agents usually pursue independent gradient descent on the specific value function. Lerer & Peysakhovich (2017) and Leibo et al. (2017) point out that reciprocity among agents results in a higher average reward. Peng et al. (2017) and Evans & Gao (2016) find that even in strongly adversarial settings, reciprocity shows its nontrivial value.

In traditional Q methods, each agent's policy changes over time, resulting in a non-stationary environment. In a non-stationary environment, agents are not able to make good use of naive experience replay. Recent years Lowe et al. (2017) propose the actor-critic framework(also called MADDPG), which combines offline and online learning, which enhances the ability for multi-agent learning. Then, (Yang et al., 2018)(MF-MARL), Iqbal & Sha (2018)(MAAC) and Jiang & Lu (2018) explore policy and communication optimizations within the Actor-Critic framework.

We here propose the Scholastic-Multi-Actor-Critic method(SMAC), which aims to improve the ability of the critic. We want to train a more powerful critic, the headmaster critic that enables actors to communicate more efficiently during training. The SMAC learns to control when and how an agent receives information from others. That is, the access of observations of an agent depends on the critic. This optional additional term when applied to a group of agents, leads to extra reciprocity and cooperation. The policy gradient is consistent with prior works presented by Sandholm and Crites Sandholm & Crites (1996) and Foerster et al. (2018a).

Our approach enables high dimensional settings. We deploy experiments on the Coin Game4.1.1, the Cooperative Treasure Collection4.1.2 and the MAgent(Zheng et al., 2018). Our algorithm leads to the overall highest average return on these games. All agents using our method achieve the stable equilibrium with less training resources.

2 RELATED WORKS

As mentioned above, interactions between agents can either be cooperative, competitive or usually both. Model-free reinforcement learning algorithms in this domain could be concluded to value-based methods, policy-based methods and actor-critic methods.

MADDPG (Lowe et al., 2017) combines offline and online learning that enhances the ability of multi-agent learning. It allows the policies to use extra information to ease the training. The critic is enlarged with extra information about the policies of other agents, while each actor only has access to local information. Local actors are used at the execution phase after training.

COMA(Counterfactual Multi-Agent Policy Gradients) raised by Foerster et al. (2018b) is aimed to solve multi-agent credit assignment in cooperative settings. Before, each agent trains with his own critic so that the information sharing between them is insufficient, resulting in poor cooperation between agents. Therefore, the centralized critic firstly introduced in COMA to give a preliminary solution to this problem.

MF-MARL, the Mean Field Multi-Agent Reinforcement method developed by Yang et al. (2018) try to model opponents by the use of Mean Field Theory under Q-learning and Actor-Critic methods. It uses numerical techniques that greatly reduce the cost of modeling opponents.

Somewhat like COMA(Foerster et al., 2018b), MAAC (Iqbal & Sha, 2018)(Multi-Actor-Attention-Critic) considers to make full use of information and takes the attention mechanism within the centralized critic network. The experiment result shows that as the scale is growing, this method demonstrates its great effect. However, the requirement of computing is too high. On the other hand, ATOC (Jiang & Lu, 2018)(Learning Attentional Communication) decides to find a good communication group for the initiator agents by attention methods, too. Nevertheless, the determination of the initiator is very vague, and as the decisive role, if the initial selection is not appropriate, the entire model will collapse.

3 METHODS

3.1 BACKGROUND

3.1.1 STOCHASTIC GAME AND DEEP Q-NETWORKS

A multi-agent stochastic game G is formulated by a tuple $G = \langle S, A, P, O, R, n, \gamma \rangle$. S denotes the state space, the configurations for all agents. Each agent takes $a_i \in A$ at every time step, forming joint actions $\mathbf{a} \in \mathbf{A} \equiv A^n$. To choose actions, each agent uses a policy $\pi_{\theta_i} : \mathcal{O}_i \times \mathcal{A}_i$, which produces the next state according to the state transition function. $P(s'|s, \mathbf{a}) : S \times \mathbf{A} \times S \to [0, 1]$ denotes transition probabilities of states, and $o_i \in \mathbf{O}$ denotes observations. The reward function $r^i(s, \mathbf{a}) : S \times \mathbf{A} \to \mathbb{R}$ specify rewards and $\gamma \in [0, 1)$ is the discount factor, and for each agent, $R^i_t = \sum_{l=0}^{\infty} \gamma^l r^i_{t+l}$. Policy gradient methods update an agent's policy, parameterised by θ^i.

Provided and initial state s, the value function of agent i under the joint policy π could be formulated as:

$$v^j_\pi(s) = v^j(s; \pi) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi,p} \left[r^j_t | s_0 = s, \pi \right] \tag{1}$$

We define the Q-function within the framework of N-agent games based on the Bellman equation in (1) such that the Q-function Q_π^i for agent i under policy π could be recursively formulated as

$$Q^\pi(s,a) = \mathbb{E}_{s'}\left[r(s,a) + \gamma \mathbb{E}_{a'\sim\pi}\left[Q^\pi\left(s',a'\right)\right]\right] \tag{2}$$

, and deep Q-networks learn the action-value function $Q*$ by minimizing the loss in (3):

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(Q^*(s,a|\theta) - y\right)^2\right], \tag{3}$$

and

$$y = r + \gamma \max_{a'} \overline{Q}^*\left(s',a'\right) \tag{4}$$

where $\overline{Q}^*$ is the target Q function and its parameters update periodically with the most recent θ, which stabilize the learning. Besides, the experience replay buffer $D = (s,a,r,s')$ also used to stabilization. However, because agents are independently updating their policies as learning progresses, the environment appears non-stationary from the view of any one agent, violating Markov assumptions required for convergence of Q-learning. Foerster et al. (2017)'s approach point out, another difficulty is that the experience replay buffer cannot be used in such a setting since in general.

3.1.2 Policy Gradients

Policy gradient techniques (Sutton et al., 2000) aims to estimate the gradient of an agent's expected returns with respect to the parameters of its policy. This gradient estimate takes the following form as (5):

$$\nabla_\theta J\left(\pi_\theta\right) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log\left(\pi_\theta\left(a_t|s_t\right)\right)\sum_{t'=t}^{\infty}\gamma^{t'-t}r_{t'}\left(s_{t'},a_{t'}\right)\right] \tag{5}$$

3.1.3 Actor-Critic Methods

The term $\sum_{t'=t}^{\infty}\gamma^{t'-t}r_{t'}\left(s_{t'},a_{t'}\right)$ in the policy gradient estimator leads to high variance, as returns can vary drastically between training episodes. The Actor-critic method (Konda & Tsitsiklis, 2000) aims to ameliorate this issue by using a function to approximate the expected returns. Moreover, it replacs the original return term in the policy gradient estimator with this function. Siven a state and action, an agent under actor-critic methods learns a function to estimate expected discounted returns as: $Q_\psi\left(s_t,a_t\right) = \mathbb{E}\left[\sum_{t'=t}^{\infty}\gamma^{t^{-}t}r_{t'}\left(s_{t'},a_{t'}\right)\right]$, it updates by minimizing the regression loss of:

$$\mathcal{L}_Q(\psi) = \mathbb{E}_{s,a,r,s'}\left[\left(Q_\psi(s,a) - y\right)^2\right] \tag{6}$$

where

$$y = r(s,a) + \gamma \mathbb{E}_{a'\sim\pi(s')}\left[Q_{\overline{\psi}}\left(s',a'\right)\right] \tag{7}$$

in which $Q_{\overline{\psi}}$ is the target Q-value function. A recent approache(Haarnoja et al., 2018) applies a soft value function by modifying the policy gradient to incorporate an entropy term to encourage exploration and avoid converging to non-optimal deterministic policies. It could be formulated as:

$$\nabla_\theta J\left(\pi_\theta\right) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log\left(\pi_\theta(a|s)\right)\left(\alpha\log\left(\pi_\theta(a|s)\right) - Q_\psi(s,a) + b(s)\right)\right] \tag{8}$$

where $b(s)$ is a state-dependent baseline. The loss function for temporal difference learning is also revised with a new target, that is:

$$y = r(s,a) + \gamma \mathbb{E}_{a'\sim\pi(s')}\left[Q_{\overline{\psi}}\left(s',a'\right) - \alpha\log\left(\pi_\theta\left(a'|s'\right)\right)\right] \tag{9}$$

3.2 Scholastic-Actor-Critic

Our method obeys the same paradigm of training critics centrally and executing learned policies distributedly. That is proposed to overcome the challenge of non-stationary environments. The main idea behind our approach is group discussion, which encourages agents to emulate those better than themselves with high efficiency. We design a more powerful critic, the headmaster critic, to learn how to group agents and determine when to communicate, that has the same effect of the attention mechanism. The additional critic has a global perspective of all agents and focuses on agents with highest and lowest rewards. Accounting for the impacts from opponents, observations and actions incorporate information into the estimation of each agent's value function in the same group.

3.2.1 Assignment of groups

Expand the setting of MAAC(Iqbal & Sha, 2018), we introduce a headmaster critic to assign communication groups. The critic randomly selects n collections with random size s and changes every k epchos. After selecting n collections, we take the average contirbutions from each group(super agent, sa), and apply the following loss function:

$$\mathcal{L}_Q(\psi) = \sum_{i=1}^{N} \mathbb{E}_{(o,sa,r,o')\sim D} \left[\left(Q_i^{\psi}(o, sa) - y_i \right)^2 \right] \tag{10}$$

where

$$y_i = r_i + \gamma \mathbb{E}_{sa'\sim\pi_{\overline{\theta}}(o')} \left[Q_i^{\overline{\psi}}(o', sa') \right] \tag{11}$$

The action-value $Q_i^{\psi}(o, sa)$ function estimates outcomes in group i from 1 to n, which receives observations and actions of agents. To avoid the degradation, we set threshold for n as $n/2$ and s as $s > 1$.

3.2.2 cirtics in groups

Critics within the same group updated together to minimize a joint regression loss function:

$$\mathcal{L}_Q(\psi) = \sum_{i=1}^{N} \mathbb{E}_{(o,a,r,o')\sim D} \left[\left(Q_i^{\psi}(o, a) - y_i \right)^2 \right] \tag{12}$$

Note that $Q_i^{\psi}(o, a)$,the action-value estimate for agent i, receives observations and actions for partial agents. Where,

$$y_i = r_i + \gamma \mathbb{E}_{a'\sim\pi_{\overline{\theta}}(o')} \left[Q_i^{\overline{\psi}}(o', a') - \alpha \log \left(\pi_{\overline{\theta}_i}(a_i' | o_i') \right) + \Gamma \right] \tag{13}$$

$$\Gamma = \omega \log \left(\pi_{\overline{\theta}_i}(a_i' | o_{others}') \right) + \sigma \log \left(\pi_{\overline{\theta}_i}(a_i' | o_{others}') \right) \tag{14}$$

in which ψ and θ are the parameters of the target critics and target policies, respectively.

3.2.3 Agents in groups

To calculate the Q-value function $Q_i^{\psi}(o, a)$ for agent i, the critic receives the observations $o = (o_1, \ldots, o_N)$ and actions $a = (a_1, \ldots, a_N)$ for all agents in a group. Then other agents' contributions could be formulated as 15. where g_i is a two-layer MLP(multi-layer perceptron) embedding function and f_i is a softmax function. It could be formulated as:

$$Q_i^{\psi}(o, a) = f_i \left(g_i \left(o_i, a_i \right) \right) \tag{15}$$

As shown in Foerster et al. (2018b), an advantage function using a baseline that only marginalizes out the actions of the given agent from Q. It helps in credit assigning. In other words, by comparing the value of specific actions to an average action, an agent could learn whether the action he made would cause an increase in expected return. Thus the individual policies are updated with the following gradient:

$$\nabla_{\theta_i} J(\pi_\theta) = \mathrm{E}_{a\sim\pi_\theta} \left[\nabla_{\theta_i} \log \left(\pi_{\theta_i}(a_i | o_i) \right) \left(\alpha \log \left(\pi_{\theta_i}(a_i | o_i) \right) - Q_i^{\psi}(o, a) + b(o, a_{others}) \right) \right] \tag{16}$$

$$\begin{aligned} A_i(o, a) &= Q_i^{\psi}(o, a) - b(o, a_{others}) \Big) \\ b(o, a_{others}) &= \mathbb{E}_{a_i\sim\pi_i(o_i)} \left[Q_i^{\psi}(o, (a_i, a_{others})) \right] \end{aligned} \tag{17}$$

$b(o, a)$ is the multi-agent baseline that used to calculate the advantage function.

We implement a more general and flexible form of a multi-agent baseline. We do not apply a global reward, but naturally decompose an agent's encoding observations and the average of encodings of other agents.

$$\mathbb{E}_{a_i \sim \pi_i(o_i)} \left[Q_i^{\psi} \left(o, (a_i, a_{others}) \right) \right] = \sum_{a_i' \in A_i} \pi \left(a_i' | o_i \right) Q_i \left(o, (a_i', a_{others}) \right) \tag{18}$$

As shown above, we output the value for every action and add an observation-encoder as $E_i = g_i (o_i)$. For each agent, using these encodings in place of the $E_i = g_i (o_i, a_i)$ described above, and modify f_i such that it outputs a value for each possible action. We can estimate the expectation by sampling actions from our policy and averaging their Q-values. So we do not need to add any parameters in the case of continuous policies.

4 Experiments

4.1 Setup

We operate our algorithms in various settings, including the Coin Game 4.1.1, Cooperative Treasure Collection(CTC) (Lerer & Peysakhovich, 2017) 4.1.2 and MAgent(Zheng et al., 2018) (a cooperative-competitive battle game in the Open-source MAgent system) that tests capabilities of our approach and baselines. The three games we raised, from simple to complex, are all facing iterated prisoners dilemmas(Luce & Raiffa, 1958). For each setting, we study the scalability of different methods as the number of agents grows and evaluate their ability to attend to information relevant to rewards.

4.1.1 Coin Game

The Coin Game is a higher dimensional alternative of IPD (iterated prisoners dilemma), which is convenient to make comparisons to previous works. As shown in 1, two agents with red and blue colors are tasked to collect coins which are either red or blue on the grids. A new coin with random color appears randomly after the last one is picked up. Agents move to a coin's position and both receive a point after picking it up while the agent with a different color loses 2 points. When they only pick up coins with their own color, the total return is maximized. While players usually pick up different ones. Therefore the maximum achievable collective return is approximately 50 in expectation if neither agent chooses to defect and both agents collect all coins of their own color. In this game we define niceness as $n(s_t, a_t)$ to be part of the measurement. If an agent takes action a_t^i, picks up a coin which penalizes the other players, $n(s_t, a_t) = -1$. We use recent defections as the measure of niceness $N(T) = \sum_{i=1}^{t} \lambda^{t-i} n(s_i, a_i)$ at time T.

4.1.2 Cooperative Treasure Collection

Cooperative Treasure Collection(CTC), as shown in 1, is a variant of Coin Game in which agents play roles as hunter or bank. "Hunter"s are tasked to collect the treasure of any color and deposit them into the corresponding colored bank. The "Bank"s are tasked to gather as much treasure as possible from the "Hunter"s simply. Agents could see each others' positions and concern their own. "Hunter"s receive a global reward for the successful collection of treasure, and all agents receive a global reward of the depositing amount. "Hunter"s will additionally penalized for colliding with each other. As such, the task contains a mixture of shared and individual rewards. It requires different "modes of attention" which depends on the agent's state and other agents' potential actions that affects its rewards.

4.1.3 MAgent

The mixed cooperative-competitive battle game, MAgent(Zheng et al., 2018),is a more complex multi-player environment. Agents are devided into armies, and required to take a series of actions while exact discounted reward cannot be assessed. Each army consists of homogeneous agents, and the goal of them is to get more rewards by collaborating with teammates to defeat all opponents.

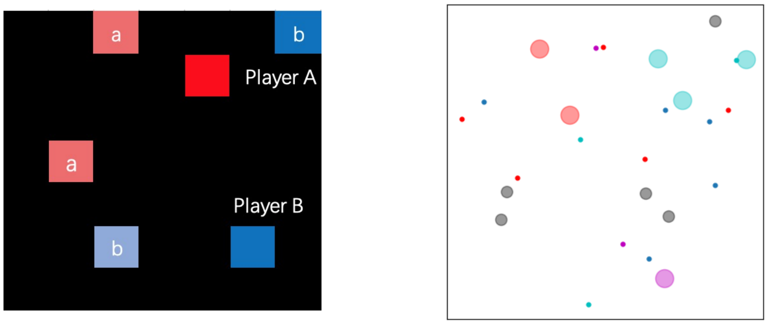


Figure 1: The Coin Game and the Cooperative Treasure Collection Game

Table 1: The average rewards compared to other methods with growing of the scale in the convergent training stages.

Game	Agents	MADDPG+SAC	MARL	MAAC	ATOC	Ours(SMAC)
CTC	8	-3.9	3.4	-4.7	3.1	2.8
	16	17.6	11.7	0.8	1.5	3.4
	32	32.1	14.8	10.1	13.0	13.2
	64	41.2	18.9	23.3	24.2	24.5
	128	77.3	29.5	64.1	65.8	78.1
MAgent	8	-	3.4	4.9	-2.7	0.8
	16	-	14.7	27.9	26.5	27.0
	32	-	32.5	29.5	28.6	30.7
	64	-	34.8	35.4	39.1	41.5
	128	-	35.6	56.1	40.6	57.7

**Note that the number of agents for each group in MAgent is half of the total. And all values are normalized into 0 to 100.*

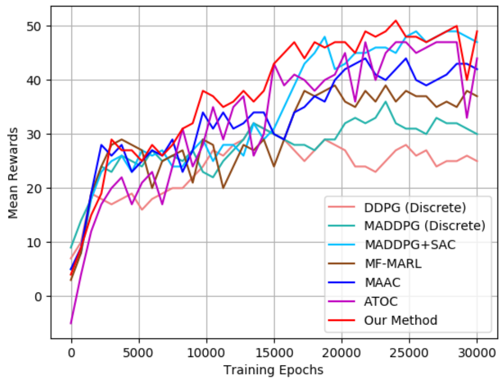
Agents can take actions to either move to or attack others on nearby grids. Ideally, the agents are able to learn skills such as chasing to hunt, escaping from enemies or working with teammates.

4.2 BASELINES

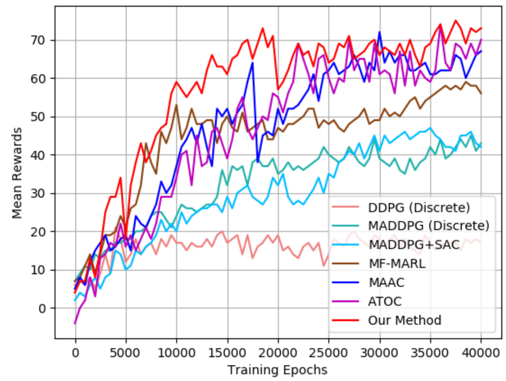
We have compared our method to recently proposed state-of-art methods in the multi-agent learning field: (1)DDPG(Lillicrap et al., 2015), (2)MADDPG(Lowe et al., 2017), (3)MF-MARL(Yang et al., 2018), (4)MAAC(Iqbal & Sha, 2018) (5)ATOC(Jiang & Lu, 2018).

As mentioned in MAAC(Iqbal & Sha, 2018), we do some modifications on some algorithms for experiments. Since deterministic policies are not possible, we use the Gumbel-Softmax reparametrization trick for learning in discrete action spaces for both MADDPG(Lowe et al., 2017) and DDPG(Lillicrap et al., 2015). The modified versions are referred to as MADDPG (Discrete) and DDPG (Discrete). For a detailed description of this reparametrization, we use a soft actor-critic method (Haarnoja et al., 2018) to optimize. We implement MADDPG with Soft Actor-Critic, named as MADDPG+SAC. Then the baselines are (1)DDPG (Discrete) (2)MADDPG (Discrete) (3)MADDPG+SAC (4)MF-MARL (5)ATOC.

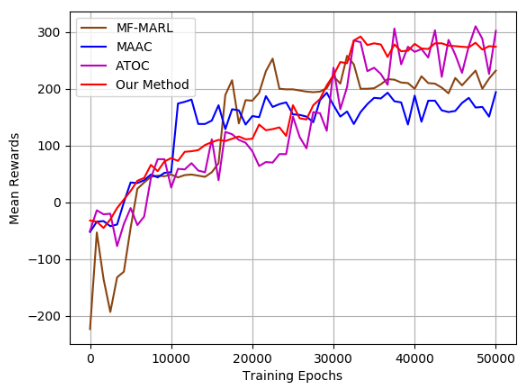
Hyperparameters are tuned based on performance and kept constant across all variants of critic architectures. All methods are re-implemented such that their approximate total number of parameters (across agents) is close to our approach. These models are trained with eight random seeds each.



(a)



(b)



(c)

Figure 2: Results of our methods and others. In Coin Game(a) and CTC(b), methods involve DDPG(Discrete)4.2, MADDPG(Discrete)4.2, MAAC(Iqbal & Sha (2018)) , MF-MARL(Iqbal & Sha (2018)) and ATOC(Jiang & Lu, 2018). In MAgent(c)(Zheng et al., 2018), we compare our method to MF-MARLYang et al. (2018), MAACIqbal & Sha (2018) and ATOC(Jiang & Lu, 2018).

4.3 RESULTS AND DISCUSSION

We first compare the average rewards attained by all approaches. We normalized by the range of awards achieved in an environment, as the number of agents changes. The proposed approach (SMAC) is competitive with other state-of-the-art approaches as shown in 4.3. In the Coin Game, most algorithms show a pleasing result while the MARL method shows less poorly performance. MAAC is competitive with our approach in both the Coin Game and the CTC environment. On the other hand, DDPG(Discrete), MADDPG (Discrete), MADDPG+SAC and MARL don't perform well on CTC. We infer that due to the simplicity of action modes and the limited scale of agents, it's not hard for agents to learn tricks. Moreover, each agent's local observation provides enough information to make a decent prediction of its expected rewards.

However, agents within MAgent(Zheng et al., 2018) dynamics over time so that it's not capable for DDPG(Discrete), MADDPG (Discrete), MADDPG+SAC break down. Thus we compare our method to MF-MARL(mean field-MARL,Yang et al. (2018), MAAC(Iqbal & Sha (2018)) and ATOC(Jiang & Lu, 2018). For all methods, rewards firstly are under zero, but along with the process of training, the reward gradually grows and finally stop in different levels. In this game, subgroups of agents are interacting and performing coordinated tasks with separate rewards while the components are changing over time. Thus it exemplifies why dynamic attention can be beneficial. MAAC(Iqbal & Sha (2018) and ATOC(Jiang & Lu, 2018) take more iterations to reach a stationary state.

Further, we explore the improvements with growing scale as shown in Table 1. DDPG(Discrete) and MADDPG(Discrete) could not handle a hige dimentional learning. MADDPG with SAC and MF-MARL(mean field-MARL,Yang et al. (2018) are barely satisfactory. But MAAC(Iqbal & Sha (2018)), ATOC(Jiang & Lu, 2018) and SMAC(ours) steadily performs when the number of agents increases. In future research, we will continue to improve the scalability when the number of agents further increases by sharing policies among agents and performing attention on sub-groups (of agents). We anticipate that in complicated scenarios, our method could work well.

5 CONCLUSIONS

We propose an algorithm, the SMAC(Scholastic-Actor-Critic) for training decentralized policies in multi-agent settings. We design a more powerful critic, the headmaster critic to learn how to group agents and when to communicate besides conventional ones. We also adapt useful advantage functions that avoid converging to non-optimal deterministic policies. We analyze the performance of the proposed approach compared the state-of-the-art methods on the Coin Game, CTC(Lerer & Peysakhovich, 2017), and MAgent(Zheng et al., 2018), concerning the number of agents. Thanks to the flexible setting, our results are promising in dynamic occasions with small training expenses. We intend to explore more to highly complex and dynamic environments.

REFERENCES

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2951–2960, 2017.

Richard Evans and Jim Gao. Deepmind ai reduces google data centre cooling bill by 40%. *DeepMind blog*, 20, 2016.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2137–2145, 2016.

Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1146–1155. JMLR. org, 2017.

Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018a.

Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 66–83. Springer, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1804–1813, 2016.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. *arXiv preprint arXiv:1810.02912*, 2018.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, pp. 7254–7264, 2018.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 464–473. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*, 2017.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

Robert Duncan Luce and Howard Raiffa. *Games and decisions: Introduction and critical survey*. Wiley New York, 1958.

Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69. IEEE, 2007.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Junhyuk Oh, Valliappa Chockalingam, Satinder Singh, and Honglak Lee. Control of memory, active perception, and action in minecraft. *arXiv preprint arXiv:1605.09128*, 2016.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2681–2690. JMLR. org, 2017.

Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2, 2017.

Tuomas W Sandholm and Robert H Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37(1-2):147–166, 1996.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pp. 2244–2252, 2016.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Gerald Tesauro. Extending q-learning to general adaptive multi-agent systems. In *Advances in neural information processing systems*, pp. 871–878, 2004.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 417–426. ACM, 2018.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. *arXiv preprint arXiv:1802.05438*, 2018.

Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.