Multi-Agent Reinforcement Learning with Clipping Intrinsic Motivation

K. Charoenpitaks and Y. Limpiyakorn

***Abstract*—Intrinsic motivation is one of the potential candidates to help improve performance of reinforcement learning algorithm in complex environments. The method enhances exploration capability without explicitly told by the creator. This is suitable for the case of multi-agent reinforcement learning where the environment complexity is beyond standard. In this paper, the Random Network Distillation method is applied to implement intrinsic motivation in the multi-agent environment. Two intrinsic motivation architectures are developed and compared with the benchmark in different scenarios. The experiments show an increase in performance of the very complex environments while little to no improvement over the non-complex ones. Although there exists some overhead which results in less sample efficiency, the centralized intrinsic motivation architecture shows a long-term on par or even better optimization performance as it could explore on more states. The performance of the centralized architecture shows a solid improvement in 2s3z environment and achieves almost 70%win rate over the benchmark of 43%.**

***Index Terms*—Reinforcement learning, multi-agent learning, curiosity exploration, intrinsic reward.**

I. Introduction

Reinforcement learning (RL) is becoming more and more popular as it can solve very complex problems given a direction in terms of reward functions. Although, it is not easy to interpret the thought process of the algorithm, the self-learning without given specific labels is less labor intensive and it may come up with the alternative solutions that achieve the same or even better performances.

The potential for a wide-open sequential solution is come at the cost of complexity optimization. In RL, the objective function is to maximize the expected sum of reward [1]. Hence, it is required a good walkthrough path that achieves a good reward to get an optimization performance. Then, the better policy will get a better walkthrough path again. The process repeats until achieving an acceptable solution. However, getting the first good one is challenging. It is not easy to apply reinforcement learning for most complex environments as the rewards are either sparse [2] or too complex. In case of complex rewards, it is probable that agents would get stuck in the loop of suboptimal rewards as they have never seen any better paths. In literature, there are many studies related to this issue. One of the solutions is the reward shaping where the researcher handcrafted the reward function that would be abundant and easy to pick up and optimize. However, the simulation may be very complex, and it may not be a good idea to handcraft the reward as it may lead to the suboptimal solution.

Intrinsic Motivation is an alternative method that could enhance exploration capability without relying on domain knowledge of the creator [3], and thus scalable for the usage [4]. The idea of Intrinsic motivation is from the self-generated curiosity of humans where it encourages a software agent to explore new things while no effect to the known states [2], [5]. The concept can be applied to encourage a software agent to better explore unknown areas and to find the optimal path in the process.

In real-world applications, there are many simulations that are multi-agent in nature. This incurs even more complexity to find the optimal solution for such a task. The problem is not only very challenging, but also has high impact on real-world use cases. While other researchers are interested in communication protocol between the agents [6], some disagree as the cooperative centralized learning may prevent inefficiency by sharing information internally and certainly avoid exploring on the same area [7].

This research aims to explore the effect of intrinsic motivation using random network distillation (RND) on multi-agent setup. The experimental investigation has been carried out on different RND architectures of multi-agent reinforcement learning (MARL), namely centralized intrinsic motivation architecture and individual intrinsic motivation architecture, each of which using different clipping ratios. The ratio is used to optimize the agent policy. It is the limit ratio between environment reward and self-generated. The preliminary experiments were conducted on The StarCraft Multi Agent Challenge (SMAC) environment to compare the performance in many multi-agent scenarios with some selected clipping ratios.

II. Background

A. StarCraft Multi-Agent Challenge (SMAC)

The SMAC is a customized multi-agent environment based on StarCraft 2 game engine. The environment is rich and complex where it focuses on unit’s micromanagement perspective instead of full resources macro-management on the standard StarCraft2 game [8]. The SMAC allows independently individual unit control to the extreme level. Each unit has its own local observation and actions which are crucial to multi-agent experiments. Nevertheless, the goal of

Manuscript received August 13, 2020; revised February 2, 2021. This work was supported in part by Chulalongkorn University under Ratchadaphiseksomphot Endowment Fund.

The authors are with the Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand (e-mail: korawatcharoenpitaks@gmail.com, Yachai.l@chula.ac.th).

doi: 10.18178/ijmlc.2022.12.3.1084

the SMAC is to achieve the highest team rewards which definitely require agents to cooperative in complex strategy.

B. Counterfactual Multi-Agent (COMA)

The COMA (Fig. 1) is the baselined algorithm of policy-based MARL that is implemented in the SMAC research paper [8]. The algorithm is based on the actor-critic model with the modification of the critic network which is adjusted based on each individual agent's contributions instead of all equally weighted [9]. The COMA is categorized as centralized learning and decentralized execution paradigm where the critic is centralized taking all information from many agents while the agents individually act based on their local observations. In details, the critic network calculates agents weighting using counterfactual baseline which is the action value of an agent when the particular agent is idle.

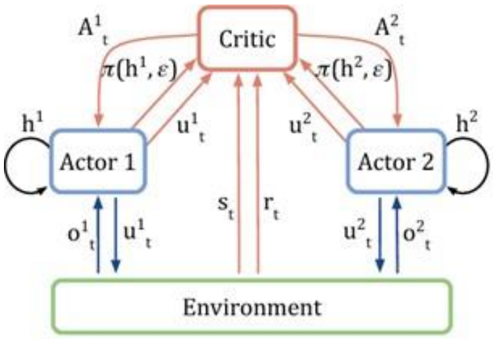

Fig. 1. COMA architecture [8].

III. Methodology

A. Random Network Distillation

The RND is one of the methods to create intrinsic motivation. The method is the subset of the prediction-based method where the model generates the prediction error between the benchmark and the predictor that will be used to represent intrinsic motivation. The idea is that in the unfamiliar state the curiosity is high and otherwise in the familiar state. Therefore, the algorithm is trained to achieve better prediction or reduce the error of the state it had been through. As a result, the prediction yields low error on the familiar state and high error on the unfamiliar state.

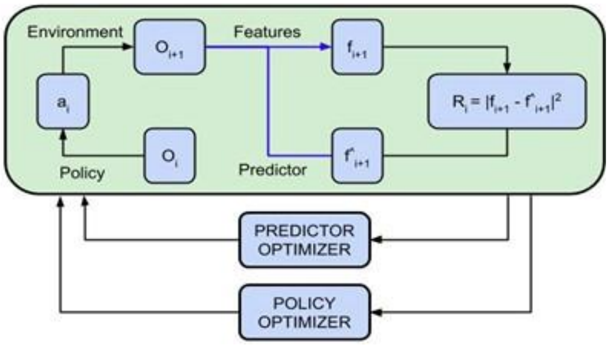

Fig. 2. Random network distillation architecture.

In case of the RND (Fig. 2), the input is the converted features from the next state and passed through the networks. There are two neural networks called Target network and Predictor network [10], [11]. The Target network is used as the benchmark hence the parameter is fixed, so the only adjustable network parameter is Predictor network. During the training process, the objective function, which is also an intrinsic reward, is the Squared Error of the two outputs. However, the error is only propagated through the predictor parameters.

B. Intrinsic Advantage with Clipping Ratio

The notion of Clipping Intrinsic Motivation is based on the intrinsic motivation of RND and Advantages value in Advantage Actor Critic (A2C) setup. To begin with, the advantage is the value of how the action is better compared to the others at a given state and it uses to optimize policy neural network in policy based RL. In details, the advantage is derived from the state action value, which is, in turn, derived from the reward. In this case, we would call it extrinsic advantage. To combine the intrinsic term, we should make it in the form of intrinsic advantage first. The idea for Clipping Intrinsic Advantage is that we should scale the Intrinsic advantage to the same scale as extrinsic advantage. Using Clipping method helps clipping the excessive amount to suit the needed level after having been normalized by other methods. Therefore, it allows intrinsic advantage component to never have too much impact on extrinsic advantage, and thus, appropriate for optimization. The steps of computing intrinsic advantage are as follow:

$$\mathrm{I}ntrinsic\ Advantage\ =\ \mathrm{Int\_}Adv$$
$$Extrinsic\ Advantage\ =\ Ext\_Adv$$
$$Intrinsic\ Reward\ =\ Int\_Rwd$$
$$Extrinsic\ Reward\ =\ Ext\_Rwd$$
$$Clipped\ Ratio\ =\ CR$$

$$Int\_Rwd =\ Int\_Rwd\_batch/std\_dev(Int\_Rwd\_batch)$$
$$Int\_Adv\ =\ Int\_Rwd\ -\ Baseline\ \text{Predictor}$$

$$f(Int\_Adv) = \begin{cases} -Ext\_Adv \times CR\ , & Int\_Adv < -Ext\_Adv \times CR \\ Int\_Adv, & -Ext\_Adv \times CR < Int\_Adv < Ext\_Adv \times CR \\ Ext\_Adv \times CR\ , & Int\_Adv \geq Ext\_Adv \times CR \end{cases}$$

$$Total\_Adv = Ex\mathrm{t}\_Adv +\ Int\_Adv$$

The algorithm starts with sampling a batch of experiences from interactions with the environment. In a batch, sampled extrinsic rewards is equivalent to episode length multiplied by the number of agents in parallel setting to eight in this case. The sampled batch will be used for a single iteration of optimization. The intrinsic reward generated by the RND is recorded in the batch in the same length. However, the intrinsic reward is scaled down by standard deviation of the batch as shown in the benchmark research [11]. The intrinsic advantage is derived from the different between the Intrinsic reward and the baselined predictor which is a prediction from neural networks. The predictor attempts to predict the scale down intrinsic reward by using observation as an input. In the next step, the intrinsic advantage is clipped by product of extrinsic advantage of each individual agents and clipping ratio of that particular element in the rollout. Lastly, the total advantage is computed as the sum of intrinsic advantage and extrinsic advantage, where extrinsic advantages are the sum of all agent extrinsic values in case of Centralized Intrinsic Motivation, while separate for each agent in case of

Individual Intrinsic Motivation.

For optimization, the algorithm uses standard COMA formula except that the variable advantages is defined as a combination of both components, extrinsic and intrinsic part.

C. Individual Intrinsic Motivation Architecture (IIMA)

The IIMA is the simple architecture extended from the single agent version. Previously, the single agent RND networks contain the Predictor and Target networks used to predict the intrinsic reward from the output deviation. Based on IIMA (Fig. 3), the individual set of RND networks is built for each individual agent, resulting in different intrinsic motivation advantage for each agent. IIMA is one version used to explore the multi-agent setup.

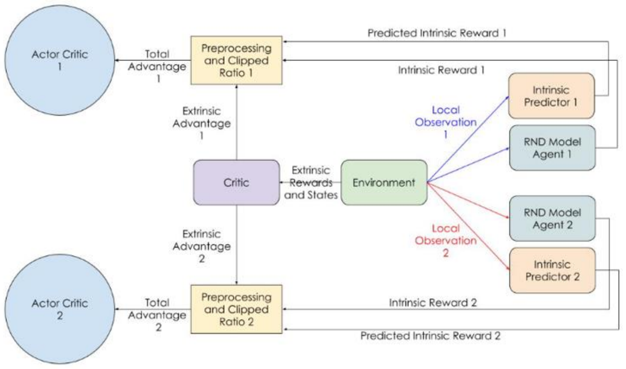

Fig. 3. Individual intrinsic motivation architecture.

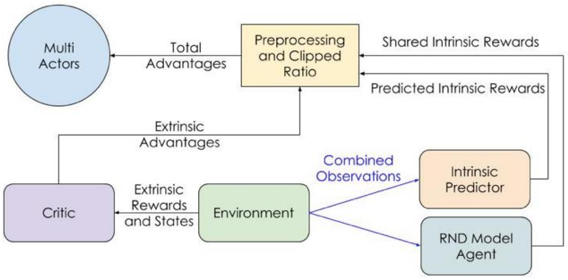

Fig. 4. Centralized intrinsic motivation architecture.

D. Centralized Intrinsic Motivation Architecture (CIMA)

The CIMA, as shown in Fig. 4, is built by following the idea of the centralized learning and decentralized execution paradigm where the learner is centralized and can learn off-policy. In this case, we would like to centralize the intrinsic of all the agents by combining all of their observations to obtain an integrated observation and use it as an input to create the intrinsic reward. It is expected that the architecture should be the more suitable for multi-agent in the long run as it is built close to the paradigm.

E. Environments

Due to the perfect setups with rich and complex states for exploration, the SMAC Environment stages are selected as the main environments for multi-agent experiments. Based on the benchmark experimental results, we selected the top 5 easiest environments (see Table I) which also contain all types from the list to improve with intrinsic motivation.

The environments are about the micromanagement task where an individual software agent has to collaborate with other agents to attack enemy units. The SMAC have 6 action spaces which are move up, move down, move left, move right, attack, and idle. It is necessary for the active unit to choose one of the actions. Each individual unit also has its own local observation space which is used as inputs for the agent policy.

To win the game, it is required to attack enemy units to reduce enemy hit point to zero before the opposite happened. The name of the map explicitly tells us about the unit in the game. The followings are the brief description for the unit in experiments. “s” stands for stalker which is a range unit, “sc” stands for spine crawler which is a power tower, “z” stands for zealot which is a powerful melee units, “c” stands for colossus which is a power area attack range unit, and “m” stands for marine which is weak range unit. These unit combination and match up require a special technique of collaboration to win. For example, the 2s_vs_1sc requires both stalkers to take turn attack the spine crawler that will result in its take turn attack stalkers. This causes the stalker to stay in game longer thus higher damage overall.

There are three main micromanagement types which can be used to evaluate how the intrinsic motivation helps improve cooperation of complex tasks. The first type is symmetric which is the most basic where the units on both sides are at equal number. This type only requires the agent to perform just better than AI from in-game game engine. The next type is asymmetric which is to give a handicap to opposite AI by reducing our units by one. There is another type which is called micro-trick where the agent must learn the clue or some specific patterns to win the scenarios, although some patterns are hard to find.

TABLE I: SMAC ENVIRONMENT DESCRIPTION

Name	Ally Units	Enemy Units	Type *c*
2s_vs_1sc	2 Stalkers	1 Spine Crawler	micro-trick
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots	symmetric
3s5z	3 Stalkers & 5 Zealots	3 Stalkers & 5 Zealots	symmetric
1c3s5z	1 Colossus, 3 Stalkers & 5 Zealots	1 Colossus, 3 Stalkers & 5 Zealots	symmetric
10m_vs_11m	10 Marines	11 Marines	asymmetric

IV. EXPERIMENTS

The algorithm is implemented on different scenarios to compare the effectiveness of different architectures and clipping ratios. Based on the assumptions mentioned in the environments section, the investigation on 5 different maps: 2s_vs_1sc, 2s3z, 3s5z, 1c3s5z, and 10m_vs_11m was carried out. There are 3 different scenarios: no intrinsic Motivation (green), Individual Intrinsic Motivation or IIM (blue), and Centralized Intrinsic Motivation or CIM (red). These scenarios aim to compare the performances of each different intrinsic motivation architectures in multi-agent setup in combination with various clipping ratios: 0.2, 0.5, and 1 to find out how the magnitude of intrinsic advantages impacts on different scenarios.

In case of clipping ratio=0.5, the graphs (Fig. 5-Fig. 9) show that IIM and CIM may not outperform the standard benchmark unless the simulation is very hard to explore in which the CIM shows the win rate up to 2% instead of 0%. The CIM shows a more powerful exploration in the case of multi-agent setup as shown in 3s5z environment. However, the CIM usually requires larger training episodes to achieve the same performance compared to the IIM and No intrinsic motivation in the easier environment.

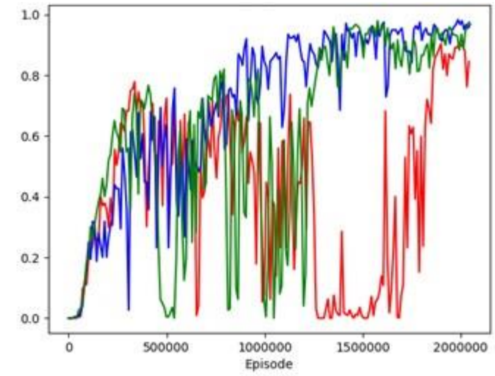

Fig. 5. Win rate of 2s_vs_1sc with Clipping Ratio = 0.5.

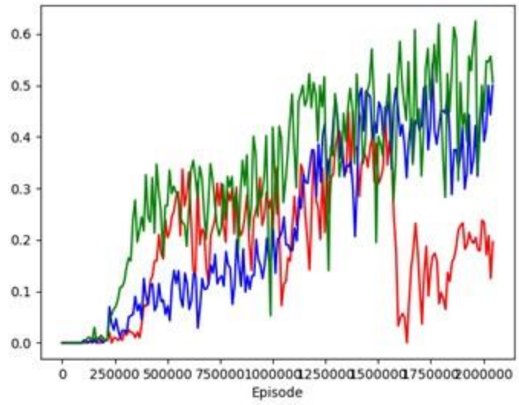

Fig. 6. Win rate of 2s3z with Clipping Ratio = 0.5.

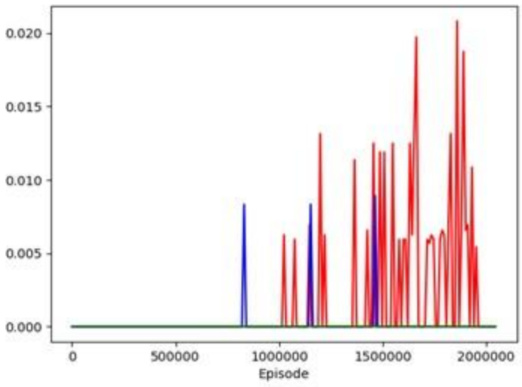

Fig. 7. Win rate of 3s5z with Clipping Ratio = 0.5.

The results with 0.2 clipping ratio are shown in Fig. 10-14. Observing that the performance of IIM and CIM should be more similar with the benchmark (green line) as the ratio is decreased. The results show overall higher performance for intrinsic motivation. Observing that CIM shows a solid trend to outperform IIM in the long run, but it always shows a higher overhead cost. The intrinsic motivation also shows a significant higher result than the benchmark especially in the map of 2s3z where the benchmark results only show 43% win rate as shown in [8]. Overall, the type of environments still has no significant relationship on the condition of difficulty of the environment. The findings show that the Win rate performance is merely the reflection of environment complexity in terms of exploration and optimization, not the human defined environment type.

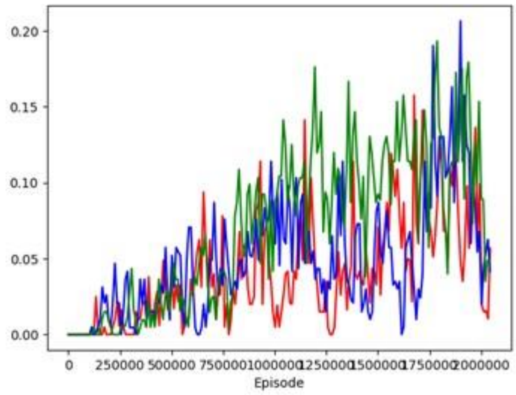

Fig. 8. Win rate of 1c3s5z with Clipping Ratio = 0.5.

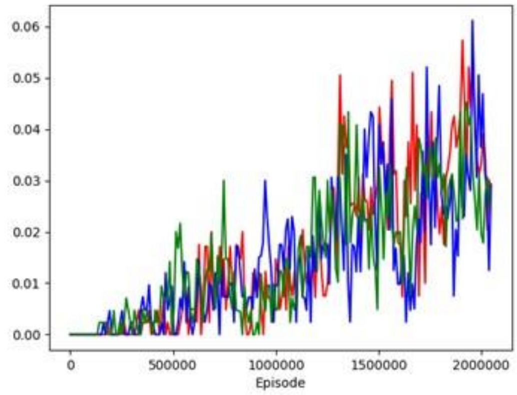

Fig. 9. Win rate of 10m_vs_11m with Clipping Ratio = 0.5.

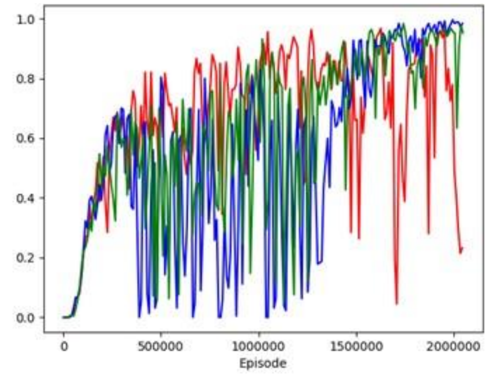

Fig. 10. Win rate of 2s_vs_1sc with Clipping Ratio = 0.2.

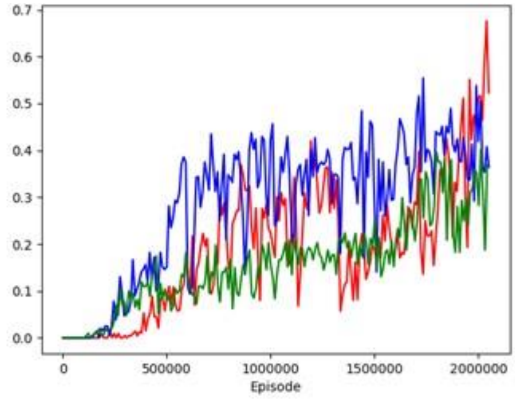

Fig. 11. Win rate of 2s3z with Clipping Ratio = 0.2.

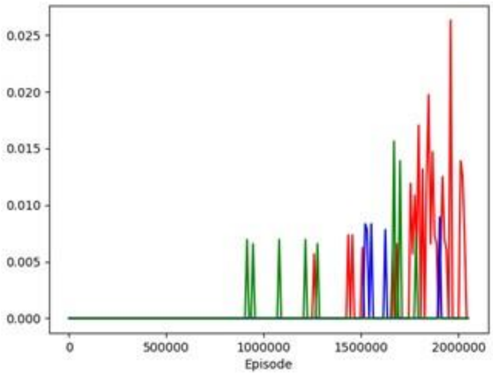

Fig. 12. Win rate of 3s5z with Clipping Ratio = 0.2.

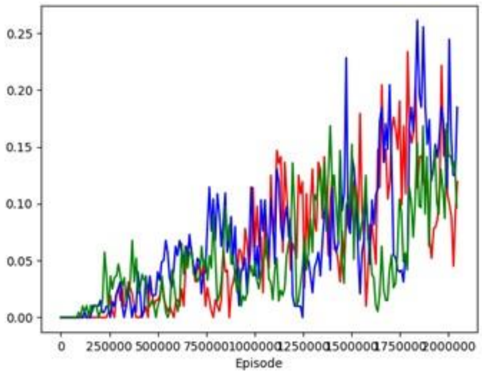

Fig. 13. Win rate of 1c3s5z with Clipping Ratio = 0.2.

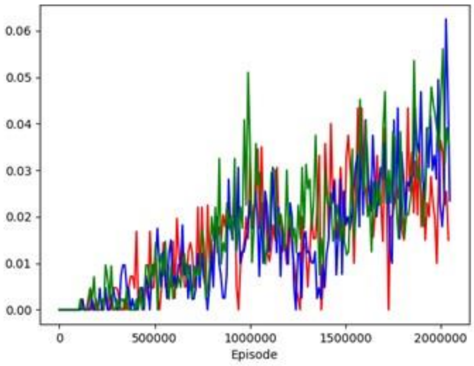

Fig. 14. Win rate of 10m_vs_11m with Clipping Ratio = 0.2.

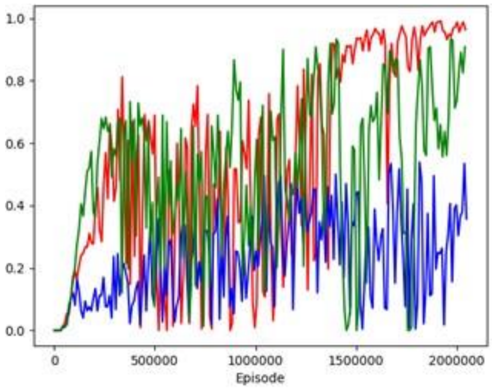

Fig. 15. Win rate of 2s_vs_1sc with Clipping Ratio = 1.

Fig. 15-Fig. 19 illustrate the comparisons of win rates using clipping ratio of 1. The results show diminishing performances of IIM compared to those using lower clipping ratio situations. The broader clipping value allows more extrinsic advantage of an individual agent to the optimizer and its directions, intuitively, are less likely to align with each other's. Imagine that each individual agent has different curiosity, some different states may have the same of intrinsic motivation but with the different intrinsic distribution. In the case of CIM, the combined intrinsic value is generated from associated observations. Its value thus reflects on the overall curiosity at once. Therefore, the intrinsic advantages are gradually changed and become more deterministic than the other setups.

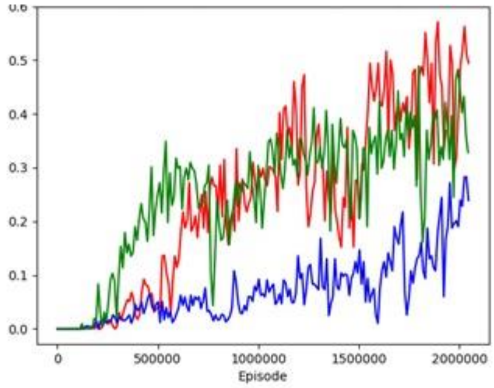

Fig. 16. Win rate of 2s3z with Clipping Ratio = 1.

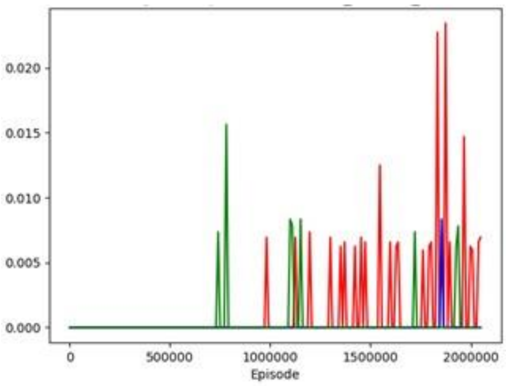

Fig. 17. Win rate of 3s5z with Clipping Ratio = 1.

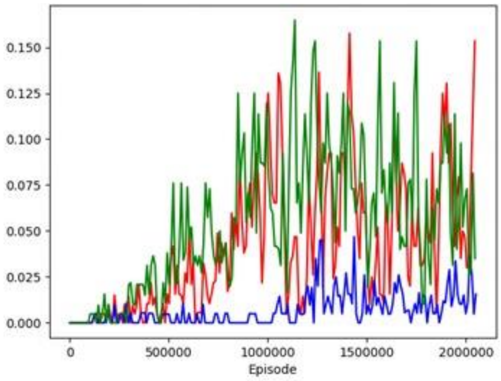

Fig. 18. Win rate of 1c3s5z with Clipping Ratio = 1.

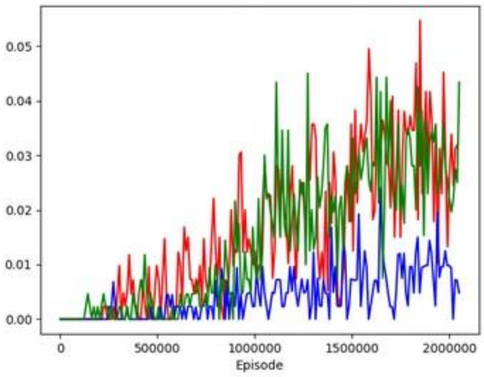

Fig. 19. Win rate of 10m_vs_11m with Clipping Ratio = 1.

V. Conclusion

The research has explored the effectiveness of Intrinsic Motivation on multi-agent setups in 2 different dimensions: architecture and clipping ratio of the advantage values. The intrinsic motivation derived from the RND technique is constructed in two architectures: Individual and Centralized Intrinsic Motivation and tested on each architecture with three different values of clipping ratios: 0.2, 0.5, and 1.

The experimental results show the win rate improvement over the benchmark algorithm (COMA) in the more complex environment as the intrinsic motivation helps better exploring an unknown state and it is more likely to move out of local optimal path. The CIMA even achieved almost 70% win rate on 2s3z with 0.2 clipping ratio while the benchmark record is only 43% in the benchmark, SMAC paper. Although, the optimal clipping ratio is not yet explored, we do know that the appropriate scale of intrinsic advantage could help improve overall long run performance of the algorithm. The main optimization objective relies on the designed reward function in terms of extrinsic motivation. It is intuitive that the magnitude of extrinsic motivation should be more than that of intrinsic motivation in the long run, especially at the end. Therefore, the value of clipping ratio greater than one shows a significant drop in performance for both CIM and IIM because this allows intrinsic motivation to have a greater magnitude than extrinsic motivation. Whereas setting intrinsic ratio to zero in either CIM or IIM means the case of none of intrinsic motivation.

Further investigation on other clipping ratios or some other aspects rather than win rate may reveal more underlying insight of intrinsic motivation in the optimization. There are also many more aspects of the intrinsic motivation on multi-agent reinforcement learning for further research conduct. For example, the area of architecture of intrinsic network, the distribution of intrinsic motivation to each individual agent, and the study on impact of shared knowledge agents, competitive agent with intrinsic motivation in general.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

In the research, the first author, Charoenpitaks, initiated ideas and designed the research processes. Furthermore, he conducted the experiments and wrote the research paper. Limpiyakorn, as the corresponding author, advised the methodology, shared insight, analyzed the results, in addition to review and revise the research paper.

Acknowledgment

C. Korawat would like to thank Chulalongkorn University for financial support on the research. The completion of this research could not have been possible without the support from the University.

References

[1] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," arXiv preprint arXiv: 1611.05397, 2016.

[2] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16–17.

[3] A. Aubret, L. Matignon, and S. Hassas, "A survey on intrinsic motivation in reinforcement learning," arXiv preprint arXiv: 1908.06976, 2019.

[4] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

[5] O. Pierre-Yves, and K. Frederic, "What is intrinsic motivation? A typology of computational approaches," *Frontiers in Neurorobotics*, vol. 1, no. 6, Nov. 2007.

[6] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. J. Strouse, J. Z. Leibo, and N. De Freitas, "Social influence as intrinsic motivation for multi-agent deep reinforcement learning," arXiv preprint arXiv: 1810.08647, 2018.

[7] S. Iqbal and F. Sha, "Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning," arXiv preprint arXiv: 1905.12127, 2019.

[8] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. Rudner, C. Hung, P. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," in *Proc. the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 2186–2188.

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

[10] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," arXiv preprint arXiv: 1909.07528, 2019.

[11] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," arXiv preprint arXiv: 1810.12894, 2018.

Korawat Charoenpitaks was born in Bangkok. He holds the bachelor of electrical engineering in 2013 and master of finance in 2017 from Curtin University, Australia. He earns the degree of master of science in computer science from Chulalongkorn University in 2020. His interests are in new emerging technology such as AI, Blockchain and Fintech. His previous research is in the fields of reinforcement learning and renewal energy. Currently, he works as a data scientist in Bangkok.