Recurrent Deprivation Curiosity for the Knowledge Graph Compression in Reinforcement Learning Navigation

Li Shude[1,†], Liu Wen[1,*,†], Deng Zhongliang[1,†], Song Xudong[1,†] and Zhang Haoyu[1,†]

[1]*Beijing University of Post and Telecommunication (BUPT), 10 Xitucheng Road, Haidian district, Beijing, China*

Abstract

In the end-to-end navigation exploration of unknown regions based on reinforcement learning, the curiosity mechanism is usually used as the setting of the intrinsic reward mechanism of reinforcement learning. During the exploration process, agents may be excessively curious about noise or dynamic scenes and be repeatedly attracted to the dynamic region, resulting in the problem of navigation loop, which affects the efficiency and performance of navigation. To solve this problem, this paper proposes the graph compression cyclic deprivation curiosity algorithm (KRD-Curiosity) as an intrinsic reward mechanism for the exploration and navigation of reinforcement learning. Firstly, the algorithm relies on the distribution rule of deprivation degree and uses the upper and lower local memory sequences to predict the habitual behavior of agents, to avoid the wrong judgment of noise location. Secondly, the time fading mechanism is used to punish the agent to alleviate the problem of the agent indulging in dynamic scenes. Finally, knowledge graph compression is used to reduce the entropy of network storage information and improve the speed of information processing during navigation. In this paper, a complex multi-room environment was created for the agent navigation experiment, and the ablation experiment was set to compare and evaluate the navigation training efficiency of the proposed method, and the correlation between the sense of deprivation and the exploration logic was found. At the same time, the influence of different compression accuracy on navigation efficiency was proved. The experiment showed that our method improved the navigation performance compared with traditional methods.

Keywords

Reinforcement Learning, Deprivation Curiosity, Knowledge Graph Compression

1. Introduction

Mobile robots are widely used in industries, hotels, security, hospitals and other industries, and have a wide range of application scenarios in the field of positioning and navigation assistance [1, 2].Accurate autonomous navigation is a prerequisite for mobile agents to apply other auxiliary functions, and exploratory navigation in unknown regions is still a challenge for the agents. Because there is no prior map guidance, compared with the traditional global

Proceedings of the Work-in-Progress Papers at the 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN-WiP 2023), September 25 - 28, 2023, Nuremberg, Germany

*Corresponding author.

†These authors contributed equally.

lishude123654@126.com (L. Shude); liuwen@bupt.edu.cn (L. Wen); dengzhl@bupt.edu.cn (D. Zhongliang); 13789695330@163.com (S. Xudong); zhy15935467108@163.com (Z. Haoyu)

0009-0007-0319-0910 (L. Shude)

navigation, the exploration and memory ability of the agent are required to be higher. Reinforcement learning navigation [3, 4, 5] has shown excellent exploration success rate and navigation performance in recent studies. Agents learn strategies to maximize the total number of future rewards by interacting with environmental states, so that agents can achieve the most effective action behaviors. However, setting different reward mechanisms in the same navigation task will affect the efficiency of navigation and the accuracy of the results, so it becomes very important to design a reward function that conforms to the navigation task. Moreover, in the objective reality scenario, the agent itself cannot obtain effective immediate rewards in the process, and can only judge the effectiveness of the behavior in the previous process through the final result. The exploration behavior in the navigation process cannot get timely reward feedback, and such sparse rewards further improve the difficulty of constructing the reward function of the reinforcement learning model. To sum up, the reward function plays a key role in the autonomous navigation of mobile agents in reinforcement learning. Therefore, how to design an appropriate reward function becomes a key issue in the autonomous exploration task of the agents.

In recent years, in order to overcome the problem of sparse rewards, two branches have emerged: imitation learning and the method of setting intrinsic incentives. Imitation learning [6] guides the exploration process by making the agent observe and imitate the behavior pattern in the actual process of human processing, so that the state-action trajectory distribution generated by the model matches the trajectory distribution of the input. It focuses on obtaining better performance through artificial pre-training, but there are still some limitations. For example, a model that is pre-trained by manual will inevitably be affected by pre-training data, which makes it difficult to be transferred and applied to other scenarios. The other is to set up intrinsic motivation, which does not need the limitation of pre-training data and assists agents to complete each sub-task by providing intrinsic rewards in the process. This auxiliary reward makes up for the absence of external reward, encouraging the agent to explore new environment and learn various possible opportunities. However, such methods are usually faced with a serious problem [7]. Agents usually choose the maximum prediction error for action judgment. When the environment changes frequently, agents may mistakenly judge the current environment as the new position state. For example, if there is a TV set with regular screen changes in the environment, every time the intelligent body visits the location, it will judge the dynamic picture as the new location instead of the previously recorded location. The worst case is that the intelligent body is attracted by the images of constantly changing programs and gets stuck in a deadlock state, which is the problem of "cotch-potato".

In this paper, to solve the cotch-potato problem, the graph compression cycle deprivation curiosity algorithm (KRD-Curiosity) is proposed as the intrinsic reward mechanism of reinforcement learning model. It can predict the position of the agent through the context memory sequence to determine whether the agent is in a state of addiction. At the same time, the time fading sequence is set to punish the addicted state, and the path length of state network nodes with lower clustering is explored as dynamically as possible, so as to increase the information processing speed in the navigation process and improve the navigation efficiency. The main contributions of this paper are as follows:

- By taking advantage of the different sensitivities of agents to poor information at different

stages, a curiosity algorithm based on dynamic deprivation was designed to predict navigation behavior by local context storage unit, and time fading was added to avoid the problem of agents being addicted to dynamic scenes.

- By referring to human learning mechanism, the above algorithm of knowledge graph model compression optimization is proposed, which abstracts the prior knowledge clustering class into a more simplified model, which can be used to compress and simplify old memory, reduce the information entropy of stored data, increase the speed of information processing of agents, and improve the efficiency of agents exploring the environment.
- We created a complex multi-room navigation environment, and set ablation experiments on it to compare and evaluate the navigation success rate of our method, test the navigation generalization effect of our method at different starting points, explore the influence of curiosity behavior of different deprivation on navigation efficiency, and analyze the correlation between different degrees of compression precision and navigation performance.

2. Realtion Work

Autonomous navigation of mobile agents: As the basic function of mobile robots to execute various instructions, autonomous navigation technology requires robots to move to the designated position safely, which has attracted the attention of researchers in the field of robotics [8]. With the improvement of computer hardware performance, more and more algorithms are applied in mobile robot navigation [9], such as rapid exploration random tree (RRT)[10], simultaneous localization and mapping (SLAM)[11], artificial potential field method [12], fuzzy logic [13], etc. These algorithms have achieved satisfactory robot navigation effects to a certain extent. However, for the unknown scene, the agent is susceptible to various noises and environmental dynamics, resulting in wrong judgment and navigation failure. The recent reinforcement learning framework can successfully explore the target point without too much prior knowledge and achieve relatively advanced performance in the field of mobile robot navigation [5].

Reinforcement learning based on strategy gradient: In recent years, reinforcement learning (RL) is almost all based on strategy gradient. The advantage of strategy gradient based method is that it can learn to select different strategies according to different observation states and output action values directly and stably, eliminating the intermediate step of value based method. At present, a relatively popular algorithm is Actor-Critic algorithm [14]. It can use the idea of confrontation to conduct real-time learning environment and reward relationship, and it performs well in a small environment. In order to further improve the performance of the model, researchers have proposed many improved algorithms based on strategy gradient, such as A2C[15], A3C[16], PPO[17], etc. Among them, the A2C algorithm reduces the risk of high variance by taking the value function as the reference value of cumulative reward. Moreover, for the storage speed of large memory network, A2C trains faster, showing better navigation efficiency. In addition, OpenAI researchers found that A2C showed better performance than A3C when using a single GPU machine, and A2C trained faster when the designed strategy network was larger. For the above reasons, A2C is chosen as the benchmark algorithm in this

paper.

Intrinsic reward mechanism in reinforcement learning: In the paper on reinforcement learning, several approaches are proposed using intrinsic rewards to explore. For example, Ng et al.[18] proposed a reward shaping function to improve the optimal strategy, and put forward some auxiliary reward design methods to obtain the strategy with desired attributes. Sorg et al. [19] introduced the strategy gradient of reward design, which is only applicable to planning agents based on forward search. Rewards shaped or designed can help agents explore the environment, but it is still a big challenge to calculate the optimal reward. Imitation learning based on exploration method has been applied to sparse reward tasks [20, 21] and achieved good results. Other attempts [22, 23] introduce expert demonstrations and trajectory preferences, and train agents to mimic the demonstrator's behavior; In addition, Schmidhuber et al. [24] studied a method to improve performance by reward conversion. However, reward switching is designed by experts, not learned. Getting quality expert information is difficult. These methods are not suitable for sparse reward setting. They often require the support of experts or prior data sets, and are highly dependent on training samples, which makes it difficult to adapt to navigation and exploration tasks in unknown scenes. Therefore, the setting of intrinsic reward function plays a decisive role in reinforcement learning navigation tasks.

Curiosity mechanism in psychology: The mechanism of human curiosity has also been extensively studied in the field of psychology. Curiosity is a mind-driven behavioral mechanism for searching for knowledge. The information obtained through curiosity may provide you with new and challenging stimuli in the future and contribute to personal happiness. The research divides curiosity into two categories [25], namely "busyman" and "hunter", which have completely different behavior patterns in the face of unknown environment. The former is more inclined to explore knowledge in new fields. The latter are more inclined to explore undiscovered knowledge in the same field, so they show different feelings of deprivation, that is, curiosity. busyman has a strong sense of information deprivation and has a large breadth of knowledge acquisition but a small depth. "hunter" has a weak sense of information deprivation and a greater depth of knowledge acquisition. At the same time, George Loewenstein, a psychologist and economist at Carnegie Mellon University, has proposed an "information gap theory" of curiosity: [26] when knowledge is scarce or abundant, it doesn't excite us. In other words, the navigation agent should be endowed with different deprivation curiosity for different stages in order to play a better navigation performance.

In this paper, drawing on the cutting-edge research theories of psychology, we design the graph compression cycle deprivation algorithm as an intrinsic reward mechanism, and use A2C algorithm to construct the framework of reinforcement learning autonomous navigation. Meanwhile, the navigation efficiency of agents affected by different deprivation curiosity was studied. Specifically, we mainly take A2C reinforcement learning algorithm based on strategy gradient as the framework, and use dynamic deprivation curiosity rule to dynamically adjust the navigation strategy of agents according to the degree of knowledge acquisition. Local memory context storage unit is used to predict the location and state information of the arrival point. And set the time fading mechanism, so that the agent to a certain range of repeated visits to the scene of the reward gradually reduced, can prevent the agent because of curiosity and addiction into a dead cycle state; Finally, this paper proposes a knowledge graph compressed sensing method to extract and store memory features in a high-dimensional model, which reduces

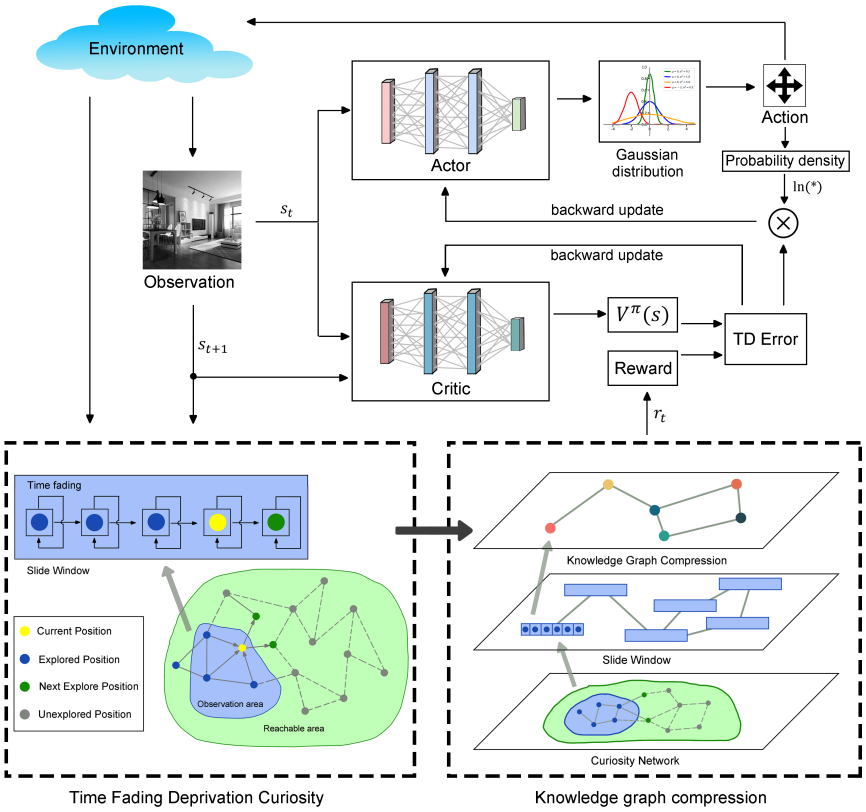

Figure 1: Model mechanism diagram. The bottom left is a curiosity deprivation mechanism based on time fading, and the bottom right is a knowledge graph compression method

the pressure of network storage and improves the speed of information processing. Finally, the relationship between the sense of deprivation and the success rate of navigation and the correlation between the precision of knowledge compression and the success rate of navigation were tested. Experiments show that the rule of curiosity deprivation in psychology is also applicable to navigation agents, and the reinforcement learning based on strategy gradient can improve navigation efficiency and navigation performance.

3. Proposed Approach

3.1. Navigation Framework

For a standard reinforcement learning navigation process interacting with the environment, at each time step t, the agent senses the environment state s_t by external sensor, and processes the observed state s_t by strategy network $\pi(s)$ to generate the next action a_t, which enables the agent to move to the next position s_{t+1}. At the same time, the reward weight r_t of the previous step is obtained, and the final goal of navigation is reached after multiple rounds of training.

The reinforcement learning model in this paper is based on A2C method. Since there is a problem of high variance in the calculation process of traditional strategy gradient algorithm or traditional actor-critic method, it is difficult for the model to converge. A2C algorithm reduces the risk of high variance by taking the value function as the reference value of cumulative reward. The initial state of this algorithm generates an approximate Gaussian distribution through the Actor network, and the action is obtained through sampling and pruning. The agent performs the action and interacts with the environment to generate the next observation state s_{t+1}. The environmental interaction parameters are input to the KRD-Curiosity algorithm at the same time to get the reward r_t. Input s_t and s_{t+1} into Critic network to get a value function $V^{\pi}(s)$, calculate TD error, and carry on backward updating Critic network all the way. At the same time, TD error and the action probability density output by Actor network are multiplied by logarithm, and finally backward updates the Actor network parameters.

In order to solve the problem of catastrophic forgetting and the problem of "cotch-potato" in traditional cursity-based reinforcement learning navigation, more effective intrinsic reward mechanisms are needed. The intrinsic reward in this paper is divided into two parts: (a) The curiosity module based on the dynamic sense of deprivation of time fading, which can predict the future state by combining environmental and time factors through short-term memory, so as to reduce the attraction of dynamic scenes to agents. (b) Knowledge graph compressed sensing can store old memory features in high-dimensional states, reduce the storage pressure of agents, and delay the occurrence of forgetting, as Figure 1.

3.2. Recurrent Deprivation Curiosity

Curiosity based exploration basically solves the problem of lack of intrinsic reward in navigation behavior of agents, but there is usually a "cotch-potato" problem. When navigating the maze, the navigation agent is likely to be fascinated by the TV that keeps changing the screen. In this case, the intrinsic reward will be great and the agent will be more inclined to stop and watch TV rather than continue to explore the maze, leading to the occurrence of "addictive" behavior.

We propose a dynamic deprivation-driven curiosity module based on time fading. Deprivation-driven curiosity refers to the degree to which individuals seek information in order to overcome the feeling of being deprived of knowledge. When individuals feel knowledge gap or knowledge gap, they will have the feeling of being deprived, which drives them to have strong curiosity to learn new knowledge. This sense of deprivation also decreases with the narrowing of the knowledge gap, and the preference for behavioral strategies gradually shifts from high exploration to high utilization. At the same time, by setting rewards that are inversely proportional to time, agents can avoid being attracted by interesting scenes, so as to complete navigation exploration behavior faster.

By comparing two different method can predict state output characteristics $\hat{\phi}(s_{t+1})$ and $\hat{\phi}'(s_{t+1})$, to output reward mechanism, as shown in Figure 2. The Model contains two child prediction module, the first module is the Forward Model, the position of the above step state $\phi(s_t)$ and the step on the action of a_t for input, forecast the explicit position of time step in the $\hat{\phi}'(s_{t+1})$; The second submodule, Slide Window and Time Fading, is based on input of the position status of the next step on the intelligence and the current position status $\phi(s_{t+1})$.

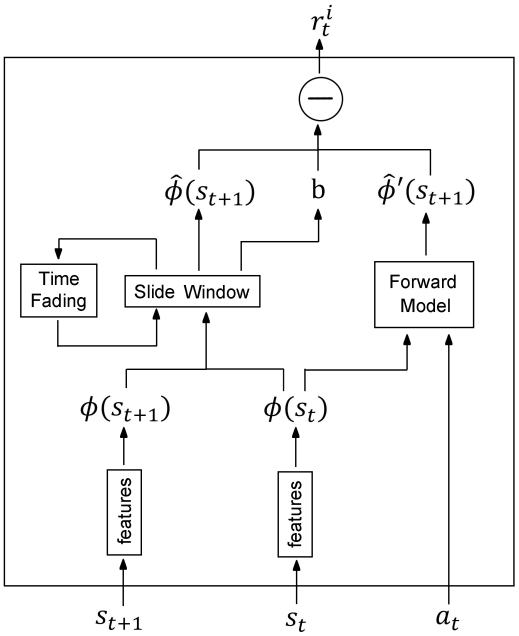

Figure 2: A curiosity deprivation mechanism based on time fading

By matching the memory in the sliding window, while assisting with temporal attenuation, Output location potential energy state $\hat{\phi}(s_{t+1})$ and weight deviation b. The intrinsic reward r_t^i algorithm of curiosity is:

$$r_i^t = ||\hat{\phi}'(s_{t+1}) + \hat{\phi}(s_{t+1})|| + b$$

Among them, the formula of each component is:

$$i = SW.find(\phi(s_t))$$

$$\hat{\phi}(s_{t+1}) = SW[i+1]$$

$$b = \frac{1}{n}\sum_{t=1}^{n}(\hat{\phi}(s_{t+1}) - t_{i+1} \times \phi(s_{t+1}))^2$$

$$L_F = \frac{1}{2}||\hat{\phi}'(s_{t+1}) - \phi(s_{t+1})||^2$$

Where, SW represents the Slide Window vector, $find()$ represents the index number returned from the corresponding value found in the slide window, and t represents the existence time of elements in the window, which is stored as a vector format. L_F is the loss function of the Forward Model network in the first submodule. The second submodule's algorithm flow is that the Slide Window is first matched with $\phi(s_t)$, and updates the contents of the Slide Window if it doesn't have one in the memory. If it exists, the next position matching the position is found as the agent's habitual action output. Our ultimate goal is to make a big difference between

the agent's actual action and the habitual action, so that the agent can explore as much area as possible.

We also use dynamic deprivation mechanisms so that the ratio of exploration to exploitation decreases as the success rate of exploration increases:

$$e_{greed} = \frac{explore}{use}$$

$$explore = (0.8 - \frac{0.8 - 0.2}{100} \times min(10, \frac{successTimes}{10})) \times use$$

At the beginning, the ratio of exploration to utilization is set to 0.8, indicating a state of high exploration. At this time, the curiosity of the agent on the unexplored area is greater than the prior knowledge. With the increase of successful exploration times, the agent gradually mastered the navigation method of this map, and the ratio of exploration to utilization gradually decreased from 0.8 to 0.2, Set the control range to 100. transforming from high exploration to high utilization. It makes the agent use the prior experience to navigate while ensuring a certain curiosity.

3.3. Knowledge Graph Compression

The existing deep reinforcement learning shows obvious advantages in predicting the navigation behavior of agents, but there are still many defects. Among them, it is an unavoidable problem that agents are too curious to forget the previously learned knowledge. That is, after learning new knowledge, they almost completely forget the previously learned content, which makes artificial neural networks unable to constantly adapt to the environment and carry out continuous learning like human beings. For curiosity-driven deep reinforcement learning, this phenomenon will inevitably occur as agents continue to input new states into the network.

In psychological studies, the memory mechanism of mammalian brain may avoid this problem. They usually summarize a large number of messy knowledge into a systematic knowledge framework for storage. The advantage of this is that new information can be stored in a smaller order of magnitude, and the reduction of information entropy can bring more benefits.

Based on this mechanism, we propose a knowledge graph model compression algorithm, which compresses the knowledge in local memory into a more compact model for storage. After high-dimensional transformation of the environmental observation quantity s_t, KNN network is used to cluster into a certain region for storage. The distance calculation formula of KNN is as follows:

$$d(x, y) = \sqrt{(\sum_{k=1}^{n}(x_k - y_k)^2)}$$

x_k and y_k are coordinate values of two points. When the agent explores the same cluster region, a smaller reward value r_t is given; when the agent detects a new cluster region, a larger reward value is given. This approach further improves the intensity of the curiosity mechanism while avoiding catastrophic forgetting.

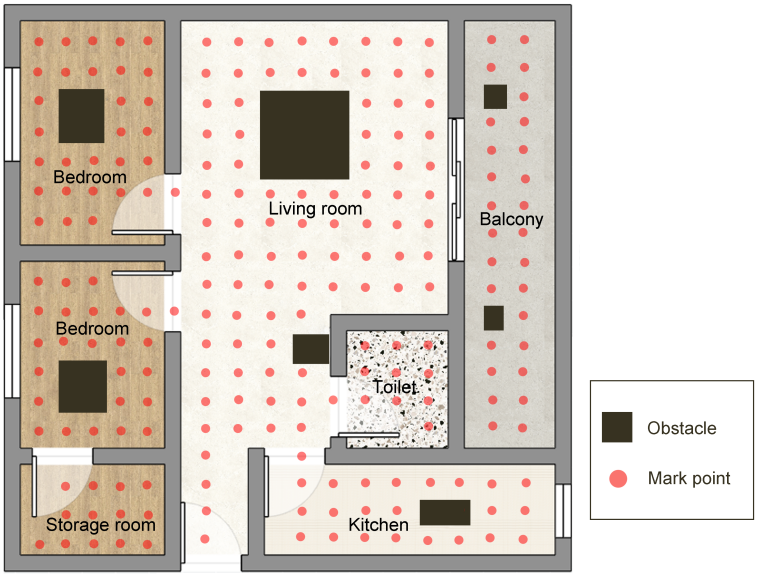

Figure 3: Schematic diagram of experimental environment

4. Experience and Results

In this section, we autonomously create a complex multi-room scene "Multiroom" on which we will evaluate our algorithm. The main purpose of navigation is to find the minimum step from the starting position to the target position. We first evaluated the navigation results using our model, and then conducted an additional experiment to investigate the effect of knowledge extraction capability on navigation efficiency. All models were implemented in Python and trained on a GeForce RTX 2080 Ti GPU.

4.1. The Multioom Enviorment

In order to verify and evaluate our model, we independently created a multi-room complex scene framework by referring to the "CliffWalking" class module in the gym library. This framework refers to the common structure of 80 square meter family environment, which can well represent the general indoor scene in the real world. Within this environment are Spaces for bedrooms, balconies, kitchens, bathrooms, living rooms and dining rooms. Each scene space is represented as a grid diagram, and the spacing between grid points is 4-meter, with a total of 20×20 points. The navigation agent is able to move through five independent actions: forward, backward, left, right, and finish. I'm going to move it in 4-meter units. The end point has been set before the test, and the agent executes the completion action when it judges that the end point has been reached.

4.2. Implementation details

For the training process, we made the agent perform the navigation task in the above environment, and the model was set in 500,000 episodes for training, testing the total reward of the output environment in each stage. The total training time was about 96 hours. End the episodes when the navigation agent has reached the target position, or when more than 1000 steps have passed. In the initial state, the ratio of exploration to utilization is set at 0.8, and the maximum change limit is 0.2. The starting point of the curiosity reward is -100, and the reward for navigating to the end is 100. The intrinsic reward for each step is determined by the curiosity mechanism. In order to encourage the agent to explore and navigate more effectively, the reward range is -5 to 10.

4.3. Evaluation metrics

Our navigation experiments were evaluated using the following two indicators, average track length (ATL) and success rate (SR), which describe navigation efficiency from different dimensions. ATL represents the average number of steps required by the agent to reach the target, which is used to evaluate the path optimization ability of the navigation agent. The SR is defined as $\frac{1}{N}\sum_{i=1}^{N} S_i$, used to assess the convergence efficiency of navigation agent training process.

4.4. Baselines

The reinforcement learning model in this paper is the basic model, which can also be replaced by other reinforcement learning models based on strategy gradient. Therefore, the influence of different models on navigation performance is not within the comparison scope of this paper. We only evaluate and explain the curiosity mechanism proposed by ourselves. In order to prove the effectiveness of our proposed curiosity mechanism based on sense of deprivation on navigation efficiency, we set up a comprehensive ablation experiment according to three technical points and drew a graph, and used the above three indicators to evaluate each baseline. Table 1 shows our baseline and the description of technical points.

Table 1
Baseline and description of ablation methods

Algorithm metrics	Dynamic deprivation	Time fading	Knowledge graph compression
Random exploration	✓	×	×
Only time fading	×	✓	×
Only compression	×	×	✓
No compression	✓	✓	×
No time fading	✓	×	✓
No dynamic deprivation	×	✓	✓
Ours	✓	✓	✓

4.5. Result

In this section, we mainly set up five experiments: ablation curves of each method, data tables based on two indexes, and the relationship between different deprivation senses, compression clustering accuracy and navigation success rate.

4.5.1. Sample efficiency

To analyze the navigation efficiency of our proposed method, we first tested a comparison experiment based on the ablation method. We compared our method to other methods that have undergone ablation. ATL and SR are used to reflect the navigation performance. For each method, we set a maximum of 300,000 test sessions. In each turn, the turn ends when the navigation agent reaches the goal or more than 1000 moves.

Figure 4 shows the comparison between other ablation methods and our method in the ATL index. Most of the other methods complete convergence between 150,000 and 200,000 episodes. It shows that the memory capacity of the agent is an important factor restricting the navigation performance. Our method can improve the entropy of stored information and alleviate the forgetting mechanism.

4.5.2. Generalization across new targets

In order to analyze the generalization ability of navigation, we use the same training model, set the same starting point but different ending point, and conduct navigation generalization test on the navigation body. We took the distance between the new endpoint and the original endpoint (step) as the X-axis to explore the relationship between navigation generalization ability and exploration success rate.

During the test, we randomly set 5 new target points within a certain step distance from the original target point, and set an upper limit of 1000 steps for each new evaluation point. When the agent reaches the target or completes 1000 steps, the navigation is completed, and the average value of 5 success rates is taken to draw. As shown in Figure 5, these ablation methods have a certain generalization ability with our method. In summary, the method proposed by us has the best generalization effect.

4.5.3. The deprivation of curiosity

According to the research of psychology, regardless of the influence of innate factors and individual differences, each individual in different stages of knowledge acquisition is also sensitive to curiosity. George Loewenstein, a psychologist and economist at Carnegie Mellon University, has proposed the "information gap theory" of curiosity [26] : when knowledge is scarce or abundant, it does not inspire curiosity. In other words, the navigation agent should be endowed with different deprivation curiosity for different stages in order to play a better navigation performance. In order to verify the different degree of demand for curiosity deprivation of navigation agents at different training stages, we used the proposed method to extract five training time steps of 50,000, 100,000, 150,000, 200,000 and 250,000 respectively as the starting point of the test, and started from these time steps. Take different deprivation

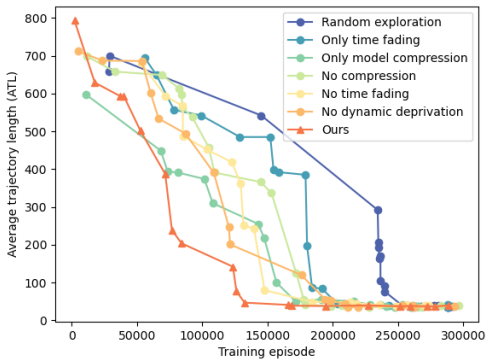

Figure 4: Ablation experiment

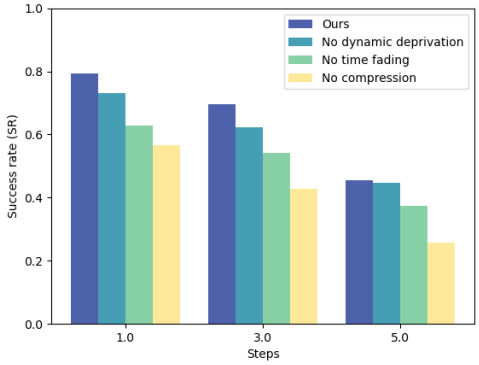

Figure 5: Generalization experiment

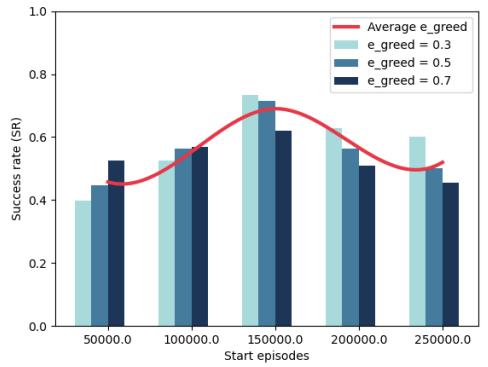

Figure 6: Deprivation curiosity experiment

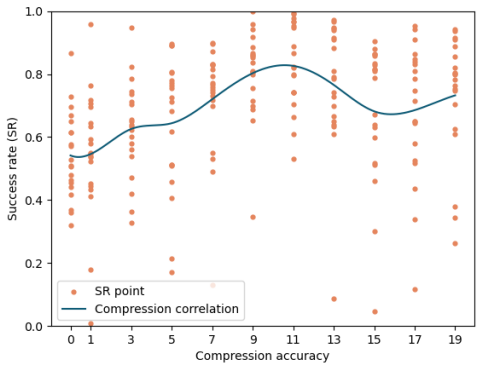

Figure 7: compression accuracy experiment

strategies e_{greed} for continuous training until the 300,000th step, and evaluate their success rate (SR), as shown in Figure 6.

The orange curve represents the average success rate (SR) of each training step. The distribution curve of the sense of deprivation of curiosity shows a trend of "rising in the middle and declining on both sides" on the whole. Among them, agents of 50,000 and 100,000 steps are more sensitive to exploration, and navigation with prior knowledge is more effective after 150,000 steps. The author analyzed that this might be due to the convergence to the optimal path after 150,000 steps, and curiosity exploration at this time could not bring positive rewards, leading to a gradual decline in the success rate. In accordance with this law, our method has set higher e_{greed} in the early stage of navigation, that is to say, given a higher weight value for curiosity exploration. When intelligent agents have grasped certain navigation knowledge, they will gradually reduce the proportion of exploration and increase the proportion of prior knowledge to achieve the optimal navigation strategy.

4.5.4. Navigation efficiency on Compression Degree

In order to test the correlation between the compression accuracy of knowledge graph and navigation performance, equispaced tests were set on the proposed method. Each compression accuracy was tested 20 times, and the mean value was calculated to draw the correlation curve, as shown in Figure 7. In this environment, when the compression accuracy is 9 and 11, the effect reaches the peak, indicating that appropriate data compression is helpful to improve the success rate of navigation. When the compression accuracy reaches 19, the correlation curve shows an upward trend. The author believes that the reason is that when the compression accuracy becomes smaller, the agent shows a state of global memory, that is, it does not compress and store in the memory bank at all. Although the global storage can fully retain the memory of the agent, the storage capacity will become large, which is a challenge to the size of the network model.

5. Conclusion and Future Work

This paper proposes a navigation method combining the time fading mechanism of curiosity deprivation and knowledge graph compression. By depriving the curiosity mechanism to predict navigational behavior, and punishing the state of prolonged addiction, knowledge graph compression was used to reduce the storage of redundant information. The proposed navigation system not only improves the efficiency of data processing during training, but also improves the navigation performance. In addition, the reinforcement learning framework of the navigation system in this paper is replaceable, and it can also be applied to other models based on policy gradient, such as DDPG and PPO. This method aims to explore target points in the scene without prior map as efficiently as possible, so it can provide a way of thinking for navigation search and rescue. Our future work includes investigating more complex curiosity mechanisms to further improve the navigation performance of our approach, conducting real-world validation tests on mobile robots, and evaluating the feasibility and navigation efficiency of our model in real-world navigation environments.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Transactions on robotics 32 (2016) 1309–1332.

[2] A. Wahid, A. Stone, K. Chen, B. Ichter, A. Toshev, Learning object-conditioned exploration using distributed soft actor critic, in: Conference on Robot Learning, PMLR, 2021, pp. 1684–1695.

[3] D. Wu, Y. Lei, M. He, C. Zhang, L. Ji, Deep reinforcement learning-based path control and optimization for unmanned ships, Wireless Communications and Mobile Computing 2022 (2022) 1–8.

[4] Q. He, J. L. Liu, L. Eschapasse, E. H. Beveridge, T. I. Brown, A comparison of reinforcement learning models of human spatial navigation, Scientific Reports 12 (2022) 13923.

[5] K. Zhu, T. Zhang, Deep reinforcement learning based mobile robot navigation: A review, Tsinghua Science and Technology 26 (2021) 674–691.
[6] P. Kormushev, S. Calinon, D. G. Caldwell, Robot motor skill coordination with em-based reinforcement learning, in: 2010 IEEE/RSJ international conference on intelligent robots and systems, IEEE, 2010, pp. 3232–3237.
[7] J. Kober, J. Peters, Imitation and reinforcement learning, IEEE Robotics & Automation Magazine 17 (2010) 55–62.
[8] J. Van den Berg, M. Lin, D. Manocha, Reciprocal velocity obstacles for real-time multi-agent navigation, in: 2008 IEEE international conference on robotics and automation, Ieee, 2008, pp. 1928–1935.
[9] W. van Toll, N. Jaklin, R. Geraerts, et al., Towards believable crowds: A generic multi-level framework for agent navigation, ASCI. OPEN 2015 (2015).
[10] H.-T. L. Chiang, L. Tapia, Colreg-rrt: An rrt-based colregs-compliant motion planner for surface vehicle navigation, IEEE Robotics and Automation Letters 3 (2018) 2024–2031.
[11] Q. Fang, X. Xu, X. Wang, Y. Zeng, Target-driven visual navigation in indoor scenes using reinforcement learning and imitation learning, CAAI Transactions on Intelligence Technology 7 (2022) 167–176.
[12] T. Weerakoon, K. Ishii, A. A. F. Nassiraei, An artificial potential field based mobile robot navigation method to prevent from deadlock, Journal of Artificial Intelligence and Soft Computing Research 5 (2015) 189–203.
[13] E. Korkmaz, Adversarial robust deep reinforcement learning requires redefining robustness, arXiv preprint arXiv:2301.07487 (2023).
[14] R. S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, Advances in neural information processing systems 12 (1999).
[15] Y. Kwon, B. Saltaformaggio, I. L. Kim, K. H. Lee, X. Zhang, D. Xu, A2c: Self destructing exploit executions via input perturbation, in: Proceedings of The Network and Distributed System Security Symposium, 2017.
[16] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, PMLR, 2016, pp. 1928–1937.
[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).
[18] A. Y. Ng, D. Harada, S. Russell, Policy invariance under reward transformations: Theory and application to reward shaping, in: Icml, volume 99, Citeseer, 1999, pp. 278–287.
[19] J. Sorg, R. L. Lewis, S. Singh, Reward design via online gradient ascent, Advances in Neural Information Processing Systems 23 (2010).
[20] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, et al., Deep q-learning from demonstrations, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.
[21] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, M. Riedmiller, Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, arXiv preprint arXiv:1707.08817 (2017).
[22] J. Ho, S. Ermon, Generative adversarial imitation learning, Advances in neural information

processing systems 29 (2016).
[23] F. Torabi, G. Warnell, P. Stone, Behavioral cloning from observation, arXiv preprint arXiv:1805.01954 (2018).
[24] J. Schmidhuber, Formal theory of creativity, fun, and intrinsic motivation (1990–2010), IEEE transactions on autonomous mental development 2 (2010) 230–247.
[25] D. M. Lydon-Staley, D. Zhou, A. S. Blevins, P. Zurn, D. S. Bassett, Hunters, busybodies and the knowledge network building associated with deprivation curiosity, Nature human behaviour 5 (2021) 327–336.
[26] R. Golman, G. Loewenstein, An information-gap theory of feelings about uncertainty, 2016.