Cognitive Map Construction Based on Grid Representation

Yuxin Du
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1064139488@qq.com

Hongge Yao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 835092445@qq.com

***Abstract*—This paper investigates a grid-representation-based approach to spatial cognition for intelligent agents, aiming to develop an effective neural network model that simulates the functions of the olfactory cortex and hippocampus for spatial cognition and navigation. Despite progress made by existing models in simulating biological nervous system functions, issues such as model simplification, lack of biological similarity, and practical application challenges remain. To address these issues, this paper proposes a neural network model that integrates grid representation, reinforcement learning, and encoding/decoding techniques. The model forms a grid representation by simulating the integration of grid cells in the medial entorhinal cortex (MEC) with perceptual information from the lateral entorhinal cortex (LEC), which encodes and retains spatial location information. By leveraging attractor networks, convolutional neural networks (CNNs), and multilayer perceptrons (MLPs), the model achieves the storage of spatial location and environmental information, as well as the construction of cognitive maps. The experimental results show that after using this model, the map generation accuracy increased by 15%, the navigation accuracy of the agent in complex environments by 20%, and the target localization error was reduced to less than 10%, demonstrating a significant overall performance improvement in the grid-based cognitive map construction.**

Keywords-Grid representation; Cognitive map ;Infere-n-ce and generation

I. INTRODUCTION

Cognitive map is an internal structure in biological brain used to represent and understand environmental information, similar to the map in the city, which can guide people to find the destination [1]. In animal experiments, scientists have found that as animals move through their environment, the brain builds an internal cognitive map through interactions between neurons and synaptic adjustments that help animals find food, water, and more. The hippocampus and entorhinal cortex play key roles in memory [5]. The hippocampus is primarily responsible for short-term memory and spatial navigation, while the entorhinal cortex is involved in object recognition and spatial memory. Understanding the role of these two regions in memory and the relationship between them is crucial to understanding how brain-like cognitive maps are implemented.

The aim of the research on the construction of cognitive maps using artificial neural networks is to design a neural network model to simulate the learning and navigation abilities of humans and animals in the real world. This model can help people better understand how the human brain works, but also can provide new ideas and methods for the application of machine learning and artificial intelligence.

In the study of spatial cognition of agents, the construction method of cognitive map based on grid representation is an effective technical means [3]. By simulating the functions of the entorhinal cortex and hippocampus, agents can obtain spatial position, direction and target information, and use this information for spatial cognition, such as agent navigation and decision making [2]. This technique can not only improve the navigation accuracy and efficiency of the agent, but also enhance the adaptability and robustness of the agent in complex environments [4]. At the same time, it can be applied to robot navigation, autonomous vehicles, virtual reality and

DOI: 10.2478/ijanmc-2024-0037

augmented reality and other fi-elds to improve the intelligence level and adaptability of these systems.

II. Type Style and Fonts

A. VAE network

VAE works by sampling the potential space and then using a decoder to convert the sampled potential vector into a new data sample. Since the VAE's encoder maps the raw data onto a Gaussian distribution in the potential space, the potential vector can be obtained by sampling that Gaussian distribution. The sampled potential vector is then decoded to generate new data similar to the original data distribution.

One advantage of VAE in terms of augmentation of generated data is that it can control the degree of variation of generated data. By manipulating dimensions in different directions in the underlying space, selective changes can be made to t-he generated data. For example, you can interpolate or scale specific dimensions to generate data with specific properties or degrees of variation.

It is important to note that when applying VAE for data augmentation, it is necessary to ensure that the new data generated is similar to the distribution of the original data set, which can be achieved through the objective function when training the VAE model. VAE models are often trained with KL divergence of the underlying vector to minimize reconstruction errors, which ensures that the new data generated is similar to the original data set distribution, thus guaranteeing the quality of the new data generated

B. Reinforcement learning and agent cognition

The agent spatial cognition methods based on reinforcement learning usually employ end-to-end training, where action selection and path planning are performed directly from raw sensor readings [4]. Although this method can feed both dynamic and static obstacles into the network through a single frame, it is insufficient in the processing of interactive information. In order to solve this problem, some researchers use the received signal strength to define the return value and adopt Q learning method to complete path planning, but this method is not suitable for uncertain environments with a lot of dynamic obstacles, and the model is difficult to be transferred to the actual environment. Some researchers have trained UAVs by using Soft Actor-Critic (SAC) algorithm to perform autonomous obstacle avoidance in continuous action space using only image data, but the model has poor generalization ability and is difficult to adapt to the new environment [6]. This paper comprehensively considers the problem of multiple obstacles in complex and unfamiliar scenes, processes data through the method of simulated cognitive neuroscience, and further trains the representation vector obtained by grid representation to obtain the current position signal and target position signal of the cognitive map as the input of reinforcement learning network, optimizes the action selection of the agent, and improves the generalization ability of the model as a whole [7].

III. Network model

A. Grid Representation

The grid representation model is a two-branch fusion model (as shown in Figure 1). Branch one extracts the key information of the current environment from the environment through the attractor network to simulate the visual information perceived by animals in the environment, and then extracts its features through the deformable convolution layer. Branch two obtains grid cells through the exploration of the environment by agents. Grid cells are believed to code the relatively unchanged spatial structure information that can span different environments and are input into the Hebbian competitive neural network for competition among position cells [8]. After multiple competitions, key position information is obtained, and then feature extraction is carried out through the deforming convolution layer. Feature fusion is performed on the features extracted from the double branches, and the final output is a kind of representation vector after fusion, which is called grid representation.

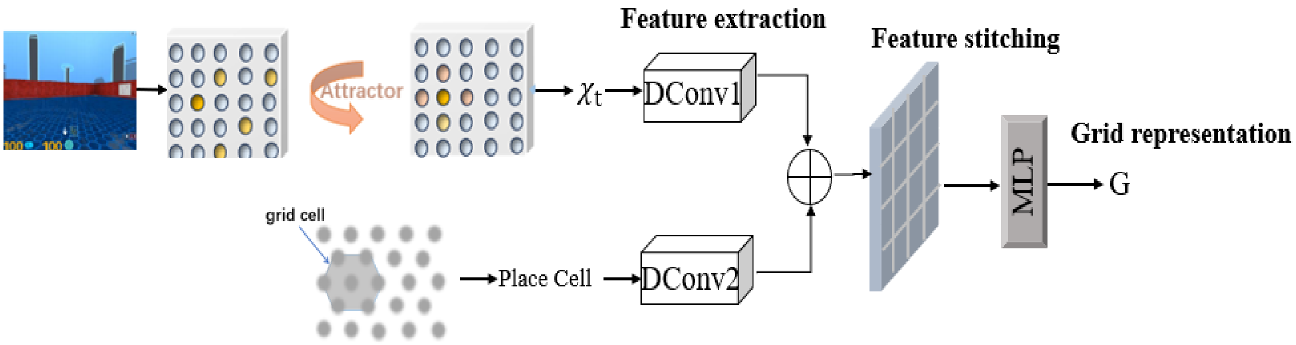

Figure 1. Grid representation model

The grid representation network model is mainly divided into the following parts:

- Visual perception branch: It is used to extract key information about the environment, similar to the visual information perceived by animals.
- Spatial coding branch: It is used to encode location information in the environment to obtain grid cells and location cells.
- Double-branch feature fusion: the features obtained from the visual perception branch and the spatial coding branch are fused to obtain the spliced feature vector.
- MLP training: Input the spliced feature vector into the multi-layer perceptron (MLP) for training, and finally output grid expression G.

The following is a step-by-step explanation of the composition and function of each part.

Branch of Visual Perception (Branch 1) ， the branch of visual perception extracts key information about the current environment from the environment through the attractor network [9]. This branch uses the deformable convolution layer 1 to perform feature extraction on the extracted information to capture important visual features in the environment. The features obtained through this branch can capture the important visual features in the environment and provide the basis for the subsequent feature fusion.

Spatial Coding Branch (Branch 2) ， the spatial coding branch explores the environment through agents and obtains grid cells. Grid cells are thought to encode information about spatial structures that can span different environments and remain relatively unchanged. The grid cells are entered into the position cells in the Hebb competitive neural network to make them compete with each other. After many competitions, get key position information. Then, the deformation convolution layer 2 is used to extract the features of these position information. The features obtained through this branch can capture the spatial structure information of the environment and provide the basis for the subsequent feature fusion.

Two-branch feature fusion is after obtaining the features of deformable convolution layer 1 and deformable convolution layer 2, double-branch feature fusion is carried out. The features obtained from the two branches are concatenated into a feature vector. This feature vector reflects key visual and spatial structure information in the environment, providing more comprehensive features for subsequent training.

MLP training is fusion vectors after processing are then processed by the multi-layer perceptron (MLP), and the vectors output by the activation function are linearly weighted and summed to obtain the final grid representation. The output result of this output layer is a high-dimensional vector representation G, reflecting the distribution and position information of the input information in space, as well as additional information after MLP processing. This output can be used as input to VAE (variational autoencoder) for further processing and encoding through inference models.

B. Inference and generation

The output of grid representation networks can be used as input to variational autoencoders (VAE). VAE is a generative model capable of learning potential representations of data and generating new data samples.

After the grid expression output G at step=t and the output gt-1 of the generated model at step= T-1 at the previous moment are combined as the input of VAE, the inference model of VAE will receive these two vectors as input (as shown in Figure 2) [10].

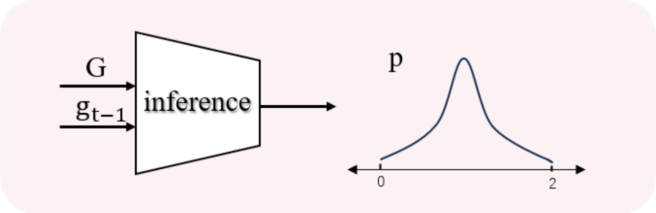

Figure 2. Inference model

Inference models map the input vectors G and gt-1 into the potential space using a nonlinear transformation function, a process that can usually be implemented by one or more neural network layers. The function learns a mapping relationship from the input space to the potential space so that the distribution of the input data in this space is as close as possible to its true potential distribution. In the potential space, the distribution of the data is modeled as a probability distribution, which is usually a multidimensional Gaussian distribution.

After obtaining a representation of the potential space, the inference model calculates the mean and variance of the potential variables. These parameters describe the distribution of potential Spaces and are used to generate new data samples in the decoder. This process is usually implemented using a regularization term, such as KL divergence or ELBO (Lower bound of evidence). The function of the regularization term is to make the potential distribution learned by the model as close as possible to a prior distribution, such as a Gaussian distribution. In order to ensure that the generated distribution can be as close as possible to the true posterior distribution of the task, we construct the variational lower as shown in equation (1), the Loss_Critic in the loss (2) function serves as a constraint for generating latent variables.

$$ELBO = E_T[E_{z_c \sim q_\varphi(z_c|g^T)}[R(T,z) + D_{KL}(q_\varphi(z_c \mid g^T) \| p(z_c))]] \quad (1)$$

$$Loss\_Encoder = \frac{1}{n}\sum_{i=1}^{n} D_{KL}(q_\varphi, p) + Loss\_Critic \quad (2)$$

Then sample a potential variable from the learned potential distribution. This latent variable mimics the place cell, or p, in the hippocampus, which represents spatial cognitive information. The sampling process is usually implemented using a reparameterization trick, by randomly sampling a sample from a potential distribution and then mapping this sample into the output space via a nonlinear transformation function.

In order to ensure the accuracy of the trained state distribution, a generation model is constructed. The input p is sampled from the posterior distribution q(p|g) obtained from the Encoder, and the output is the cognitive map reconstructed state feature g corresponding to the current p. We hope that the specific characteristics of the original state g can be restored according to the sampled p, and the accuracy of the generated p can be measured as a constraint term for generating potential variables.

In order to make the generated distribution as close as possible to the true posterior distribution of the state space, we construct the variational lower bound as in equation (3):

$$ELBO = E_T[E_{Zs \sim q_\phi(p_s|g)}[R(T,z_s) + D_{KL}(q_\phi(z_s \mid g) \| p(z_s))]] \quad (3)$$

In the decoding process, the output we want to get is the cognitive map g of the current moment. To achieve this, we need to add an output layer to the last layer of the decoder, which should be the same size as the dimensions of the cognitive map g. In the output layer, we can use nonlinear activation functions such as sigmoid function or softmax function to map the neuron's output to the range of [0,1] to get the final cognitive map g (as shown in Figure 3).

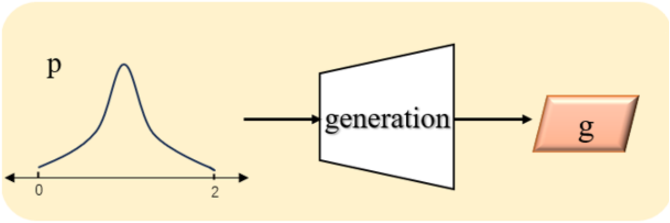

Figure 3. Generating the model

C. Cognitive map construction

The cognitive map construction network is divided into four parts: grid representation, inference and generation, target signal acquisition, and cognitive map construction.

Grid expression: In this part, the sensory perception in the memory is extracted through the attractor network, and the information obtained by grid discharge is used as input, feature extraction is carried out through convolution, and then feature fusion is carried out through MLP to obtain the fused feature vector as grid expression.

Inference and generation: In this part, the grid expression output and the output of the previous generation are taken as inputs together, and the input data is mapped to the potential space through inference to obtain the mean and variance of the potential variable, and then the distribution of the potential variable p. p sampled from the posterior distribution q(p|g) obtained by inference is taken as the input of generation, and after decoding by generation, g_t is output as the cognitive map reconstruction state feature corresponding to the current p.

Spatial cognitive information p is the core inferential information of cognitive map, and also an important basis for constructing cognitive map. Latent variables can be decoded to generate new cognitive maps based on these features.

Target signal acquisition: In order to extract walks from the environment and map them to the potential space as the inference input, the potential distribution p2 of the quadruple [s, a, r, s'] is obtained. The potential distribution contains the core feature distribution of the quadruple, which is goal code, that is, the target signal (as shown in Figure 4).

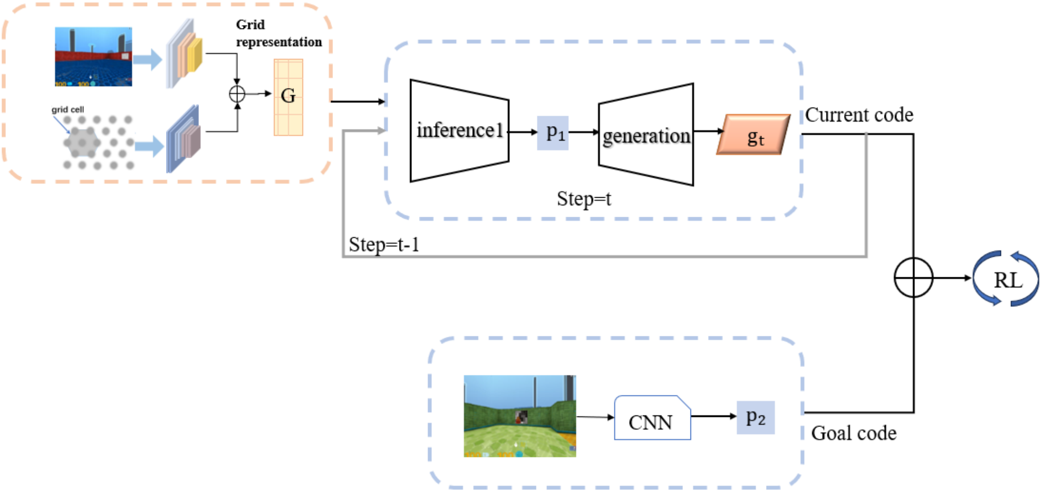

Figure 4. Overall model

IV. Experiment

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Data

In this experiment, a public data set containing road scenes is selected, which contains pictures of various road scenes and corresponding target annotation information, including vehicles, pedestrians and other targets. Such a data set can provide a variety of road environments and help evaluate the model's performance in different scenarios.

B. Navigation and obstacle avoidance performanc -e evaluation experiment

To verify the effectiveness of the cognitive map model in navigation and obstacle avoidance, we construct a virtual maze environment. The maze scene is a two-dimensional plane containing several static obstacles. The test object (the simulated agent) navigates the environment with the goal of getting from the start to the end while avoiding all obstacles.

Obstacle distribution: Obstacles are static and randomly distributed in different locations in the maze. Obstacles vary in shape and size to increase navigation difficulty and simulate diversity in a real environment.

Target point setting: set the starting point and end point of the maze, and require the tested object to reach the end point as quickly as possible without colliding with obstacles.

To visualize and analyze the performance of cognitive map models in navigation, we generate attention heat maps. The heat map shows the location and frequency distribution of the tested object throughout the maze.

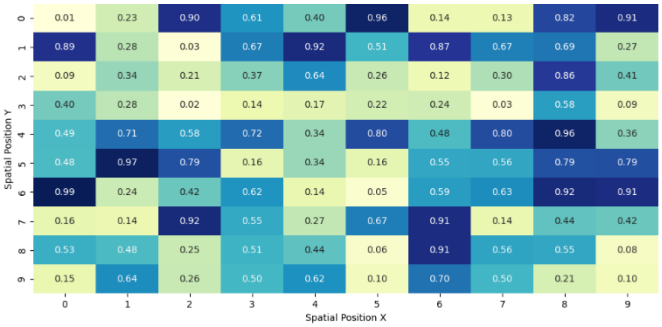

Figure 5. Location attention heat map

Through the heat map in the Figure 5, we can visually observe the position frequency distribution of the tested object in the maze. This heat map is generated by counting the occurrence frequency of the measured object in each spatial position (x, y). The darker the color, the higher the activity frequency of the measured object in this area; conversely, the lighter the color, the lower the activity frequency.

- High frequency areas:

In the heat map, certain areas (such as the (6, 0) and (0, 7) positions in the image) that show darker colors are hotspots of attention. This indicates that the subject stays in these positions for a long time or passes several times. The reason may be that these areas are critical turning points, bottleneck locations in the maze, or more complex path selection areas. These hot spots may have an important impact on navigation decisions, and the model may have performed more calculations or adjustments at these locations to choose the best path or avoid obstacles.

- Low frequency region:

Lighter colored areas indicate places where the subject passes less or spends less time. It may be because the path selection in these areas is relatively simple, or these areas are the edge of the maze or dead end, so the subject does not need to do too much stop in these areas.

The low-frequency activity in these regions shows how confident and efficient the model is on these partial paths, able to pass quickly.

From the distribution of the overall heat map, it can be seen that the behavior of the tested object is more concentrated in some specific areas, which may reflect the effectiveness of the path selection and obstacle avoidance strategy of the cognitive map model in this area. For example, in regions with more complex paths (such as the center or turning point of a maze), the color of the heat map changes more dramatically, reflecting that the model makes more path selection judgments in these regions.

To measure the accuracy of the model prediction, a heat map of the coincidence degree between the model prediction path and the actual navigation path is generated (as shown in Figure 6).

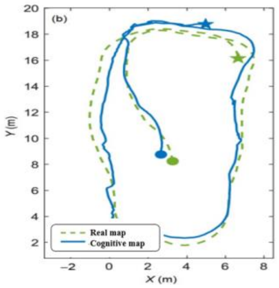

Figure 6. Trajectory comparison diagram

- Method: The predicted path of the model was compared with the actual path, and the areas with high coincidence between the two were marked.
- Objective: To analyze the predictive ability and execution ability of the model in different environments, and determine the difference between the path planning of the model and the actual execution.

C. Contrast experiment

In order to demonstrate the advantages of the cognitive map construction model based on grid representation design in the environment construction task, we compared the object recognition performance of different object detection models on the public data set and the cognitive map image generated by the cognitive map construction model. Common target detection models such as YOLOv5, Faster R-CNN and SSD were selected and deployed, trained and tested in a rigorous experimental environment to ensure the accuracy and reliability of the evaluation results. By comparing and analyzing the performance of the model in different scenarios, the purpose is to verify the data enhancement effect of the cognitive map building model, and explore its advantages and limitations in practical application.

In order to conduct comprehensive and effective data comparison, we choose the following common object detection models as data comparison models:

- YOLOv5: As a fast and accurate target detection model, YOLOv5 shows good performance and low computing cost.
- Faster R-CNN: Faster R-CNN is one of the classic target detection models in the industry, with high accuracy and strong robustness.
- SSD: SSD (Single Shot MultiBox Detector) is a target detection model that can detect multiple frames in a single time, with fast detection speed and high accuracy.
- DETR: DETR is an innovative target detection model with unique performance and advantages in the field of target detection.
- RT-DETR: RT-DETR is also an excellent target detection model with good adaptability to various scenarios.

These commonly used target detection models are selected as data comparison models, and each target detection model is deployed in the environment for training and testing to ensure the accuracy of the model effect.

This paper recorded average accuracy (mAP), accuracy (P), recall rate (R), F1 score and average accuracy (APcar) for each model.

In order to show the performance of each model on different data sets more intuitively, line charts for each index are drawn. Figure8 shows the comparison of the performance indicators of the target detection model under different data sets, and Figure9 shows the comparison of the execution rate of the target detection model under different data sets.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar (Figure 7 and Figure 8):

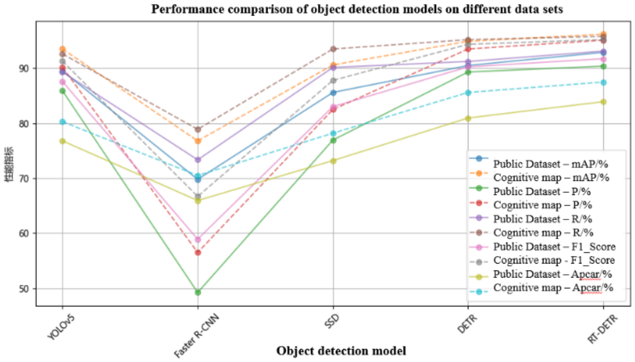

Figure 7. Display of target detection results

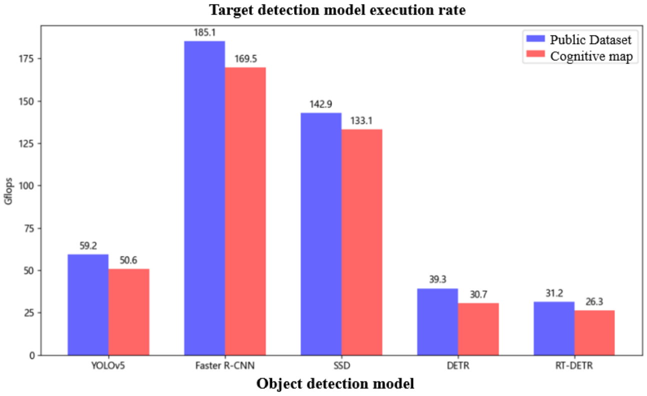

Figure 8. Execution rate diagram of target detection model

V. CONCLUSIONS

Based on the grid representation method, this paper discusses the performance of the agent in spatial cognition and navigation planning. By building a deep learning model that includes a grid representation generator, an encoder, a decoder, an inference model (including the hidden variable p), and a Goal Code generator, we have succeeded in providing the agent with a whole-process solution from environment awareness to target location to path planning.

Experimental results show that the model performs well in navigation and path planning tasks. Grid representation not only provides the agent with clear spatial structure information, but also enhances its ability to understand the layout of the environment. The introduction of hidden variable p enables the model to capture the main distribution of the environment, and improves the environmental adaptability and continuous decision-making ability of the model. The generation of Goal Code provides the agent with clear target guidance, which helps to generate accurate and efficient navigation path.

By combining deep learning techniques, we give the agent powerful spatial perception and decision-making capabilities, enabling it to navigate autonomously and complete tasks in complex environments. This not only provides strong support for the development of intelligent robots, autonomous driving and other fields, but also provides new ideas and methods for the research of future agents in the field of spatial cognition.

In the future, we will continue to optimize the model architecture and parameter Settings to improve the performance and generalization of the model. At the same time, we will also explore more advanced technologies and methods to promote the research and application of agents in the field of spatial cognition. We believe that in the near future, agents based on grid expression will be able to show their strong potential and value in more fields.

REFERENCES

[1] Whittington, J. C., McCaffary, D., Bakermans, J. J., & Behrens, T. E. How to build a cognitive map. Nature Neuroscience, 1-16. (2022).

[2] Farzanfar, D., Spiers, H. J., Moscovitch, M., & Rosenbaum, From cognitive maps to spatial schemas.Nature Reviews Neuroscience, R. S. (2022).

[3] Rueckemann, J. W., Sosa, M., Giocomo, L. M., & Buffalo, E. A. (2021). The grid code for ordered experience.Nature Reviews Neuroscience, 22(10), 637-649.

[4] Andrew Szot and Alex Clegg et al. Habitat 2.0: Training Home Assistants to Rearrangetheir Habitat, Advances in Neural Information Processing Systems, 2021.

[5] Foster, D., Morris, R., Dayan, P. et al. A model of hippocampally dependent navigation,using the temporal difference learning rule. Hippocampus 10, 1–16 (2021).

[6] The Tolman-Eichenbaum Machine: Unifying Spaceand Relational Memory through Generalization in the Hippocampal Formation. Whittington et al., 2020, Cell 183, 1249–1263.

[7] Everett M, Chen Y F, How J P. Collision avoidance in pedestrian-rich environments with deep reinforcement learning[J]. IEEE Access, 2021, 9: 10357-10377.

[8] Shanshan Qin, Shiva Farashahi, David Lipshutz, Coordinated drift of receptive fields in Hebbian/anti-Hebbian network models during noisy representation learning, Nature Neuroscience, pages339–349 (2023)

[9] Laura Cantini, Hope4Genes: a Hopfield-like class prediction algorithm for transcriptomic data, Scientific Reports, 337 (2019)

[10] Zhila Agharezaei, Reza Firouzi, Samira Hassanzadeh, Computer-aided diagnosis of keratoconus through VAE-augmented images using deep learning, Scientific Reports, 20586 (2023).