Wei *et al. Brain Informatics* (2024) 11:23
https://doi.org/10.1186/s40708-024-00237-8

Brain Informatics

RESEARCH Open Access

Modeling biological memory network by an autonomous and adaptive multi-agent system

Hui Wei[1*], Chenyue Feng[1] and Fushun Li[1]

Abstract

At the intersection of computation and cognitive science, graph theory is utilized as a formalized description of complex relationships description of complex relationships and structures, but traditional graph models are static, lack the dynamic and autonomous behaviors of biological neural networks, rely on algorithms with a global view. This study introduces a multi-agent system (MAS) model based on the graph theory, each agent equipped with adaptive learning and decision-making capabilities, thereby facilitating decentralized dynamic information memory, modeling and simulation of the brain's memory process. This decentralized approach transforms memory storage into the management of MAS paths, with each agent utilizing localized information for the dynamic formation and modification of these paths, different path refers to different memory instance. The model's unique memory algorithm avoids a global view, instead relying on neighborhood-based interactions to enhance resource utilization. Emulating neuron electrophysiology, each agent's adaptive learning behavior is represented through a microcircuit centered around a variable resistor. Using principles of Ohm's and Kirchhoff's laws, we validated the model's efficacy in memorizing and retrieving data through computer simulations. This approach offers a plausible neurobiological explanation for memory realization and validates the memory trace theory at a system level.

Keywords Memory modeling, MAS, Memory simulation, Decentralized algorithm

1 Introduction

1.1 Deficiencies in research on brain memory mechanisms

Memory is a cognitive function produced by the activity of the brain's cortical neural system, and the ability to form memory is the foundation for accumulating knowledge and making reasoning judgments. As an extremely complex information processing system, humans have not yet developed a complete theoretical system for the brain's memory mechanisms. Exploring how the brain encodes, stores, and retrieves information remains a significant challenge in memory research. Modern neuroimaging technology is rapidly advancing, and through methods such as fluorescent tagging and two-photon imaging, humans are gradually unveiling the neural system's connectivity and other details. However, such analyses can only offer limited insights, and we are still unfamiliar with the neural mechanisms and information flow directions underlying memory activities. From a neurobiological perspective, memory refers to neural system activities or physical changes in neuronal connections triggered by external stimuli or brain states [1]. Neurobiological research on memory mechanisms focuses on synaptic plasticity [2–5], neurotransmitters [6, 7], and electrochemical signals [8, 9] at the cellular and molecular levels, while cognitive psychology likens memory to an information processing system responsible for encoding, storing, and retrieving information. These theories provide a reasonable starting point to

*Correspondence:
Hui Wei
weihui@fudan.edu.cn
[1] Laboratory of Algorithms for Cognitive Models, School of Computer Science, Fudan University, No. 2005 Songhu Rd, Yangpu District, Shanghai 200438, China

Table 1 Comparison of computer storage management and brain memory management

Category	Computer storage management	Brain memory management
Minimum Functional Unit	Data are stored in binary form, with hard drives using polarity of magnetic particles, charges in capacitors, etc., to represent 1s or 0s	In neural cells, membrane possesses resting and action potentials. In resting state, signal is inhibited; if activated, it generates an action potential, conducting signal
Storage Device	An HDD records data by changing the distribution of the two polarities of magnetic particles using an electromagnet on read/write head; an SDD stores 0s and 1s by altering the number of electrons in the floating gate layer through the application of an electric field	Memory refers to neural system activities or physical changes in neuronal connections triggered by external stimuli or brain states [1]. Many studies have begun to define engrams as basic units of memory, but there are unresolved issues such as how structure of engrams affects quality of memory, how multiple engrams interact with each other, and how engrams change over time [11]
Data Allocation	Operating systems allocate disk space for files with blocks as the basic unit, using methods such as contiguous allocation, linked allocation, and indexed allocation	We have a very limited understanding of how a single memory item is structurally stored at the level of biological neurons, and how multiple memory items are allocated among neural cells (groups) in cerebral cortex
Data Access	Control circuits translate logical addresses into physical addresses. Read/write head of an HDD moves to the cylinder, track, and sector where the data are located for data access; for an SDD, controller directly performs read and write operations on storage units at physical address	When brain stores or recalls a memory, we know little about how it locates engram of memory content among vast number of neural cells in cerebral cortex, replacement of old memories with new ones, and neuronal network mechanisms during recall

map the brain's coarse-scale organization using functional imaging technologies including EEG and fMRI [10]. Neurobiology and cognitive psychology investigate the brain's memory mechanisms from different levels, yet there remains a significant gap in our understanding of the actual biological processes behind memory, making it challenging to accurately model the brain's memory mechanisms.

Neurobiology provides a rich array of dynamic components, and cognitive psychology outlines the global functional layout of the brain, yet neither fully elucidates the complete process of brain information processing. Many unknown details still exist about how biological neural networks encode, store, and retrieve information, which cannot be fully explained by electrophysiological or anatomical experiments, nor by cognitive experiments alone. The challenge in memory research at the level of biological neural networks is to develop a computational model of brain information processing that adheres to neurobiological constraints and can execute memory tasks. Modern digital computers possess powerful storage capabilities, and comparing them with the brain can further our understanding of what is still missing in memory research. Table 1 compares computer storage and brain memory management, from which the implementation process of computer storage, from the bottom to the top layer, is clear. However, the implementation process of brain memory involves numerous unknowns, which motivates us to simulate and emulate the brain's memory process based on neurobiological mechanisms.

1.2 The memory trace theory

The concept of how memory is stored and retrieved has always been a focal point of exploration for neuroscientists. Semon introduced the term "engram" to describe the neural substrate that stores memories [12]. He suggested that experienced events activate a group of neurons to produce chemical or physical changes, thereby forming engrams, with the cells generating memory traces known as engram cells. The reactivation of these engrams can induce the recovery of memories. Hebb proposed the theory of cell assemblies, positing that cells activated by the same event and having intrinsic connections would form cell assemblies, and the synaptic connections between these assemblies would be strengthened [13]. Based on the memory trace theory and the theory of cell assemblies, we have gained further understanding of the neural basis of memory storage and retrieval-that the encoding and storage of memory information depend on the concurrent activation of neurons during memory formation. During the memory storage phase, external stimuli cause a group of neurons in the brain to discharge together, initiating changes in relevant signaling pathways and gene expression. These neurons undergo lasting chemical or physical changes, and the memory information is believed to be stored in the network formed by these neurons.

Thanks to a combination of various techniques such as molecular and cellular neurobiology, physiological recording and multiphoton imaging, transgenic and viral vector-mediated gene insertion, and optogenetics and chemo-genetics, neuroscientists have begun to identify and manipulate memory engram cells. Kitamura and colleagues demonstrated that memories are stored in engram cells through engram cell labeling and optogenetic manipulation, showing that engram cells reactivate during memory retrieval, and artificial activation or inhibition of this group of engram cells can directly trigger or suppress memory expression, proving that fear memories exist in engram cells [14]. Many laboratories' research results have begun to define engrams as the basic unit of memory. Recent studies on memory engram cell populations suggest that the memory trace of a given memory is not necessarily located in a single anatomical location but is distributed across multiple locations connected by a specific memory pattern, thus forming memory engram cell pathways [15].

Currently, three types of evidence support the rise of the memory trace theory:

1. Observational studies provide correlational evidence between the physiological and structural characteristics of neurons in specific brain regions and memory behaviors
2. In functional loss studies, animals or humans with physical or chemical damage to specific brain regions exhibit impairments in certain aspects of memory behavior.
3. The use of transgenic, optogenetics, and chemogenetic techniques to identify specific subgroups of cells related to specific memory behaviors. The ability to identify and manipulate engram cells and the whole-brain engram complex has advanced the study of the memory neural substrate. However, many unknown details remain about how biological neural networks store information, such as how engram structures affect memory quality, how multiple engrams interact, and how engrams change over time [11].

1.3 Directed graph and MAS for brain modeling

The brain can be conveniently represented as a network of neurons and their directed interconnections, and memory maybe represented by those connection patterns according to the trace theory, making directed graph a mathematical tool for studying its structure and memory system. Directed graph, as a branch of graph theory, which provides tools for processing and analyzing network structures. Tracing back to Euler's solution to the Seven Bridges of Konigsberg problem in the 18th century [16], it has been applied such as to design solutions for the Traveling Salesman Problem(TSP) [17], construct knowledge graphs [18], and create databases using graph structures [19], where graphs usually serve as structured representations of data or knowledge, inherently lacking dynamic behavior, with their functionality reliant on externally applied algorithms. In these applications of graph theory, algorithms typically operate on the graph structure from a global view, meaning that the executor of the algorithm (like a CPU) has access to global information and makes decisions.

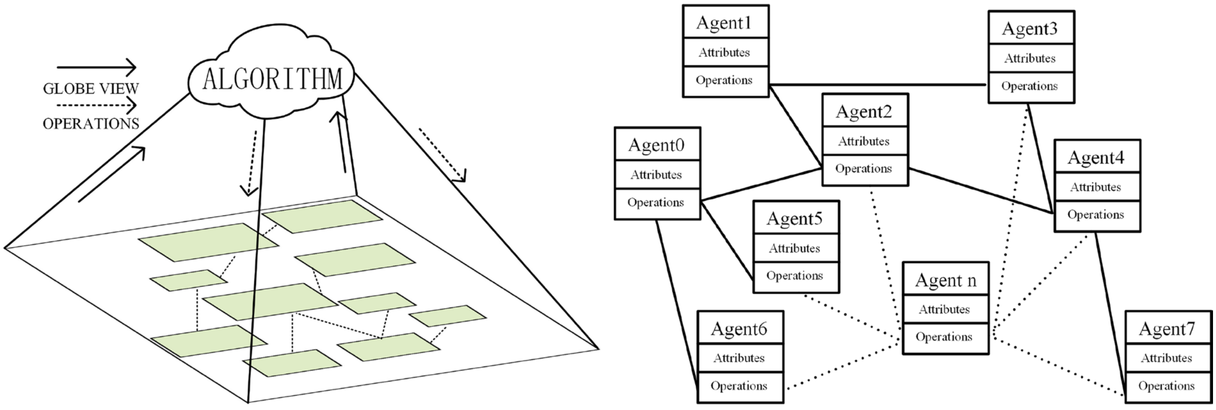
The modeling of brain by directed graph is centralized, needs a "God's eye" view, while biological neural networks lack such a global view or central controller and are characterized by decentralization, consisting of many simple units that are only connected to their neighbors. But if we upgrade the static nodes in the directed graph to dynamic agents, creating a network system by multiple agents, it can better align with biological neural networks and more accurately model and simulate biological memory processes. We propose an algorithm that does not rely on a "God's eye" view, focusing on implementing a memory function in a multi-agent system (MAS). Just as neurons have only local connections, agents in a MAS can only see their connected neighbors. Each agent makes decisions based on its local field of information. Agents are no longer passive data storage nodes, but are active, autonomous units, which can adaptively learn how to respond in different contexts, simulating the behavior of biological neurons. Passive nodes with a single global algorithm and active agents with numerous independent small algorithms represent two completely different paradigms. Contrasting these two modes in Fig. 1, we see the difference between traditional centralized processing and the proposed decentralized processing. In the latter, each agent stores information and can also process and transmit information, forming complex dynamic patterns across the entire MAS.

We focus on the information processing mechanisms behind brain memory activities, using MAS to comprehensively model and simulate memory, abstracting memory instances into directed paths in the MAS. Through MAS and the pervasive learning algorithms inherent in each agent, it dynamically learns and optimizes these paths, achieving a memory mechanism based on MAS, enabling efficient information memory and retrieval. MAS can provide a detailed implementation algorithm for imprinting hypothesis at the directed graph network level. Perhaps providing an intermediate level between the low-level electrochemical level and the high-level behavioral level, offering a novel perspective on the brain's memory mechanisms.

2 Related works

2.1 Hopfield and BAM network

The Hopfield network [20] is a type of recurrent neural network that can be abstracted as a directed complete graph, with many variants [21–26]. After a finite number of iterations, the input and output signals of the Hopfield network no longer change, and the state of the network is the memory vector obtained through recall. Each node



(a) Centralized Mode: Single Global Module Operates on Graph from Detached, Overarching Perspective

(b) Decentralized Mode: Each Agent Possesses Attributes and Autonomous Capabilities, Such as Establishing Connections with Other Agents

Fig. 1 Comparison of centralized and decentralized modes

receives inputs from all nodes except itself and outputs them to all nodes except itself, which is the contradiction between Hopfield networks and biological neural networks.

Bidirectional associative memory networks [27] implement bidirectional associative functionality and have undergone many modifications [28–31]. Bidirectional association refers to the corresponding patterns stored in a network. In the BAM model, for successfully stored pattern pairs in the network, inputting any corresponded pattern can obtain a corresponding pattern. Like Hopfield networks, the input and output layers in BAM networks are fully connected, which is different from actual biological neural networks.

2.2 Multi-agent systems

Multi-agent systems have gained widespread attention in recent years. These decentralized systems accomplish a task through the collaborative effort of many small entities, each with independent behavior. The small entities make distributed independent decisions with individual tasks assigned to autonomous entities called agents, each determining the correct action for solving the task based on their inputs. Individual agents only have partial information and communicate with their neighbors. These are applied in areas such as computer networks, drones, and robot swarms [32].

MAS are extensively used in robotic systems and to some extent can solve issues such as local positioning, obstacle detection, path planning, and navigation in multi-robot systems [33]. For example, a collision avoidance problem was solved for multiple robots in a decentralized, distributed framework, where input data were only collected from onboard sensors [34]. Each robot is an agent that shares its decision with all other robots, thereby achieving an optimal strategy among them.

Although multi-agent systems utilize distributed and decentralized decision-making, the connections between agents in these applications are relatively weak and do not form a fixed pattern for memory or other applications, currently there is no research on the implementation mechanism of biological memory in MAS.

3 Using multi-agent system to model biological neural network

3.1 Modeling of the agent network

Inspired by the graph structure of Hopfield network and independent decision making abilities of agents, our proposed MAS model can more accurately simulate and understand the complex memory processes in the human brain, whose memory system consists of many structurally similar neurons connected by synapses, which can be abstracted as agents and its connections in a MAS, as shown in Fig. 2. Abstractly modeling this as a locally connected MAS agent network aligns with biological reality. The process of signal transmission between neurons can be abstracted as a dynamic network flow.

We utilize a two-dimensional matrix of agents to form the MAS and abstractly model the biological neural network. Figure 3a depicts a generated 10×10 scale agent matrix of a MAS network. Agents, which possess their own behavior, are connected to neighbor agents through directed connections. Edge and internal agents are shown in orange and gray, respectively. Arrows between agents represent connections and directions of signal transmission. Notably, although there are bidirectional connections between edge and internal agents, only one direction is activated at a time. When an edge agent's connection is directed toward an internal agent, it is used as an input, representing the abstract definition of a memory instance, such as of an object's name. When a connection is directed toward an edge agent, it acts as an output, representing specific features of a memory instance such as color or smell. A non-activated connection indicates that the memory instance lacks a certain specific feature. The proposed model's underlying MAS network should be capable of performing memory functions across different scales and topological structures.

According to the memory trace theory, the content of memory is physically the propagation of neural excitation

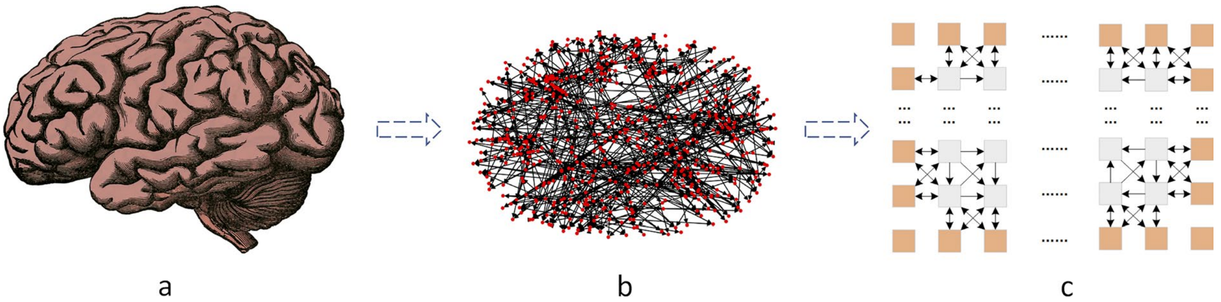


Fig. 2 Abstraction of human brain memory system into a locally connected MAS (**a**. Brain **b**. Connected neurons in the brain **c**. Neurons abstracted by nodes in a directed graph)

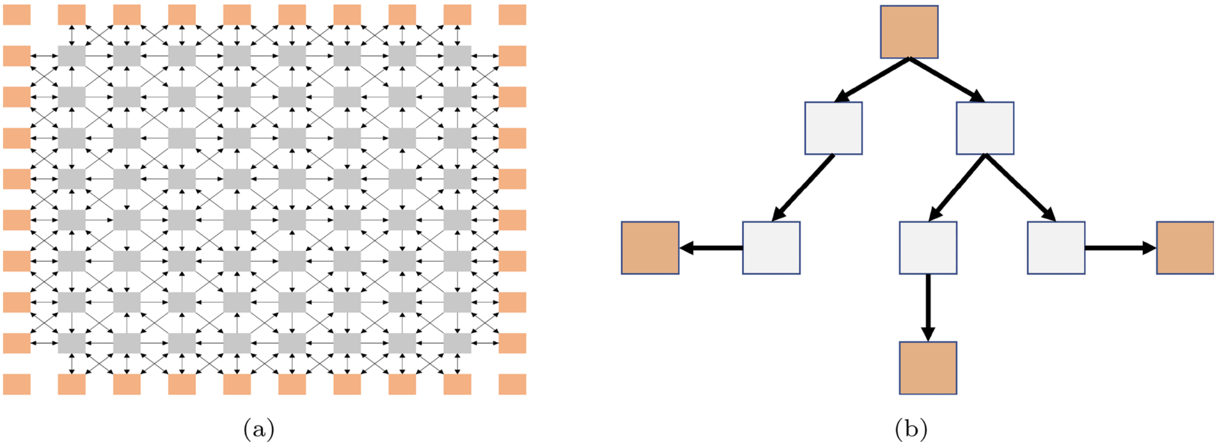


Fig. 3 **a** Abstract modeling of neuronal network as sparsely connected MAS. **b** MAS path modeling representing memory instances

in a biological neural network caused by input stimuli, along with the reinforcement and awakening of propagation traces. The repeated occurrence of input stimuli can strengthen the connectivity between neurals. Inputs correspond to different propagation path traces, meaning that connected paths are specific physical carriers. This process does not require an overarching "God's eye" view and is reasonable within the context of a biological neural network, which is massive and dynamic. Therefore, in our modeling, different paths are formed by arcs in the MAS as the physical implementation of memory. For example, in Fig. 3b, the agents at the top, bottom, left, and right, connected by some intermediate agents and directed connections, form a path that can be considered a memory instance in the MAS.

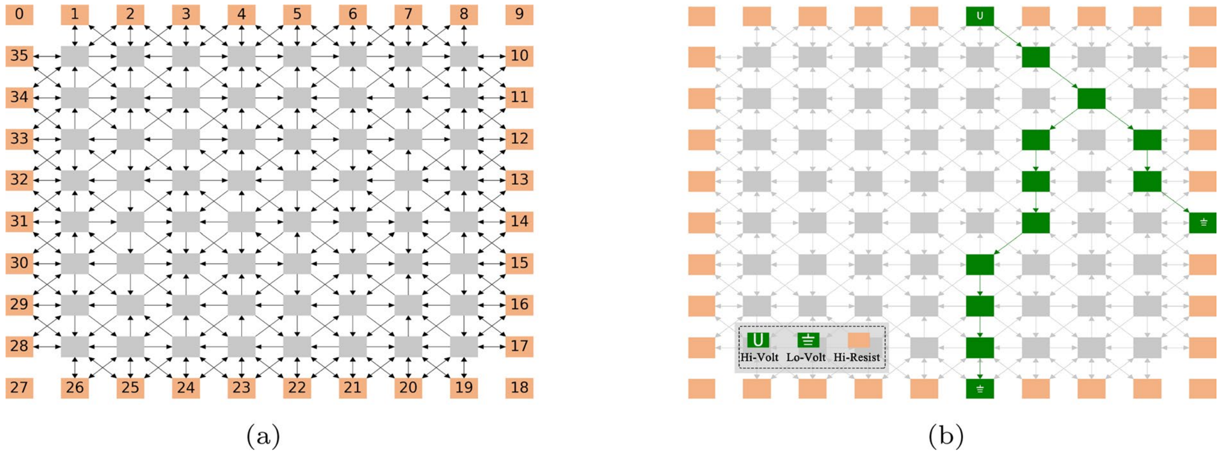
3.2 Formation of memory paths

Consider the implementation process of memory as the formation and consolidation of paths between agents in a MAS network is a natural approximation. Humans perceive external information through multiple senses to understand the characteristics of an object. Neurons, upon receiving stimuli, transmit signals to different downstream neurons in the network, forming various transmission paths. Repeated stimulation solidifies these paths, creating a memory. When similar signals are received again, they are transmitted along these established paths, in a process known as path consolidation. First, we will show formation of memory paths in our model.

Based on this process, our model, as shown in Fig. 3a, classifies agents into edge and internal agents. Edge agents represent the characteristic information of memory instances, serving as inputs and outputs of information. An activated edge agent indicates the presence of a feature. Activated edge agents can be in an input or output state (i.e., the respective starting point and endpoint of signal transmission), while inactivated agents cannot input or output signal. In the neuronal network of the biological cerebral cortex, action potential pulses are transmitted along directed paths. Therefore, in our design, the MAS network is endowed with the attributes of a circuit, using electric field theory to explain the establishment and consolidation of paths. This process is verified through computer simulation.

We model the signal transmission of the neural network as the flow of electric current in a circuit, in which current always flows from a higher to a lower voltage level. The paths in the MAS are abstracted as current transmission paths between agents. Voltage levels are set depending on the states of edge agents. An edge agent in the input state is set to high voltage and in the output state is set to low voltage, which is seen as grounding. Inactive agents are set to a high resistance state, which is equivalent to disconnection. Input edge agents can connect and transmit currents to output edge agents through internal agents. The current enters from an input agent and continuously propagates outward, forming different paths, and the transmission ends when it reaches an output agent or there is no further path. As the current can only be output through output agents, a stable current flow is formed only on the paths that reach the output agents; these interconnected paths form the path traces of an instance on the MAS. Longer paths have higher resistances, so more currents are transmitted through the shortest path, which is the physical realization of memory, i.e., the dominant path of the memory instance or simply the memory instance path.

Figure 4a shows the numbering of each edge agent. A vector can be constructed by connecting the states



(a) (b)

Fig. 4 **a** Numbering of edge agents in MAS. **b** Setting different voltage states for edge agents

of edge agents according to their numbers. This can be seen as a memory instance vector, representing different external stimulus features by altering the vector's bits, forming different interconnected memory instance paths that can be remembered by the MAS. The memory instance vector combines the states of edge agents. Vector bits are changed to construct current transmission paths corresponding to different memory instances, representing their connected paths. Table 2 shows a vector of edge agent states, where the first and second rows show agent numbers and agent states, respectively, where 1 indicates an agent activated in the input state, −1 indicates the output state, and 0 represents inactivation, this vector is a memory instance.

In Fig. 4b, memory instance of Table 2 is uploaded, some edge agents are set to high voltage, some to low voltage, and the rest to a high resistance state. Under the influence of the electric field, positive charges move from high to low potential, forming connected paths through internal agents. Current flows from high-voltage agents to low-voltage agents. A memory instance is distributed in the MAS in the form of current transmission paths. The green path in the graph has the fastest potential drop and is the dominant path of the memory instance.

As can be seen from Fig. 4b, the paths in the MAS network are a limited resource. Therefore, uploading multiple memory instances into one MAS can result in competition for paths and agent resources. When a path is occupied by one memory instance, other instances that also need this path can create resource competition, which can be incompatible and destructive. An ideal solution is to find a compatible path-sharing scheme for multiple memory instances. Then, during retrieval, additional features may be needed, not just the shared path for correct retrieval. This is like the human brain's memory process, where the Hippocampus-Cortical two-tier system temporarily stores information. The hippocampus may undertake batch processing of different memory instances, forming an intensive, path-compatible, multi-instance composite representation to conserve resources, which is later separated and transferred to the cortex for long-term memory [35].

4 Autonomous agent design

4.1 Internal structure design of agents

A single biological neuron cell can differentiate various signal inputs and output different signals to various neighboring neurons. Based on this, we model a single agent of the MAS network, as shown in Fig. 5. There exists a path between each input and output, with diodes and variable resistors on each path. The total resistance value of the variable resistors is fixed, and they have collaborative change rules. Diodes ensure unidirectional current flow. These components enable

Table 2 Vector representation of edge agent state

Edge agent	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Status	0	0	0	0	0	1	0	0	0	0	0	0	0	0	– 1	0	0	0
Edge agent	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Status	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0

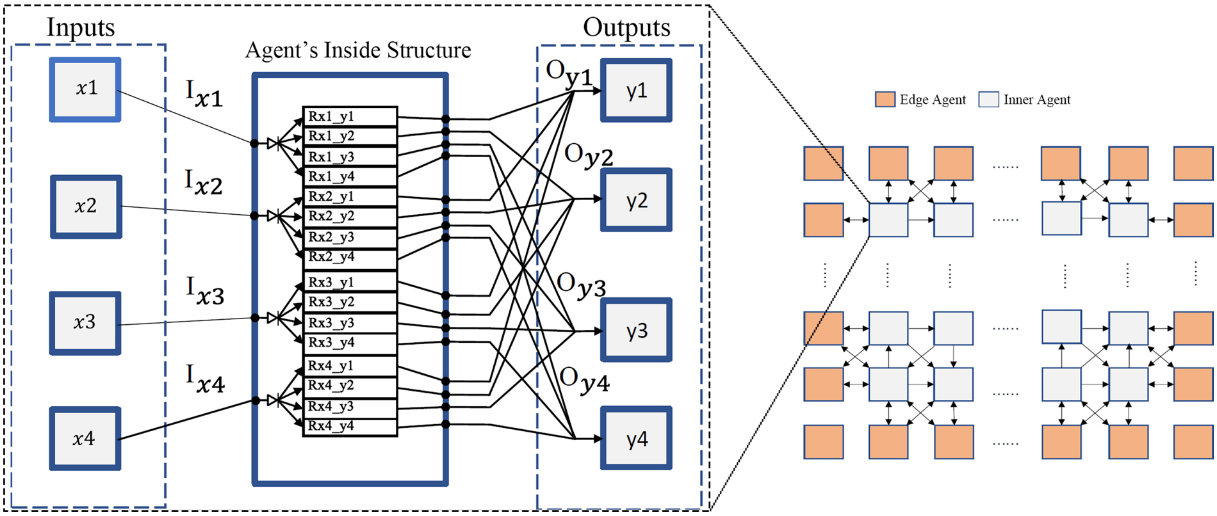


Fig. 5 Modeling of simulation circuit structure inside agent of MAS

path consolidation and resource competition learning between agents.

The input and output of the agent in Fig. 5 is each connected to four agents, where input agents $x1$, $x2$, $x3$, $x4$ have respective incoming current values of $I_{x1}, I_{x2}, I_{x3}, I_{x4}$, and output agents $y1$, $y2$, $y3$, $y4$ have respective outgoing current values of $O_{y1}, O_{y2}, O_{y3}, O_{y4}$. The variable resistors are $R_{x1\_y1}, R_{x1\_y2}, ..., R_{x4\_y4}$, representing resistances on different paths inside agents from input to output. Variable resistors have the function of distributing the output current.

Edge agents input by connecting to an external power source and output by grounding, or they can disconnect. Internal agents receive input and output currents from neighboring agents. Unlike internal agents, the resistors inside edge agents do not have collaborative change rules, and they only function to limit current and prevent short-circuiting.

4.2 Agent's adaptive resource competition learning algorithm

According to the trace theory of memory, a neural network can reinforce recurring signal flow paths to achieve memory. After constructing the memory network through a MAS and agent circuit, converting memory instances into current transmission paths through potential differences, the memory function requires simulation of the process of consolidating the dominant path into the MAS. This relies on the changing rules of variable resistors in each agent. Variable resistors follow changing rules. The sum of the resistances of multiple variable resistors in a single agent remains constant, indicating that the total resource amount is constrained, i.e.,

$$\sum_{i=1}^{m}\sum_{j=1}^{n} R_{xi\_yj} = R_c, \tag{1}$$

where R_c is a constant, and m and n are the respective numbers of input and output paths. Like a rectangle, if the area on the left side increases, the area on the right side decreases. The resistance of variable resistors through which current greater than a threshold value I_t flows gradually decreases. As the total resistance within an agent is constant, the reduced resistance is evenly distributed to the other variable resistors in the same agent,

$$\begin{cases} R_{xi\_yj}(t+1) = R_{xi\_yi}(t) + \Delta r_{xi\_yj}(t) \\ \begin{cases} k_{xi\_yj} = 1, \quad if \quad I_{xi\_yj} > i_t \\ k_{xi\_yj} = 0, \quad otherwise \\ k = \sum_{i=1}^{m}\sum_{j=1}^{n} k_{xi\_yj} \\ C = \sum_{i=1}^{m}\sum_{j=1}^{n} R_{xi\_yj}(t) \times k_{xi\_yj} \times L_r \end{cases} \\ \Delta r_{xi\_yi}(t) = \begin{cases} -R_{xi\_yj}(t) \times L_r, \quad if \quad I_{xi\_yj} > I_t \\ \frac{C}{m\times n-k}, \quad otherwise \end{cases} \\ \sum_{i=1}^{m}\sum_{j=1}^{n} \Delta r_{xi\_yj}(t) = 0 \end{cases} \tag{2}$$

where t is the number of iterations, $R_{xi\_yi}(t)$ is the resistance of the variable resistor between input agent xi and output agent yi at the t^{th} iteration, $\Delta r_{xi\_yi}(t)$ is the resistance change at the t^{th} iteration, I_{xi_yj} is the current between input agent xi and output agent yi, k_{xi_yj} indicates whether I_{xi_yj} exceeds the threshold I_t, k is the total number of variable resistors where current is greater than the threshold, L_r is the proportion of resistance reduction in resistors with current flow after one iteration, L_r can be considered the learning rate, and C is the total value of the reduced resistance.

5 Multi-agent system's memory function

5.1 Memory and retrieval process of current flowing paths

Inspired by the complex regulation mechanisms of biological neuronal signal transmission, each agent in the MAS possesses independent and autonomous path learning capabilities, enabling the network to record transmission paths. The MAS can be divided into two working modes: memory and retrieval. The variable resisters only change in memory mode.

The memory and retrieval of paths simulate the respective biological processes of memory and recall. According to an agent's adaptive learning algorithm, in memory mode, the resistance of variable resistors through which a substantial current continually flows will gradually decrease. Continuously uploading the same memory instance vector to the edge agents will establish stable current transmission paths between agents from high to low voltage. Among the paths within each agent, the path with the fastest potential drop will dominate, and the resistance of variable resistors on this dominant path will gradually decrease through natural competition, while the resistance on other paths increases. This change solidifies the current transmission path traces in the MAS. The longer the current flow, the lower the resistance, making it easier for current to flow, for the path to be more likely activated, and for the stored content to be more profound, reinforcing the path.

In retrieval mode, some bits of the memory instance vector may be lost, such as remembering an object's shape but not its color. When such a partial valid vector, the so-called probe vector, is loaded onto the MAS, only certain edge agents are set to their corresponding states, and lost edge agents are set to low voltage. This assumes the possible existence of uncertain features, and later judgments are made based on the current level, with larger output currents indicating retrieved paths. The input state bits in the probe vector must not be lost; otherwise, no retrieval can occur due to the lack of input. In extreme cases, all bits except those corresponding to the input agents in the probe vector may be lost. From Ohm's Law, sub-paths with smaller resistance receive more current. Since the resistance on the dominant path of the memory instance is relatively low, a larger proportion of current will flow along this path. The longer the duration of the memory process, the less the resistance on the path, and the greater the proportion of current flowing along the dominant path, making selective current flow the basis for path retrieval.

Notably, when more than one memory instance is stored in a MAS and their current flow paths overlap, new stored instances will disrupt the path traces of the original instances solidified in the network, in a process known as retroactive inhibition. In such cases, for accurate retrieval, the number of permissible lost bits in the probe vector decreases, which is like requiring more detailed features to distinguish similar objects. For example, from Fig. 6, it can be seen that, after sequentially uploading green and red instances into the MAS, since the paths overlap (black path), the graph stores the composite paths of the two memory instances.

5.2 Path current calculation based on Kirchhoff's laws

To evaluate the effectiveness of memory, it is necessary to calculate the current flowing through each path. For complex meshed circuits, Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL) can be applied. KVL states that the algebraic sum of the potential differences (voltages) across all elements in a closed loop is zero, and KCL states that the sum of currents entering a node is the sum of currents leaving it.

In the MAS, we know the variable resistors of each path and the input voltage. The positive pole of the input power source is connected to the agents in the input state and the negative pole to the agents in the output state. By applying KVL in the loop from the output-state edge agents to the input-state edge agents and using KCL at the internal agents, a set of equations can be established to solve for the current values in each branch. Agents that are not activated can be considered to have no current flow and can be ignored in the calculations. Taking the current flow paths in the MAS in Fig. 7 as an example, the input and output currents of each agent are numbered. The equations obtained by applying KCL to internal agents are

$$\begin{cases} I_1 - I_3 = 0 \\ I_2 - I_4 - I_5 = 0 \\ I_3 - I_6 = 0 \\ I_4 - I_7 = 0 \\ I_5 - I_8 = 0 \end{cases} \tag{3}$$

and, applying KVL to loops, we obtain

$$\begin{cases} I_1 \times R_1 + I_3 \times R_3 + I_6 \times R_6 - U_{input} = 0 \\ I_2 \times R_2 + I_5 \times R_5 + I_8 \times R_8 - U_{input} = 0 \\ I_2 \times R_2 + I_4 \times R_4 + I_7 \times R_7 - U_{input} = 0 \end{cases} \tag{4}$$

The voltage is calculated using Ohm's Law, $U = IR$, where U_{input} is the input voltage, and R_i is the resistance value of the variable resistor on the corresponding path within the agent for current I_i.

Through the above process, we can evaluate the retrieval effect of memory instances by calculating the current flowing through each path in retrieval mode. A significant current still flowing through the dominant path in retrieval mode indicates successful retrieval. We next discuss the memory and retrieval processes.

6 Simulation experiment

We conducted a series of simulation experiments to investigate the role of MAS in modeling memory systems. Our focus was on exploring the generation, functionality, and capacity of these MASs, simulating the varied microstructural complexities found in biological memory systems. The experiments ranged from initial network generation to comprehensive memory functionality tests, including capacity testing for single and multiple memory instances. We also conducted comparative analyses with other models.

6.1 Generation of MAS network

Considering that the memory systems of biological entities have many microscopic structural differences, for better simulation, we first initialize the MAS, to generate MAS networks of varying scales and connection relationships. For ease of visualization, the MAS is set in a grid format, with its width and depth determining its size and scale. After determining the number of agents based on the side length, the agent matrix is traversed, and connections in random directions are established between adjacent agents to generate MAS networks with different topological structures, as shown in Algorithm 1. Figure 3a shows an example of randomly generated networks with a side length of 10.

Algorithm 1 MAS Generation Algorithm

```
Result: MAS Matrix Represented by Adjacency List
Input: Width and Depth of MAS $x, y$;
Output: MAS Matrix;
Function genAgentMatrix($x, y$):
    Initialization: Agent matrix agentMatrix with width and depth $x, y$. For each
      agent in the matrix, attributes include input agent list inList, output agent list
      outList, bidirectional agent list 2wList;
    for $i \leftarrow 1$ to $x$ do
        for $j \leftarrow 1$ to $y$ do
            The current agent is represented by selfAgent;
            neighbors ← getNeighbor(selfAgent);
            neiLen ← len(neighbors);
            for neighbor ← neighbors[0] to neighbors[neiLen - 1] do
                $xN, yN$ represent the horizontal and vertical coordinates of the agent
                  neighbor in the matrix;
                if the selfAgent and neighbor has not been allocated a direction then
                    if not (isEdgeAgent(selfAgent) and isEdgeAgent(neighbor)) then
                        direct ← Random Boolean value;
                        if direct then
                            agentMatrix[i][j]['inList'].add(neighbor);
                            agentMatrix[xN][yN]['outList'].add(selfAgent);
                        else
                            agentMatrix[i][j]['outList'].add(neighbor);
                            agentMatrix[xN][yN]['inList'].add(selfAgent);
                        end
                    else
                        agentMatrix[i][j]['2wList'].add(neighbor);
                        agentMatrix[xN][yN]['2wList'].add(selfAgent);
                    end
                end
            end
        end
    end
    return agentMatrix;
End Function;
```

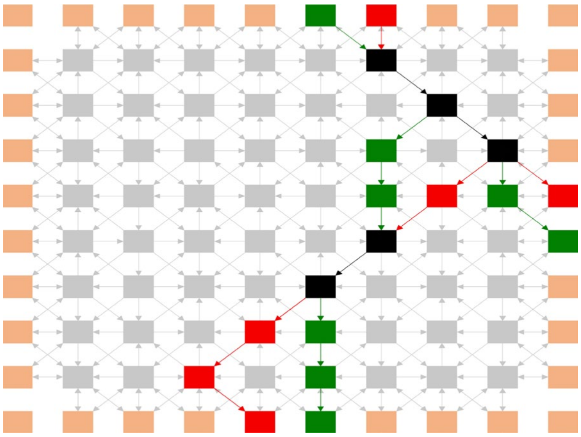


Fig. 6 Distribution of two memory instance paths in MAS

6.2 Assigning activation states to edge agents

In biological neural networks, different stimuli lead to the formation of memories. In our simulation, a memory instance is uploaded through the edge agents of the MAS. For visualization purposes, the top edge of the network is defined as the input edge, where activated agents are assigned a high-voltage state. The others are output edges, with activated agents assigned a grounding state. Agents corresponding to nonzero vector bits are activated. Inactive edge agents are assigned a high-resistance state. Each memory instance is divided into component A, which corresponds to the input edge, and components B, C, and D, corresponding to the three output edges. In the MAS network, electric current flows from A to B, C, and D. Like water, the current always flows downstream, where A acts as the upstream, and B, C, D as the downstream. Figure 8 shows a schematic representation of the correspondence between each component of a memory instance and the MAS.

In memory mode, stable current can only form on the paths between agents in a high-voltage state and grounding state. According to their changing rules, as the current continues to flow, the resistance values of variable resistors on the dominant path with the fastest potential drop decrease continuously. The path is solidified into the MAS through the reduction of resistance values of variable resistors on the path.

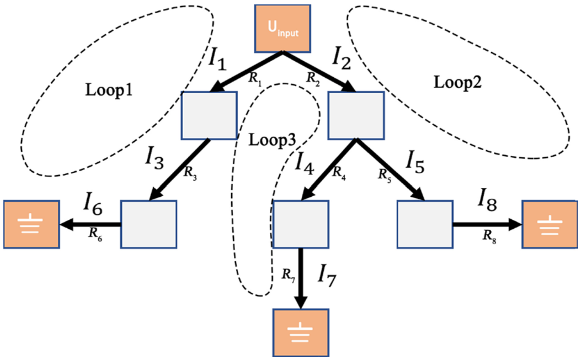


Fig. 7 Currents on multiple connected path branches in MAS

Retrieval has two modes: awakening based on upstream information, and awakening based on both upstream and downstream information. However, in either mode, the edge agent corresponding to component A, serving as the energy source of the MAS, must not be omitted from the probe vector. The difference between the probe vector and the memory instance vector is quantified by the Hamming distance, $d = \rho n$, where n is the length of the vector, and ρ is the proportion of differing bits.

When using awakening based on upstream information, all bits in the probe vector, except for the input bit corresponding to component A, are set to be missing, testing the effect of retrieval in the most extreme case. In this mode, the mapping method of the edge agents corresponding to component A remains unchanged, and all edge agents corresponding to the output state of B, C, and D are grounded to simulate the complete loss of this characteristic information, intending to activate all output paths. Here, ρ reaches its maximum, $\rho_{max} = 1 - \frac{x+f\times(2y+x)}{n}$, where x, y are the respective width and depth of the MAS, and f is the proportion of activated agents in the memory instance.

When multiple instances are stored in the MAS, there is interference between the paths of each instance, including partial sharing, partial destruction, or even breakage. The original paths are no longer complete, and continuing to use awakening based on upstream information may lead to retrieval failure. Therefore, awakening based on both upstream and downstream information is introduced. Compared with awakening based on upstream

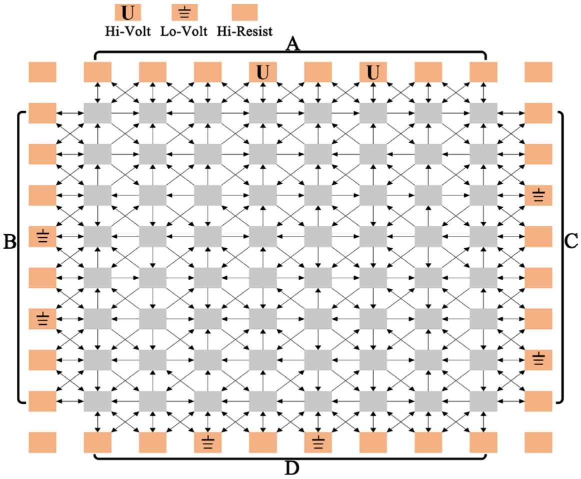


Fig. 8 Correspondence of memory instance vector to edges of MAS

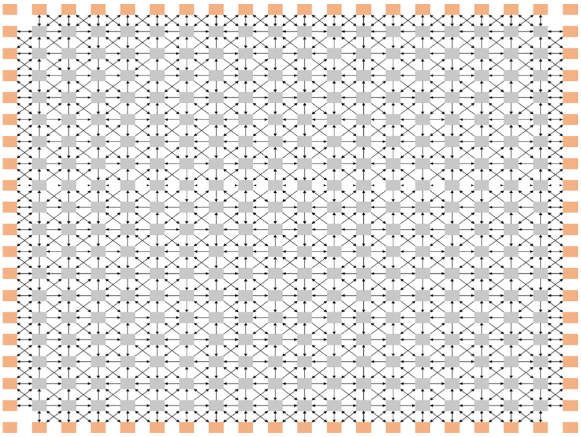


Fig. 9 Randomly generated 20 × 20 MAS where directed connections between agents form the basis for path creation

information, the probe vector in this mode retains more information and has fewer missing bits, i.e., some edge agents corresponding to the output state of components B, C, and D remain in their original high-resistance state. This aims to reduce available output paths, better retrieve the composite paths stored in the MAS, and improve the retrieval effect of each instance path.

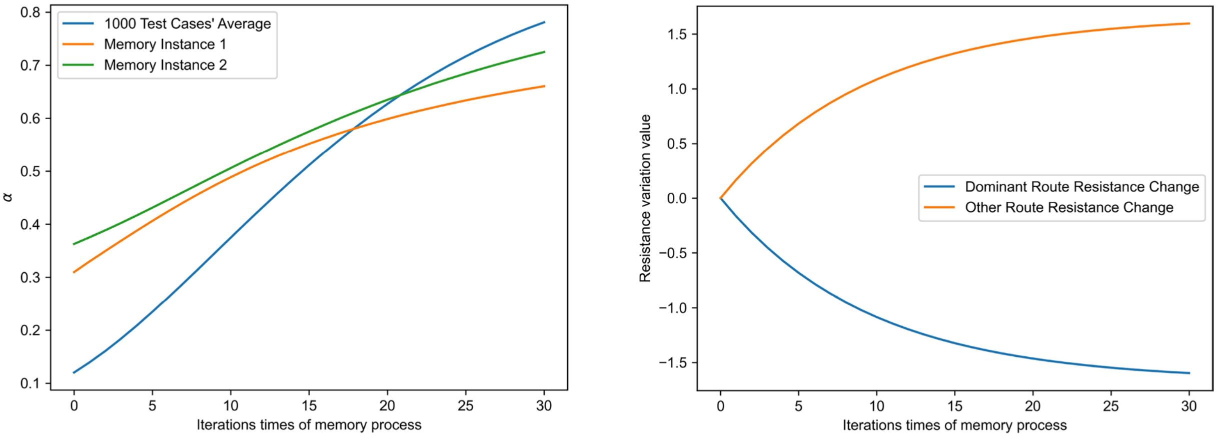
6.3 Memory functionality testing

We tested the memory functionality of the MAS when storing a single instance. A memory instance was loaded into the MAS network to observe if it could be successfully stored. The experiment checked if the current on the connected path of the memory instance in the MAS during upstream information-based awakening was significantly greater than that of other paths. This allowed for a clear distinction between old and new paths. For a MAS with only one memory instance, if the memory is successful, it should be able to accurately restore the original path. Since the resistance values on the original path are relatively small, only the previously reinforced dominant path of the memory instance will maintain a larger proportion of current, while the current on other paths will be smaller. This is an important criterion to differentiate between established and temporary traces.

In this experiment, the learning rate L_r was set at 10%. The memory functionality was tested in a 20 × 20 MAS network, as shown in Fig. 9, which was randomly generated using Algorithm 1. In each test, a memory instance, with 30% of the edge agents randomly activated, was uploaded to the MAS, for 30 iterations of the memory process, like rehearsing to reinforce memory. This learning process iteratively adjusts the resistance values of the variable resistors on connected paths, gradually forming a so-called dominant path with relatively low resistance. After each iteration, the graph was switched to retrieval mode, the probe vector was uploaded, and the output current of each path was calculated. The proportion of the current of the memory instance's dominant path to the total current of all paths is

$$\alpha = \frac{\sum_{i=1}^{k} i_j}{\sum_{j=1}^{m} i_j}, m > k,$$

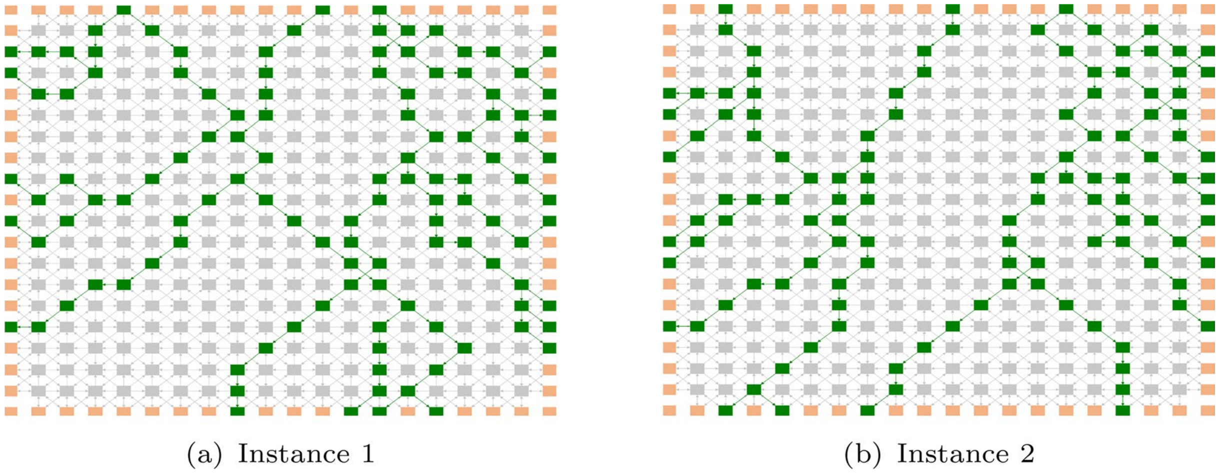
where i_j represents the output current values of each path, m is the total number of paths, and the dominant path consists of the first k paths. At this point, ρ=0.525.



(a) Proportion of Current Output on Dominant Path of Memory Instance

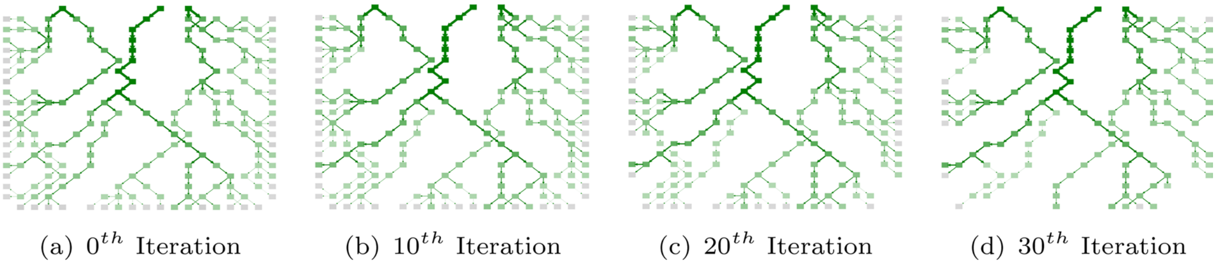
(b) Changes in Variable Resistors within Single Agent

Fig. 10 Changes in MAS metrics at different iteration counts


(a) Instance 1

(b) Instance 2

Fig. 11 Dominant path of memory instance in MAS


(a) 0^{th} Iteration

(b) 10^{th} Iteration

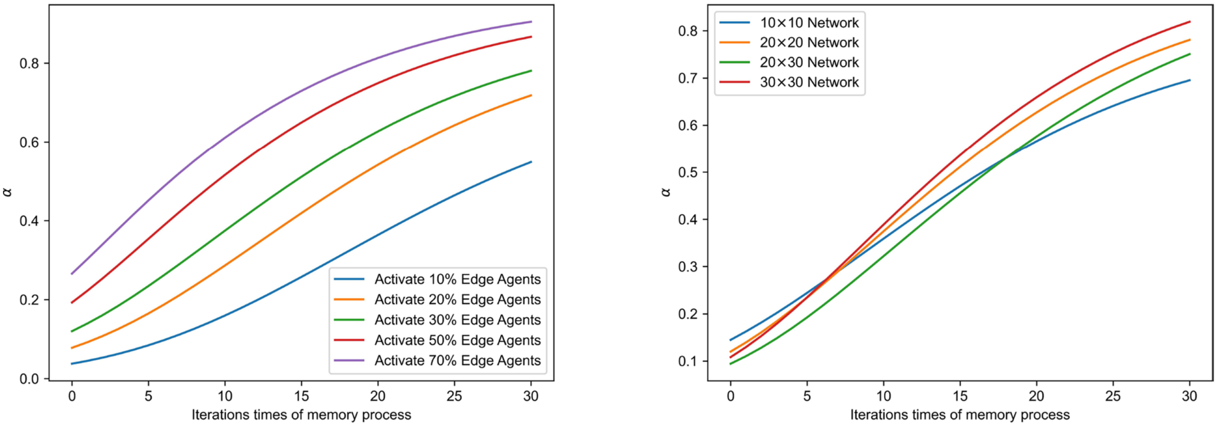
(c) 20^{th} Iteration

(d) 30^{th} Iteration

Fig. 12 Currents of various paths at different numbers of iterations

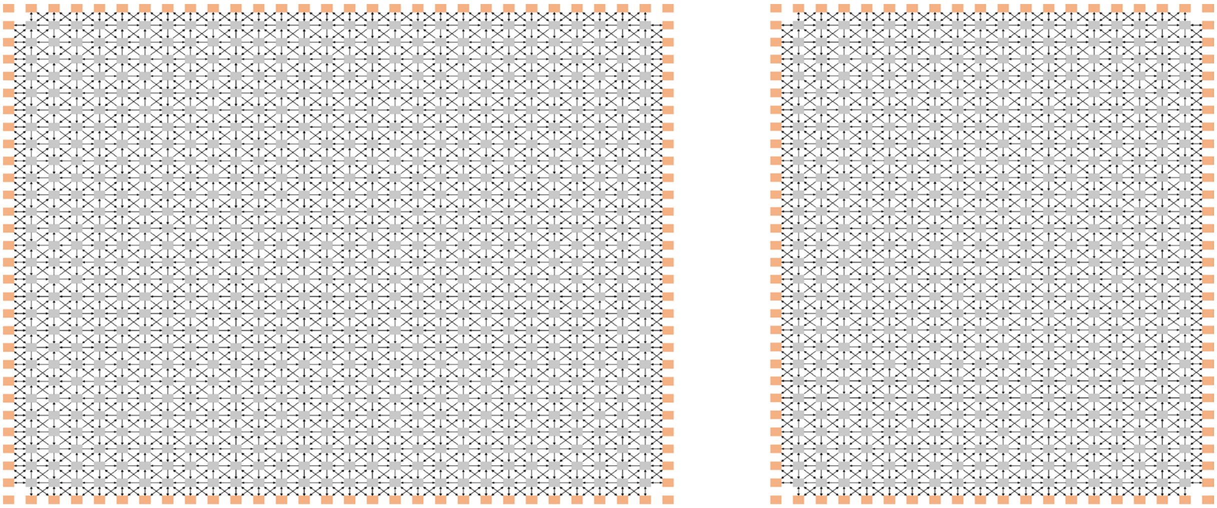
The results, as shown in Fig. 10a, indicate that with increasing iterations, randomly generated memory instances 1 and 2 have differences in the α curve after iterative reinforcement, but both achieve significant improvement, similar to the varying efficiency humans experience when memorizing different content. A random agent was selected to observe the changes in the resistance values of its internal variable resistors, as depicted in Fig. 10b. The resistance values of variable resistors on the dominant path inside the agent gradually decreased, and the reduced resistance was evenly distributed to the variable resistors on the other paths of the same agent, causing the resistance on these paths to increase gradually. The dominant paths of memory instances 1 and 2 in the MAS are shown in Fig. 11a, b. Figure 10a also shows the average value of α from experiments with 1000 test cases, which showed an increase after iteration. This demonstrates that the memory function of the MAS operates well. With different memory instances, after a certain number of iterations, the current flowing through the original dominant path in retrieval mode is significantly greater than that of other paths, solidifying this path as the representation of the memory instance, thereby realizing the memory function. The experimental results indicate that memory instances 1 and 2 require a similar number of iterations to achieve the same memory effect. Random testing on another 1000 examples yielded similar results, indicating that the MAS and path-finding algorithms are not dependent on specific examples and are universally applicable to a wide variety of content.

Taking memory instance 2 as an example, the currents in various paths were calculated and visualized at the 0th, 10th, 20th, and 30th iterations of memory, visually demonstrating the process by which the dominant path gradually becomes more prominent among many paths with increasing iterations, as shown in Fig. 12, where the thickness of paths and opacity of agents represent the magnitude of the current. Thicker paths and lower opacity indicate greater current. Green paths have the fastest potential drop from the activated input agents to each connected output agent. Paths with a current less than 0.4 mA are not displayed. Green edge agents are activated output agents corresponding to the memory instance, and gray agents are inactivated output agents. It can be observed that as iterations proceed, the current on the dominant path continuously increases, and the path is gradually solidified into the graph.



(a) (b)

Fig. 13 **a** Proportion of current output on dominant paths under different agent activation ratios. **b** MASs of different sizes and topological connections demonstrate similar learning capabilities, indicating universality



(a) 30 × 30 (b) 20 × 30

Fig. 14 MASs randomly generated with significant variations in the number of agents and connectivity relationships

The activation proportion of edge agents affects the number of MAS resources occupied by a memory instance. A higher activation proportion may lead to the occupation of more paths, and the dominant path may be more expansive. We conducted experiments on memory instances with different activation proportions of edge agents. The experimental results are the average value of α with 1000 test cases, as shown in Fig. 13a, indicating that memory instances with higher proportions have higher proportions of current output α on their dominant paths after memory. This may be because such instances can access more path resources for output during retrieval. At the same time, according to the definition of ρ_{max}, the higher the proportion of activated agents, the smaller ρ_{max} becomes, reducing the difference between the probe vector and the original vector.

Neural networks in different people's brains vary in terms of the number of neurons, topological connections, and other microscopic details. Despite these numerous and fine structural differences, they do not seem to affect memory function, indicating universality. To verify the effectiveness of the memory function of the MAS network under different topological structures and sizes, Algorithm 1 was used to generate randomly topologized MASs of sizes 10 × 10, 30 × 30, and 20 × 30, as illustrated in Figs. 3a, 14a, and 14b, respectively.

Using the parameters from Fig. 10, memory experiments were conducted in these MASs, and the average α was calculated from 1000 test cases. The results, as shown in Fig. 13b, indicate that despite the different sizes and topologies of the three MASs, they all achieve improvements after iterations, meaning they all can perform the memory function.

The results show that as the number of iterations in the memory process increases, the current flowing through the dominant path of the memory instance gradually increases. This indicates that the path is progressively solidified into the MAS after the sample is repeatedly uploaded. The learning algorithm may exhibit slight differences in memory effectiveness in MAS networks of different sizes and topological structures, but it can generally achieve memory of path traces. This demonstrates that MASs generally possess memory functionality when combined with path representation and path solidification algorithms.

6.4 Incremental memory and retrieval testing of multiple memory instances

Both the human brain and computer memory devices contain a large amount of varied content, and their memory mechanisms must ensure compatibility. In computers, file management systems ensure that different files exclusively occupy different spaces, making compatibility issues relatively simple. However, in MASs, there is an issue of sharing small units. When more than one memory instance is stored in the same MAS, the dominant paths of different memory instances will likely partially overlap. Subsequent stored memory instances will interfere with and potentially disrupt paths already stored due to resource competition, a phenomenon known as retroactive inhibition.

We conducted a two-part experiment in a 20 × 20 MAS, as shown in Fig. 9. First, testing was done using awakening based on upstream information, followed by tests using probe vectors with different ρ values for awakening based on both upstream and downstream information. Memory instances with 30% of edge agents randomly activated were generated and incrementally stored in the same MAS in a certain order. The learning rate L_r was set at 10%, and each instance was iterated 30 times in the memory process.

In the retrieval of each instance, the proportion of the output current of each branch on the dominant path to the total output current on the dominant path is calculated as

$$\beta_o = \frac{i_o}{\sum_{j=1}^{k} i_j}, 1 \leq o \leq k.$$

The primary evaluation metric was α, while secondary metrics included the similarity of $\beta_1 - \beta_k$ at this point to its value when the instance was stored alone in the same MAS for the same number of iterations. Similarity was quantified using cosine similarity; the values of $\beta_1 - \beta_k$ were arranged in the same order to form vectors, and the cosine similarity between them was calculated. This was used to assess the deviation in the output balance of each branch. These β values form a vector,

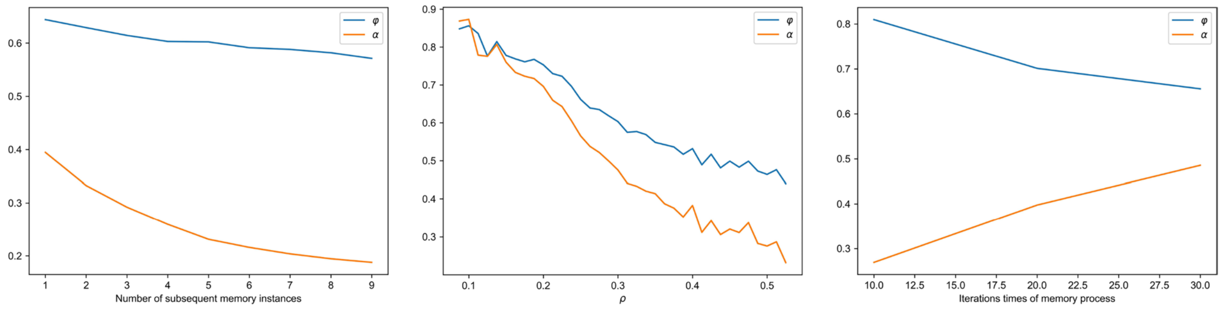
$$V = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_k].$$

The cosine similarity is

$$\varphi = \frac{V_{pre} \cdot V_{curr}}{\| V_{pre} \| \| V_{curr} \|},$$

where V_{pre} and V_{curr} refer to the vectors of β values from the previous (stored alone) and current (with other instances stored) retrieval scenarios, respectively.

In the same MAS, 10 memory instances were incrementally stored, and after completing the memory process of each instance, retrieval based on upstream



(a) (b) (c)

Fig. 15 **a** Retrieval effectiveness with different subsequent memory instance counts. **b** Impact of probe vectors with different ρ values on retrieval effectiveness. **c** Different iteration counts in memory process influence retrieval effects information was conducted for the stored instances. The experiment was repeated 1000 times with different test cases, and the results were averaged. As shown in Fig. 15a, the retrieval effectiveness significantly decreased compared with when stored alone, as discussed earlier. The number of subsequent instances (reflecting the original position of the extracted instance) showed a negative correlation with both α and φ, indicating that the memory of new instances continuously interferes with paths already stored in the graph, but the impact on φ is relatively minor.

The results of the experiment in Fig. 15a also indicate that the MAS retains the ability to distinguish whether a particular instance has been stored before, even after storing 10 instances. During retrieval, α remains higher than the average value shown in Fig. 10a without iterative memory, indicating that the current still preferentially flows through the previously established paths. This is akin to recognizing a person whose name we cannot recall.

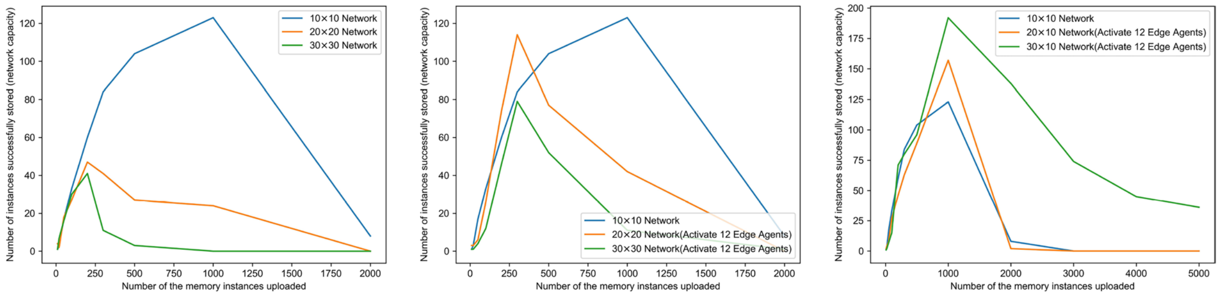
Testing based on both upstream and downstream information was conducted after all 10 memory instances were stored in each experiment. Probe vectors with different ρ values were uploaded to test the retrieval effectiveness. The experiment was repeated 1000 times with different test cases, and the results were averaged. As shown in Fig. 15b, the retrieval effectiveness gradually improves as ρ decreases, with both metrics showing significant improvement. The results are markedly better than those from awakening based on upstream information alone. This is like a human better recalling things after receiving some hints.

From the previous experiments and the definition of the memory learning algorithm, it is known that the number of memory iterations and the learning rate affect the proportion of the dominant path output during retrieval. In an incremental memory process, the number of iterations for subsequent instances can also be considered an indicator of the degree of interference of preceding instances on solidified paths. To test the impact of the number of memory process iterations on the retrieval effectiveness of multiple memory instances, two memory instances with 30% of edge agents activated were randomly generated and uploaded to a 20 × 20 network, with the learning rate L_r maintained at 10%. The number of iterations in the memory process was set to 10, 20, and 30. The first and second instances stored were the disturbed and disturbing instance, respectively. After memory, the disturbed instance was subjected to upstream information-based awakening retrieval testing. The experiment was repeated 1000 times under different iteration counts, and the results were averaged. The results, as shown in Fig. 15c, indicate that the number of iterations in the memory process positively correlates with α, and negatively with φ, which is consistent with theoretical analysis. The increase in α is due to the deeper solidification of the paths of the preceding instance, while the decrease in φ is due to increased interference from subsequent memory instances on the preceding instance.

In summary, the experiments observed the retroactive inhibition phenomenon in the graph and verified the impact of probe vectors with different ρ values on retrieval effectiveness through awakening based on both upstream and downstream information after storing multiple instances. ρ is negatively correlated with retrieval effectiveness, confirming the influence of the number of iterations in the memory process on retrieval effectiveness.

6.5 Capacity testing

As previously mentioned, different instances stored in the MAS interfere with each other to some degree, affecting their retrieval. In the capacity testing experiment, a certain number of memory instances were randomly generated, and all were subsequently stored in the same MAS. Each stored instance was then individually



(a) (b) (c)

Fig. 16 **a** Capacity measured by individually retrieving instances after storing different quantities. **b** Capacity measured after changing number of activated agents in memory instances. **c** Capacity measured with consistent network depth

Table 3 Attributes of dominant paths of memory instances uploaded to networks of different depths

Size of MAS network	Number of connections between agents in connected subgraph of instance	Average number of agents constituting path
30 × 10	364.459	54.439
30 × 20	813.775	83.721
30 × 30	1385.789	113.686

Table 4 Capacity of our method and hopfield network

Our method		Hopfield	
Size	Measured capacity	Size	Measured capacity
30 × 10	108	300	20
30 × 20	32	600	22
30 × 30	15	900	25

retrieved, and α was calculated. Memory was considered successful if $\alpha \geq 90\%$. Retrieval was conducted using the awakening method based on upstream information, which reflects the number of instances that can be successfully and completely stored in a MAS after storing multiple instances.

The experiment was conducted in 10 × 10, 20 × 20, and 30 × 30 networks. Several sets of memory instances with 30% activated edge agents were randomly generated, with varying numbers in each set. The instances were incrementally stored in the same MAS, group by group. The learning rate L_r was set at 10%, and each instance underwent 30 iterations of the memory process. The results, as shown in Fig. 16a, indicates that the memory capacity of MAS networks is limited, and larger networks are not necessarily better; smaller-scale networks seem to have a larger capacity. For the same network, more samples can be stored when internal resource competition is not intense. Once the memory limit is reached, storing more samples can disrupt the paths occupied by previous instances, leading to retrieval failure of instances that were originally remembered.

We conducted further testing regarding the issue of larger MAS networks having smaller capacities. To control variables, the number of activated edge agents in memory instances randomly generated for 20 × 20 and 30 × 30 networks was 12 (10×4×30%) in both cases, and experiments were conducted in these networks with the same parameters as in Fig. 16a. The results, as shown in Fig. 16b, indicate that reducing the number of activated agents can increase the memory capacity of MASs. However, larger networks still have a lesser capacity than smaller-scale networks.

In capacity tests based on square MASs, the phenomenon of larger network sizes corresponding to smaller capacities was observed. A possible reason for this is that the square network topology, with equal length and width, is not conducive to resource conservation, as the paths formed tend to be deeper. Therefore, we tested networks with a consistent depth but varying widths. Capacity tests were conducted in 20 × 10 and 30 × 10 networks, with experimental parameters consistent with those in Fig. 16b. The results, as shown in Fig. 16c, indicate that in networks with consistent depth, larger network sizes correspond to greater capacities. The cerebral cortex has a similar structure, with consistent depth across different areas (six layers of cells), but also has the ability to laterally expand, forming a flat structure.

A total of 1000 memory instances with 12 activated edge agents were randomly generated and uploaded to 30 × 10, 30 × 20, and 30 × 30 MASs. The average numbers of agents and agent connections in the paths formed were calculated, to compare the resource consumption of networks with different depths. The experimental results are shown in Table 3, and indicate that, with the same width, networks with smaller depths consume fewer resources per instance, forming paths with fewer agents. This explains why the MAS networks with smaller depths performed better in the earlier experiments. A related biological fact is that the cerebral cortex has relatively few layers, forming a flat structure that is not deep, but that has extensive horizontal expansion capabilities.

6.6 Capacity comparison

We compare the capacity of the proposed model with that of the Hopfield network measured in experiments. Like the criterion of $\alpha \geq 90\%$ set in this paper, in the Hopfield network, a retrieved vector is considered successfully remembered if its similarity with the original vector exceeds 90%. The memory efficiency of the Hopfield network is optimal when each binary bit of a memory instance is independently generated with a 50% probability of being one of two values [36], which corresponds to randomly activating 50% of the edge agents in the proposed model. To control variables, other parameter settings remained the same as in subsection 6.5, but we randomly activated 50% of the edge agents. At the same time, ρ for the probe vector used in the Hopfield network was the same as ρ_{max}.

The experimental results, as shown in Table 4, are similar to our earlier conclusions. The proposed model shows higher memory efficiency in networks with smaller depths and outperforms the Hopfield network in capacity. However, as the depth increases, the capacity gradually becomes less than that of the Hopfield network. Furthermore, in the experiments, the capacity of

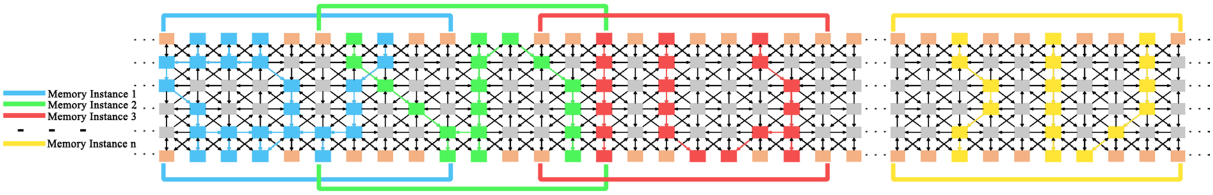


Fig. 17 Compactly and randomly uploading memory instances to different positions on MAS

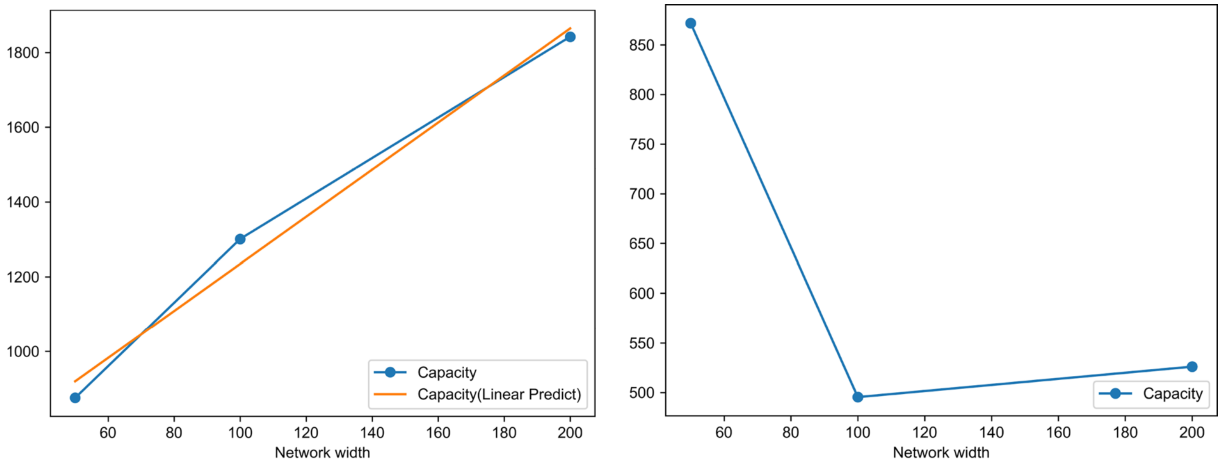
the Hopfield network dropped rapidly when the binary bits of the memory instance were unevenly distributed, which corresponds to a decrease in the activation ratio in the proposed method. As mentioned above, reducing the activation ratio in our method can increase the memory capacity. Therefore, our model can flexibly adjust memory capacity by varying the proportion of activated agents, whereas the Hopfield network cannot. In addition, compared with the fully connected network of the Hopfield model, our model's sparsely connected network consumes fewer resources and achieves greater memory capacity, especially in networks with smaller depths. As the number of connections in a fully connected network grows quadratically, a 30 × 30 Hopfield network contains $900 \times 899 = 809100$ connections, way higher compared with the one in Table 3.

However, our method of retrieval is based on extracting downstream agents from upstream agents, and it cannot retrieve information represented by downstream agents if upstream agents are lost, meaning it cannot extract upstream agents based on downstream agents. Meanwhile, the Hopfield network does not differentiate between upstream and downstream agents.

6.7 Capacity testing of networks with six layers of depth

The cerebral cortex of higher primates has a vertical depth of six layers of cells and the capability for lateral expansion. Mimicking this form of the brain's cortex, the depth of the MAS network was set to six layers. In this configuration, the memory instance retains components A and D, with no B or C. Component A is used for input and D for output, consistent with previous capacity testing experiments, and retrieval is based on upstream information. To control variables, the number of bits in the memory instance vectors was maintained at 20 bits (10 bits each for A and D), with an activation ratio of 50%. Random starting positions were selected on the top edge of the MAS, with component A continuously uploaded from the starting position and D uploaded to the symmetric position on the bottom edge, as illustrated in Fig. 17. In this case, the memory instances are said to be compactly distributed on the MAS, with each instance being uploaded on consecutive edge agents of the graph.

The experiment was conducted in networks of sizes 50 × 6, 100 × 6, and 200 × 6. Several groups of memory instances were randomly generated, each with varying numbers of instances. Instances were incrementally stored in the same MAS, group by group. The learning rate L_r was set at 10%, and each instance underwent 30



(a) Compactly Distribution

(b) Dispersed Distribution

Fig. 18 Capacity of MAS networks with six layers of depth and different widths

Table 5 Resource occupation of single instances in 6-layer depth MASs with compact and dispersed distribution of different widths

Network width	Number of connections between agents in connected subgraph of instance	Average number of agents constituting path
Compact distribution		
50	50.174	35.735
100	50.171	35.934
200	50.957	35.447
Dispersed distribution		
50	63.042	53.373
100	90.687	73.097
200	131.151	101.649

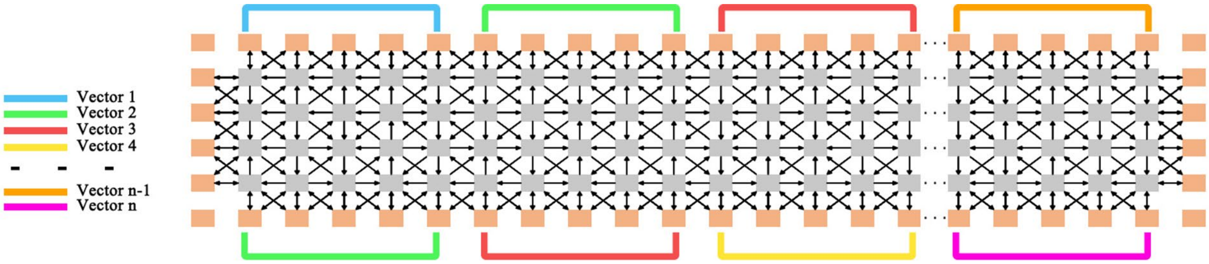


Fig. 19 Uploading pairs of vectors as memory instances to non-overlapping positions on MAS

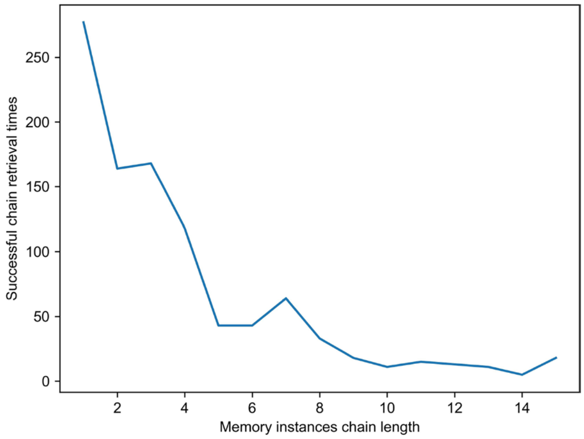


Fig. 20 Distribution of chain lengths in chain awakening across 1000 experiments

iterations of the memory process. The capacity testing rules were the same as in subsection 6.5, and the test results were as shown in Fig. 18a, indicating that as the network width increases, its capacity gradually improves, roughly linearly.

We also experimentally tested the memory capacity of the MAS when instances were uploaded in a dispersed distribution. In this case, A and D components were no longer continuously uploaded from a certain position to the top and bottom edges. Instead, bits of these components were randomly mapped to different positions on the top and bottom edges. The corresponding agents of each memory instance were no longer concentrated in a continuous area, to test the memory capacity of the MAS when instances were stored randomly.

The experimental results of the dispersed distribution, as shown in Fig. 18b, indicate that except for the network with a width of 50, whose capacity remained largely unchanged compared with the compact distribution, the capacities of the other MAS networks showed a significant decrease and almost stopped increasing as the network widened.

To explore the reasons for this phenomenon, the resource occupation of individual instances was analyzed. The results, as shown in Table 5, indicate that in the case of a compact distribution, the resource occupation of individual instances does not increase as the network widens. In a dispersed distribution, the horizontal distance between activated agents is typically greater, requiring more resources to connect, leading to a linear increase in resource occupation per instance with increasing width. The resource occupation in a dispersed distribution at a width of 50 was still relatively close to that of a compact distribution, which may explain why the network capacity in a dispersed distribution remains close to that of a compact distribution at this width. Subsequently, as the gap in resource occupation widens and shows linear growth, the capacity becomes significantly lower than in a compact distribution and almost stops growing with the widening of the network.

6.8 Chain awakening experiment with long chains

Many brain activities, such as sequential working memory, are related to temporal information [37]. A long-chain sequential awakening experiment examined a series of instances with sequential associations, where the output component of a memory instance is the input for the following instance, i.e., it acts as an awakening cue. The experiment observed whether such a chain-style awakening can be successful. Multiple vectors of the same length were randomly generated and combined in pairs to form memory instances, with the output vector (D component) of one instance serving as the input vector (A component) of the next. The method of uploading memory instances is similar to Fig. 17 and illustrated in Fig. 19, the edge agents used by each instance do not overlap, indicating that the instances do not share input or output agents.

The experiment was conducted in a 200 × 6 network. Sixteen vectors of length 10 were randomly generated, with adjacent vectors paired to form memory instances, totaling 15 pairs. In this setup, the output of one pair of instances served as the awakening cue for the next pair. The instances were incrementally stored in the same MAS in sequential order, with the learning rate L_r set at 10%, and each instance underwent 30 iterations of the memory process. After memory, each instance was sequentially retrieved in order. Retrieval of the next instance proceeded only if the previous retrieval was successful; otherwise, the retrieval chain was broken, and subsequent instances were not retrieved. In this experiment, $\alpha \geq 80\%$ was considered successful memory, and other capacity testing rules were the same as in subsection 6.5. The experiment was repeated 1000 times. The test results are shown in Fig. 20 and indicate that the model can successfully chain-awaken sequentially stored memory instances, but the probability of successful awakening decreases for instances closer to the end of the chain. This is because the failure to retrieve any instance leads to the failure of all subsequent instances in the chain.

7 Conclusion

We modeled the memory mechanism of the cerebral cortex by leveraging the known principles of synaptic plasticity and the behavior of engram cells, which have been extensively studied in neurobiology. Specifically, the model simulates the strengthening of synaptic connections based on Hebbian plasticity and the dynamics of neuron activation patterns as described in the engram theory of memory storage. Additionally, the electrophysiological principles guiding our model are grounded in the propagation of action potentials and synaptic transmission laws, such as Ohm's and Kirchhoff's laws, which have been validated through computer simulations. These simulations align with empirical observations of neuron activation and memory recall dynamics in biological neural networks, such as those demonstrated in Kitamura et al.'s work on memory engram cells [14].

The MAS model in the algorithm is active, with each agent capable of personalized adaptive learning based on local neighborhood information, modeling the memory system of biological neural networks. This breaks away from the reliance of the traditional graph model on a global view for operation, lacking autonomous parallel distributed processing capabilities, and is closer to actual biological neural networks. Inspired by the capability of biological neurons to transmit electrical signals, this adaptive learning behavior was simulated through microcircuits centered on variable resistors, successfully realizing the simulation of the memory and retrieval processes in the entire MAS network on a computer. In simulations, the model could distribute memory instances across the MAS, transforming them into connected paths, and achieving path memory and retrieval within the network. The model's generalizability was verified, and it was shown to be capable of achieving memory functions in different topological structures of MASs. Tests determine the capacity of networks of various scales, verifying factors influencing capacity size, as network scales, edge agent activation ratios, and dispersion of memory instances all affect network capacity. Particularly, referencing the structure of the cerebral cortex in higher primates, the memory capabilities of networks with six layers of depth were verified, exploring the relationship between the capacity of six-layer networks and their width.

The process of memorizing and retrieving instances simulates the biological processes of memory and recall. Experimental results showed certain similarities between the model and biological neural networks. For example, the number of iterations required for a MAS network to memory different instances and achieve the same retrieval effectiveness varies, as does the time required for humans to remember different content. Furthermore, our model universally possesses memory functions in MASs of different sizes and topological structures, similar to how the microscopic differences in people's brain networks do not affect the effectiveness of memory functions. Our model performed better and had a larger capacity in networks with smaller depths, with capacity increasing linearly with width, resembling the flat morphology and horizontally expansive cortical structure of the brains of higher primates.

In summary, we proposed a self-learning MAS model that does not require a global perspective, based on the simulation of biological neural networks. The MAS achieves memory functionality while decentralized, based on local information and adaptive learning capabilities. The model demonstrates biological plausibility by aligning its simulated memory processes with known neural activity patterns, the activation of engram cells during memory recall. It also serves as an inspiration for further research into memory neural mechanisms, enriches neural computational models, and offers new perspectives and ideas for neural computation research.

Author contributions

Hui Wei carried out conceptulization, algorithm， analysis and supervisor, funding, experiment design and writing the paper. Chenyue Feng coded simulation program and collected experimental data, wrote the paper. Fushun Li wrote the paper. All authors read and approved the final manuscript.

Funding

This study was funded by National Natural Science Foundation of China (Grant No. 61771146).

Data availability

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable to this study.

Competing interests

The authors declare that they have no competing interests.

Received: 3 June 2024 Accepted: 2 September 2024
Published online: 14 September 2024

References

1. Chaudhuri R, Fiete I (2016) Computational principles of memory. Nat Neurosci 19(3):394–403. https://doi.org/10.1038/nn.4237
2. Abraham WC, Jones OD, Glanzman DL (2019) Is plasticity of synapses the mechanism of long-term memory storage? NPJ Sci Learn 4(1):9. https://doi.org/10.1038/s41539-019-0048-y
3. Humeau Y, Choquet D (2019) The next generation of approaches to investigate the link between synaptic plasticity and learning. Nat Neurosci 22(10):1536–1543. https://doi.org/10.1038/s41593-019-0480-6
4. Jeong Y, Cho H-Y, Kim M, Oh J-P, Kang MS, Yoo M, Lee H-S, Han J-H (2021) Synaptic plasticity-dependent competition rule influences memory formation. Nat Commun 12(1):3915. https://doi.org/10.1038/s41467-021-24269-4
5. Martin SJ, Grimwood PD, Morris RG (2000) Synaptic plasticity and memory: an evaluation of the hypothesis. Annu Rev Neurosci 23(1):649–711. https://doi.org/10.1146/annurev.neuro.23.1.649
6. De Rossi P, Nomura T, Andrew RJ, Masse NY, Sampathkumar V, Musial TF, Sudwarts A, Recupero AJ, Le Metayer T, Hansen MT et al (2020) Neuronal BIN1 regulates presynaptic neurotransmitter release and memory consolidation. Cell Rep 30(10):3520–3535. https://doi.org/10.1016/j.celrep.2020.02.026
7. Feld GB, Born J (2020) Neurochemical mechanisms for memory processing during sleep: Basic findings in humans and neuropsychiatric implications. Neuropsychopharmacology 45(1):31–44. https://doi.org/10.1038/s41386-019-0490-9
8. Otero TF (2022) Exploring brain information storage/reading for neuronal connectivity using macromolecular electrochemical sensing motors. Adv Intell Syst 4(1):2100058. https://doi.org/10.1002/aisy.202100058
9. Wang Y, Gu C, Ewing AG (2022) Single-vesicle electrochemistry following repetitive stimulation reveals a mechanism for plasticity changes with iron deficiency. Angew Chem Int Ed 61(20):e202200716. https://doi.org/10.1002/anie.202200716
10. Kriegeskorte N, Douglas PK (2018) Cognitive computational neuroscience. Nat Neurosci 21(9):1148–1160. https://doi.org/10.1038/s41593-018-0210-5
11. Josselyn SA, Tonegawa S (2020) Memory engrams: recalling the past and imagining the future. Science 367(6473):eaaw4325. https://doi.org/10.1126/science.aaw4325
12. Semon RW (1911) Die mneme als erhaltendes prinzip im wechsel des organischen geschehens. Engelmann
13. Hebb DO (2005) The organization of behavior: a neuropsychological theory. Psychology Press
14. Kitamura T, Ogawa SK, Roy DS, Okuyama T, Morrissey MD, Smith LM, Redondo RL, Tonegawa S (2017) Engrams and circuits crucial for systems consolidation of a memory. Science 356(6333):73–78
15. Tonegawa S, Liu X, Ramirez S, Redondo R (2015) Memory engram cells have come of age. Neuron 87(5):918–931
16. Gao S (2009) Graph theory and network flow theory. Higher Education Press
17. Cook WJ, Applegate DL, Bixby RE, Chvatal V (2007) The traveling salesman problem: a computational study. Princeton University Press, Princeton. https://doi.org/10.1515/9781400841103
18. Ji S, Pan S, Cambria E, Marttinen P, Philip SY (2021) A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans Neural Netw Learn Syst 33(2):494–514. https://doi.org/10.1109/TNNLS.2021.3070843
19. Jouili S, Vansteenberghe V (2013) An empirical comparison of graph databases. Int Conf Soc Comput 2013:708–715. https://doi.org/10.1109/SocialCom.2013.106
20. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79(8):2554–2558. https://doi.org/10.1073/pnas.79.8.2554
21. Kobayashi M (2017) Chaotic pseudo-orthogonalized Hopfield associative memory. Neurocomputing 241:147–151. https://doi.org/10.1016/j.neucom.2017.02.037
22. Krotov D, Hopfield JJ (2016) Dense associative memory for pattern recognition. Advances in neural information processing systems. 29. http://papers.neurips.cc/paper/6121-dense-associative-memory-for-pattern-recognition.pdf
23. Mungai PK, Huang R (2017) Chunking mechanisms for a self improving associative memory model. IEEE Sympos Ser Comput Intell (SSCI) 2017:1–6. https://doi.org/10.1109/SSCI.2017.8285215
24. Mungai PK, Huang R, Chen Z, Zhou X (2017) Semantic Neuron Networks Based Associative Memory Model. In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 1–8. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.18
25. Oku M, Makino T, Aihara K (2013) Pseudo-orthogonalization of memory patterns for associative memory. IEEE Trans Neural Netw Learn Syst 24(11):1877–1887. https://doi.org/10.1109/TNNLS.2013.2268542
26. Shriwas R, Joshi P, Ladwani VM, Ramasubramanian V (2019) Multi-modal associative storage and retrieval using Hopfield auto-associative memory network. Int Conf Artif Neural Netw 57–75. https://doi.org/10.1007/978-3-030-30487-4_5

27. Kosko B (1988) Bidirectional associative memories. IEEE Trans Syst Man Cybern 18(1):49–60. https://doi.org/10.1109/21.87054
28. Cholet S, Paugam-Moisy H, Regis S (2019) Bidirectional associative memory for multimodal fusion: a depression evaluation case study. Int Jt Conf Neural Netw (IJCNN) 2019:1–6. https://doi.org/10.1109/IJCNN.2019.8852089
29. Kosko B (2021) Bidirectional associative memories: unsupervised Hebbian learning to bidirectional backpropagation. IEEE Trans Syst Man Cybernet Syst 51(1):103–115. https://doi.org/10.1109/TSMC.2020.3043249
30. Singh MP, Saraswat V (2017) Multilayer feed forward neural networks for non-linear continuous bidirectional associative memory. Appl Soft Comput 61:700–713. https://doi.org/10.1016/j.asoc.2017.08.026
31. Zhao Y, Ren S, Kurths J (2021) Synchronization of coupled memristive competitive BAM neural networks with different time scales. Neurocomputing 427:110–117. https://doi.org/10.1016/j.neucom.2020.11.023
32. Dorri A, Kanhere SS, Jurdak R (2018) Multi-agent systems: a survey. IEEE Access 6:28573–28593. https://doi.org/10.1109/ACCESS.2018.2831228
33. Rasheed AAA, Abdullah MN, Al-Araji AS (2022) A review of multi-agent mobile robot systems applications. Int J Electrical Comput Eng (2088-8708) 12(4). https://doi.org/10.11591/ijece.v12i4.pp3517-3529
34. Long P, Fan T, Liao X, Liu W, Zhang H, Pan J (2018) Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. IEEE Int Conf Robot Automat (ICRA) 2018:6252–6259. https://doi.org/10.1109/ICRA.2018.8461113
35. Winocur G, Moscovitch M, Bontempi B (2010) Memory formation and long-term retention in humans and animals: Convergence towards a transformation account of hippocampal-neocortical interactions. Neuropsychologia 48(8):2339–2356
36. McEliece R, Posner E, Rodemich E, Venkatesh S (1987) The capacity of the Hopfield associative memory. IEEE Trans Inf Theory 33(4):461–482. https://doi.org/10.1109/TIT.1987.105732
37. Xie Y, Hu P, Li J, Chen J, Song W, Wang X-J, Yang T, Dehaene S, Tang S, Min B et al (2022) Geometry of sequence working memory in macaque prefrontal cortex. Science 375(6581):632–639. https://doi.org/10.1126/science.abm0204

Publisher's Note