applied sciences

MDPI

Article

Attention-Based Mechanisms for Cognitive Reinforcement Learning

Yue Gao [1], Di Li [2], Xiangjian Chen [1,*] and Junwu Zhu [1]

[1] College of Information Engineering, Yangzhou University, Yangzhou 225009, China
[2] China Shipbuilding Industry Corporation, Yangzhou 225001, China
* Correspondence: cxj831209@163.com; Tel.: +86-177-5136-0234

Abstract: In this paper, we propose a cognitive reinforcement learning method based on an attention mechanism (CRL-CBAM) to address the problems of complex interactive communication, limited range, and time-varying communication topology in multi-intelligence collaborative work. The method not only combines the efficient decision-making capability of reinforcement learning, the representational capability of deep learning, and the self-learning capability of cognitive learning but also inserts a convolutional block attention module to increase the representational capability by using the attention mechanism to focus on important features and suppress unnecessary ones. The use of two modules, channel and spatial axis, to emphasize meaningful features in the two main dimensions can effectively aid the flow of information in the network. Results from simulation experiments show that the method has more rewards and is more efficient than other methods in formation control, which means a greater advantage when dealing with scenarios with a large number of agents. In group containment, the agents learn to sacrifice individual rewards to maximize group rewards. All tasks are successfully completed, even if the simulation scenario changes from the training scenario. The method can therefore be applied to new environments with effectiveness and robustness.

Keywords: deep reinforcement learning; cognitive learning; attentional mechanisms; multi-intelligent body collaboration

Citation: Gao, Y.; Li, D.; Chen, X.; Zhu, J. Attention-Based Mechanisms for Cognitive Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 7361. https://doi.org/10.3390/app13137361

Academic Editor: Alexander N. Pisarchik

Received: 30 May 2023
Revised: 15 June 2023
Accepted: 15 June 2023
Published: 21 June 2023

1. Introduction

In today's society, there are many problems that need to be solved cooperatively. The advantage of cooperation is that it can accomplish many complex tasks that cannot be accomplished by a single intelligence. A large task can be divided into many small parts to complete, and each intelligence performs its own task while taking into account the cooperation with other intelligence, finally making the task successfully completed. In recent years, deep reinforcement learning (DRL) has made significant advances in single-intelligent environments [1–4]. To facilitate the cooperative behavior of multiple intelligence, multi-agent reinforcement learning (MARL) [5–9] based on has emerged.

However, they all have their limitations. In multi-subject reinforcement learning, each intelligence has to interact with other intelligence and the environment, which may lead to a dynamic and unstable environment. As a result, the learning strategy may change frequently during the training process, causing the behavior of the intelligence to become unstable. Moreover, multi-subject reinforcement learning requires multiple intelligence to collaborate in order to maximize the joint rewards obtained. However, the interactions between the intelligence during training may lead to an increase in the complexity of the problem, making collaborative learning more difficult. There is also the high computational complexity of multi-subject reinforcement learning, as the strategies and value functions of multiple intelligence need to be processed simultaneously. This can lead to training times becoming very long, making real-time applications infeasible. Information sharing

between intelligence is important in multi-subject reinforcement learning. However, in some cases, the intelligence may not be able to share information, resulting in less efficient learning. As the network for each intelligence is trained in a fixed number of intelligence environments, it needs to be trained again when the number of intelligence changes.

To address this problem, inspired by attentional mechanisms [10], we propose an attentional mechanism-based approach, the attention-enhanced cognitive reinforcement learning method.

2. Research on Multi-Intelligence Reinforcement Learning Methods

2.1. Application of Reinforcement Learning to Multi-Intelligent Collaboration

Reinforcement learning has made significant breakthroughs in recent years in games such as Go, Poker, and Starcraft, as well as many advances in areas such as robot control and natural language processing. The main components of reinforcement learning [3,11] include the intelligent body, the environment, the state, the action, and the reward. An intelligent body interacts with its environment by observing the current state and performing actions. The environment responds to the intelligence's actions and current state, and the intelligence then receives a reward signal to determine whether it has acted correctly. The goal of the intelligence is to find an optimal strategy that maximizes the long-term reward obtained in the environment [12].

Reinforcement learning applies to dynamic environments because it can adaptively adjust its strategy to adapt to changes in the environment. Reinforcement learning can also handle uncertainty as its decisions are based on the current state and possible future states. Reinforcement learning can also handle long-term decisions because its decisions are based on the expectation of possible future rewards. Reinforcement learning can learn optimal policies, i.e., policies that maximize rewards and can handle continuous actions and state spaces [13–15], because it uses function approximation to estimate value functions and policies. This is an advantage of reinforcement learning.

The feasibility and effectiveness of reinforcement learning in multi-intelligent collaboration [16] have been initially demonstrated, but there are still some remaining issues. Firstly, in multi-intelligent collaboration, the interactions and influences between the intelligence are very complex, and therefore, designing suitable reinforcement learning algorithms to optimize the overall gain of all the intelligence is a very challenging problem. Secondly, the non-stationarity and convergence of reinforcement learning algorithms are challenged by the fact that strategies and behaviors of the intelligence constantly interact and adapt to each other. In addition, due to possible competition and conflict, the multi-intelligent collaboration problem may have multiple locally optimal solutions for the Nash equilibrium point [17], which also adds to the algorithm's complexity. Third, in multi-intelligent body collaboration, the intelligence need to communicate with each other to coordinate their actions, which increases the communication cost. In addition, the computational cost can be very high for large-scale intelligent body systems. Fourthly, there may be competitive relationships between intelligence in multi-intelligence collaboration, i.e., their action goals are not always the same. This competitive relationship can lead to algorithms in trouble, as each intelligence tries to maximize its own payoff without considering the overall payoff of the whole system. Fifth, in multi-intelligent collaboration, the interactions between intelligence are complex and dynamic. Reinforcement learning algorithms need to be able to adapt to such interactions and consider how to work with other intelligence to maximize the overall payoff. This requires the algorithm to be able to understand the actions and strategies of other intelligence and make decisions accordingly.

2.2. Deep Reinforcement Learning for Multi-Intelligence Collaboration

In recent years, thanks to big data, increased computing power, and algorithmic breakthroughs, deep learning techniques have made impressive achievements. Combined with the advantages of deep neural networks, there have also been breakthroughs in reinforcement learning, particularly deep reinforcement learning [18]. In deep reinforce-

ment learning, machine learning models use neural networks to learn a mapping between state space and action space in order to select the best action in the environment. The model learns by interacting with the environment, obtaining feedback signals from the environment, and adapting its action strategy based on these signals in order to achieve a specific goal.

Deep reinforcement learning can be applied to multi-intelligent collaboration tasks, such as collaboratively carrying objects or maintaining a certain distance during collaboration [19]. Each intelligence can learn its own strategy and collaborate with other intelligence to complete the task. It can also be applied to multi-intelligence adversarial tasks, where each intelligence learns its own strategy and competes with the others to achieve the highest score. Alternatively, distributed reinforcement learning in multi-intelligence collaboration can be distributed across different computers to help the intelligence learn the task together. In multi-intelligent collaboration, it is sometimes difficult to obtain enough labeled data to train deep reinforcement learning models.

Nevertheless, there are also unsolved challenges in deep reinforcement learning. For example, DRL algorithms often require a large amount of training data in order to learn high-quality strategies. This may not be feasible for some application scenarios as the cost of collecting data can be very high or take a very long time. If the initial state is set poorly, it may cause the algorithm to fall into a local optimum and fail to learn the global optimum. In addition, DRL algorithms often use neural networks to approximate value functions or policies. However, neural networks can only handle discrete action spaces and require special treatment for continuous action spaces, such as using a Gaussian distribution to sample actions, which can lead to the increased complexity of the algorithm. Additionally, DRL algorithms often use black box models such as neural networks for learning, which makes the decision process of the algorithm uninterpretable and makes it difficult to understand why a particular action or decision was chosen. Also, DRL algorithms often use parametric models such as neural networks for learning, which can be prone to overfitting.

2.3. Cognitive Learning in Multi-Intelligence Collaboration

Multi-agent collaboration (MAC) is a process in which multiple intelligence collaborate and interact in a common task to achieve a common goal. Cognitive learning is an important learning method that can help intelligence to better understand their environment, learn and make decisions in MAC.

I. The Concept of Cognitive Learning

Cognitive learning is an interdisciplinary discipline based on cognitive psychology and computer science that aims to study how intelligence learns from external information and understands and responds to problems in complex environments [20]. Cognitive learning is concerned with the following aspects: how to represent and store knowledge and how to learn from data; how to obtain information from the environment and how to select important information and process it; how to make decisions based on environmental states and goals and to realize these decisions through planning; how to exercise cognitive control and coordination in a multitasking and uncertain environment; how to use language and interactions to help intelligence communicate and collaborate with each other.

II. Application Scenarios of Cognitive Learning in MAC

In multi-intelligence collaboration, each intelligence needs to make decisions and plans based on the current environmental state and task goals. These decisions and planning need to take into account the behaviors and decisions of other intelligence in order to achieve overall collaboration. Cognitive learning helps the intelligence to learn appropriate strategies based on historical experience and environmental information and to update and optimize them in new environments. For example, in robot collaboration, each robot needs to plan its path considering the positions and movements of other robots to avoid conflicts and coordinate movements. In multi-intelligent collaboration, task allocation is a

very important issue. Through cognitive learning, intelligence can learn how to allocate tasks in order to achieve optimal results.

3. Attention-Enhanced Cognitive Reinforcement Learning (CRL-CBAM)

3.1. Cognitive Learning

The cognitive learning algorithm (CLA) [21], proposed by Qihui Wu et al., is a machine learning algorithm inspired by cognitive models in neuroscience. The core idea of the CLA is to encode input data as sparsely distributed active units using a neuronal model called a "perceptron", and then to classify these units using a set of neuronal models called "clusters". These active units are then combined into high-level features using a set of neuronal models called "clusters". These features are then passed to another set of neuronal models, called "classifiers", to achieve the classification task.

The framework consists of five modules, including a cognitive feature extraction module, a cognitive control module, a learning network module, a cognitive evaluation module, and a memory module. The memory module consists of three spaces: the database (DB), the cognitive case base (CCB), and the algorithmic hyperparameter base (AHB). The core modules of the framework are cognitive feature extraction, cognitive control, cognitive evaluation, and cognitive case spaces. The cognitive feature extraction module captures features of dynamic environments and tasks and can reflect changes in the environment and tasks. The cognitive feature extraction module helps the cognitive control module quickly select the right type of algorithm and hyperparameters when the environment and task change. The cognitive control module creates matching relationships between a dynamic environment and task features to select the appropriate algorithm type and hyperparameters so that the framework can adapt to changes in the environment and task. During the offline self-learning process, the matching relationships can be continuously updated, and thus knowledge is accumulated, which facilitates the selection of the most appropriate algorithm type and hyperparameters. The cognitive evaluation module evaluates the performance of the selected algorithm types and hyperparameters so that the cognitive case space can accumulate better knowledge. The cognitive case space can accumulate knowledge of the relationships between the dynamic environment and task characteristics, as well as knowledge of the relationships between the selected algorithm and hyperparameters, thus reducing the impact of bad knowledge of inappropriate matching relationships.

3.2. A Mathematical Framework for Attention-Enhancing Cognitive Reinforcement Learning

Building a mathematical framework for attention-enhancing cognitive reinforcement learning requires consideration of several aspects. First, a basic reinforcement learning framework must be established, including elements such as environment, intelligence, actions, states, and rewards. Standard reinforcement learning frameworks such as the Markov decision process (MDP) or partially observable Markov decision process (POMDP) can be used. Secondly, within the basic reinforcement learning framework, attention mechanisms are introduced to allow intelligence to autonomously attend to and process important information. Deep reinforcement learning models such as deep Q networks (DQN) and policy gradients (PG) can be used and added to them with attention mechanisms such as adaptive attention and multi-headed attention. Attention mechanisms can help the intelligence process information more effectively and thus improve their cognitive abilities. To further enhance the cognitive ability of intelligence, other techniques in deep learning, such as convolutional neural networks (CNN), recurrent neural networks (RNN), etc., can be used to process and extract different types of information.

We inserted the convolutional block attention module (CBAM) [22], an attention mechanism model for visual tasks in deep learning, on top of the originally proposed cognitive reinforcement learning framework. The CBAM module consists of two components: the channel attention module (CAM) and the spatial attention module (SAM). As shown in Figure 1. CAM determines the importance of each channel by learning the relationships between channels, and SAM uses the learned spatial weights to emphasize or suppress the

response of each spatial location. Combined, these two components can effectively enhance the model's ability to model different channels and spatial information.

(**a**) (**b**)

Figure 1. (**a**) Channel attention module. (**b**) Spatial attention module.

Specifically, CBAM first uses global average pooling to calculate importance weights for each channel and then applies the weights to the channel feature map to strengthen the response of channels with higher importance. Next, CBAM uses a similar approach to calculate importance weights for each spatial location and then applies these weights to the spatial feature map to either strengthen or suppress the response at different locations. The ultimate goal is to effectively aid the flow of information in the network.

Adding the convolutional block attention module (CBAM) to the cognitive reinforcement learning (CRL) framework can improve the performance and efficiency of an intelligent body during learning. CBAM can improve the performance of a model by introducing an attention mechanism into a deep neural network to increase the model's attention to the input data. First, the state space, action space, and reward function of the intelligence are defined. These definitions will enable the intelligence to perceive their environment and act within it. Secondly, a CBAM module is added to the state representation, which will extract important features from the state representation so that the intelligence can better understand its environment and make more accurate decisions. Next, the intelligence is trained using an experience replay mechanism. In experience replay, the intelligence learns from previous experiences and updates them so that it can perform better in future decisions. Then come the tuning parameters; during the training process, the hyper-parameters of the CBAM module need to be adjusted, such as the number of channels of the attention mechanism. After training is complete, the model is evaluated using a test set to assess the impact of the CBAM module on model performance. It is important to note that adding the CBAM module may increase the computational complexity and training time of the model, so there is a trade-off between performance and efficiency.

As shown in Figure 2, the attention-enhanced cognitive reinforcement learning framework consists of online and offline self-learning processes, indicated by the solid black and purple lines, respectively. At the same time, the inputs to the mathematical framework are data related to the dynamic environment and the dynamic task (denoted by d), denoted by e and x, respectively. The d, e, and x denote the data set, dynamic environment, and dynamic task, respectively, all of which are finite sets. Note that "dynamic" means that the environment and tasks are dynamically changing, which may or may not be the same as the existing environment and tasks. That dynamic environment and dynamic task data (denoted as e^* and x^*) are also stored in the data space of the memory module for future use, where * represents historical data rather than real-time data of the environment and tasks. The memory module also has a cognitive case space and an algorithmic hyper-parameter space. The cognitive case space consists of a learning outcome set denoted by Y and a cognitive space denoted by $[f(e,x),(a^*,\lambda^*)]$, where $f(e,x)$ denotes the dynamic environment and the characteristics of the task, and (a^*,λ^*) denotes the selected algorithm type and hyper-parameters. The algorithm and hyper-parameter space consists of the set of available algorithm types denoted by A and the set of hyper-parameters denoted by Λ. Table 1 illustrates the work of each step.

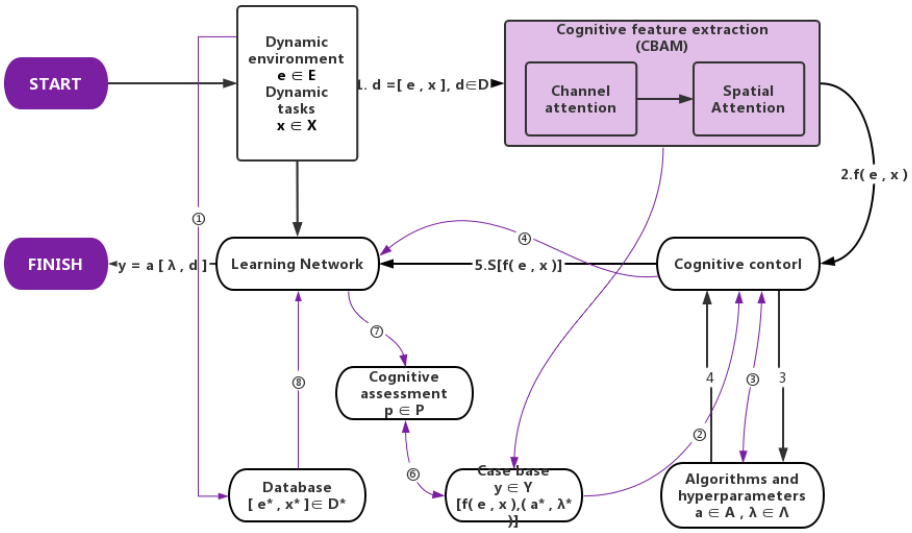


Figure 2. Cognitive reinforcement learning framework for attentional enhancement.

Table 1. Instructions for each step of the work.

Online process	1. Perceive the external environment and the task; 2. Extraction of the external environment and task features; 3. Selecting algorithms and hyperparameters; 4. Invoke; 5. Produce results.
Self-learning process	1. Store current learning results in the library; 2. Sampling cognitive cases; 3. Selecting algorithms and hyperparameters; 4. Call the algorithm and hyperparameters; 5. Get learning results; 6. Recall historical learning results for the same case; 7. Compare the better results and save them; 8. Update the case library and retrain.

The mathematical framework has four modules in addition to these two. First, we have a cognitive feature extraction module that extracts features of the dynamic environment and dynamic task and stores these features in a cognitive case space, denoted by $f(e, x)$ for the features of the dynamic environment and dynamic task. Next is a cognitive control module that establishes the matching relationship between the features of the dynamic environment and the task and selects the most suitable algorithm type and hyperparameters. These parameters can be updated during the offline self-learning process to alleviate the local optimum solution problem that may be encountered during reinforcement learning. During the offline self-learning process, we initialize multiple neural networks several times and select the result with the lowest error as a parameter. We start from different starting points and can select the optimal local optimum solution even if we get stuck in a local optimum. Next, we have a learning network module, which is used to perform algorithmic operations on the input data to derive learning results. Finally, we have a cognitive evaluation module used to evaluate the current learning results and feed the evaluation values back to the cognitive control module to adjust the type of algorithm and hyperparameters chosen.

Attention-enhanced cognitive reinforcement learning has the following advantages over traditional reinforcement learning: Firstly, attention-enhanced cognitive reinforcement learning can improve learning efficiency and reduce training time. This is because the method can find the optimal strategy faster by selecting more meaningful information when making decisions. Secondly, attention-enhanced cognitive reinforcement learning can help intelligence to better understand its environment and choose the optimal action. This is

because the method allows the intelligence to focus more on task-relevant information and thus make more accurate choices when making decisions. Attention-enhanced cognitive reinforcement learning can then make the intelligence more robust, i.e., more able to cope with environmental changes and noise. This is because the approach helps the intelligence to better distinguish between important information and noise when faced with complex environments. Finally, attention-enhanced cognitive reinforcement learning can make an intelligence's behavior easier to interpret. This is because the method allows intelligence to select task-relevant information, thus making its behavior more interpretable.

4. Attention-Enhancing Cognitive Reinforcement Learning Algorithms

The attention-enhanced cognitive reinforcement learning algorithm is an approach that combines the attention mechanism in deep learning with the optimization of the value function in reinforcement learning. It aims to solve complex decision problems by learning how to allocate attention. The basic principle of the algorithm is to introduce the attention mechanism into the estimation of the value function for reinforcement learning. Specifically, the algorithm uses a deep neural network to learn a value function that predicts the value of an action based on the state of the environment and the current allocation of attention. To enable attention to be adaptively allocated to the most useful features of the environment, the algorithm also uses an attention mechanism to adjust the parameters in the neural network. In this way, the algorithm can adaptively choose which state features to focus on according to the needs of the task at hand, thus making decision-making more accurate and efficient. Specifically, the training process of the algorithm can be divided into the following steps:

Step 1: Calculate the attention allocation through the attention mechanism based on the current state and the parameters in the neural network.

Step 2: Based on the attention allocation and the current state, the value of the current action is predicted by the neural network.

Step 3: Based on the environmental feedback and the predicted value, update the parameters of the neural network so that the predicted value is closer to the actual value.

Step 4: Repeat steps 1 to 3 until the algorithm converges or reaches a pre-determined number of training steps.

In summary, the attention-enhanced cognitive reinforcement learning algorithm improves the efficiency and accuracy of reinforcement learning by introducing an attention mechanism to adaptively select the most useful state features.

5. Experiment

We designed two different multi-intelligent cooperation tasks, including formation control and group containment, to validate the effectiveness of attention-enhanced cognitive reinforcement learning and also designed a containment task to verify the method's robustness [23].

In all tasks, the only way to obtain more information is through limited communication, as this is the only way to obtain information from other intelligence. The environmental map here has a side length of 3 meters, a detection distance of 0.8 meters, and a communication distance of 1 meter. The radius of the smart body is 0.1 m, and the radius of the obstacle is 0.2 m. The mass of the smart body is 1 kg, and the action space is discrete. Each smart body can control plus or minus velocity units in the X and Y directions. The boundary conditions are the four boundaries of the map. These simulation environments are built based on [24], where the intelligent body can move freely using a first-order system model.

In simulation experiments, we compared the performance of the CRL-CBAM algorithm with that of the MADDPG [24], R-MADDPG [24], and TRANSFER [25] algorithms. MADDPG requires information about the state of the intelligence during training to construct the critic network, while R-MADDPG is a recurrent version oriented towards a partially observable environment. The TRANSFER algorithm ignores the temporal relationships between the intelligence. In the comparison, we found that the CRL-CBAM

algorithm performed well, achieving stable and high rewards after training in the presence of initial instability.

5.1. Formation Control

(1) Task setup: CRL-CBAM was compared with the formation control task algorithm in two different scenarios, as shown in Figure 3. These scenarios include scenario (a) with five intelligence and scenario (b) with fifteen intelligence. In these scenarios, the goal of all the intelligence is to be evenly distributed around the center of the stratum and to be collision-free. Where the obstacles are fixed, the positions of the intelligences are randomly generated, and the geometric center of the intelligences' formation is marked as the center of the formation.

(**a**) (**b**)

Figure 3. Illustration of formation control scenarios. (**a**) Formation control-5 intelligence. (**b**) Formation control-15 intelligence.

(2) Simulation results. For these five intelligence, they performed very similarly regardless of which method was used. This is because the relationships between these intelligence are relatively simple, and all methods are able to learn satisfactory strategies regardless of whether a graph convolution layer or attention mechanism is used. Specifically, the CRL-CBAM method can achieve more rewards than all baseline methods at the end of the training, but it converges more slowly. This suggests that CRL-CBAM is relatively difficult to train because it uses a graph convolution layer and an attention mechanism, which requires more data for training.

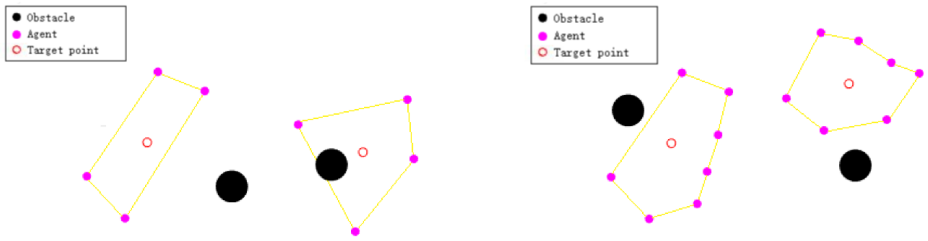
As the number of intelligence increases, CRL-CBAM has a better ability to handle complex interactions and dynamic graph structures. Compared to other methods, CRL-CBAM converges faster and has a more stable performance. In particular, the CRL-CBAM method converged to a steady state after 6200 update sets, while the other methods converged to a steady state after 7500 update sets. The results show that CRL-CBAM can handle complex interactions between a large number of intelligence. In contrast, without the help of the graph convolution layer, the attention mechanism and other easily trained methods do not perform well in complex environments with an increasing number of intelligence.

Our method performs similarly to other methods in some scenarios but performs better in others. In particular, CRL-CBAM can achieve more rewards and more efficient performance than other methods in scenarios with 10 or more intelligence, suggesting that CRL-CBAM can better handle scenarios with a high number of intelligence.

5.2. Group Containment

(1) The purpose of this task is to evaluate the performance of the scenario. In this task, the environment contains n intelligent bodies and m landmarks, where the relationship between n and m can be expressed as (n/m) = k, where k is a set of positive integers. Based on these constraints, two scenarios were designed: one containing 8 intelligence and 2 obstacles, and the other containing 14 intelligence and 2 obstacles (as shown in Figure 4). The CRL-CBAM algorithm was used to compare with other algorithms in these

two scenarios. In these scenarios, all the intelligence had to be divided into two groups and evenly distributed around the two landmarks to avoid collisions.



(**a**) (**b**)

Figure 4. Group containment scenario. (**a**) Group containment-8 intelligence. (**b**) Group containment-14 intelligence.

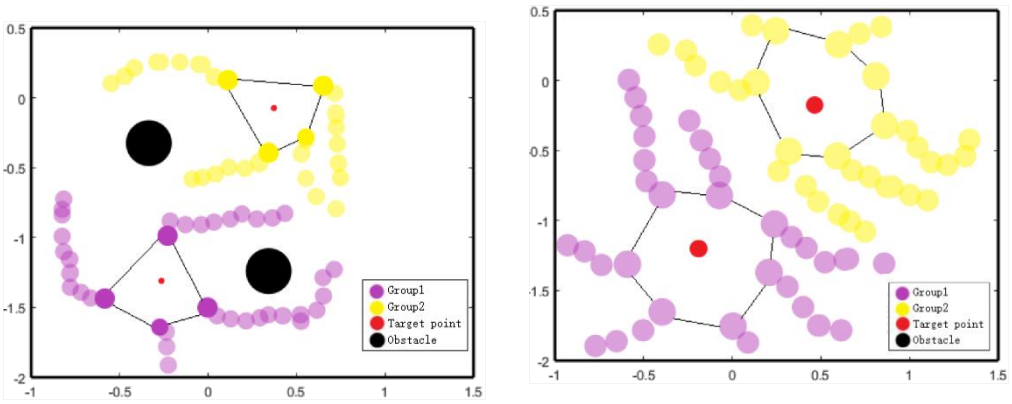
(2) Simulation results. As shown in Table 2, all methods except MADDPG and R-MADDPG have completed the task.

Table 2. Evauluation results of group containment.

Method	N = 8			N = 14		
	Steps	**Success Rate**	**Rewards**	**Steps**	**Success Rate**	**Rewards**
CRL-CBAM	13.8	100	−0.49	13.1	100	−0.62
MADDPG	80.21	0	−1.58	80	0	−4.62
R-MADDPG	31.07	70	−0.94	80	0	−2.9
TRANSFER	17.7	95	−0.67	14.58	87	−0.84

In this case, CRL-CBAM showed superior performance to the other methods, being able to obtain the 15 best measurements. In contrast, MADDPG and R-MADDPG yielded the smallest rewards, especially in complex environmental settings where the cyclic strategy of MADDPG and R-MADDPG was unable to handle complex interactions between intelligences. As we have mentioned, CRL-CBAM is able to handle complex interactions and dynamic spatial structures efficiently. When the number of intelligences was increased to 14, the CRL-CBAM algorithm outperformed the other methods significantly and also gained greater rewards. This suggests that communication is crucial in the cooperation between intelligences. Furthermore, the higher the complexity of CRL-CBAM in representing the relationships of the intelligences, the better the execution.

As shown in Figure 5, the goal of the agents is to find the most suitable position while taking into account mutual avoidance. In Figure 5b, not all agents are trying to find the closest position around the landmark. The location where an agent is located does not necessarily maximize its individual reward, but it does maximize the joint reward. The results suggest that agents have learned a complex policy whereby agents sacrifice individual rewards to maximize group rewards.

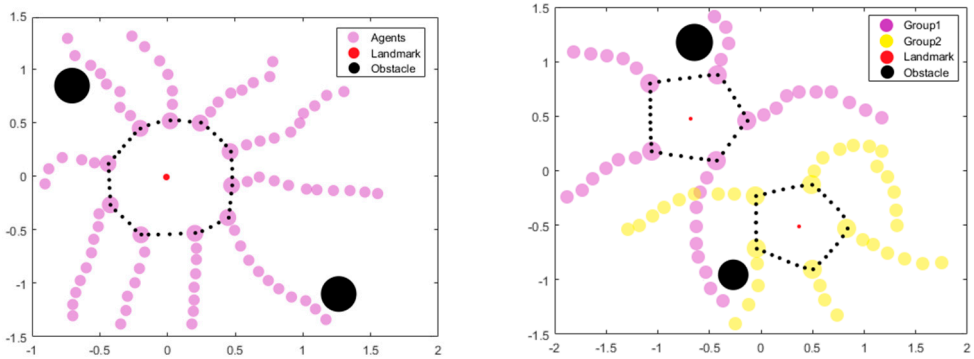


(**a**) (**b**)

Figure 5. Trajectory of group containment. (**a**) Group containment-8 intelligence. (**b**) Group containment-14 intelligence.

5.3. Robustness of CRL-CBAM

To test the generalization and robustness of CRL-CBAM, we evaluated it for two different scenarios from the training scenario. Specifically, we scaled up the two evaluation scenarios to twice the size of the original training scenario. Even though the evaluation scenarios differed from the simulation scenarios, all tasks were successfully completed. Thus, these results show that CRL-CBAM is suitable for the new environment and has reliable generalization performance (see Figure 6).



(**a**) (**b**)

Figure 6. Trajectories of scenarios. (**a**) Formation control-10 agents. (**b**) Group containment-10 agents.

6. Conclusions

This paper presents a cognitive reinforcement learning method (CRL-CBAM) based on attentional mechanisms. The method increases representational capacity by using attentional mechanisms to focus on important features and suppress unnecessary ones. Two modules are used to emphasize meaningful features on two main dimensions: the channel and the spatial axis. Thus, our modules effectively aid the flow of information in the network by learning which information to emphasize or suppress. Experimental results show that the method can effectively improve the performance of intelligence in multiple reinforcement learning tasks and has better learning efficiency and performance stability compared to traditional reinforcement learning methods. In addition, the method can help us to better understand the decision-making process of intelligence and improve the interpretability of their decisions.

Author Contributions: Conceptualization, D.L. and J.Z.; Methodology, X.C.; Writing—original draft, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.I.; Abbeel, P. Trust Region Policy Optimization. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1889–1897.
2. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Hassabis, D. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
3. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
4. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
5. Singh, S.P.; Kearns, M.J.; Mansour, Y. Nash convergence of gradient dynamics in general-sum games. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial, Stanford, CA, USA, 30 June–3 July 2000.
6. Hu, J.; Wellman, M.P. *Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1999.
7. Lauer, M.; Riedmiller, M. *An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2000.
8. Littman, M.L. Value-function reinforcement learning in Markov games. *Cogn. Syst. Res.* **2001**, *2*, 55–66. [CrossRef]
9. Anastasios, G.; Sotirios, S.; Nikolaos, K.; Panagiotis, K.; Panagiotis, T. Deep Reinforcement Learning for Energy-Efficient Multi-Channel Transmissions in 5G Cognitive HetNets: Centralized, Decentralized and Transfer Learning Based Solutions. *IEEE ACCESS* **2021**, *9*, 129358–129374.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
11. Coles, M. Formats: The Neural Basis of Human Error Processing: Reinforcement Learning, Dopamine, and the Error-Related Negativity. *Psychol. Rev.* **2002**, *109*, 679.
12. Shalev-Shwartz, S.; Shammah, S.; Shashua, A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv* **2016**, arXiv:1610.03295.
13. Wang, X.; Sandholm, T. Reinforcement Learning to Play an Optimal Nash Equilibrium in Team Markov Games. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–14 December 2002.
14. Tesauro, G. Extending Q-Learning to General Adaptive Multi-Agent Systems. In Proceedings of the Advances in Neural Information Processing Systems 16, Neural Information Processing Systems, NIPS 2003, Vancouver, BC, Canada, 8–13 December 2003.
15. Bowling, M.H. Convergence and No-Regret in Multiagent Learning. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2004.
16. Liu, D.K.; Wang, D.; Dissanayake, G. A Force Field Method Based Multi-Robot Collaboration. In Proceedings of the Robotics, Automation and Mechatronics, Bangkok, Thailand, 1–3 June 2006.
17. Etesami, S.R. Optimal versus Nash Equilibrium Computation for Networked Resource Allocation. *arXiv* **2014**, arXiv:1404.3442.
18. Li, H.; Kumar, N.; Chen, R.; Georgiou, P. Deep Reinforcement Learning. In Proceedings of the ICASSP 2018—2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018.
19. Hüttenrauch, M.; Šošić, A.; Neumann, G. Learning Complex Swarm Behaviors by Exploiting Local Communication Protocols with Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1709.07224.
20. Mohsin, S.; Zaka, F. A New Approach to Modeling Cognitive Information Learning Process using Neural Networks. In Proceedings of the International Conference on Artificial Intelligence (ICAI), Las Vegas, NV, USA, 16–19 July 2012.
21. Wu, Q.; Ruan, T.; Zhou, F.; Huang, Y.; Xu, F.; Zhao, S.; Liu, Y.; Huang, X. A Unified Cognitive Learning Framework for Adapting to Dynamic Environment and Tasks. *IEEE Wirel. Commun.* **2021**, *18*, 208–216. [CrossRef]
22. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. *CBAM: Convolutional Block Attention Module*; Springer: Cham, Switzerland, 2018.
23. Wang, H.; Pu, Z.; Liu, Z.; Yi, J.; Qiu, T. A Soft Graph Attention Reinforcement Learning for Multi-Agent Cooperation. In Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, 20–21 August 2020.

24. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
25. Agarwal, A.; Kumar, S.; Sycara, K.P.; Lewis, M.J. Learning Transferable Cooperative Behavior in Multi-Agent Teams. *arXiv* **2020**, arXiv:1906.01202.