Learning Distributed Coordinated Policy in Catching Game with Multi-Agent Reinforcement Learning

Xiangyu Liu and Ying Tan
Key Laboratory of Machine Perception (MOE), Peking University
Department of Machine Intelligence, School of Electronics
Engineering and Computer Science, Peking University, Beijing, 100871, China
Email: {xiangyu.liu, ytan}@pku.edu.cn

***Abstract*—Although learning-based methods such as reinforcement learning have been applied to multi-agent systems design successfully, it is still difficult to learn efficient coordinated policies for agents in partially observed environment settings. Centralized learners contain much more information, but add more complexity, while independent learners suffer from partial observation. To address these problems, we propose a directed multi-agent actor-critic algorithm to directly learn the coordinated policy from experience. The directed critic model can obtain all information including global information and actions, which provides effective learning signals for distributed learning actors. We take Multi-Agent Catching Game as the test scenario, where the task is to coordinate multiple moving paddles to catch balls dropping from the top of the screen. We perform several experimental evaluations and show that our method leads to superior results in learning performance, coordination effect and scalability, compared with both centralized and independent learning approach.**

I. Introduction

Real world applications often require multiple agents to work in a collaborative fashion in order to maximize the overall profits. For example, a group of robots collectively transport an item from its source to the nest [1], [2]. Biological swarms can act in collaboration that exceed the capability of an individual [3], [4]. Human beings also have the ability to promote cooperation in the complex social environment settings, and the emergence of "reciprocity" has been significant for the success of human societies [5]. However, how to design collaborative behaviours for artificial learning agents is still a challenging problem, because the complexity arise from the increase of agent number. Manual design of collaborative rules is a tedious work. It must be carefully tuned for achieving satisfactory performance, limiting its potential use in real applications. Thus, learning-based methods, such as reinforcement learning (RL) gathers more importance.

Recent work on integrating reinforcement learning and deep neural network, namely, deep reinforcement learning (DRL), has shown convincing results on solving complicated problems, including video games [6], Go [7], robotics locomotion [8] and so on. There are also a plethora of works on multi-agent reinforcement learning (MARL). In MARL, two main approaches are centralized RL and independent RL. However, the former approach suffers from the curse of dimensionality with the exponential growth of state and action space. While in the latter approach, each agent learns its own policy function in parallel, with no explicit mode for learning coordinated effect.

In this paper, we address the intractability of these problems by proposing a directed multi-agent actor-critic method. We extend the conventional actor-critic model with a directed centralized critic to evaluate the global board state, which takes global information and integrated actions as inputs. While each distributed actor calculates an advantage of the individual action by keeping other agents' actions invariant and updates by policy gradient. We leverage a game, Multi-Agent Catching Game, to be the benchmark, that need coordination of multiple agents. This task has following properties: (i) homogeneous agents, (ii) local observation, (iii) fully cooperative, (iv) no explicit communication.

We also formulate several baselines for this task and perform experimental evaluations to analyse how these algorithms learn or fail to learn coordinated policy. We illustrate that our method leads to superior results compared with both centralized and independent learning approach. We also demonstrate that our proposed method successfully scales to large space, showing strong scalability.

To summarize, our contribution is threefold. First, we propose a directed multi-agent actor-critic algorithm for learning distributed coordinated policy, which extends the conventional actor-critic model to multi-agent settings. Second, we propose a new design framework using centralized training and distributed execution for the scalability of multi-agent reinforcement learning in large space and many agents. Third, we present a new benchmark, Multi-Agent-Catching, for testing the coordinated effect of multiple learning agents.

This paper is organized as follows. We start by discuss related works in section II. We describe our problem domain in section III. In section IV, we show in detail about the proposed method for learning distributed coordinated policy. We evaluate our method with multiple baselines in section V. Finally, we conclude in section VI.

II. Related Works

Research on multi-agent reinforcement learning (MARL) has a long history and it is always active. In this section, we only review those methods on fully cooperative settings. Littman [9] introduced a Team-Q algorithm and proved its

Ying Tan is the corresponding author.

978-1-7281-2009-6/$31.00 paper N-19070.pdf

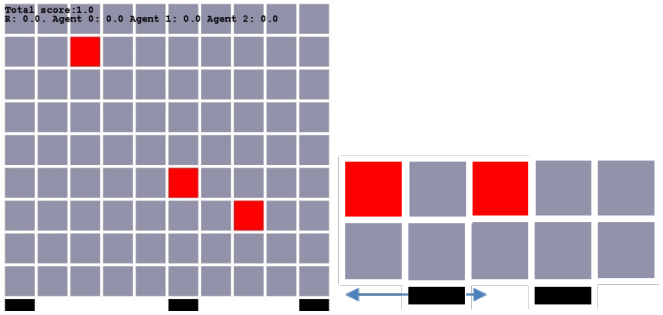
convergence to Nash Equilibrium. Lauer [10] proposed a model-free distributed Q-Learning algorithm for cooperative multi-agent decision process. This algorithm considers its teammates' behaviors and finds an optimal policy in deterministic environment. Other similar works can be found in [11]. We notice that although the research interests in MARL has an early start, the majority of the work focuses on value-based methods and tabular cases. Nevertheless, as the environment complexity increases, these traditional approaches are no longer work well.

The revival of deep neural network in the last decade brings new power to RL [6], [7]. In deep RL, the deep network serves as the function approximators for the policy. Deep RL has a stronger ability to learn high-level features automatically from raw input and environment rewards end to end with back-propagation. For multi-agent settings, there are also new advances towards Deep MARL. Recently, centralized training and decentralized execution has been adopted by many researchers in Deep MARL [12]–[15], and our work also stands in this view. [12] presented an independent learner, differentiable inter-agent learning (DIAL), that learned communication protocol among multiple agents. CommNet [13] designed a joint action learners (centralized approach) for sharing internal information inside the model. Peng et al. [14] used a bidirectional RNN to model a group of agents, and trained the model with off-policy deterministic actor-critic algorithms. Sunehag et al. [15] introduced a learned additive value-decomposition approach over individual agents. This model aims to decompose the team reward signal to each agent's utility. These methods show good performance in their domains, but they all restrict the problem with specified agent numbers. The identity becomes a feature of the state, which limits its potential values for large scale real world applications.

From the perspective of learning distributed coordinated policy with deep MARL, the closest method to ours is the work of Foerster [16]. This work aims to tackle the problem of credit assignment. It constructs a counterfactual baseline that marginalises out a single agent's action using a centralized critic. The critic is designed to receive all other agents' states and actions, and output each q-values for itself. However, this also requires agents with fixed identities and asymmetry functions. However, we consider all agents to be identical, aiming to learn a distributed coordinated policy with high scalable property.

III. Problem Formulation: Multi-Agent Catching Game

Multi-Agent Catching Game is inspired by the game "Catch" introduced in [17]. This game is played on a screen of binary pixels, and the goal is to move a paddle to catch balls that are dropped from the top of the screen. If a ball is successfully catched by the paddle, a reward of 1 will be given, and -1 otherwise. We extend this game to the multi-agent settings, as figure 1a shows. Multiple paddles need to coordinate with each other to catch as many balls as possible.



(a) An example: three paddles are trying to collect balls.

(b) An illustration of coordination effect.

Fig. 1: Test Bed: Multi-Agent Catching Game

For each time step, all balls move down one unit. The available actions for each paddle are $\{left, no-op, right\}$. We assume the game generates random amount of balls from the top of the screen, but no more than the number of agents. An illustration of coordination effect between multiple paddles is illustrated in figure 1b. If the left paddle agent observes two targets in each side of its view and also another agent on its right side, it will choose to move left, allowing its neighbor to catch its only target. Only in this plan, the overall reward is optimal.

We consider this game as a cooperative extension of stochastic game G, which is defined by a tuple $< S, \mathbf{U}, P, r, Z, O, N, \gamma >$. N is the set of agents ($|N| = n$). $s \in S$ is the global state of the environment. At each time step, all agents simultaneously take discrete actions from finite action set yielding a joint action $u \in U$. $P(s'|s,u) : S \times \mathbf{U} \times S \rightarrow [0,1]$ is the state transition probability function. All agents in the system share a reward function $r(s,u) : S \times U \rightarrow \mathbb{R}$. γ is the discount factor. All agents take the goal of maximizing the discounted reward of r_t. We consider partial observability for all agents. In each time step, agents can only draw observations $z \in Z$ from local viewpoints, which is determined by $O(s,a) : S \times A \rightarrow Z$. Each agent conditions a stochastic policy on its observations $\pi_i(a_i|o_i) : O_i \times \mathbf{U}_i \rightarrow [0,1]$.

As for a scalable and flexible solution of this problem, we consider all agents to be identical, namely they share an identical policy model. This is beneficial because any number of well-trained agents can be deployed in a large scenarios straightforwardly, regardless of any accidental failure.

IV. Methods

In this section, we first introduce the advantage actor-critic method and the difficulties when directly applied to multi-agent settings. Then we elaborate the proposed directed multi-agent actor-critic model that incorporates a centralized directed critic module, evaluating the global board state and providing learning signals for distributed actors.

A. Advantage Actor-Critic

Advantage actor-critic method (A2C), and also its asynchronous analogue, asynchronous advantage actor-critic (A3C)

[18], are popular methods in reinforcement learning and have been applied in multi-agent settings in previous works [16], [19]. A2C is an on-policy method, combining the merits of value-based method and policy-based method. The policy $\pi(a|s)$ and value function $V(s)$ reinforce each other. The value function $V(s)$ evaluates a state and constructs an advantage function $A(s,a)$ that figures out whether an action should be reinforced. The policy is trained by following the gradient that depends on the advantage function with policy gradient. In order to avoid the highly-peaked influence brought by a few trajectories (usually at the beginning stage of training), it is critical to add an entropy loss to the policy training [18]. The update rules of value and policy are as follows:

$$\theta_\pi \leftarrow \theta_\pi + \alpha A(s_t,a_t)\nabla_{\theta_\pi}\log\pi(a_t|s_t) + \beta\nabla_{\theta_\pi}H(\pi(\cdot|s_t)). \quad (1)$$

$$\theta_V \leftarrow \theta_V - \alpha\nabla_{\theta_V}(R - V(s_t))^2. \quad (2)$$

Simply applying this method to multi-agent system falls into two categories: the individual perspective, and the group perspective, corresponding to independent learning and centralized learning. The independent learning agent learns based on observed state. From the view of optimization, this means identical policies are optimized by the joint reward signal, which has a speedup mechanism as a parallel collection of training samples. However, it suffers from the partial observation and no explicit coordinated behaviors are promoted. While for centralized learner, it assumes a joint control model for all agents. The model allows inner information exchange, and it surely has the ability to capture coordinated strategy between multiple agents, because it contains all necessary information. But the major drawback is the curse of dimensionality. If an agent has $|A|$ actions available for each step, the output space for the joint policy would be $|A|^n$, where n is the number of agents. This brings a huge challenge to learning scheme.

B. Directed Multi-Agent Actor-Critic

To address these problems, we propose a directed multi-agent actor-critic (Directed-MA-A2C) model for learning distributed coordination policy. This approach captures the good aspects of the centralized approach and independent approach, while minimizing the drawbacks of the curse of dimensionality and partial observation. We use a centralized critic, $Q(s,u)$, taking the global state and joint actions for all agents as inputs. This critic is updated based on TD(λ), which is the mixture of n-step Q-value with discount that substantially reduces the variance of estimation [20], [21]:

$$g^n(s_t) = \sum_{l=0}^{n}\gamma^l r_{t+l} + \gamma^n\hat{V}(s_{t+n}). \quad (3)$$

$$y_t^\lambda = (1-\lambda)\sum_{n=1}^{T}\lambda^{n-1}g^n(s_t). \quad (4)$$

The remaining return evaluation, $\hat{V}(s_t)$, is calculated based on the expectation of each agent's policy conditioning on the remaining current observations:

$$\hat{V}(s_t) = \mathop{\mathbb{E}}_{a_i\sim\pi_i}[Q(s_t,u_t)] = \sum_{a_k^i}[\prod_{i\in N}\pi_i(a_k^i|o_t^i)]\hat{Q}(s_t,u_t). \quad (5)$$

We use target network to stabilize the training [6]. It is intuitively obvious that the centralized directed critic reflects the long-term reward given the current global state and the combination of agents' actions.

However, we take this critic model not only as a state-action evaluation function, but also a virtual simulator directing each agent to reason about its advantage function and reinforce a better policy move itself. The centralized model evaluates the "big picture" of current situation, and transmits the learning signals to each distributed actors. When an episode is over, agents will query the critic with evaluations of each possible actions, keeping other agents' actions invariant. Then the advantage function computes the difference between the exact Q value of executed actions and the weighted average of Q values for all actions based on the current policy:

$$A^i(o^i,a^i) = Q(s,u) - \sum_{a_k^i}\pi^i(a_k^i|o^i)Q(s,(u^{-i},a_k^i)). \quad (6)$$

This idea of marginalization is similar to the difference reward techniques in RL literature [22]. With advantage function, all distributed actors can be trained by policy gradient. To be more concrete, the pseudo code is elaborated in Algorithm 1. A key insight of this method is that the centralized critic has a global vision. Though actors have only local vision, the critic has the ability to direct each actor how to coordinate with neighbors based on their observations, because it can evaluate a long-term effect with different combinations of actions. The coordination is implicated in the critic. Also, the distributed design of actors can address the intractability of high dimension in part. We also notice that the homogeneous agents can speedup the training.

V. Experimental Results and Analysis

In this section, we evaluate the directed multi-agent actor-critic method on multi-agent catching game. The results are compared against independent and centralized learning-based methods and against a heuristic hand-crafted baseline as well as a random policy baseline. Unless stated otherwise, the presented results are the average performance across 20 runs with different random seeds determining the spawning balls per step in the game, and also the initializations of the model. The episode length is set to 100 steps. We aim to investigate the model from three perspective: the learning performance, coordination effect, and scalability. We first introduce the state representation for the multi-agent catching game and the model architecture, then introduce the settings of baseline algorithms, and finally demonstrate the comparison results.

Algorithm 1 Directed Multi-Agent Actor-Critic

Initialized $\theta_Q, \hat{\theta}_Q, \theta_\pi, \lambda, \gamma, T, \alpha_Q, \alpha_\pi$
for each episode e **do**
  Randomly initialize the game, get the global state s_0 and observation $\{o_t^i\}_n$
  Empty buffer ϕ
  for $t = 0$ to $T-1$ **do**
    for each agent i **do**
      Sample action a_t^i from $\pi(a_t|o_t;\theta_\pi)$
    end for
    Get r_t, s_{t+1}, $\{o_{t+1}^i\}_n$ from the game
    $\phi \leftarrow \phi + (s_t, \{o_t^i\}_n, u_t, r_t, s_{t+1}, \{o_{t+1}^i\}_n)$
  end for
  Transform this episode to a batch data
  for $t = 0$ to $T-1$ **do**
    Compute y_t^λ using equation (5)(3)(4)
    Update critic $\theta_Q \leftarrow \theta_Q - \alpha_Q \nabla_{\theta_Q}(y_t^\lambda - Q(s_t, u_t))^2$
    Synchronize target network $\hat{\theta}_Q \leftarrow \theta_Q$
  end for
  for $t = 0$ to $T-1$ **do**
    for each agent i **do**
      Compute advantage $A(o^i, a^i)$ using equation (6)
      Update actor: $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi A(o_t^i, a_t^i)\nabla_{\theta_\pi} \log \pi(a_t|o_t) + \beta \nabla_{\theta_\pi} H(\pi(\cdot|o_t))$
    end for
  end for
end for

A. Experimental Setup

State Representation In the multi-agent catching game, we define a paddle agent can only observe the local state of 5×5 pixels and the position of nearby agents with 5 bits one-hot vector, from its central view. For the marginal exception case, we pad the empty space with -1. Each agent samples an action from the actor policy for each step. While the centralized critic have the access to the full game state, we simplify the global state feature with lower half of the screen to reduce the difficulty of finding more useful features, because agents should focus on those closer targets which are about to fall down.

Model Architecture Both of the actor and critic models are composed of fully connected network. The actor has two hidden layers consisting of 256 and 64 hidden units with relu nonlinearities. It outputs a categorical distribution over multiple actions. While for the critic, the architecture differs for different agent number settings. We conduct three set of experiments with 3, 5, 7 agents. The layers architectures are [256, 128], [512, 128] and [512, 256, 128], respectively, which we found work best. For all experiments, we use Adam as the optimizer, with the initialized learning rate of $1 * 10^{-4}$ for actor network, and $1 * 10^{-3}$ for critic network. The eligibility trace parameter λ is set to 0.85, and discount factor γ is set to 0.99.

B. Baselines

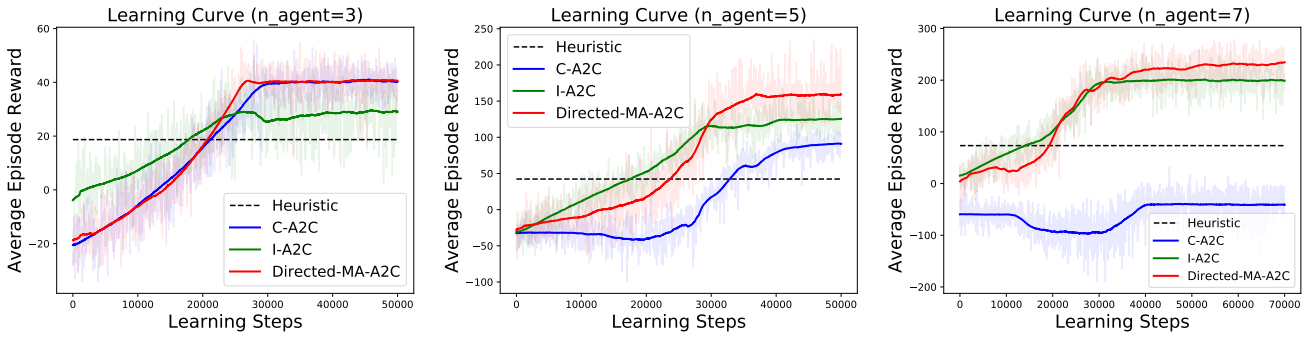
- *Random controller* Random policy generates a random action, regardless of the state.
- *Heuristic policy* The heuristic agent only steps towards the nearest target. Apparently, as long as two agents accidentally meet together, they will never be separated, which is unfavorable for collecting more targets.
- *Independent advantage actor-critic (I-A2C)* We take the independent form of A2C as a baseline algorithm, as stated in section IV-A. For I-A2C, the network architecture is the same with the actor model of directed multi-agent A2C.
- *Centralized Actor-Critic (C-A2C)* The input space of the centralized model is the concatenation over all agents' states. The output space is the cartesian product for all agents' discrete action set. We take the centralized form of A2C as another baseline algorithm. For C-A2C, the model architecture is more complicated as the joint action of policy output. The layers architectures are [256, 64], [1024, 1024, 512], [5000, 5000, 3000] for 3, 5, 7 agents respectively. Both of the two learning baseline approaches use shared network between actor and critic, except the last output layer.

C. Learning Effect

We first examine the learning effect on 10×10 multi-agent catching game described in section III. We plot the learning curves of directed multi-agent A2C algorithm against other learning-based approaches, as well as the performance of heuristic approach in figure 2. Table I shows the comparison of episode average reward performance.

Compared to the baseline methods, the directed multi-agent A2C algorithm get the best average reward performance, especially when more agents get involved. For the independent A2C, it is more likely a speedup mechanism by parallel collection of training samples. However, the bottleneck comes from the partial observation. The independent agents fail to reason about a coordinated agreement between each other. The "lazy" agents move greedily for maximizing its own interests. We will demonstrate in further experiments in section V-D that this approach doesn't stem coordinated effects. Meanwhile, for centralized approach, we notice although the performance is well in 3-agents game (in figure 2a), it struggles to master a good performance in 5-agent game and 7-agents game (in figure 2b and 2c). It even fails to converge to a positive episode reward in 7-agents game. This is due to the curse of dimensionality because the model becomes more heavy and is difficult to train.

It is also noteworthy for directed multi-agent actor-critic algorithm, that as more agents involved, the architecture of actor remains unchanged. The critic's I/O complexity is $\mathcal{O}(n)$, while for centralized approach it is $\mathcal{O}(|A|^n)$. The distributed actor can also benefit from the speedup of parallel training,



(a) Agent Num=3 (b) Agent Num=5 (c) Agent Num=7

Fig. 2: Learning curves for the Multi-agent Catching Game. The curves show the average episode reward for different agents number settings.

TABLE I: Comparison between multiple policies on Average Episode Reward

# of agents	Random	Heuristic	C-A2C	I-A2C	Directed-MA-A2C
3	-26.1	19.1	**43.6**	36.5	**42.9**
5	-51.0	49.7	80.9	135.5	**157.0**
7	-66.9	73.4	-20.1	202.6	**231.1**

because all agents share a policy model, and receive learning signals from the directed central critic.

D. Coordinated Performance

We further test the coordinated effect performance in this task. We quantify two measurements, p_1 and p_2, to demonstrate the coordinated ability of our model. One obvious measurement is the percentage of view range, which is the ratio between all agents' observed state and the global state:

$$p_1 = \frac{\bigcup_{i \in N} |O_i|}{|S|}. \tag{7}$$

As the game randomly generates targets, the agents should be apart with each other to maximize the coverage, in order to collect more targets. The other measurement p_2, is the ratio between the average amount of targets collected per agent (G_i) and the total amount of targets collected (G_{game}), in one episode:

$$p_2 = \frac{E_{i \in N}[G_i]}{G_{game}}. \tag{8}$$

Note that the G_{game} is not the simple summation of G_i, because there's a possibility that multiple agents collect the same target. Thus for a better policy, p_2 should be minimized. We also notice the best performance for p_2 should be $\frac{1}{|N|}$. However, it cannot be guaranteed that an arbitrary target is always accessible to an agent.

Table II compares our proposed algorithm against the baselines under 3-agents and 5-agents scenarios. We can see that our method and the centralized approach have a similar effect on keeping margin between agents and avoiding overlap to collect more targets. This verifies that the centralized approach have the ability to find a coordination policy, but it suffers from the curse of dimensionality as shown in section V-C. While the independent learning approach has a similar effect with greedy heuristic policy on these two measurements, which doesn't stem a strong coordinated effects.

E. Scalability

We next investigate how well the method scales to large space and many agents. We extend the game scenario to a 20×20 map. In this experiment, we only investigate the scalability performance against the independent learning approach and heuristic approach, because it is hard to train a centralized policy with a larger state-action space. For both independent approach and directed multi-agent actor-critic approach, we use well-trained model for 7-agents scenarios in 10×10 game with random generated targets no more than 15 per step.

Figure 3 shows the scalability performance with increasing agent number settings, from 3 to 15. We can see for all approaches, the insertion of new agents increases the episode reward. Our method outperforms other two approaches with a higher reward growth of rate. Despite large amount of agents are present in the environment, they coordinate with each other to collect more targets, resulting in an intelligent behaviours. While for independent approach, it is more likely a greedy policy with similar ascent rate with heuristic approach.

F. Further Discussions

We emphasize some design considerations here for this algorithm. Although the directed centralized critic succeed in learning a rational evaluation for the global state, it is sensitive to the agent number in the learning phase. When the agent number increases, the critic will also be more complex and hard to train, inevitably. However, there's an alternative way that we can first use appropriate number of agents to co-evolve the actor model and critic model during learning phases, and then deploy the distributed actors in large-scale environment. The effect of this training mode is demonstrated in section V-E.

TABLE II: Comparison between multiple policies on coordination measurements

	N=3		N=5	
π	p_1 (%)	p_2	p_1 (%)	p_2
Random	77.7%	0.66	79.2%	0.54
Heuristic	68.1%	0.81	79.4%	0.79
C-A2C	**84.1%**	**0.50**	90.9%	0.44
I-A2C	70.8%	0.74	76.2%	0.71
Directed-MA-A2C	**83.8%**	**0.52**	**93.5%**	**0.40**

Note: p_1 and p_2 are defined in equation (7) and (8)

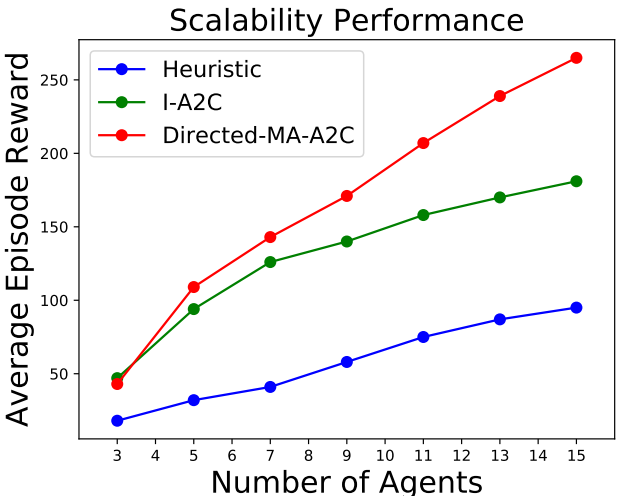


Fig. 3: Scalability Experiment

There's also another way to solve the partial observation problem, which use the history information with memory. Incorporating a recurrent model has been a popular choice to solve complex tasks with deep reinforcement learning techniques [23]. This model can also be easily extended to a recurrent form with recurrent actors. And we'll leave this for future work.

VI. Conclusion

In this paper, we presents a directed multi-agent actor-critic model to learn distributed coordinated policy in multi-agent catching game. The proposed model addresses the intractability of the curse of dimensionality and partial observation by using a directed critic to calculate a marginalised advantage function for distributed actors. The empirical evaluations have shown that this algorithm has better performance on both the learning effect and coordinated ability than other baseline algorithms. We also show that it scales to large-scale scenarios with a higher reward growth rate. Future work will concentrate on more complex tasks including continuous control and real-world applications. We also aim to integrate a hierarchical module in reinforcement learning algorithms to learn coordinated policy.

Acknowledgment

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

[1] Y. Tan and Z. Zheng, "Research advance in swarm robotics," *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013.

[2] Z. Zheng and Y. Tan, "Group explosion strategy for searching multiple targets using swarm robotic," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 821–828.

[3] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[4] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *International Conference in Swarm Intelligence*. Springer, 2010, pp. 355–364.

[5] S. Bowles and H. Gintis, *A cooperative species: Human reciprocity and its evolution*. Princeton University Press, 2011.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[8] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1889–1897.

[9] M. L. Littman, "Value-function reinforcement learning in markov games," *Cognitive Systems Research*, vol. 2, no. 1, pp. 55–66, 2001.

[10] M. Lauer and M. 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.

[11] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*, 2008.

[12] J. Foerster, Y. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 2137–2145.

[13] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems*, 2016, pp. 2244–2252.

[14] P. Peng, Q. Yuan, Y. Wen, Y. Yang, Z. Tang, H. Long, and J. Wang, "Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games," *arXiv preprint arXiv:1703.10069*, 2017.

[15] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[16] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," *arXiv preprint arXiv:1705.08926*, 2017.

[17] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[19] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.
[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
[21] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.
[22] A. Agogino and K. Turner, "Multi-agent reward analysis for learning in noisy domains," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 2005, pp. 81–88.
[23] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," 2015.