Explainable and Adaptable Augmentation in Knowledge Attention Network for Multi-Agent Deep Reinforcement Learning Systems

Joshua Ho[1,2,3], Chien-Min Wang[2]
[1]Social Networks and Human-Centered Computing Program, Taiwan International Graduate Program,
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan 115
[3]Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu Taiwan, 30013
{jho, cmwang}@iis.sinica.edu.tw

***Abstract*—The scale of modern Artificial Intelligence systems has been growing and entering more research territories by incorporating Deep Learning (DL) and Deep Reinforcement Learning (DRL) methods. More specifically, multi-agent DRL methods have been widely applied to address the problems of high-dimensional computation, which interpret the conditions that real-world systems mainly encounter and the issues that require resolving. However, the current approaches of DL and DRL are often challenged for their untransparent and time-consuming modeling processes in their attempt to achieve a practical and applicable inference based on human-level perspective and acceptance. This paper presents an explainable and adaptable augmented knowledge attention network for multi-agent DRL systems, which uses game theory simulation to tackle the problem of non-stationarity at the beginning while improving the learning exploration built upon the strategic ontology to achieve the learning convergence more efficiently for autonomous agents. We anticipate that our approach will facilitate future research studies and potential research inspections of emerging multi-agent DRL systems for increasingly complex and autonomous environments.**

Keywords-component; Artificial Intelligence; Knowledge Engineering; Transfer Learning; Multi-agent Deep Reinforcement Learning; Explainable and Adaptable Knowledge Attention Network

I. INTRODUCTION

The method of Reinforcement learning (RL), a branch of Machine Learning (ML), was proposed last century and was inspired by the behavioral sciences and psychology. RL is the closest form of human and pet learning path because it can learn from its own experience by exploring the unknown environment with or without prior knowledge. RL trains the agent to learn how to react in the environment via a trial and error method, while offering rewards according to the agent's taken action, given and future state, and the feedback from the environment. The RL methods are essentially applied to many contemporary user-interactive applications specified by Markov Decision Process (MDP) for decision-making problems optimally specified, such as robot learning and controlling systems, human-computer interaction, and intelligent assistant systems, which cannot be handled well by notably supervised and unsupervised learning mechanisms.

In the proposed experiments, we focus on *Q-learning,* which is a popular RL method, with its goal to maximize the discounted reward. Moreover, we use deep *Q-learning* (DQL) which adopts the RL method and Deep Neural Network (DNN) architecture for the prospective Deep Q-network (DQN). With the statistical estimation of DQN, deep *Q-learning* can learn a relatively covenanted and low-dimensional representation from originally high-dimensional raw data in a functional approximation, to adequately deal with the optimal convergence. Also, the proposed multi-agent DRL (MADRL) aims at game theory simulations for the corporative agents in the experiments, where transfer learning and prior knowledge sharing methods are applied to investigate the learning efficiency through ontology mappings for multiple agents' communication in the dynamic environment.

In this paper, we present an explainable and augmented knowledge attention network, according to *Prioritized Experience Replay* [13] for random batch sampling, which helps to identify the error samples and update weighted samples in the learning process. Our approach learns to achieve the improvements of its learning process and memory explorations through ontological hierarchy mappings, which builds up the inquiry interfaces and makes the learning more eligible and adaptable during the RL modeling process given valuable and effective opinions. Also, by applying the prior and sharing knowledge transferred among agents, the explainable and advanced knowledge bases are induced in the MADRL simulations, while the autonomous agents are capable of adapting their corporative behaviors and cope with the corporative or competing agents for a new environment and/or a possible target task.

The remainder of this paper is organized as follows: Section II reviews the basics and related literature about MADRL. Section III and IV present the system architecture and design with *Knowledge Engineering* (KE) concepts and ontological mapping interfaces, considered methodologies in the system design; Section V respectively report the experimental results and discussion of the experiments. Section VI concludes our work and discusses its use, method, contribution, and the related future work.

II. Literature Review

A. RL and MADRL

RL methods based on multi-agent (MA) simulations have become applied to solve challenging MDP problems in recent years, but some works were applied to *Policy Architecture Network* [1] and *Curriculum Learning* [19] last century. However, RL may require higher quality environments with well-described observations, explicit action spaces, rewards and states to manipulate the engineering process. MADRL usually requires more customized observations among agents that interact with each other in a mixed cooperative-competitive condition [6]. This is often regarded as more realistic and closer to real-world problems [2]; here the number of agents is also estimated to be between only a single agent and an effective number of average agents for optimization purposes [3]. More recent work [5] involves human feedback [16] in the *human-in-the-loop* RL method to augment binary responses with state salient information to boost performance.

B. Deep Q-Learning

Learning an RL model with high-dimensional sensory inputs is often a challenging and time-consuming process. By applying DNN to overcome the control of policy and shorten the delay between action and resulting rewards, the network of deep *Q-learning* [20] based on the design with a single agent is trained as an off-policy DRL method according to the Bellman equation (1), where the *discount factor* γ, manages the future rewards during its learning to reach the convergence. The discounted value is mathematically necessary because the environment can be fully observable, partially observable or completely uncertain.

$$Q(s_t, a_t) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t, a_t], \quad (1)$$

Since real-world problems consist of large, complex and continuous state and action spaces, ordinary *Q-learning* is unable to efficiently resolve these problems due to the large memory required to store rewards (*Q-values*) in its *Q-table*; this leads researchers to consider taking advantage of popular Deep Neural Networks for simplifying parametric estimations and the reduction *of Q-values* in the *Q-table* to accelerate the learning with experience replay [21]; at the same time, *Q-learning* directly optimizes the action-value function with the updating rule (2):

$$Q_t(s_t, a_t) \leftarrow (1-\alpha) Q_t(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a} Q(s_{t+1}, a) \right], \quad (2)$$

where s_t is the current state, a_t is the action taken at the current instant, r_t is the reward which is received after executing a_t, and $Q_t(s_t, a_t)$ is the evaluation of the action a_t in the state s_t. The parameters of the *learning rate* α and *discount factor* γ are required to be adjusted in the system.

C. Prior Knowledge Sharing and Transfer Learning

In the earlier works of the 1980s, Knowledge Engineering (KE) was designed to transfer human knowledge into an implemented Knowledge Base System (KBS). The goal is to model a KBS to form an expert-system, which is usually based on knowledge from interviewed experts in order to model how humans solve specific tasks within disciplines and according to rules. The heuristic of problem-solving methods and representations of modeling an expert-system became the foundation of KBSs, which address KE as a modeling process achieving formalizations and implementations [12]. A well-known KE framework, *Protégé* [22], was designed to perform knowledge acquisitions as a durable and extensible platform for KBS research and development. *Protégé* aims to transform instances of domain concepts into a reusable knowledge-acquisition framework through ontology mappings. In this framework, background knowledge is absorbed and practiced in the application of RL for specifying agent behaviors within a language, in order to speed up the training process and increase the asymptotic performance [11]. In another study, similar tasks were executed by a physical robot agent to navigate in a maze-like environment [10]: the proposed DRL algorithm reused transferred knowledge and solved the problems more quickly.

D. Game Theory Simulation

Knowledge transfer has become one of the core concepts in modern game theory simulations of RL. A *General Game Playing* system implements techniques of feature transfer with a value-function to extract general game features among vastly disparate state spaces, thus expediting the agent's learning in many games [14]. Recent game theory experiments demonstrate not only cooperative-competitive simulations, but also multi-agent strategical learning to solve a hierarchical learning scheme and sequential social dilemmas [24] by using DRL on DQL. The game of *hide-and-seek* has been addressed by [4] recently in an emergent tool for MA interactions, which shows that agents learn to develop explicit strategies, counterstrategies as well as more '*exceeding expectation*' complex strategies. Recent work has focused on games like *predator-prey* in [24], which shows that policies of other agents and *Policy Ensembles* can be inferred through observations. The method of *Q-learning* and the *Actor-Critic* [23] policy gradient approach explicitly requires the modeling procedure by the decision-making process of coordinate agents, although the performance of the proposed algorithms may need further improvement for a MADRL environment.

III. System Architecture

The proposed system aims at providing an explainable and adaptable interface to learn a game simulation of MADRL, while considering the interactions with not only other agents, but with humans [15] for building a feasible KBS to improve the poor sample efficiency of DRL. Since DRL has been criticized for the limitation of designed algorithms and the slowness of convergence in real-world situations, the techniques of sharing and transferring knowledge [17] can be applied to achieve an increased number of agents to adapt to relevant behaviors [7, 8, 9].

The agents use the knowledge shared among themselves in the episodes either with the *Prioritized Experience Replay* (PER) or *Uniform Experience Replay* (UER) method: the former proposes the sample batch priorities according to errors in the memory, while the later one formulates the sample batch according to randomly appended samples in the memory. In our DQL, PER is pursued and prioritizes the weighted sample from the memory tree in each update and normalizes the feature dataset to train the DRL model gradually. The non-stationary sample and the latency between *action-rewards* in the DRL are our major challenges to achieve a more efficient learning

convergence. The goal of the proposed system, shown in Fig 1, is to deal with the sample inefficiency to get close to human-level compliance, since real-world problems are much more complex, crucial and challenging.

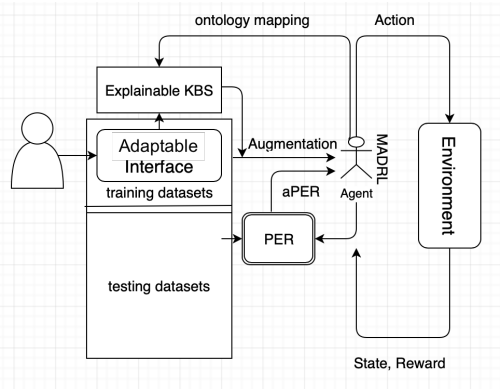


Fig 1. The proposed system shows an explainable KBS and an adaptable interface for augmentation in MADRL systems.

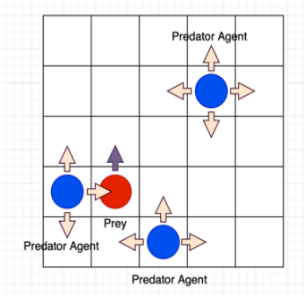


Fig 2. The prey-predator game sets an arbitrary number of predator (blue) and prey (red) agents in a N x N grid experimental world.

IV. System Design

A. Pursuit Game

Previous works have adopted multiple agents in various simulations such as *StarCraft*, *IC3Net,* and *hide-and-seek* games [18]. The pursuit game, *prey-predator* (Fig 2), sets an arbitrary number of predator and prey agents in an *N* x *N* grid (i.e. *N* x *N* cells) as the experimental world. We evaluated the agent performance when all prey was captured by the predator agents, by measuring the time or step counted toward the capture and the success rate within a limited time spent in the episodes. The capture is the final stage occurring when the prey has no empty neighboring cells that are occupied by predators, or when the prey and a predator are located in the same cell and adjacent cells are not empty due to predators occupying the neighboring cells. We randomly set the initial locations for the agents deployed in the grid world to choose an action in the *action-space* for each step; at this point, prey can be optionally configured as either the default '*fixed state'* (F) or the '*random escape'* (R) targeted prey mode. A cell cannot be simultaneously occupied by two or more agents, unless a prey and a single predator may be occasionally located in the same cell in the episode. The boundaries cannot be crossed by agents, and predators can learn to cooperate in the capture while prey can only stand still or escape arbitrarily. A single episode has a limited number of steps, and when it reaches the capture before the limit step (100 steps), the episode will end. If no capture occurs, the episode will end and restart a new one at the limit step of 100 till the end of the game.

B. Multi-Agent DRL

The proposed MADRL consists of a set of cooperative n agents denoted by $A = \{0, 1, 2 \ldots . . n-1\}$, while at time t, each agent $a \in A$ observes the current state $S_t \in S$ in the environment, with the taken action $u_t^a \in U$ being based on the stochastic policy π'. The resulted reward is received as $R_t = \{S_t, u_t^a\}$ at the same time the environment moves to $S_{t+1} \leftarrow S_t$. The objective is to search a set of policy π for all agents in order to maximize the total reward in a Q-learning probability function $R_T = \sum_{j=T}^{\infty} \gamma^{j-T} R_j$, where R_j is the reward received at time j and the *discount factor* γ is required to describe the observable environment.

The policy set π can be classified into two types of method: 1. the joint policy among all predators at every π'^j, and 2. an independent policy π'^i for each agent. The method of π'^j considering the joint actions $\prod_{a \in A} U^a$ of all agents encounters the space complexity that grows exponentially as the number of agents increases. The proposed alternative approach, the '*independent policy'* π'^i, is based on each agent that can eventually reduce the space complexity. However, the '*independent policy'* also suffers from the non-stationarity of the RL environment, even though the experience memory replays; at the same time, PER is applied for the memory replay and DQN in the deep neural network helps to stabilize the sample inefficiency. The PER of obsolete memory cannot notably catch the dynamics of the environment, which the agents should learn in order to take accurate action to achieve rewards and a more effective convergence. Thus, the proposed augmented PER (aPER) is explored and verified in the MADRL simulations to address this issue.

C. Knowledge Engineering

To formulate and explain the *action-space* and *state-space* in our experiments, we designed two ontologies: the action ontology in Fig 3 and state ontology in Fig 4. In the latter, the predator agent can use the action ontology to choose the next action such as either '*up'*, '*down'*, '*left'*, '*right'* or '*stay'* in the *action-space* and learn other predators' action by reusing the knowledge of other predator agents. Meanwhile, the predator agent can observe its status as well as that of other agents, which are referred to as the state ontology to avoid a conflict of interest when occupying the same cell as other predators.

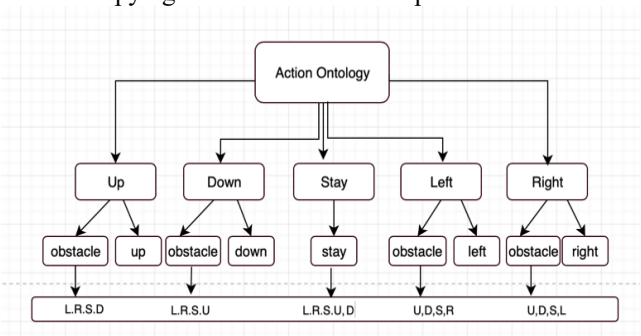


Fig 3. Action ontology provides agents with directions based on the condition, where actions are 'up' (U), 'down' (D), 'left' (L), 'right (R), and 'stay' (S).

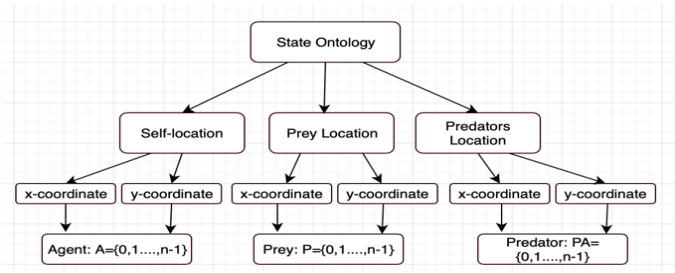


Fig 4. State ontology shows the corresponding x-coordinate and y-coordinate for self-location, prey location, and other predators' locations.

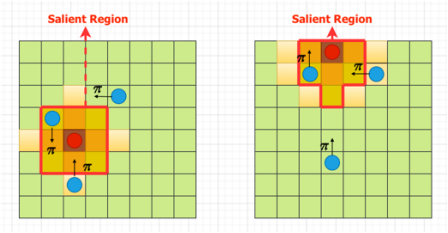


Fig 5. *Adaptable Interface* provides human guidance for training the predator agents (blue) to target the salient region and cells of prey (red).

The prey mode can be set to either F or R as the target, whereas a single predator agent can use the state ontology to decide whether to occupy the designated cell that the prey is simultaneously located in to prevent conflict with other predators. Thus, for a feasible KBS, we employed two hierarchy mappings over two ontologies that are built upon knowledge sharing and knowledge transfer mechanisms, and the necessary search function and related inquiring interfaces are estimated and preprogrammed for all predators in the RL environment.

D. Adaptable Interface

The proposed *Adaptable Interface* aims at integrating the human inputs in the MADRL training process, to improve the sample inefficiency at the training stage, which will result in a performance boost in the following testing episodes. One of the ways to do this may be to offer augmented information to the system through natural language, but this presupposes that the system is fully built and can promptly on time understand the meaning of natural languages in a dynamic environment. In another more ideal way would involve conveying the real-time and straightforward information to allow humans to point at salient regions and cells of the environment state. This augmented input referred to as *human-in-the-loop* attention for predator agents, guides predators' actions based on the salient information of the *state-space* shown as Fig 5, in addition to the original *action-space*. The guiding signal can be offered as frequently as in every step to interact with dynamic prey with the '*random escape'* mode in particular. The proposed *Adaptable Interface* helps to examine the three metrics of environment sample efficiency, human augmented sample efficiency, and agent observations for targeting prey.

V. Experimental Results and Discussion

A. Experimental Settings

The experiments are based on the *MADRL* and *Adaptable MADRL* methods shown in Table I, where the *Configuration* settings are arranged in the current deployment. We executed 10 epochs and each epoch consisted of 50 episodes in the evaluation. In each epoch, the proposed system defined the *Training* (1st to 10th episode) and *Testing* (11th to 50th episode) datasets for both methods. A maximum number of steps were counted and limited to 100 to restart in each episode if no capture occurs; otherwise, the next episode restarts at any time of capture within the limit. Here, we adopted aPER to remember the sampling batch and weigh the important replay experience, which is ranked in the priority list by filtering the error samples. *Augmentation* represents the '*human-in-the-loop*' training courses for each episode in addition to the statistic *action-space* expected by the proposed *Adaptable MADRL* method. The experiments were conducted according to the settings illustrated in the list of hyper-parameters and related values (Table II), and the number of exploration steps was set to 100 steps.

TABLE I. Datasets

setting / method	Configuration, Training vs. Testing		
	Configuration	***Training***	***Testing***
MADRL (Baseline)	▪ Prioritized Experience Replay (PER) ▪ Adam optimizer ▪ ReLu softmax function ▪ 3 predator agents ▪ 1 prey agent (F/R) ▪ Initially random state agents ▪ N x N grid ▪ Total 10 epochs ▪ 50 episodes / epoch ▪ 100 steps limited / episode ▪ aPER (*human-in-the-loop)**	1st ~ 10th episode / epoch	11th ~ 50th episode / epoch
Adaptable MADRL		1st ~ 10th episode* / epoch	11th ~ 50th episode / epoch

TABLE II. Hyperparameters and Values

Parameter	Value
RMSPro learning rate	0.00005
Adam learning rate	0.0000625
Initial ε for exploration	1.00
Final ε for exploration	0.01
Number of exploration steps	100
Steps between target network updates	1000
Replay memory size	1000000
Minibatch size	64
Discount factor γ	0.95
Prioritization exponent ω	0.5
Prioritization importance sampling β	0.4 →1.0

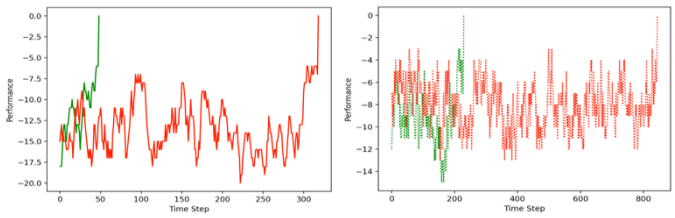
TABLE III. Experiment Results

	ASR	ASST	ASE
MADRL (Baseline)	17.9%	**-321.77**	-1043.84
Adaptable MADRL	**30.35%**	-353.732	**-840.10**

B. Experimental Results and Discussions

Here we report the results (Table III) of the three metrics of ASR, ASST, and ASE. The ASR indicates the *Average Success Rate* of our method, *Adaptable MADRL*, which outperforms the baseline to achieve a capture within a total of 400 episodes of

testing. Though the *Average Score for Success Test* (ASST) of *Adaptable MADRL* is slightly lower than the baseline's average success score, our method is still competitive in providing knowledgeable advice, according to the records of success episode due to the greater success observed in aPER, and the sample learning deficiency during the initial states. This is especially the case in memorable conditions both in training and the testing episodes, such as when the salient region is related to a corner of cells and boundary cells; here, our method learns much more quickly than the arbitrary batch samples in the observed test episodes. Since *Adaptable MADRL* improves the success rate within a limited number of steps in each episode, it eventually boosts performance for estimating the *Average Score for Episode* (ASE) in our experiments. Moreover, we adopted the above *Configuration* settings for similar training episodes with two methods, which both have fixed initial and random initial locations for all agents, and only all prey will stay at the origin point after their initialization. We evaluated the results of the two methods in the testing of the 11th episode with a non-limited number of steps to observe the convergent performance. The outcomes are shown in Fig 6 (a) and (b), which demonstrate our method performs around 8.33 (350/42) times and 4.03 (887/220) times to explore, which, when compared to the baselines in the experiments, are more efficient with the solutions being reached more rapidly.



(a) (b)

Fig 6. In (a), our method (green) shows 8.33 times faster with 'fixed initial location' for both methods, and in (b), which shows 4.03 times faster with 'random initial location' compared to the baseline (red) for both methods.

VI. Conclusiton and Future Work

In this work, we present a novel method to integrate an explainable KE and adaptable human interfaces for the MADRL learning environment. The experiments were conducted to simulate a game with enhanced inputs in the Human-in-the-Loop paradigm, to improve the rewards and accuracy. The results show that the convergence in MADRL was achieved, and feasible non-stationarity was avoided in the episodes for training with human guidance. The proposed method outperforms the baseline in environment sample efficiency and leverages the experience memory replay for the knowledge attention network. We also verified that the human input can be important in the early sample stabilizing process, especially in some special conditions for the agent to learn more efficiently, which can be advantageous for the further integration of human-centered computing and AI. Our future work will focus on the techniques and instructing courses given to DRL by human augmented, logically implicated, and incremental learning, to decide whether the observation quality, vectors, sequence of *state-action*, constraints and budgets are essential for various types of learning algorithms.

References

[1] Long, Qian, et al. "Evolutionary Population Curriculum for Scaling Multi-Agent Reinforcement Learning." *arXiv preprint arXiv:2003.10423* (2020).

[2] Wang, Qing, et al. "Arena: a toolkit for Multi-Agent Reinforcement Learning." *arXiv preprint arXiv:1907.09467* (2019).

[3] Yang, Yaodong, et al. "Mean field multi-agent reinforcement learning." *arXiv preprint arXiv:1802.05438* (2018).

[4] Multi-Agent Hide and Seek https://openai.com/blog/emergent-tool-use/. (2020).

[5] Guan, Lin, Mudit Verma, and Subbarao Kambhampati. "Explanation Augmented Feedback in Human-in-the-Loop Reinforcement Learning." *arXiv preprint arXiv:2006.14804*(2020).

[6] Tampuu, Ardi, et al. "Multiagent cooperation and competition with deep reinforcement learning." *PloS one* 12.4 (2017): e0172395.

[7] de Witt, Christian Schroeder, et al. "Multi-agent common knowledge reinforcement learning." *Advances in Neural Information Processing Systems*. (2019).

[8] Hu, Chunyang. "A Confrontation Decision-Making Method with Deep Reinforcement Learning and Knowledge Transfer for Multi-Agent System." *Symmetry* 12.4 (2020): 631.

[9] Moreno, David L., et al. "Using prior knowledge to improve reinforcement learning in mobile robotics." *Proc. Towards Autonomous Robotics Systems. Univ. of Essex, UK* (2004).

[10] Zhang, Jingwei, et al. "Deep reinforcement learning with successor features for navigation across similar environments." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[11] Shapiro, Daniel, Pat Langley, and Ross Shachter. "Using background knowledge to speed reinforcement learning in physical agents." *Proceedings of the fifth international conference on Autonomous agents*. 2001.

[12] Studer, Rudi, V. Richard Benjamins, and Dieter Fensel. "Knowledge engineering: principles and methods." *Data & knowledge engineering* 25.1-2 (1998): 161-197.

[13] Schaul, Tom, et al. "Prioritized experience replay." *arXiv preprint arXiv:1511.05952* (2015).

[14] Banerjee, Bikramjit, and Peter Stone. "General Game Learning Using Knowledge Transfer." *IJCAI*. 2007.

[15] Ilhan, Ercüment, Jeremy Gow, and Diego Perez-Liebana. "Teaching on a Budget in Multi-Agent Deep Reinforcement Learning." *2019 IEEE Conference on Games (CoG)*. IEEE, 2019.

[16] Zheng, Guanjie, et al. "DRN: A deep reinforcement learning framework for news recommendation." *Proceedings of the 2018 World Wide Web Conference*. 2018.

[17] Ammanabrolu, Prithviraj, and Mark O. Riedl. "Transfer in deep reinforcement learning using knowledge graphs." *arXiv preprint arXiv:1908.06556* (2019).

[18] Elman, Jeffrey L. "Learning and development in neural networks: The importance of starting small." *Cognition* 48.1 (1993): 71-99.

[19] Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:1312.5602* (2013).

[20] Foerster, Jakob, et al. "Learning to communicate with deep multi-agent reinforcement learning." *Advances in neural information processing systems*. 2016.

[21] Gu, Shixiang, et al. "Continuous deep q-learning with model-based acceleration." *International Conference on Machine Learning*. 2016.

[22] Gennari, John H., et al. "The evolution of Protégé: an environment for knowledge-based systems development." *International Journal of Human-computer studies* 58.1 (2003): 89-123.

[23] Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in neural information processing systems*. 2017.

[24] Leibo, Joel Z., et al. "Multi-agent reinforcement learning in sequential social dilemmas." *arXiv preprint arXiv:1702.03037*(2017).