AFRL-AFOSR-JP-TR-2021-0010

Foundational Aspects of Machine Learning in Multi-Agent Online Games as Serious Games

Yi, Sungwon
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu
Daejeon, , 34129
KR

08/05/2021
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
05-08-2021	Final	31 Jul 2019 - 30 Mar 2021

4. TITLE AND SUBTITLE
Foundational Aspects of Machine Learning
in Multi-Agent Online Games as Serious Games

5a. CONTRACT NUMBER
FA2386-19-1-4020

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sungwon Yi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu
Daejeon, 34129
KR

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2021-0010

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This seedling project investigated the complexities of scaling reinforcement learning algorithms, using commercial-off-the-shelf strategy games as the experimental environment. A major research contribution is an multi-agent reinforcement learning (MARL) approach where an agent learns policies based on other agents, rather than attempting to learn policy by itself, independent of the other agents. The research team also investigated using the Grey Wolf Optimizer simulate human-AI teaming and training, as a simpler method to approximate training conditions with hybrid teams. The project produced three papers and two patent submissions. In addition, software developed for the experiment is openly hosted on GitHub.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			ALAN LIN
U	U	U	SAR	10	**19b. TELEPHONE NUMBER** *(Include area code)* 227-7009

Final Report

Foundational Aspects of Machine Learning in Multi-Agent Online Games as Serious Games

Grant Number: FA2386-19-4020 (AFOSR/IOA)

Date Submitted: June 25, 2021

Primary Investigator
Dr. Sungwon Yi, Managing Director
Planning Division
ETRI
218 Gajeong-Ro, Yusung-Gu, Daejon
Phone: 82-10-3002-0170
E-mail: sungyi@etri.re.kr

1 Introduction

In multi-agent systems, agents are constantly interacting with the environment as well as other agents, which can add the complexity of the problem and make it difficult in finding efficient solutions [1–3]. In this seeding project, we investigate various research issues in multi-agent systems emanating from the sources of complexity including experimental environment, Deep Neural Networks (DNN) architecture design for a real-time strategic game, and exploration strategies based on the decision-making policies of other agents.

To properly investigate different approaches in complex real-time systems such as DOTA2, the experimental environment must support massively scaled RL algorithms using hundreds of thousands of CPUs and GPUs interacting with game servers in real-time, and which requires huge engineering efforts and hardware resources.

Considering these constraints, we focused on a simpler experimental framework called DotaClient [4]. Instead of exploring all the design issues in building the entire experimental framework, we extended the understanding of the basic mechanisms required for a subsequent large-scale research work. We extended DotaClient to support multi-agent gameplay and implemented a widely used multi-agent reinforcement learning (MARL) technique called QMIX [5]. This work is supported by the Korean company RTST who has been developed machine learning-based training programs (games) for ROK Navy and Army. The developed software is publicly available from July 2021 [6].

Also, we investigated a fundamental issue in MARL that mainly comes from the interagent interactions, which are often invisible to other agents. In this work, we capture the behaviors of other agents and use them to update the agent's policy either by directly learning from others' policies or checking that others' behavior is worth learning. The first approach has produced patent applications in Korea and U.S and two papers for each approach are now prepared for a conference submission or under review for publication. For this work, two researchers each from Future Strategy Research Lab, Currently Planning Division and Artificial Intelligence Lab in ETRI collaborated through bi-weekly meetings, where research progress, results, and ideas were shared. PSU also participated in the meeting as needed. Despite our efforts to enhance collaboration, the research collaboration activities were limited due to COVID19. However, PSU provided theoretical background and analysis of our approaches, which became a valuable foundation to our research.

Furthermore, we also investigated two important concepts in MARL with two Korean universities. One of the main obstacles in MARL is in a huge state space comes from interagent interaction as mentioned before. This study used state categorization and self-attention to reduce the state complexity and to identify the relationship between agents. The work is conducted in collaboration with a research team, led by Prof. Joongheon Kim from Korea University.

Finally, a careful discussion on the best approaches to developing hybrid AI/human players was deliberated. However, adding human players among AI agents involves several challenges, such as collecting appropriate training data for human behaviors. As an alternative, we propose to adopt the idea of Grey Wolf Optimizer, where the entire pack of wolves is controlled by a small number of high-rank wolves called Alpha [7]. We believe that the idea of training Alpha, a handful of agents, instead of training the entire pack of agents is hugely beneficial in cutting computational and training-time costs. Currently, this work is in its infancy and needs follow-up research, excluding the details of this issue in this report. However, the very basic idea will be patented in Korea by the end of this year.

The research outputs of this seeding project are:

- (Software) "QMIX extension for DotaClient", https://github.com/etri/dotaclientQMIX
- (Research paper, Patent) "Curiosity-based Multi-Agent Reinforcement Learning" submitted to IEEE CoRL 2021
- (Research paper) "A Novel and Efficient Influence-Seeking Exploration in Multi-Agent Reinforcement Learning", in preparation for conference submission
- (Research paper) "Multi-Agent Deep Reinforcement Learning using Attentive Graph Neural Architectures for Real-Time Strategy Games", submitted to IEEE SMC 2021

2 MARL experimental framework for DOTA2 [6]

DOTA2 [8] is a complex multi-agent role-playing game whose dataset is much more complex than most standard multi-agent reinforcement-learning benchmarks, such as OpenAI Gym [9]. Training multiple agents in DOTA2 require an experimental environment that supports massively scaled RL algorithms using hundreds of thousands of CPUs and GPUs, interacting with real-time game servers and huge engineering efforts and hardware resources.

Considering the time and budget for this one-year seeding project, we focused on a simple experimental framework, DotaClient [4], instead of exploring all design issues in building the entire experimental framework. DotaClient, an open-source project, is a DOTA2 experimental framework supporting one vs one self-playing game, built upon DotaService [10] that enables playing DOTA2 through gRPG. In DotaClient, the DOTA2 game server resides in a docker and communicates with multiple agents so that the entire training and playing can be implemented in a single machine. The learning framework consists of four main modules: environment, agent, optimizer, and message broker. Here DotaService is used as an environment module. Additional features required for DotaClient are provided by the agent and optimizer modules, and RabbitMQ is used as a message broker module. Although DotaClient exploited a relatively simple architecture, and thus well suited for a simulation study focusing on feasibility. The core feature of the multi-agent RL study, the interaction among the agents, cannot be captured due to the limited number of agents and PPO, a widely referred RL algorithm, does not specifically target multi-agents RL.

This project extended DotaClient to support up to five agents and implemented two basic MARL algorithms [6]: QMIX [5], a widely referred algorithms in the literature, and IQL [11], a basic MARL algorithm. For a deeper understanding of DOTA2 and MARL algorithms, we implemented QMIX and IQL algorithms instead of directly using the publicly available source code. We believe that these two algorithms will be useful benchmarks in the comparative analysis on RL algorithms in the DOTA2 environment. However, the limitation of single machine-based experiments will be obstacles with the current implementation considering the complexity of DOTA2. We plan to extend the current multi-agent DotaClient to a fully distributed simulation testbed in the future.

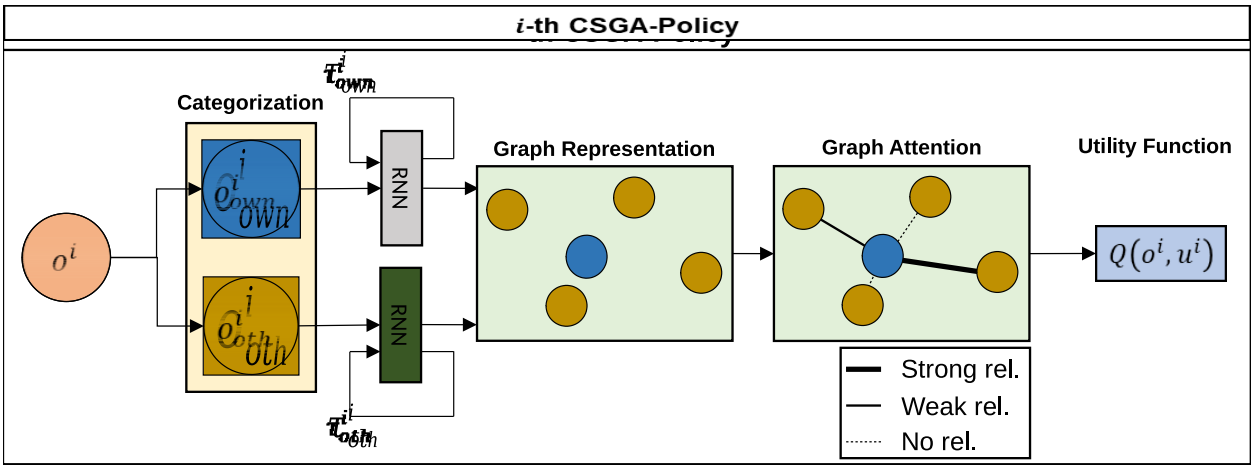


Figure 1: The pipeline of i-th agent's CSGA-policy.

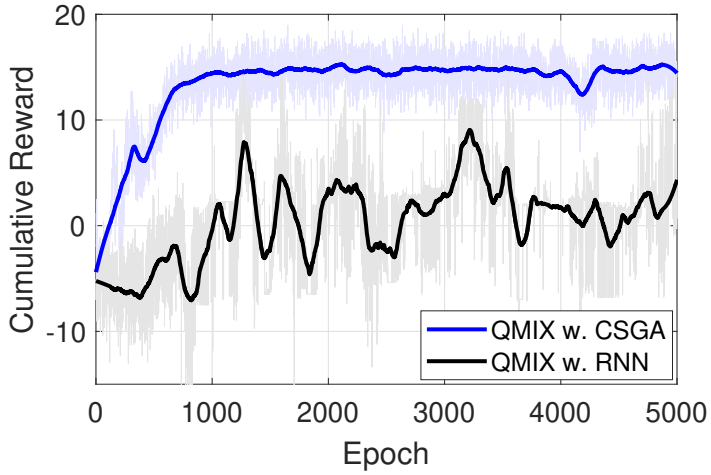
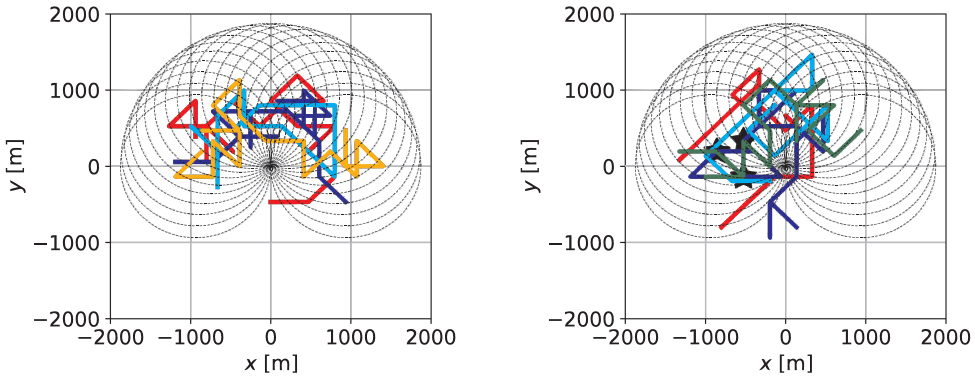


Figure 2: Cumulative reward.

3 Categorized State Graph Attention MARL [12]

In a multi-agent system, agents can obtain state information consisting of common observation and agents' unique state information. For example, multi-UAVs obtain one agent's unique information and other agents' partial state information including relative position, relative distance, and all other observable partial information. From these examples, we proposed a state categorization technique. In this technique, an RL engineer first analyzes the agent's state vector structure (e.g., from the first element to the k-th element of the state vector is the l-th agent's state information). Then, all elements are categorized into several groups (e.g., an agent's unique information and other agent's information). The state categorization technique is a simple yet efficient method in making a graph representation. With the self-attention mechanism, the edges of the graph are defined using the relationship between nodes [13]. We call this *Categorized State Graph Attention* (CSGA) MARL.

Fig. 1 shows the pipeline of CSGA. First, the state information is categorized into the agent's state and others' partial state. Second, the categorized state information is encoded with a recurrent



(a) QMIX w. CSGA. (b) QMIX w. RNN.

Figure 3: Trajectories of multi-UAVs.

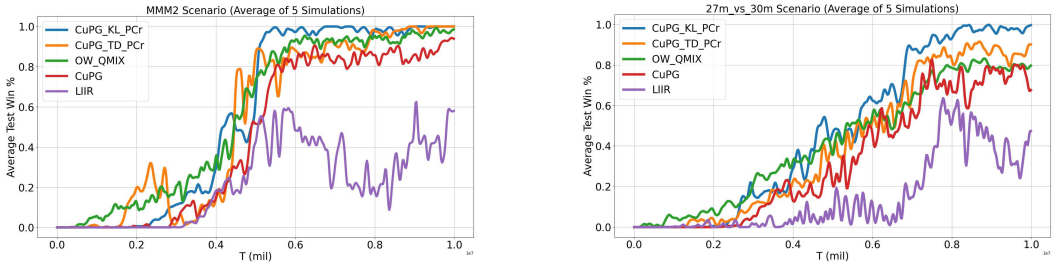
neural network (RNN). Third, agents locally make a star-topological graph via graph attention. With the graph attention mechanism, the edge of the graph is trained and affects the action-value function $Q^{\theta_i}(o_t^i, \tau^i)$. That is, the agent determines its actions using graphs and its own trajectories τ_i. All CSGA-policies are trained using centralized methods [**?**] (e.g., QMIX), whereby agents can make decisions in cooperative manners.

The contributions of CSGA policy can be summarized in two-fold:

- *Data efficiency*: Compared to fully connected layers, which are commonly used for RL policy, CSGA better exploits the agent's state information. It converts agent state information into nodes of graphs used by attention whereby the data efficiency increases.

- *Graph attention in decentralized MARL*: *G2ANet* [14] is based on graph attention in a centralized MARL system. Unlike a centralized MARL system, a policy using graph attention does not exist in a decentralized MARL system since it is constrained by unsecure interagent communication. However, CSGA-policy makes a star-topological graph via state categorization. Therefore, graph attention can be used in a decentralized MARL system if the policy consists of CSGA-policy.

For the performance evaluation, simulations of QMIX with CSGA and QMIX with RNN were conducted on a multi-UAVs environment that guaranties URLLC requirements [5, 15, 16]. In a multi-UAV environment, agents should follow the target area and evade collision simultaneously. Fig. 2 shows the cumulative reward that QMIX with CSGA outperforms QMIX with RNN regarding convergence speed, cumulative reward, and variance of reward. Fig. 3 shows the trajectory of UAV agents that QMIX with CSGA follows the target while QMIX with RNN does not follow the target and collision occurs. Although state structure analysis can be viewed as a constraint, we believe that CSGA-policy is suitable for centralized training and decentralized execution supported MARL systems particularly if the application-specific analysis is inexpensive.



(a) MM2 Scenario

(b) 27m vs 30m Scenario

Figure 4: Average test-win rate for super-hard scenarios

4 Curiosity-based MARL

Observation in social science is known as a primary source of the intellectual process of acquiring knowledge. Children often observe and imitate their older siblings and such activities are known to contribute to their cognitive development [17, 18]. In this work, we propose a noble curiosity-based learning technique inspired by this concept. The idea of curiosity-based learning is in exploiting others' knowledge as children do and which can affect an agent's learning process either in positively or negatively. However, the agent can quickly investigate others' various experiences, and thus the training time can be significantly reduced.

To compare the knowledge of each agent, we introduce the curiosity concept using conditional mutual information. More specifically, each agent manages the long-term visitation counts for others and uses these numbers in calculating the conditional mutual information among agents, used to update its decision-making policy. In addition, we extend the curiosity concept to experience sampling as well. Reply buffer is a widely adopted technique for addressing the sample efficiency problem in MARL. However, an agent selects the samples based on the maximum expected reward if any prioritization technique is used, and thus agent's experience can be biased. We address the problem using Kullback-Leibler divergence-based prioritization in sampling experiences and it helps an agent to build more various experiences and in turn to reach a better learning curve.

The proposed technique was evaluated in StarCraft Multi-Agent Challenge (SMAC) [19], a widely used simulation testbed for multi-agent systems, and compared with the following baseline techniques: LIIR

which is one of the most widely used simulation testbed for Multi-agent systems, and compared with the following baseline techniques: LIIR [20] , OW_QMIX [21], CuPG(Individual agent's curiosity-based policy gradient method), and CuPG_TD_PCr(Curiosity-based policy gradient with a temporal difference (TD)-error based prioritization). Figures 4 shows the average test-win rates for super-hard scenarios with MMM2 and 27 m vs 30 m, respectively. The results indicate that the proposed method outperforms the previous techniques in win rate and learning speed as well.

5 Influence-seeking exploration

In MARL, an agent can be significantly affected by the actions of other agents, inducing different values of rewards. Also, partially observable information about other agents, make the problem more intractable in many real-world applications. Therefore, an efficient exploration strategy to capture the effect of other agents, which is called influence, is necessary for combating the problems in MARL. In this research, the influence is defined as the variance of joint action-values with different actions of agents, which measures the effect of other agents in terms of an expected return. Since directly computing the variance of joint action-values is computationally expensive, two types of approximation techniques are proposed: variance-based and range-based influences.

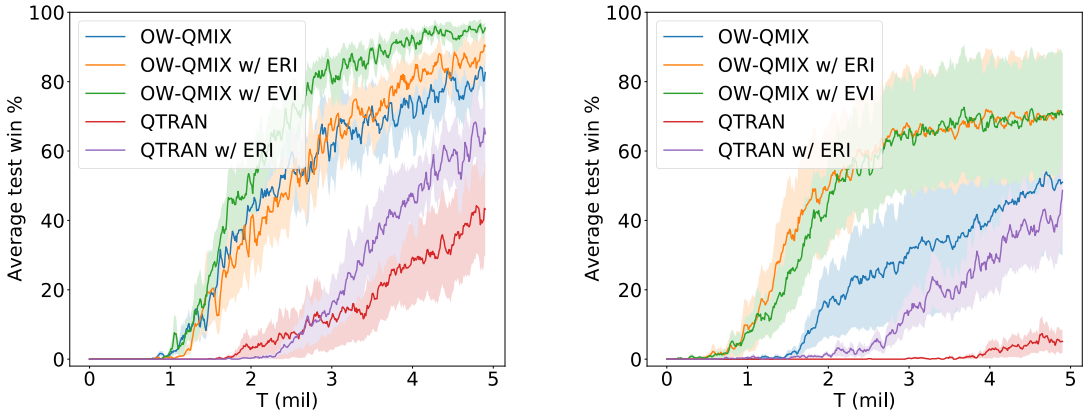
Variance-based influence calculates the variance of joint action-values from the approximated variance propagation [22]. In the approximated variance propagation, input variances are defined first, and then they are propagated through a function using the Jacobian of the function. It is analytically tractable; however, it is not without some computational costs in calculating the Jacobian of the function. For more practical use, range-based influence is also proposed. This influence is calculated as the difference between maximum and minimum of joint action-values over different actions of other agents. Finding both the global maximum and minimum of the joint action-values is analytically intractable, hence local optimal values are used when choosing the best and worst action from each individual's viewpoint.

The calculated influence is incorporated into the exploration method as an exploration bonus to find actions highly influenced by others. The exploration strategies with these two proposed influences are called *exploration by variance-based influence* (EVI) and *exploration by range-based influence* (ERI), respectively.

The proposed exploration strategy encourages agents to actively explore action spaces influenced by other agents. These action space can have complex cooperative behaviors that are difficult to find using existing exploration methods. In addition, the proposed method effectively avoids local minima using the exploration bonus. The proposed method was also verified in StarCraft Multi-Agent Challenge (SMAC) environments. OW-QMI [21] and QTRAN [23] were used as the baseline algorithms for MARL. As shown in Fig. 5, we verified that EVI and ERI improve the performance and convergence speed of the baselines. In particular, OW-QMIX with EVI shows the best performances among the methods compared.

Table 1: Final average test win rate % in different scenarios

SCENARIO	OW-QMIX	OW-QMIX W/ ERI	OW-QMIX W/ EVI	QTRAN	QTRAN W/ ERI
3S VS 5Z	92.5	95.0	**98.8**	50.0	70.6
MMM2	56.9	73.8	**74.4**	10.0	56.3



(a) 3s vs 5z (b) MMM2

Figure 5: Average test win rates with different battle scenarios in SMAC environment

6 Conclusions and Future Research

In this work, we investigated two important issues in MARL research: DNN architecture design and RL algorithms. In DNN architecture design, we focused on the optimized use of available information by each agent. To this end, state categorization was proposed and the simulation-based evaluation showed the improved performance. For RL algorithms, we focused on capturing the behaviors of other agents and using them in the learning process. For this, curiosity-based learning and influence-seeking exploration techniques are proposed. In the simulation-based performance evaluation in SMAC, the proposed techniques exhibit better performance compared to existing techniques. We plan to conduct more evaluation studies on these approaches with full DOTA2 or StarCraft in the future. During an in-depth survey on MARL, we found that Grey Wolf Optimizer can serve as a good candidate for studying human and AI collaboration. Our future work will include issues in the experimental framework for human and AI collaboration (including GWO), and effective MARL techniques for a large number of agents.

References

[1] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," 2015.

[2] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," 2017.

[3] P. Peng, Y. Wen, Y. Yang, Q. Yuan, Z. Tang, H. Long, and J. Wang, "Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games," 2017.

[4] T. Zaman, "DotaClient," https://github.com/TimZaman/dotaclient.

[5] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," 2018.

[6] D. Lee, "QMIX Extension to DotaClient," https://github.com/etri/dotaclientQMIX.

[7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965997813001853

[8] "Defense of the Ancients(DOTA) 2," https://www.dota2.com.

[9] "GYM: A Toolkit for Developing and Comparing Reinforcement Learning Algorithms," https://gym.openai.com.

[10] T. Zaman, "DotaService," https://github.com/TimZaman/dotaservice.

[11] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *In Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, 1993, pp. 330–337.

[12] W. J. Yun, S. Yi, and J. Kim, "Multi-agent deep reinforcement learning using attentive graph neural architectures for real-time strategy games," *arXiv preprint arXiv:2105.10211*, 2021.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[14] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7211–7218.

[15] W. J. Yun, B. Lim, S. Jung, Y.-C. Ko, J. Park, J. Kim, and M. Bennis, "Attention-based reinforcement learning for real-time uav semantic communication," *arXiv preprint arXiv:2105.10716*, 2021.

[16] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *arXiv preprint arXiv:1507.06527*, 2015.

[17] M. Azmitia and J. Hesser, "Why siblings are important agents of cognitive development: A comparison of siblings and peers," *Child Development*, vol. 64, pp. 430–444, 1993.

[18] G. Brody, "Sibling relationship quality: Its causes and consequences," *Annual Review Psychology*, vol. 49, pp. 1–24, 1998.

[19] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," 2019.

[20] Y. Du, L. Han, M. Fang, J. Liu, T. Dai, and D. Tao, "Liir: Learning individual intrinsic reward in multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 4403–4414.

[21] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[22] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2931–2940.

[23] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5887–5896.