Neurocomputing 377 (2020) 269–285

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A self-organizing developmental cognitive architecture with interactive reinforcement learning

Ke Huang, Xin Ma*, Rui Song, Xuewen Rong, Xincheng Tian, Yibin Li

Center for Robotics, School of Control Science and Engineering, Shandong University, Jinan, PR China

ARTICLE INFO

Article history:
Received 3 April 2019
Revised 6 June 2019
Accepted 22 July 2019
Available online 23 October 2019

Communicated by Peter. W Vamplew

Keywords:
Cognitive development
Online learning
Self-organizing neural network
Object recognition
Interactive reinforcement learning

ABSTRACT

Developmental cognitive systems can endow robots with the abilities to incrementally learn knowledge and autonomously adapt to complex environments. Conventional cognitive methods often acquire knowledge through passive perception, such as observing and listening. However, this learning way may generate incorrect representations inevitably and cannot correct them online without any feedback. To tackle this problem, we propose a biologically-inspired hierarchical cognitive system called Self-Organizing Developmental Cognitive Architecture with Interactive Reinforcement Learning (SODCA-IRL). The architecture introduces interactive reinforcement learning into hierarchical self-organizing incremental neural networks to simultaneously learn object concepts and fine-tune the learned knowledge by interacting with humans. In order to realize the integration, we equip individual neural networks with a memory model, which is designed as an exponential function controlled by two forgetting factors to simulate the consolidation and forgetting processes of humans. Besides, an interactive reinforcement strategy is designed to provide appropriate rewards and execute mistake correction. The feedback acts on the forgetting factors to reinforce or weaken the memory of neurons. Therefore, correct knowledge is preserved while incorrect representations are forgotten. Experimental results show that the proposed method can make effective use of the feedback from humans to improve the learning effectiveness significantly and reduce the model redundancy.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Cognitive robots have received increasing attention in artificial intelligence, because they can develop knowledge and skills autonomously to cope with various learning tasks and dynamic environments in humans' daily life [1,2]. Two major challenges for current cognitive robots are quickly learning new objects they encounter and appropriately communicating with humans during human-robot interaction [3–7]. Therefore, the designed cognitive system must have capabilities to acquire knowledge incrementally and correct mistakes promptly according to human feedback in an online way.

Typically, researches for developing robots' cognition mostly focus on passive perception, such as observing, listening and action imitating [8–10]. These learning ways are known as individualistic learning [11]. It is suitable for learning object concepts online and even developing associative relationships cross multi-modalities [12–16]. However, the learned knowledge sometimes includes incorrect representations, which may be caused by inappropriate clustering strategy [17], insufficient perception [18] or human mistakes [19]. All of them can affect robots' recognition performance and hinder their interaction with humans. Moreover, these methods often have no ability for robots to spot errors by themselves and do not consider getting feedback from humans to correct their mistakes.

Social learning may be a promising method to address these problems since it indeed promotes the cognitive development of humans [20]. Infants often learn object concepts and develop cognitive skills under parental scaffolding [21]. However, they have no ability to assess whether their internal representations are consistent with the ground truths. Parents can provide appropriate rewards for infants and correct their mistakes, which contributes to consolidating learned knowledge and promoting their cognitive development remarkably [22]. Parental scaffolding has inspired robots learning methods for human-robot interaction [23–25]. However, the instruction from humans has proved to be an online labeling process rather than another form of knowledge. Besides, robots cannot associate their observations with names taught by humans autonomously [26–28]. Another method leveraging trials and errors is Interactive Reinforcement Learning (IRL) [29–31]. However, the states in these methods should be pre-defined

* Corresponding author.
E-mail address: maxin@sdu.edu.cn (X. Ma).

https://doi.org/10.1016/j.neucom.2019.07.109
0925-2312/© 2019 Elsevier B.V. All rights reserved.

before learning and cannot increase when new states occur. Although there have been some methods using function approximation to store observed states and generalize unseen states [32,33], they can perform well in continuous space but are not suitable for the discrete learning tasks of object concepts. Consequently, human feedback should be coordinated with an online and incremental learning way. Additionally, two cognitive processes of learning and feedback should be concurrent and interleaved.

In our previous research [34], we have applied self-organizing incremental neural networks to learn object concepts and build audio-visual associations in an open-ended manner. Although the robot is able to assess its learned knowledge autonomously, it still cannot guarantee that the learned internal representations are completely correct without human guidance. By considering IRL as an effective method to utilize human feedback, a cognitive system should integrate a self-organizing incremental neural network with IRL. However, the main challenge is how to formulate the self-organizing neural network in the framework of Reinforcement Learning (RL) [35,36]. Although different incremental RL or IRL models based on ART [37], SOM [38] and GNG [39] have been developed, the function of self-organizing neural networks only focuses on solving the storage problem for large state space of RL. Besides, RL is mainly applied to find the best state-action maps but it does not involve correcting the knowledge learned by self-organizing neural networks.

In this paper, we propose a Self-Organizing Developmental Cognitive Architecture with Interactive Reinforcement Learning (SODCA-IRL) to online learn new objects and correct inappropriate representations by interacting with humans. The architecture integrates IRL with hierarchical self-organizing neural networks as in our previous work [34]. Taking a biologically-inspired approach [40–42], the SODCA-IRL introduces a novel computational memory model to individual neural networks for the integration. The memory model is designed as an exponential function and controlled by two forgetting factors to simulate the forgetting and consolidation processes of humans. Different with other memory models using constant forgetting factor [37,43], our method with adjustable forgetting factors can improve the efficiency of memory by adjusting the forgetting speed. We also propose a reinforcement strategy to formulate the hierarchical self-organizing neural networks for IRL and update the architecture according to human rewards. IRL can control the nodes' memories to reinforce the correct representations and forget the incorrect representations. The incremental online learning and the reinforcement process are concurrent and interleaved, which allows SODCA-IRL to realize knowledge acquisition and mistake correction simultaneously. The contributions proposed in this paper are as follows:

(1) An interactive cognitive architecture based on hierarchical self-organizing neural networks and IRL can learn and correct object concepts in a concurrent and interleaved fashion, which significantly improves the recognition accuracy;
(2) A novel memory model for nodes can dynamically adjust their forgetting factors through IRL, which provides a new way to connect self-organizing neural network with IRL;
(3) An interactive reinforcement strategy based on the proposed memory model can control knowledge consolidation and mistake correction by adjusting node's memory and cope with human mistakes.

The remainder of the paper is organized as follows: the related works are discussed in Section 2; the overview of the interactive cognitive architecture is illustrated in Section 3; Section 4 demonstrates the evaluative experiments; Section 5 draws conclusions of this paper.

2. Related work

2.1. Interactive reinforcement learning

IRL has been demonstrated to improve the learning efficiency of RL significantly under human guidance [44]. According to the feedback strategy of humans, IRL can mainly be divided into two branches: action interaction and reward interaction [45]. In action interaction, humans give the optimal action selection so that robots can reduce their exploration. Senft et al. [31] combined RL with Supervised Progressively Autonomous Robot Competencies (SPARC) so that humans could fully control the robot's actions in the task of Sophie's Kitchen. The method allowed the robot to learn faster and more safely. Cruz et al. [46] proposed an IRL approach by integrating human vocal commands and hand gestures as advice to guide the robot's action. A confidence was used to evaluate human advice to make the interactive guidance more robust. However, the reward rule should be pre-defined and the environment should change after an action is executed. In reward interaction, humans can evaluate actions more accurately and provide more reasonable rewards. Kim et al. [47] used the internal states of human brain measured by error-related potential of human electroencephalogram (EEG) as rewards to guide robots to learn the meanings of human gestures. Another researches based on extended Training an Agent Manually via Evaluative Reinforcement (TAMER) also support the learning from human rewards [48,49]. They built human rewards as predictive models by TAMER. In a word, reward interaction is often simpler to guide and requires less task expertise for humans.

Steels and Kaplan [11] presented a word-meaning learning process to develop robot's language through social learning. The robot learned object's color by Expectation-Maximization (EM) cluster algorithm and word-meaning through RL. In this case, the trainer not only gave reward signals, but also provided the correct name to guide the "action" selection. As the learning task is similar to ours, we also adopt this interactive strategy. Cohen and Billard [50] proposed another view for human-robot interaction, in which the human guessed the robot's intention and chose an object to reflect the state after an action was executed. However, the states and actions are pre-defined and not suitable for online learning. In our research, self-organizing neural networks can online and incrementally learn new objects to provide states and actions for IRL. Besides, IRL is applied to provide human guidance online, which can evaluate and adjust the sample-symbol maps in visual sample layers and the visual-audio associations in the associative layer.

2.2. Self-organizing neural network with reinforcement learning

Having in mind the need to support feedback and incremental learning in cognitive process, we investigate the ways that RL or IRL have been implemented in a self-organizing neural network. One common approach focuses on using self-organizing neural network to deal with the storage problem of the states or actions in RL. Teng et al. [37] utilized FALCON architecture to incrementally learn domain knowledge as state-action-reward tuples so that the optimal state-action maps could be explored by RL. This model can incrementally add new states and have generalization ability. Yaink et. al. [51] applied GNG for the robot to learn human gestures, labels as well as responses defined the Q value as the length of the action vector. Humans judged the robot's response to a given gesture and gave an appropriate reward to adjust the gesture-response maps of the robot. However, this method is not suitable for online learning tasks. Self-organizing neural networks can also help to solve the continuous space problem of RL. Some researches [52,53] used two SOMs to quantize continuous state and action spaces into discrete representations and built state-action maps by a Q-table. However, the Q-table may be memory

consuming as each row (column) belongs to a node in state (action) SOM.

Another approach is to fine-tune the learning mechanism of self-organizing neural network by RL algorithm. Vieira et al. [39] proposed a TD-GNG to map value functions into states, wherein the activation condition of GNG was altered by the value function of Q-Learning. Chen et al. [38] applied RL to evaluate the decision of Bayesian SOM instead of updating Q-value. Furthermore, the weight update of Bayesian SOM was modified by the evaluative result. However, these models address the maps between inputs and outputs in a single network, but are not suitable for hierarchical networks, where the input states are not directly mapped to the output actions. Cruz et al. [54] introduced a confidence to evaluate human commands learned by GWR. The robot could decide whether to select an action according to human guidance or self-exploration. The limitation of this work lies in that it cannot perform in an open-ended learning way.

In contrast to these models, the proposed SODCA-IRL formulates our hierarchical architecture into two level state-action pairs. One maps a shape (color) sample node to its class symbol. The other associates object's visual symbols with its name. Besides, the designed memory model provides a new way to integrate self-organizing neural network with IRL so that IRL can consolidate correct representations or forget incorrect representations through controlling nodes' memory strength.

2.3. Computational memory model

Over the past few decades, several researches have been conducted to establish computational memory models. Ebbinghaus' research [41] indicates that human memory is always forgotten sharply at the beginning and slowly afterwards and eventually approaches to a stable value. Therefore, most of the models are designed as a power function [40,55,56] or an exponential function [57,58] to fit the Ebbinghaus' forgetting curve [41]. However, these methods only consider the forgetting process. Human cognitive development is also accompanied by the consolidation of knowledge [59], which contributes to the formation of long-term memory. Mayer [42] designed a differential function to simulate the assimilation and forgetting process, which considered memory's review during the learning process. Zhou et al. [60] adopted an exponential forgetting model according to the result of the contrast experiment that the exponential function is more precise than a power function to describe the Ebbinghaus' forgetting curve. Besides, they also proposed a reenergizing algorithm to consolidate memory when similar things occurred. These models with both forgetting and review are more consistent with the laws of human memory. There are another two methods similar to a memory model. Teng et al. [37] proposed a confidence model to simulate natural decay and reinforcement process. Another confidence model presented by Tan [43] consists of three processes including decay, reinforcement and erosion. They all equal to an exponential function. However, the decay rate does not reduce but still stays constant after each reinforcement or erosion.

Compared with these methods, the proposed memory model not only considers memory's forgetting and review in the learning process, but also induces positive or negative rewards in the practice phase to control the forgetting speed by adapting forgetting factors. The model is the medium to integrate self-organizing neural networks and IRL.

3. Proposed method

3.1. Overview of the SODCA-IRL

The proposed SODCA-IRL is based on the cognitive architecture which was previously exploited to learn object concepts and audio-visual associations [34]. Fig. 1 shows the overview of the SODCA-IRL. It consists of a series of hierarchical self-organizing incremental neural networks and mainly engages in two processes: learning and practice. A real trainer interacts with the architecture by teaching objects, assessing its learning results and correcting its mistakes, which is similar to the way parents educate their infants. The trainer just needs to know the objects the architecture learns, and does not have to possess professional knowledge. The learning process involves bottom-up learning and top-down response. From a computational perspective, the bidirectional structure has abilities to guide cluster and solve conflicts autonomously, which improves the recognition accuracy. However, incorrect representations may be generated and affect these strategies. Hence the practice process is proposed to solve this problem. We extend the previous cognitive architecture by integrating with IRL and equipping some neurons with a memory model. Objects and the trainer are treated as the environment to provide perception and advice. The recognized shapes, colors and names of the architecture represent the interactive actions. Moreover, these two processes can be executed in a cross manner, and the trainer's role can also change with the task.

3.2. Learning process

In the learning process, the cognitive architecture is occupied with perceiving object's view as well as name and learning object concepts incrementally according to Hebbian learning. Furthermore, it performs memory's consolidation and forgetting to delete some redundant nodes to make the structure more optimized.

3.2.1. Memory model

Apart from storing objects or events, human memory also forget something not reviewed in time [61]. As the characteristics of memory curve are closer to the trend of exponential function [41,60], we propose a dynamic memory model as shown in (1). Inspired by Mayer et al. [42,60], our model simulates both the forgetting and consolidation processes of memory.

$$M_i(t) = M_i(0) - \frac{1}{v_i(z)} \cdot (1 - e^{-f_i(z)\cdot(t-t_z)}), \tag{1}$$

where t is the learning time, z represents the number of activation and t_z is the latest activation time. $f_i(z)$ and $v_i(z)$ are two adjustable forgetting factors of node i. $f_i(z)$ controls the decay time of the memory model, which directly affects memory's forgetting speed. $v_i(z)$ adjusts the memory's amplitude, which affects the forgetting speed indirectly. As the memory fades over time and eventually approaches to $(M_i(0) - v_i(z)^{-1})$, $v_i(z)$ also controls the memory retention.

During the learning process, a new node i has an initial memory strength $M_i(0) = 1$. At time t, its memory decays over time according to (1). If a node is activated by a similar input and becomes the winner b, its memory should be consolidated and decay more slowly than before. But it is on the condition that the input must cause an intro-class operation of b, such as updating or creating an intra-class node. Then, the memory of node b is strengthened to 1 again as $t = t_z$ and its forgetting speed should slow down. As forgetting speed is in direct proportion to $f_b(z)$ and in inverse proportion to $v_b(z)$, $f_b(z)$ should decline while $v_b(z)$ should increase with the node activation. The following equations are designed to simulate these tendencies and update $f_b(z)$ and $v_b(z)$ respectively.

$$f_b(z) = f(z-1) \cdot \delta^z, \tag{2}$$

$$v_b(z) = v_b(z-1) + \gamma \cdot z, \tag{3}$$

where δ, $\gamma \in (0, 1)$ are the activation rates of the forgetting factors respectively. $f(z)$ is a monotone decreasing function and its range

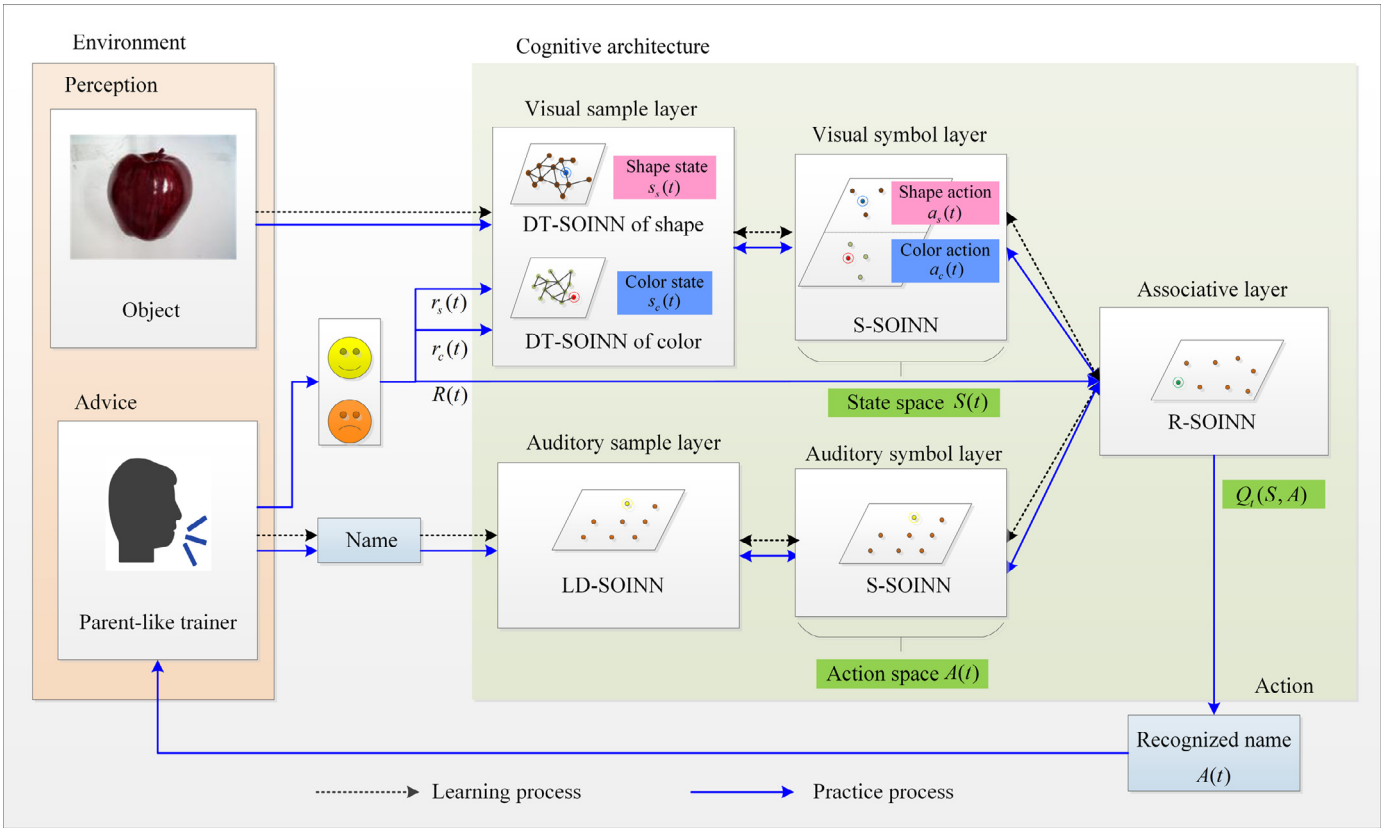


Fig. 1. Overview of SODCA-IRL.

is in(0, 1). The range of $v(z)$ should be in $[1, +\infty)$ so that the memory retention is non-negative. If node i is never activated, its memory would become too weak to be remembered. A forgetting threshold $M_{\min}$ is introduced to represent the minimum memory strength. Thus, node i is delated as long as $M_i(t) < M_{\min}$, which means its knowledge is forgotten. On the contrary, if the memory retention is higher than $M_{\min}$, the node is preserved and forms long-term memory.

Fig. 2 illustrates the memory curves of the proposed memory model in four different activation situations. The effect of the adaptive forgetting factors for our memory model is also validated by comparing with constant forgetting factors, which are set as$f(0)$ and $v(0)$. Due to the adaptive forgetting factors, the memory is strengthened and the forgetting speed becomes slower each time the node is activated. The node eventually retains a constant memory and is remembered for a long time. However, the constant one can only reinforce the memory at the activated point, and does not change the forgetting trend. No matter how many times it has been activated, the memory is still forgotten if not activated for a long time. Therefore, the memory model with adaptive forgetting factors can adjust the forgetting speed and promote the formation of long-term memory.

3.2.2. Modified hierarchical self-organizing neural networks

As shown in Fig. 1, SODCA-IRL contains three levels networks, which are proposed in our previous work [34]. The first level mainly processes visual and auditory sample representations. In the visual sample layers, the Dynamic Threshold Self-Organizing Incremental Neural Network (DT-SOINN) is dedicated to the online learning of shapes and colors. In the auditory sample layer, the Levenshtein Distance Self-Organizing Incremental Neural Network (LD-SOINN) is employed for learning word vectors. The second level based on the Symbol Self-Organizing Incremental Neural Network (S-SOINN) not only encodes the cluster numbers from each sample layer into corresponding symbols, but also decodes visual or auditory symbols from the associative layer into the original cluster numbers. The associative layer for the last level executes audio-visual integration and top-down response on the basis of the Relation Self-Organizing Incremental Neural Network (R-SOINN). The details of each network can refer to our previous work [34].

In order to integrate with IRL in practice process, we modify the hierarchical self-organizing neural networks as self-organizing developmental cognitive architecture (SODCA). In SODCA, the DT-SOINNs and R-SOINN are equipped with the proposed memory model. Nevertheless, the associations learned in R-SOINN are affected by DT-SOINNs, hence we only update the memories of DT-SOINNs during the learning process. The memories of R-SOINN are reinforced in the practice process, which is described in Section 3.3.2.

During the learning process, SODCA learns each object in an open-ended learning manner. In the visual pathway, if the received object's shape or color is novel, DT-SOINN would create a new class node with an initial memory model to learn the new sample. If the sample feature is familiar and activates a node b, DT-SOINN updates the node or creates a same class node with an initial memory model. At the same time, the forgetting factors of node b are updated by (2) and (3) to reduce its forgetting speed. After learning, the memory of node b is reinforced to 1, while the other nodes' memories in DT-SOINN decay according to (1). Once a node's memory is lower than the forgetting threshold $M_{\min}$, it would be forgotten and deleted from DT-SOINN. If a cluster in DT-SOINN disappears due to the forgetting, the related symbol in S-SOINN and associations in R-SOINN are also deleted. Then DT-SOINN outputs the recognized cluster to the visual symbol layer. In auditory pathway, LD-SOINN learns the objects' name and outputs its cluster to the auditory symbol layer. In each symbol layer,

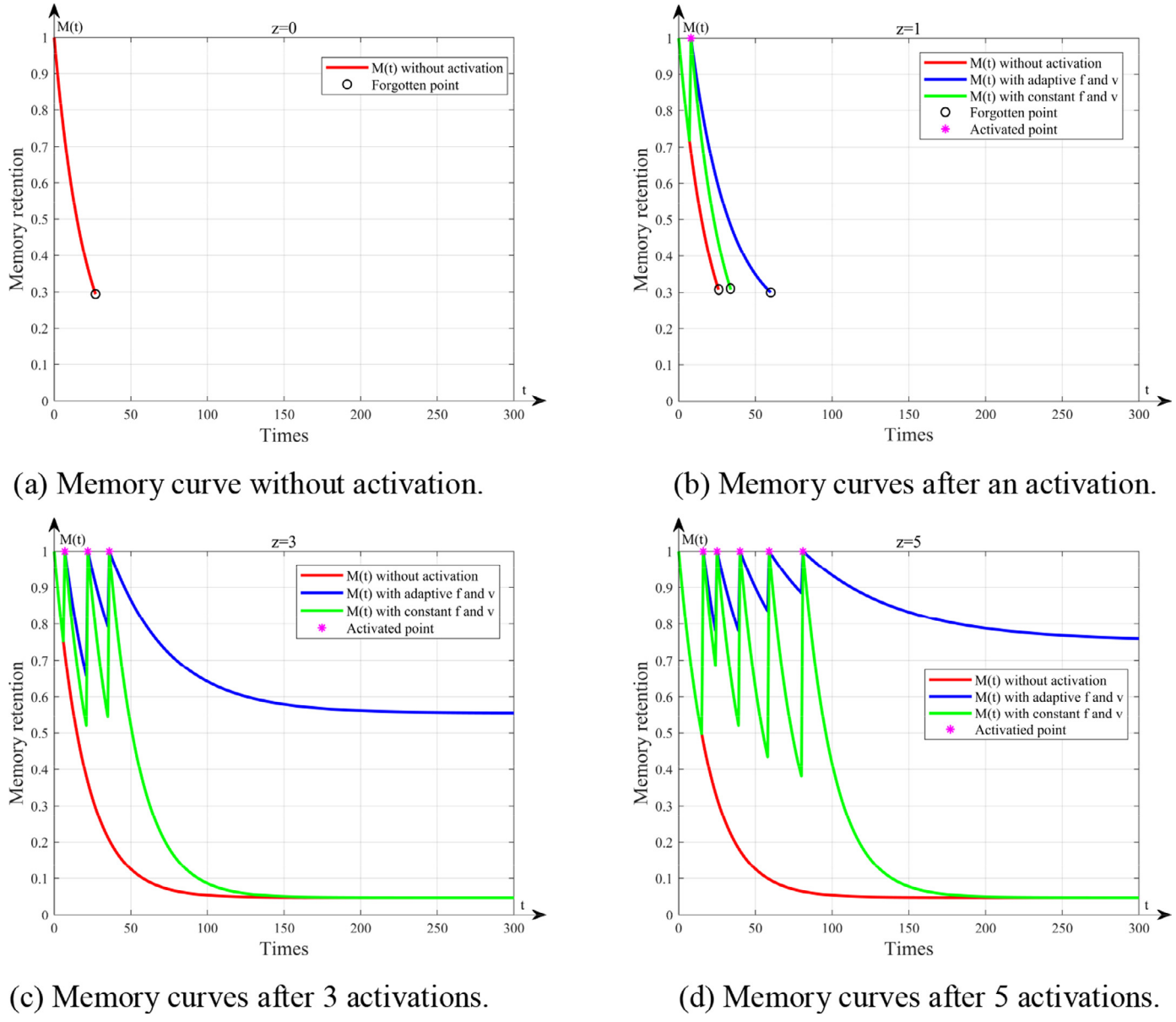


Fig. 2. Memory curves with $f(0) = 0.05$, $v(0) = 1.05$, $M_{\min} = 0.3$, $\delta = 0.8$ and $\gamma = 0.2$. (a) The memory is quickly forgotten, as it is never activated. (b) The memory with adaptive forgetting factors decays more slowly and lasts longer than the constant one after one activation. (c) The memory retention with adaptive forgetting factors is increased after each activation, while the constant one still decays as before and is eventually forgotten. (d) The memory retention with adaptive forgetting factors tends to form long-term memory with a stable and high value after 5 activations.

S-SOINN encodes the cluster as a symbol and transmits to the associative layer. Finally, R-SOINN integrates the symbols to establish an audio-visual association. The details of SODCA are illustrated in Algorithm 3 in Appendix A.

3.3. Practice process

In the practice process, the trainer tests the robot's mastery of learned knowledge, and the cognitive architecture mainly engages in recognition, recall and reinforcement.

3.3.1. Reinforcement learning formulation

The fundamental goal of RL is to find the optimum associations between states and actions. SODCA-IRL is also dedicated to associating names with objects' views to maximize the probability of obtaining rewards. Therefore, SODCA can be formulated in the framework of RL on the computational level. However, traditional RL algorithms often acquire a series of fixed states and actions, and the Q-table for all state-action pairs is memory consuming. It is not suitable for our scenario. SODCA-IRL utilizes self-organizing neural network to realize the incremental learning of states, actions and their associations in the learning process. Moreover, we apply IRL to evaluate the associations and provide rewards so that correct knowledge is consolidated and incorrect representations can be forgotten.

During the practice process, the trainer shows the robot an object. The robot recognizes the object's shape and color categories and activates the matching associative node to recall its name as the answer. Then, the trainer judges whether the result is correct or not. At time t, we treat the shape-color symbol pair as a state vector $S(t)$, the recognized name as the selected action $A(t)$, and the trainer's reward as $R(t)$, which are illustrated in Fig. 1. $R(t)$ has two values: 1 and -1. If the answer is correct, the trainer confirms it with a positive reward $R(t) = 1$ and the memory of this audio-

visual association is reinforced. Otherwise, the trainer gives a negative reward $R(t) = -1$ and tells its real name to correct the mistake. Then the incorrect relationship is gradually forgotten. However, as the state space is determined by the cluster results of the visual sample layers, the rewards for the associative layer are not sufficient to simultaneously guide two levels. Fortunately, each visual sample can be mapped into a cluster, and we can add extra IRL between visual sample layers and the visual symbol layer to further understand the source of errors. Therefore, both shape and color features can be treated as state vectors $s_s(t)$ and $s_c(t)$, and the recognized symbols equal to action vectors $a_s(t)$ and $a_c(t)$. Thus, $S(t) = \{a_s(t), a_c(t)\}$. The trainer can point out whether the visual symbols recognized from visual sample layers are correct or not and give appropriate rewards $r_s(t)$ and $r_c(t)$. If the recognized name $A(t)$ is correct while the recognized shape $a_s(t)$ (or color $a_c(t)$) is incorrect, namely, $R(t) = 1$ and $r_s(t) = -1$ (or $r_c(t) = -1$), the robot can correct the inappropriate visual feature and build a new audio-visual association by itself. The recognized association may represent another object and its memory would not be affected by IRL. As long as the recognized name is incorrect ($R(t) = -1$), the audio-visual association should be forgotten. Besides, the robot also considers assessing sample-symbol maps to find all mistakes. A new association would be created after the trainer gives its real name. Only when all rewards are positive can the robot confirm the association is correct and consolidate the knowledge.

3.3.2. The integration of self-organizing neural network and IRL

The key to the integration lies in how to introduce reinforcement signals into SODCA. In our method, the memory of each node in two DT-SOINNs and R-SOINN can be treated as a Q-value to evaluate the confidence of the state-action pair. When one of these nodes is activated and demonstrates its representation, the trainer gives an appropriate reward to the node. A positive reward means that the representation is correct and should be consolidated. Therefore, the node's memory should be strengthened to 1. On the contrary, a negative reward implies that the incorrect representation should be forgotten from the network. The memory should decline from the time receiving negative rewards. Therefore, the memory model with IRL is designed, as shown in the following equation:

$$M(t) = \begin{cases} 1, & r = 1 \\ M(t_{z_p}) - \frac{1}{v(z)}(1 - e^{-f(z)\cdot(t-t_{z_p})}), & r = -1 \end{cases}, \tag{4}$$

where t_{z_p} represents the time receiving a negative reward. z_r and z_p record the number of positive reward and negative reward respectively.

The reinforcement also affects memory's forgetting speed and retention. The update strategies of two factors in practice are designed, as shown in the following equations:

$$f(z) = \begin{cases} f(z-1)\cdot {\delta_r}^{2z_r}, & r = 1 \\ f(z-1)\cdot {\delta_p}^{2z_p}, & r = -1 \end{cases}, \tag{5}$$

$$v(z) = \begin{cases} v(z-1) + \gamma_r \cdot z_r, & r = 1 \\ 1, & r = -1 \end{cases}, \tag{6}$$

where δ_r and δ_p represent the positive reward rate and negative reward rate of $f(z)$ respectively. γ_r is the positive reward rate of $v(z)$. If $r(t) = 1$, the forgetting speed should become slower than the situation in which the node is just activated in the learning process. $f(z)$ should decline more sharply than (2) and $v(z)$ should also increase more than (3). Thus, we decrease the reward rate of $f(z)$ as $\delta_r < \delta$ and accelerate its change rate as $2z_r$. At the same time, we assign a higher reinforcement rate for $v(z)$ by $\gamma_r > \gamma$ to reduce the forgetting speed and increase the memory retention. Inversely, the forgetting speed should be accelerated if $r(t) = -1$. Thus, $f(z)$ must be a monotonic increasing function. We assign the negative reward rate as $\delta_p > 1$ and its change rate as $2z_p$ to speed up the forgetting of incorrect representations. In order to force the incorrect memories to the minimum, we set $v(z) = 1$ so that the memory retention eventually approaches to 0. As for other nodes that are not activated, their forgetting factors are not changed and their memories still decay as (1). Algorithm 1 illustrates the details of the memory model with IRL.

Four memory curves under IRL are depicted in Fig. 3 to demonstrate the reinforcement process. Apart from reducing the forgetting speed significantly, our memory model with IR can also enhances the memory retention greatly after receiving a positive reward compared with that without IRL. Besides, the memory with IRL decays more sharply after being weakened. This suggests that IRL can promote the formation of long-term memory and forget incorrect representations quickly.

In the practice process, SODCA-IRL utilizes the learning results of SODCA and interacts with the trainer to evaluate the learned knowledge. We design the reinforcement action strategy shown in Algorithm 2, which assists in adjusting the structure of SODCA-IRL. When an object learned before is shown to the architecture, DT-SOINNs find the best matching nodes for the shape and color states $s_s(t)$ and $s_c(t)$. The visual S-SOINN receives their cluster numbers and outputs the recognized shape and color symbols (actions) $a_s(t)$ and $a_c(t)$. Then the combination state $S(t) = \{a_s(t), a_c(t)\}$ is transmitted to R-SOINN and activates the best matching associative node to recall the object's name $A(t)$. The trainer evaluates the recognized shape $a_s(t)$, color $a_c(t)$ and name $A(t)$.

If the recognized name is right, SODCA-IRL receives a positive reward $R(t) = 1$. Then, the shape and color should be considered to evaluate the audio-visual association. If they are both right, namely, $r_s(t) = 1$ and $r_c(t) = 1$, the association is completely correct. Its memory would be reinforced and its forgetting speed is reduced as described by step 5 of Algorithm 1. At the same time, DT-SOINN executes a Update Action in Algorithm 2 to update the best matching node b. If the shape or color is incorrect, namely $r_s(t) = -1$ or $r_c(t) = -1$, the associative node's memory should still be weakened as described by step 7 of Algorithm 1. DT-SOINN also executes the Correct Action in Algorithm 2 for the node b. If the node never receives any positive reward, the network would repeal its class and break all connections with its neighbors. Its memory is weakened and the forgetting speed increases sharply. Besides, a new class node is created for correcting the mistake. Otherwise, this node may represent another object and should not be weakened to avoid affect the recognition of its real object.

If the recognized name is wrong, SODCA-IRL get a negative reward $R(t) = -1$. The association should be weakened no matter whether the shape or color is correct. Then, the trainer should correct the name. In order to find all mistakes, visual recognitions should also be assessed. If the shape or color is correct, DT-SOINN not only executes the Update Action, but also considers whether to execute the Reactivated Action. If the node's class has been repealed due to former advice, the network would assign a new class for the node and increase its $v(z)$ to guarantee that the correct memory is never forgotten. If the shape or color is incorrect, DT-SOINN executes the Correct Action and judges the class of b. If b has been repealed class, its memory should decay more sharply so that the incorrect representation can be quickly forgotten. Otherwise, a Weakened Action is executed. The detailed algorithm of SODCA-IRL is outlined in Algorithm 4 in Appendix B.

We also consider that humans may give wrong advice by mistakes. There are three types of mistakes that maybe occur in this paper. One is the trainer teaches wrong names or gives inappropriate rewards for names. Another mistake is that an incorrectly recognized shape (color) is treated as right. SODCA-IRL

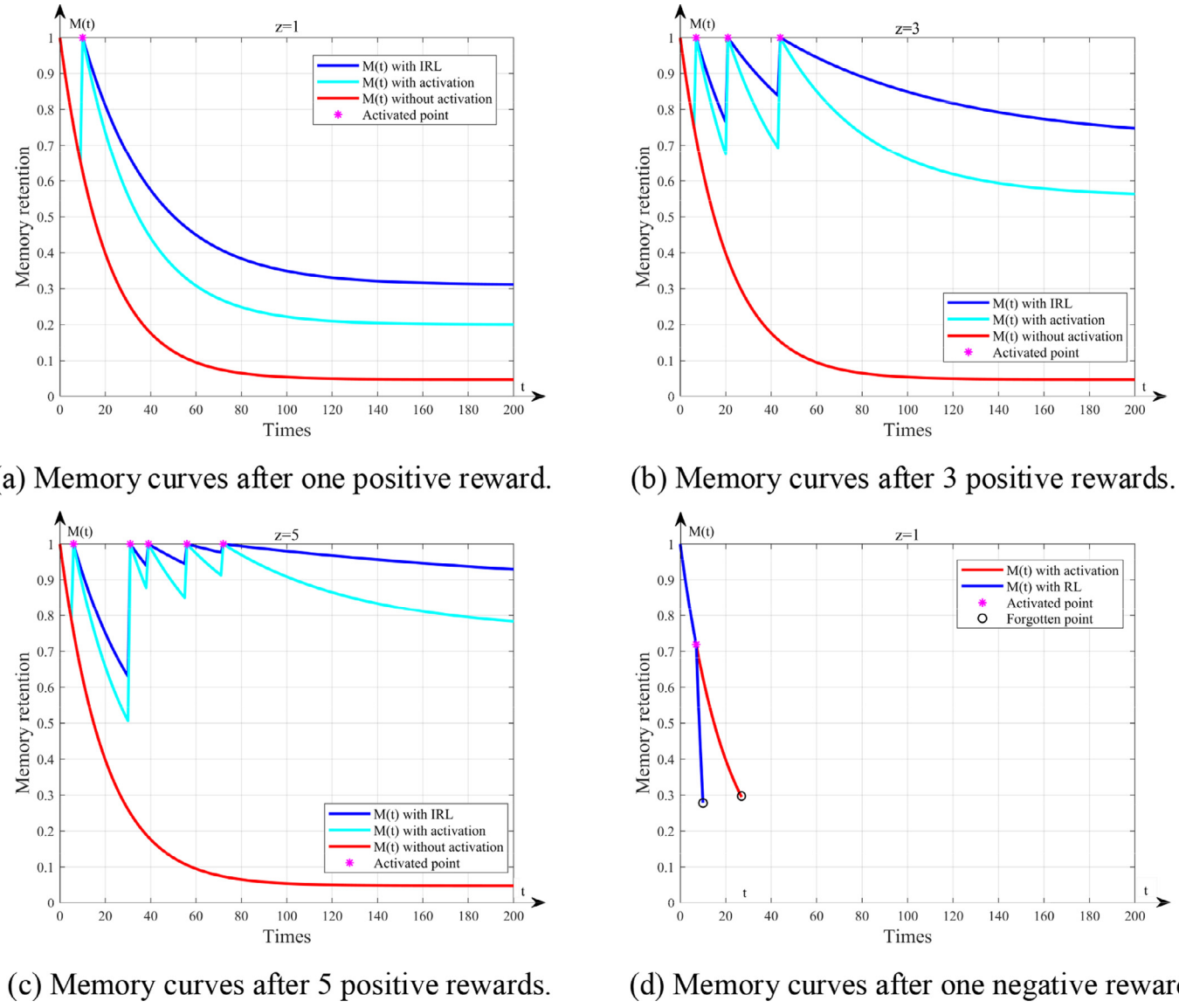


Fig. 3. Memory curves under IRL with $f(0) = 0.05$, $v(0) = 1.05$, $M_{\min} = 0.3$, $\delta_r = 0.5$, $\delta_p = 1.5$ and $\gamma_r = 0.4$. (a) Compared with other two cases, the memory with IRL decays more slowly and reserves higher retention after once positive reward. (b) After 3 positive rewards, the memory with IRL can finally reserve as many retention as that without IRL after activated 5 times. (c) The memory with IRL can reserve 0.87 retention after 5 positive rewards. (d) The memory is quickly forgotten after receiving a negative reward.

provides the trainer chances to correct these two mistakes above by giving appropriate advice when the robot meets the object again. The last one is that a correctly recognized shape (color) is misjudged as wrong. That may lead to three results. Firstly, a new node would be created for the correct representation by the Correct Action. Thus, the robot can still recognize this object by the new knowledge. Secondly, if the misjudged node has received positive rewards before, it would not be affected according to the Weakened Action. Hence the object can still be recognized correctly. Finally, if the node has not received any positive rewards so far, its class would be repealed by the Weakened Action. The robot may forget the object's name as the recognized shape (color) class is unknown. This result can also be solved through IRL. In conclusion, the robot mainly relies on humans to correct mistakes for lacking experience in the early stages of cognition. With the accumulation of knowledge and positive rewards, the robot gradually has a capacity of coping with human mistakes by itself.

4. Experimental results

Two types of experiments are performed to evaluate the proposed architecture. The first type analyzes the influence of the designed memory model on our cognitive structure (SODCA) in the learning phase. Moreover, the second type illustrates the evaluation of the complete interactive cognitive architecture (SODCA-IRL) during practice process. Both experiments are conducted on a dataset with 20 common fruits and foods (see Fig. 4), which has been previously used in PCN [14].

4.1. Evaluation criteria

Some researches on self-organizing network integrated with RL mainly assess the number of nodes, success rate and reward [37,38]. As our cognitive architecture adopts an open-ended learning way, the learning performance over time should also be evaluated. Therefore, we combine the above metrics with an effective

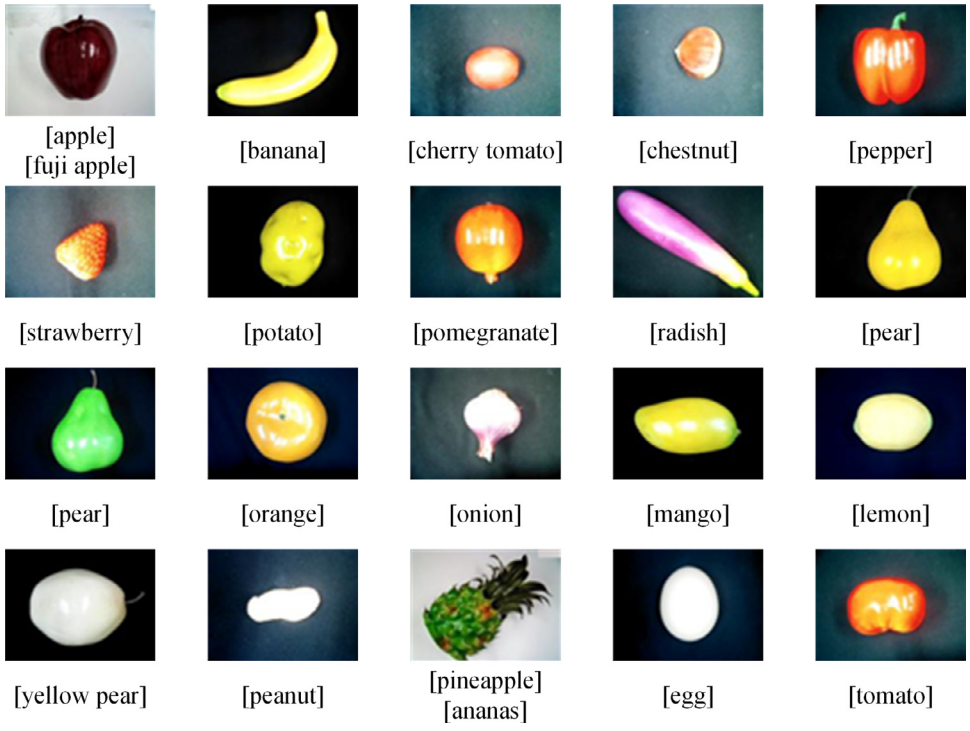


Fig. 4. The dataset of 20 common fruits and foods.

evaluation protocol for open-ended category learning algorithm applied in [26,62,63]. Moreover, there are three aspects of evaluation criteria proposed as follows.

(1) How much does it learn? We note the average number of nodes in each network after each experiment to demonstrate the learning results.
(2) How fast does it learn? We record the number of nodes in each layer and the number of categories over the online learning and practice processes to indicate the learning speed.
(3) How well does it learn? Recognition accuracy is used to evaluate the learning effectiveness. To assess the effectiveness of reinforcement learning, we analyze the memory retention, forgetting factors, and similarity thresholds of each node after learning and practice.

4.2. Results and evaluation

4.2.1. Learning evaluation

The cognitive algorithm is run in a computer and interacts with a real trainer through a user interface, as shown in Fig. 5. The Display area is used to show the input image, voice, extracted features and name. The Process area shows the execution steps of the cognitive algorithm. The Operation area provides operating buttons for the trainer. During the learning phase, the trainer gives an image and teaches the corresponding name. SODCA learns the visual features as well as the name and then builds an audio-visual association as demonstrated in Process area. All objects in the dataset are inputted one by one, and the order of instances is scrambled each time. The methods of feature extraction and automatic speech recognition have been introduced in our previous work [34].

The parameters of each self-organizing neural network are set as in our previous work [34]. Partial parameters of the memory model are set as $M_{\min} = 0.1$, $\delta = 0.8$, $\gamma = 0.2$, which are empirically found with respect to the best learning performance. Memory's forgetting speed controlled by two forgetting factors must be tuned to provide a good balance between recognition accuracy and numbers of nodes. Factor ν not only affects the forgetting speed, but also determines memory retention. We set $\nu(0) = 1.05$ so that the final memory retention tends to be a small value 0.0476. Factor f only affects the forgetting speed, hence we conduct the learning experiment under different $f(0)$. Every experiment is repeated 30 times and the learning results are shown in Tables 1 and 2.

From Table 1, the theoretical forgetting times indicate that SODCA forgets more and more quickly as $f(0)$ increases. Besides, the SODCA with $f(0) = 0$ cannot forget any knowledge and is used as a reference standard for evaluating the effectiveness of the

Table 1
Average number of nodes in each layer under different $f(0)$.

$f(0)$	Theoretical forgetting time	Average number of nodes				
		Shape sample node	Shape symbol node	Color sample node	Color symbol node	Associative node
0	N	92	39	80	27	60
0.02	146	91	36	78	26	64
0.025	117	85	37	73	22	60
0.03	97	79	36	70	22	56
0.035	83	76	37	64	23	59
0.04	73	65	32	62	22	50

Table 2
Average number of forgotten nodes in each layer under different $f(0)$.

$f(0)$	Average number of forgotten nodes				
	Shape sample node	Shape symbol node	Color sample node	Color symbol node	Associative node
0.02	5	3	3	1	3
0.025	12	4	8	1	5
0.03	21	5	12	2	7
0.035	28	10	17	4	14
0.04	32	14	30	7	21

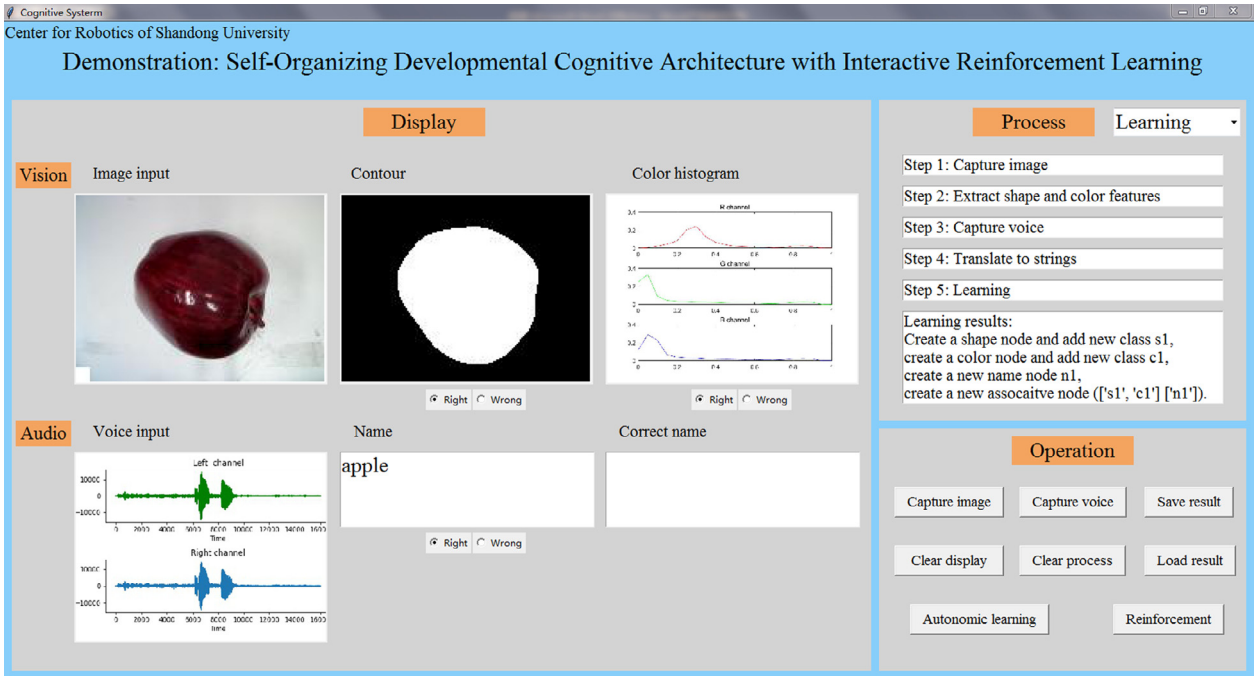


Fig. 5. An example that the trainer teaches SODCA in learning process. Firstly, the trainer clicks 'Capture image' to let SODCA see an apple and extract its contour and color histogram. Then, the trainer clicks 'Capture voice' and teaches the apple's name. SODCA translates the sound into a string. Finally, the trainer clicks 'Autonomic learning' to let SODCA learn these representations and establish an audio-visual association.

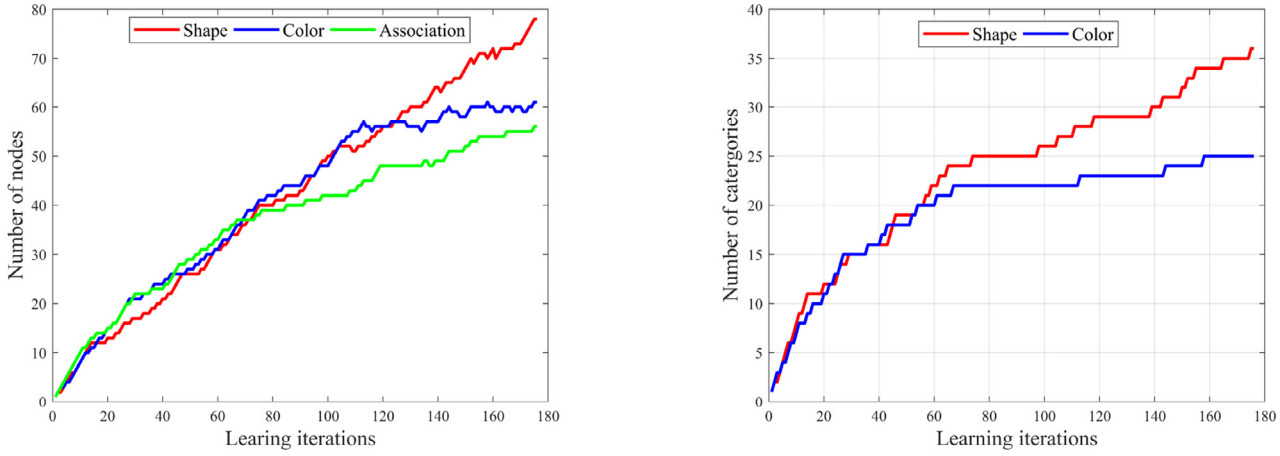
Table 3
Average accuracy under different $f(0)$.

$f(0)$	0	0.02	0.025	0.03	0.035	0.04
Average accuracy	90.79%	89.65%	89.55%	89.56%	87.73%	86.93%

memory model. The average number of nodes in Table 1 and the average number of forgotten nodes in Table 2 demonstrate that higher $f(0)$ contributes to forgetting more nodes. That suggests the SODCA with the designed memory model has the ability to reduce each network's redundancy.

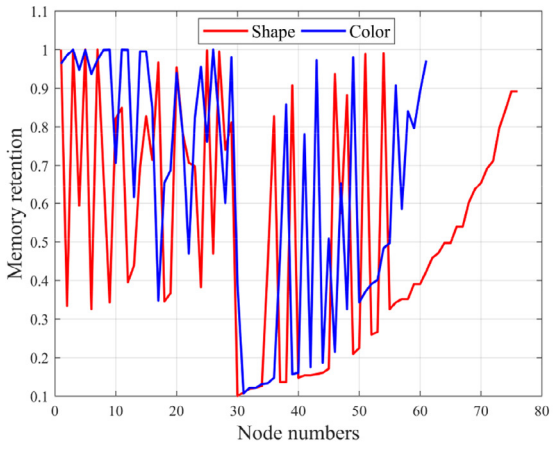
To validate the learning effectiveness, we test the average recognition accuracies under different $f(0)$. The trainer shows SODCA objects and assesses its recognized shapes, colors and names. The experimental observation manifests that SODCA can successfully distinguish all colors while all mistakes stem from incorrect shapes and names. Table 3 indicates that the SODCA with $f(0)$ from 0.02 to 0.03 can achieve comparable performance with the case of $f(0) = 0$. But the accuracies of other two cases with higher $f(0)$ are significantly poor for fast forgetting. Nodes may have been deleted before their memories are consolidated. Whereas, SODCA with low $f(0)$ can strengthen memory promptly so that the deleted nodes are mostly incorrect or redundant. Compared with other cases, $f(0) = 0.03$ obtains the optimal cognitive structure and competitive recognition accuracy simultaneously. Therefore, we select it as the default system parameter. A learning example of $f(0) = 0.03$ is illustrated in Figs. 6 and 7.

Fig. 6 shows the performance of SODCA over the online learning process. Fig. 6(a) reflects that SODCA can learn new knowledge as well as forget incorrect nodes. In Fig. 6(b), both shape and color categories increase sharply at the beginning but gradually slow

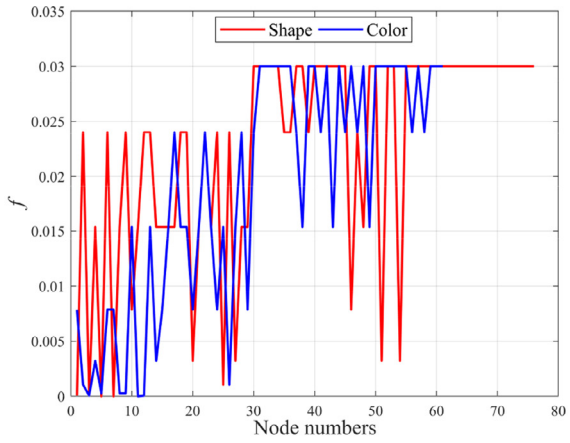


(a) Number of nodes in DT-SOINNs and R-SOINN. (b) Number of catergories in the visual S-SOINN.

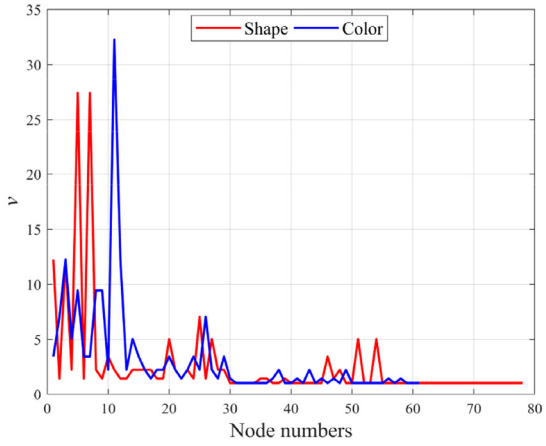
Fig. 6. The performance of SODCA with $f(0) = 0.03$ over the online learning process.



(a) Memory retention of each node after learning.



(b) Forgetting factor f of each node after learning.



(c) Forgetting factor v of each node after learning.

Fig. 7. The values of memory model with $f(0) = 0.03$ for each node in DT-SOINNs after learning.

Algorithm 1 The memory model with IRL.

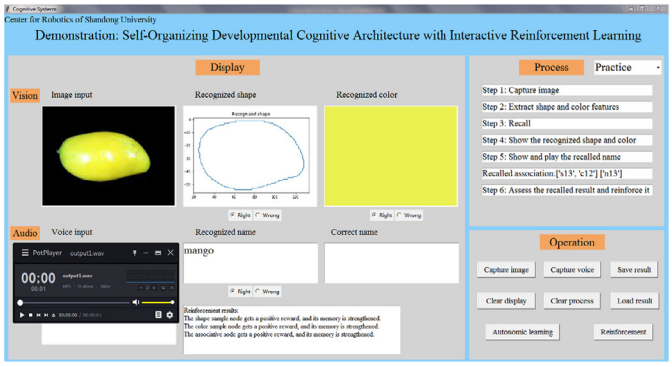
Input: reward from the trainer $r(t)$.

for each node in DT-SOINNs or R-SOINN **do**
  if the node i is activated as the best matching node b **then**
    Update the activation number of node b: $z = z + 1$.
    if $r(t) > 0$ **then**
      Strengthen: $z_r = z_r + 1$, $f_i(z) = f_i(z-1) \cdot \delta_r^{2z_r}$, $v_i(z) = v_i(z-1) + \gamma_r \cdot z_r$, $M_i(t) = 1$.
    else
      Weaken: $z_p = z_p + 1$, $f_i(z) = f_i(z-1) \cdot \delta_p^{2z_p}$, $v_i(z) = 1$, $M_i(t) =$ $\exp(-f_i(z)) \cdot M_i(t-1) + (\exp(-f_i(z)) - 1) \cdot (1/v_i(z) - 1)$.
    end if
  else
    Decay: $M_i(t) =$ $\exp(-f_i(z)) \cdot M_i(t-1) + (\exp(-f_i(z)) - 1) \cdot (1/v_i(z) - 1)$.
  end if
  if $M_i(t) < M_{\min}$ **then** delete the node i.
  end if
end for

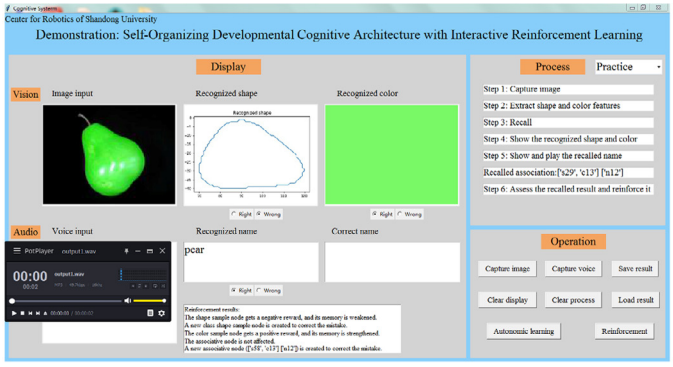
down. That suggests SODCA can quickly learn new categories and the variation of categories tends to be stable after SODCA masters enough knowledge. What is more, SODCA can learn different cate-

Algorithm 2 The action strategy of SODCA-IRL.

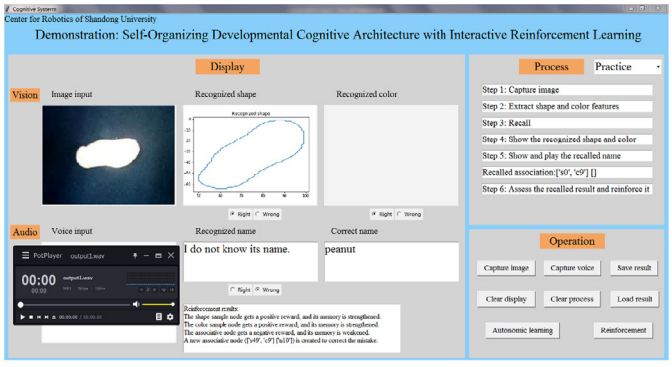
Update Action:
Update node b and its similarity thresholds TL and TH.
Correct Action:
Create a new sample node with an initial memory model.
Reinforce its forgetting factor: $v(1) = v(0) + \gamma_r$ and update its reward time $z_r = 1$.
Encode new symbol s_{c_s} for shape or c_{c_c} for color, and create a new symbol node in the visual S-SOINN.
Weakened Action:
if $z_r = 0$ for node b **then**
  Repeal its class: $c_b = 0$.
  Break up all connections between b and its neighbors.
  Reset its similarity thresholds: $TH_b = TL_b = \varepsilon_L \cdot \|w_b\|$.
  Update the memory of each node using Algorithm 1.
  if a cluster disappears as nodes are forgotten **then**
    Delete corresponding symbol node and associative nodes.
  end if
end if
Reactivated Action:
if $c_b = 0$ **then**
  Assign a new class to node b in DT-SOINNs: $c_b = \max(C) + 1$.
  Reinforce its forgetting factor: $v(z) = v(z-1) + \gamma_r$.
  Encode new symbol s_{c_s} for shape or c_{c_c} for color and create a new symbol node in the visual S-SOINN.
end if



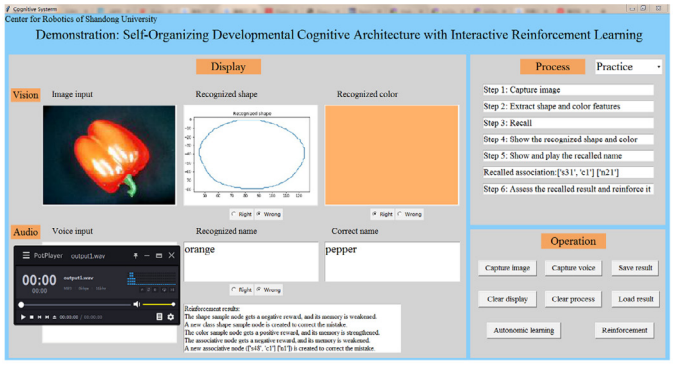
(a) SODCA-IRL recognizes the mango correctly.



(b) SODCA-IRL mistakes the pear's shape.

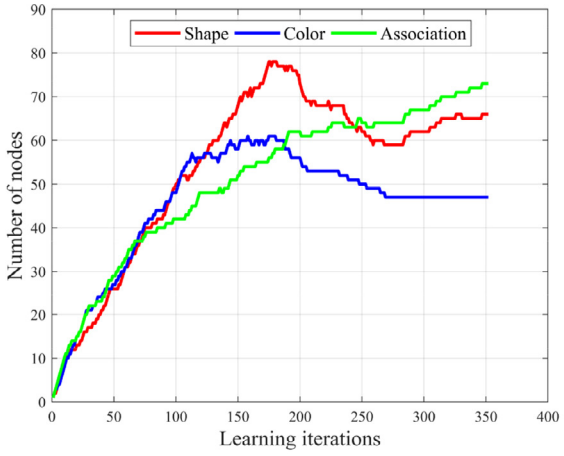


(c) SODCA-IRL forgets the peanut's name.

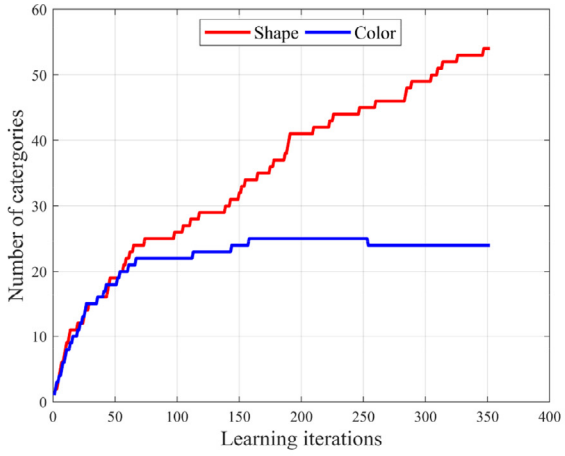


(d) SODCA-IRL mistakes the pepper's shape and name.

Fig. 8. Four situations in which the trainer interacts with SODCA-IRL in the practice process. (a) All recognized results are correct. The trainer chooses 'Right' which represents a positive reward for each result. (b) The shape of the pear is mistaken as that of a strawberry. The trainer chooses 'Wrong' which represents a negative reward for it. (c) SODCA-IRL forgets the peanut's name. The trainer chooses 'Wrong' for name and gives the correct name. (d) The pepper is mistaken as an orange. The trainer chooses 'Wrong' for shape and name, and provides its correct name.



(a) Number of nodes in DT-SOINNs and R-SOINN.



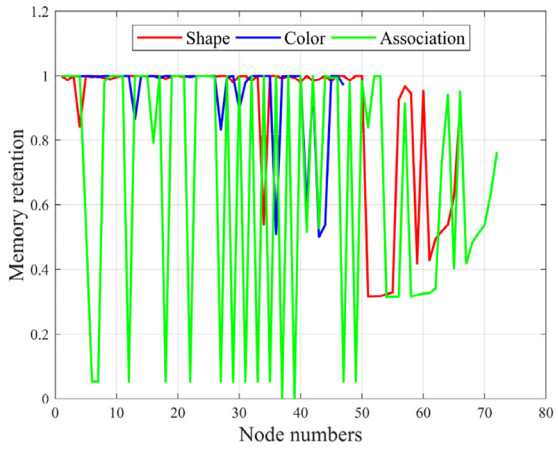
(b) Number of catergories in the visual S-SOINN.

Fig. 9. The performance of SODCA-IRL with $f(0) = 0.03$ over the online learning and practice processes.

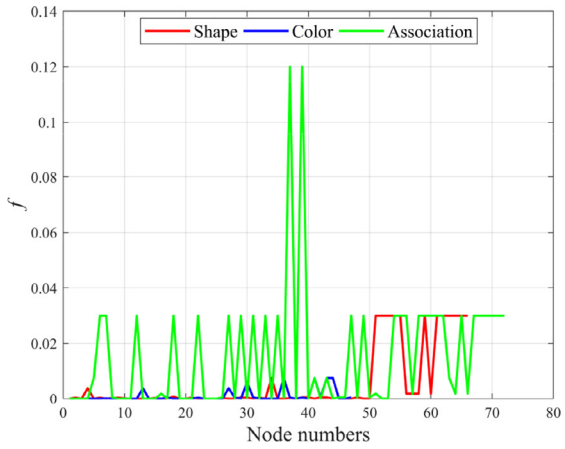
gories online and recognize the learned classes at the same time, rather than separate these two processes.

Fig. 7 demonstrates the learning results of memory models in shape and color DT-SOINNs. The memory retention and forgetting factors for most of the old nodes (whose numbers are small) are all obviously reinforced after learning. That suggests these nodes have formed long-term memory. However, some old nodes with low memory and high forgetting speed may be deleted if their memories decay to lower than the forgetting threshold $M_{\min}$. Many new nodes (whose numbers are large) still keep initial forgetting factors, which indicates they have not been activated yet. Whereas, some new nodes can maintain high memory, small f and large v, because they have received positive rewards. Therefore, the designed memory model has abilities to remember the important memory and forget redundant nodes.

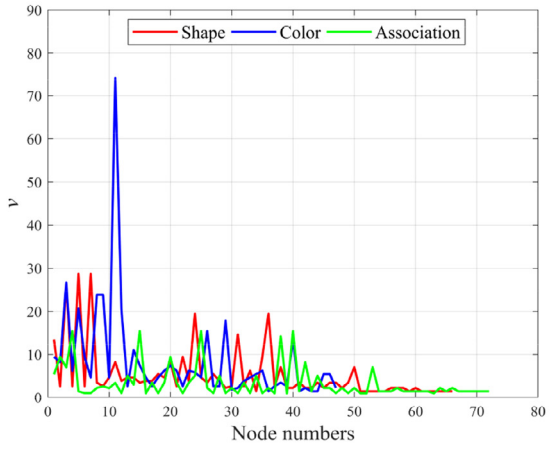
In each subfigure of Fig. 7, there is an obvious threshold at number 30 where the curve's tendency is different on its two sides. Especially, the shape nodes curve's tendency from number



(a) Memory retention of each node after practice.



(b) Forgetting factor f of each node after practice.



(c) Forgetting factor v of each node after practice.

Fig. 10. The values of memory model with $f(0) = 0.03$ for each node in DT-SOINNs and R-SOINN after practice.

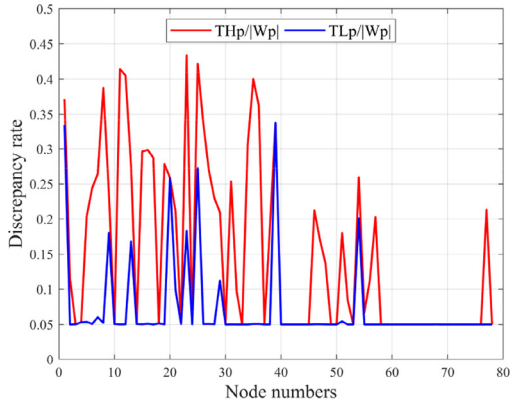
76 to number 30 is quite similar with the forgetting curve without activation in Fig. 2(c). The reason is that the memories of new nodes without activation would gradually decay over time, while the memory of number 30 is close to $M_{\min}$. Any other nodes with smaller numbers have been either forgotten for lower memory or formed long-term memory through activations. That is why the threshold appears at this point. On top of that, the threshold is not at a fixed position. It depends on the forgetting time of the memory model, learning iterations and how many similar objects the architecture meets. According to Table 1, the theoretical forgetting time of SODCA-IRL with $f(0) = 0.03$ is 97. If there are enough objects and all of them are different, the threshold would appear at the 97th node from the bottom. Once nodes are activated by similar objects, the position would be changed but not exceed the limitation of 97.

4.2.2. Practice evaluation

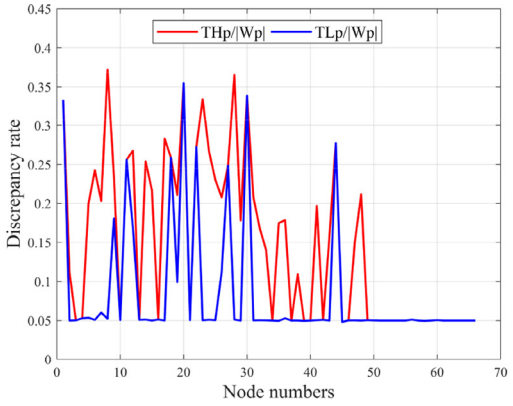
In this section, the performance of SODCA-IRL is evaluated. Four interactive situations are shown in Fig. 8. In each practice time, the trainer firstly clicks 'Capture image' to give SODCA-IRL an object. Secondly, SODCA-IRL displays the recognized shape, color and name in the user interface, and speaks the name through an audio player. Thirdly, the trainer assesses the results and gives appropriate rewards respectively. If the name is incorrect, the trainer should also provide its real name to correct the mistake. Then, the trainer clicks 'Reinforcement' to let SODCA-IRL receive these rewards. Finally, SODCA-IRL executes appropriate actions in Algorithm 2 to adapt its representations of this object and shows the reinforcement results in the Display area.

The experiments for practice are based on the results of the learning phase with $f(0) = 0.03$, which can provide initial values for each state, action and their associations of IRL. All objects learned before are tested one by one. Other parameters are set as $\delta_r = 0.5$, $\delta_p = 2$, $\gamma_r = 0.4$. We also compare SODCA-IRL with two other cognitive architectures: PCN [14] and our previous work [34] to validate its learning effectiveness. During the practice phase, the SODCA-IRL averagely deletes 41 shape sample nodes, 14 shape symbol nodes, 25 color sample nodes, 4 color symbol nodes and 16 associative nodes. At the same time, it also adds 23 shape sample nodes, 24 shape symbol nodes, 1 color sample node, 1 color symbol node and 30 associative nodes for correcting its mistakes. From 176 tests for all objects, the trainer averagely conducts 146 positive rewards and 30 negative rewards. The practice results and comparisons with two other cognitive architectures are reported in Table 4.

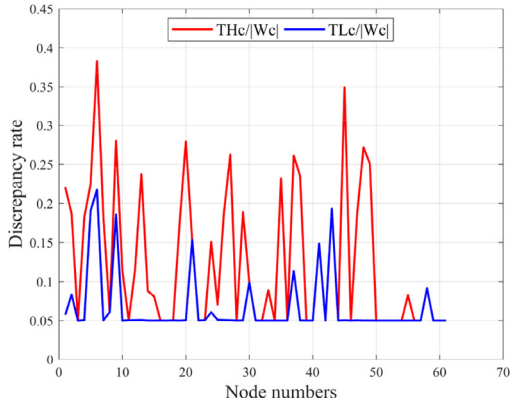
From Table 4, SODCA-IRL produces a higher average accuracy than PCN and our previous work. Specially, the value of average accuracy is enhanced to 99.24% by SODCA-IRL. That indicates our method can achieve an excellent performance of learning object concepts online. Besides, the accuracy is also higher than all results



(a) Shape similarity thresholds of each node after learning.



(b) Shape similarity thresholds of each node after practice.



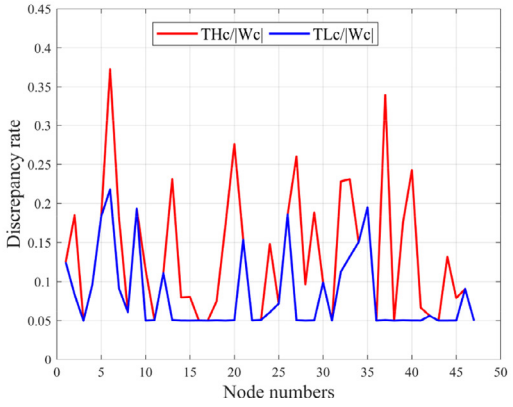


Fig. 11. The similarity thresholds of each node in shape and color DT-SOINNs after learning and practice.

Table 4
Average number of nodes in each layer after the whole cognitive process.

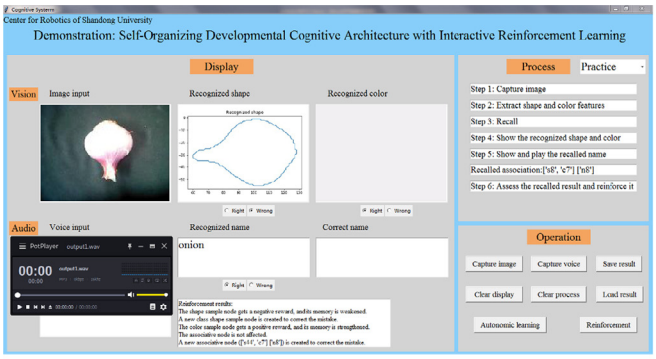
Cognitive architecture	Average number of nodes					Average accuracy
	Shape sample node	Shape symbol node	Color sample node	Color symbol node	Associative node	
PCN [14]	67	22	44	18	61	83.98%
Our previous work [34]	94	34	73	22	57	90.02%
SODCA-IRL	80	52	55	23	85	99.24%

of the learning process in Table 3. This implies IRL can significantly improve the learning effectiveness of self-organizing neural networks. The reason is that IRL can make up the disadvantage of self-organizing neural network that it cannot get feedback to correct mistakes autonomously. In addition, the numbers of two sample nodes are sharply reduced compared with our previous work, because both of the incorrect and redundant nodes are deleted. Although the numbers of two symbol nodes and the associative nodes are more than our previous work, these increased nodes are created for correcting its mistakes. A practice case is illustrated in Figs. 9 and 10.

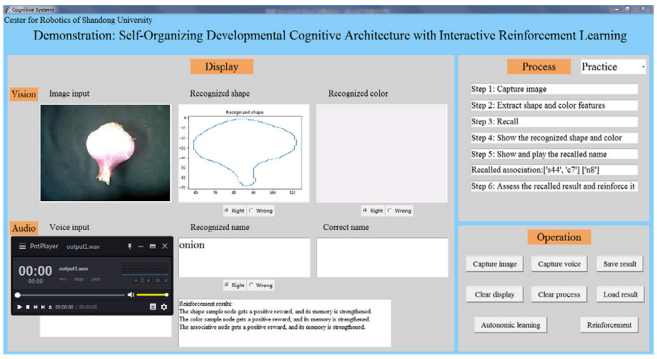
In Fig. 9, the first 176 iterations are the learning process and the remaining are the practice process. The number of shape nodes decreases significantly but the number of shape categories still increases during the practice period. The reason is that SODCA-IRL not only removes redundant nodes but also quickly forgets some incorrect representations after receiving negative rewards, and the new nodes are created for correcting inappropriate shapes. Color categories are more stable despite the decline of color nodes, which indicates that all forgotten color nodes are redundant. The number of associative nodes increases as new associations are generated after each error correction. But the incorrect associations are also quickly forgotten. Therefore, SODCA-IRL can effectively delete redundant nodes as well as incorrect representations and supplement new nodes to correct its mistakes promptly.

Fig. 10 presents the values of memory model for each node after practice. Compared with the learning results in Fig. 7, the old shape and color nodes are reinforced more heavily. They have higher memory retention and their forgetting speeds approach to 0. Such observation demonstrates that SODCA-IRL can promote the formation of long-term memory. The new nodes of shape and color, which are generated in practice process for error correction, are still going through the forgetting process. As associative nodes' memories are just adapted in the practice process, they are not as stable as shape's memories. Some associative nodes with high memories, $f < 0.03$ and $v > 1.05$ have received positive rewards and formed long-term memories. Whereas, some old associative nodes have not been activated yet as their forgetting factors still keep initial values. They would be forgotten in the next iteration because their memories have been lower than the forgetting threshold $M_{\min}$. Whereas, these new associative nodes with initial forgetting factors but high memories would not be deleted in short term. There are two nodes with extremely high f in Fig. 10(b). Correspondingly, their memories and v are also very small. The reason is that they have received a negative reward and would be quickly forgotten in the next iteration. Therefore, SODCA-IRL has abilities to remember correct knowledge and forget incorrect representations.

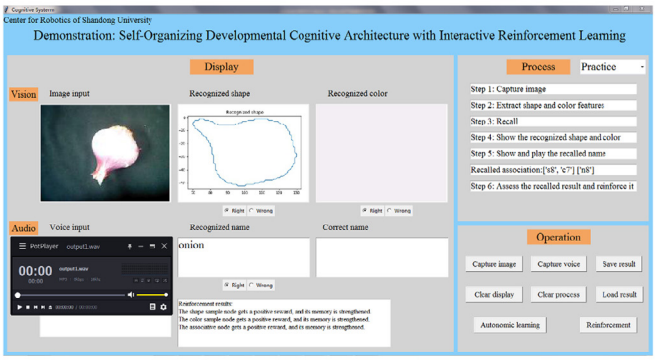
We compare the similarity thresholds of shape sample nodes and color sample nodes after learning and after practice separately.



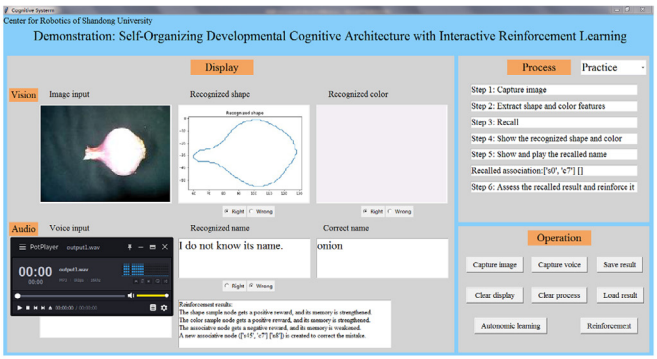
(a). The trainer gives incorrect advice to the shape.



(b) SODCA-IRL recognizes the onion by new knowledge.

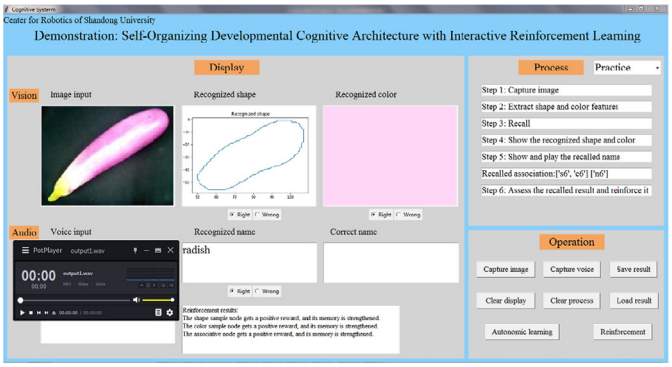


(c) SODCA-IRL recognizes the onion by old knowledge.

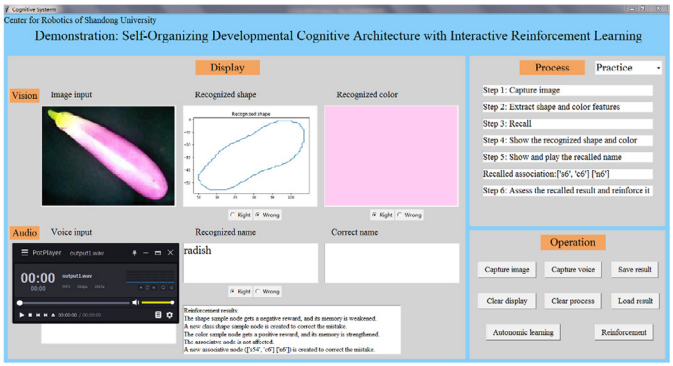


(d) SODCA-IRL forgets the name due to human mistake.

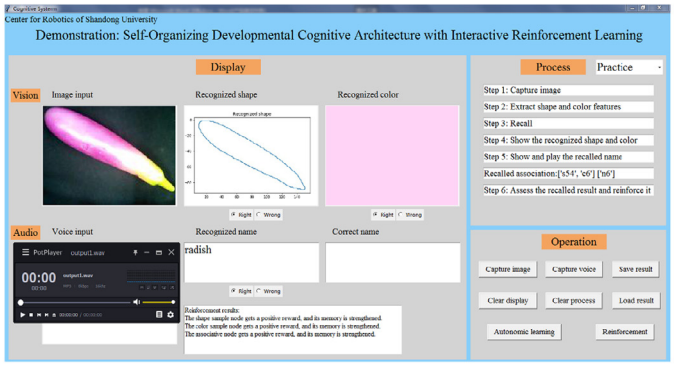
Fig. 12. An example that the trainer gives an incorrect reward to the correct shape.



(a) The trainer gives incorrect advice to the shape.



(b) The trainer corrects the mistake afterwards.



(c) SODCA-IRL recognizes the radish correctly.

Fig. 13. An example that the trainer gives a positive reward to the incorrect shape.

As shown in Fig. 11, the numbers of shape nodes and color nodes both decrease and the values of these similarity thresholds are all adjusted obviously through the practice process. In particular, the intra-class similarity threshold *TL* is extended and the between-class similarity threshold *TH* shrinks. This means the architecture tends to be mature, because it chooses to update the network rather than create a new node for the same class features. Therefore, IRL can promote the development of the cognitive architecture by introducing human guidance.

Furthermore, we validate the ability of SODCA-IRL to cope with the trainer's mistakes. A case that the trainer misjudges the correct shape of onion is shown in Fig. 12(a). SODCA-IRL creates a new class shape node and a new association to learn the correct representations. When meeting an onion again, SODCA-IRL may still recognize it correctly by the new association (see Fig. 12(b)) or the old association as shown in Fig. 12(c). But SODCA-IRL may also forget the onion's name due to the incorrect advice in Fig. 12(d). The architecture can assign a new class to the shape and build a new association after the trainer gives correct rewards and name. Fig. 12(d) can also be used to prove that SODCA-IRL is able to solve the mistakes that the trainer gives incorrect advice to name.

Another case that the trainer misjudges the incorrect shape of the radish is also conducted. In Fig. 13(a), SODCA-IRL recognizes the shape of the radish as that of a peanut. However, the trainer gives a positive reward to the shape, which allows SODCA-IRL remember this incorrect concept. When the architecture meets a radish again, it makes the same mistake (see Fig. 13(b)). Then, the trainer gives a negative reward to the shape so that the architecture creates a new class shape sample node and a new association. Finally, SODCA-IRL can recognize the radish correctly, as shown in Fig. 13(c). These experiments demonstrate that our architecture can use the proposed action strategy and interaction with the trainer to solve human mistakes.

5. Conclusion

In this paper, a self-organizing developmental cognitive architecture with interactive reinforcement learning (SODCA-IRL) is proposed for online object concepts learning and error correction through human-robot interaction. In the proposed algorithm, a memory model is designed to integrate the hierarchical self-organizing neural networks with interactive reinforcement learning. The feedback signals from IRL adjust two forgetting factors to control the forgetting speed and memory strength, which causes nodes with high memory retention to be remembered and those with low memory retention to be forgotten. Another unique property of the proposed scheme is the reinforcement strategy for the practice phase, which can help to remember the correct knowledge and quickly forget the incorrect representations. Moreover, it also contributes to coping with human mistakes. Extensive experiments carried on a common dataset elect an appropriate initialization of the forgetting factor f by the comparison with different values. Furthermore, the comparisons with two other cognitive architectures, i.e., PCN and our previous work, are implemented to evaluate the learning effectiveness of SODCA-IRL. The results show that SODCA-IRL could improve the recognition accuracy significantly and reduce the redundancy of network.

Declaration of Competing Interest

The authors have declared that no conflict of interest exists.

Acknowledgment

This work was supported by the National Key Research and Development Program under Grant no. 2018YFB1305803, the Joint Fund of National Natural Science Foundation of China and Shandong Province under Grant no. U1706228 and the Fund of National Natural Science Foundation of China under Grant no. 61673245.

Appendix A

Algorithm 3 SODCA.

Input: receive the input object and extract its shape feature $s_s(t)$ and color feature $s_c(t)$; receive name $n(t)$.

DT-SOINN:

Find the best marching node b_* ($*$ represents s or c).
if b_* dose not exist or $\|s_*(t) - w_{b_*}\|/\|w_{b_*}\| > \varepsilon_H$ **then**
  Create a new class node r_*, and the learned class $c_* = \max(C_*) + 1$ (C_* is a set of cluster).
  Initialize its memory model: $M_{r_*}(0) = 1$, $f_{r_*}(0) = f(0)$, $\nu_{r_*}(0) = \nu(0)$.
else if $\|s_*(t) - w_{b_*}\|/\|w_{b_*}\| < \varepsilon_L$ or $\|s_*(t) - w_{b_*}\| < TL_{b_*}$ **then**
  Update node b_* and its forgetting factors of using (2) and (3). Meanwhile, $c_* = c_{b_*}$.
  else if $2 \cdot TL_{b'_*} > TL_{b_*} + TL_x$ **then** go to Step 6.
    else if $\|s_*(t) - w_{b_*}\| > TH_{b_*}$ **then** go to Step 3 and 4.
      else create a same class node r_* and $c_* = c_{b_*}$. Initialize its memory model as step 4.
      Update the forgetting factors of b_* using (2) and (3), and update t_z of b_*.
    end if
  end if
end if
end if
Update the memory of each node using (1), and delete nodes whose memories are lower than $M_{\min}$.
if a cluster disappears as nodes are forgotten **then**
  Delete corresponding symbol node and associative nodes.
end if
Output the cluster c_* to the visual symbol layer.

LD-SOINN:

Find the best marching node b_n.
if the Levenshtein Distance $L(n(t), w_{b_n}) = 0$ **then** update the node b_n and $c_n = c_{b_n}$.
else create a new name node r_n, and $c_{r_n} = \max(C_n) + 1$.
end if
Output the cluster c_n to the auditory symbol layer.

S-SOINN:

Receive the clusters c_s, c_c and c_n, and encode them as symbols $a_s(t) = s_{c_s}$, $a_c(t) = c_{c_c}$ and $A(t) = n_{c_n}$.
if the nodes representing the symbols exist **then** update these nodes.
else create new symbol nodes for new symbols.
end if
Output $S(t) = \{a_s(t), a_c(t)\}$ and $A(t)$ to the associative layer.

R-SOINN:

Receive $S(t)$ and $A(t)$ from the symbol layers.
if a node b_a can be activated by the state-action pair $\{S(t), A(t)\}$ **then** update the node b_a.
else if $S(t)$ exists but $A(t)$ is not line with the node's auditory part **then**
R-SOINN returns a conflicting signal to solve the problem.
else if $A(t)$ exists but $S(t)$ is not line with the node's visual part **then**
  R-SOINN returns a guidance signal to adjust the learned knowledge in DT-SOINN and S-SOINN.
  else create a new associative node with an initial memory model.
  end if
end if
end if

Appendix B

Algorithm 4 SODCA-IRL.

Input: the shape feature $s_s(t)$ and color feature $s_c(t)$ of the new arriving object; human reward $r_s(t)$, $r_c(t)$, $R(t)$; correct name $n_{correct}$.
Output: recognized shape $a_s(t)$, color $a_c(t)$, and name $A(t)$.
1: Receive input sample $s_s(t)$ and $s_c(t)$ to DT-SOINNs.
2: Find the best matching node b_s and b_c, and transmit their cluster numbers to the visual S-SOINN.
3: Activate symbol nodes to output the shape and color actions $a_s(t)$ and $a_c(t)$.
4: Assemble the state $S(t) = \{a_s(t), a_c(t)\}$ and transmit it to R-SOINN.
5: Find the activated associative node b_a and recall the name to output action $A(t)$.
6: **if** $R(t) = 1$ **then**
7: **if** $r_s(t)$=1 (or $r_c(t)$=1) **then**
8: Update the memory of each node in DT-SOINN using Algorithm 1.
9: Execute the **Update Action** for node b_s (or b_c).
10: **else**
11: Execute the **Correct Action** for node b_s (or b_c).
12: Execute the **Weakening Action** for node b_s (or b_c).
13: **end if**
14: **if** S-SOINN has created a new symbol **then**
15: Create a new associative node and initialize its memory model.
16: **end if**
17: **else**
18: LD-SOINN learns the correct name $n_{correct}$.
19: **if** $r_s(t)$=1 (or $r_c(t)$=1) **then**
20: Update the memory of each node in DT-SOINN using Algorithm 1.
21: Execute the **Reactivated Action** for node b_s (or b_c).
22: Execute the **Update Action** for node b_s (or b_c).
23: **else**
24: Execute the **Correct Action** for node b_s (or b_c).
25: **if** $c_b = 0$ **then**
26: Update the memory of each node in DT-SOINN using Algorithm 1.
27: **else**
28: Execute the **Weakening Action** for node b_s (or b_c).
29: **end if**
30: **end if**
31: Create a new associative node and initialize its memory model.
32: **end if**
33: Update the memory of each node in R-SOINN using Algorithm 1.

References

[1] F. Ingrand, M. Ghallab, Deliberation for autonomous robots: a survey, Artif. Intell. 247 (2017) 10–44.
[2] A. Aly, S. Griffiths, F. Stramandinoli, Metrics and benchmarks in human-robot interaction: recent advances in cognitive robotics, Cogn. Syst. Res. 43 (2017) 313–323.
[3] G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: a review, Neural Netw. 113 (2019) 54–71.
[4] P. Ye, T. Wang, F. Wang, A survey of cognitive architectures in the past 20 years, IEEE Trans. Cybern. 48 (12) (2018) 3280–3290.
[5] A. Lieto, M. Bhatt, A. Oltramari, D. Vernon, The role of cognitive architectures in general artificial intelligence, Cogn. Syst. Res. 48 (2018) 1–3.
[6] A. Lieto, C. Lebiere, A. Oltramari, The knowledge level in cognitive architectures: current limitations and possible developments, Cogn. Syst. Res. 48 (2018) 39–55.
[7] P. Langley, Interactive cognitive systems and social intelligence, IEEE Intell. Syst. 32 (4) (2017) 22–30.
[8] A.F. Morse, A. Cangelosi, Why are there developmental stages in language learning? A developmental robotics model of language development, Cogn. Sci. 41 (2017) 32–51.
[9] D.H. García, S. Adams, A. Rast, T. Wennekers, S. Furber, A. Cangelosi, Visual attention and object naming in humanoid robots using a bio-inspired spiking neural network, Robot. Auton. Syst. 104 (2018) 56–71.
[10] S. Boucenna, D. Cohen, A.N. Meltzoff, P. Gaussier, M. Chetouani, Robots learn to recognize individuals from imitative encounters with people and avatars, Sci. Rep. 6 (2016) 19908.
[11] L. Steels, F. Kaplan, AIBO's first words: the social learning of language and meaning, Evol. Commun. 4 (2002) 3–32.
[12] G.I. Parisi, J. Tani, C. Weber, S. Wermter, Lifelong learning of human actions with deep neural network self-organization, Neural Netw. 96 (2017) 137–149.
[13] F. Shen, Q. Ouyang, W. Kasai, O. Hasegawa, A general associative memory based on self-organizing incremental neural network, Neurocomputing 104 (2013) 57–71.
[14] Y. Xing, X. Shi, F. Shen, J. Zhao, J. Pan, A. Tan, Perception coordination network: a neuro framework for multimodal concept acquisition and binding, IEEE Trans. Neural Netw. Learn. Syst. 30 (4) (2018) 1–15.
[15] M.U. Keysermann, P.A. Vargas, Towards autonomous robots via an incremental clustering and associative learning architecture, Cogn. Comput. 7 (4) (2015) 414–433.
[16] L. Mici, G.I. Parisi, S. Wermter, A self-organizing neural network architecture for learning human-object interactions, Neurocomputing 307 (2018) 14–24.
[17] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (2014) 1492.
[18] Y. Xing, F. Shen, J. Zhao, Perception evolution network based on cognition deepening model-adapting to the emergence of new sensory receptor, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 607–620.
[19] F. Cruz, S. Magg, Y. Nagai, S. Wermter, Improving interactive reinforcement learning: what makes a good teacher? Connect. Sci. 30 (2018) 306–325.
[20] M. Tomasello, The Cultural Origins of Human Cognition, Harvard University Press, Cambridge, MA, US, 1999.
[21] C.S. Tamis-LeMonda, Y. Kuchirko, L. Song, Why is infant language learning facilitated by parental responsiveness? Curr. Dir. Psychol. 23 (2) (2014) 121–126.
[22] M.K. Ho, J. MacGlashan, M.L. Littman, F. Cushman, Social is special: a normative framework for teaching with and learning from evaluative feedback, Cognition 167 (2017) 91–106.
[23] S. Amershi, M. Cakmak, W. Knox, T. Kulesza, Power to the people: the role of humans in interactive machine learning, AI Mag. 35 (4) (2014) 105–120.
[24] E. Ugur, Y. Nagai, H. Celikkanat, Parental scaffolding as a bootstrapping mechanism for learning grasp affordances and imitation skills, Robotica 33 (5) (2015) 1163–1180.
[25] C. Lyon, C.L. Nehaniv, J. Saunders, T. Belpaeme, A. Bisio, K. Fischer, F. Förster, H. Lehmann, G. Metta, V. Mohan, A. Morse, S. Nolfi, F. Nori, K. Rohlfing, A. Sciutti, J. Tani, E. Tuci, B. Wrede, A. Zeschel, A. Cangelosi, Embodied language learning and cognitive bootstrapping: methods and design principles, Int. J. Adv. Robot. Syst. 13 (105) (2016) 1–22.
[26] S.H. Kasaei, M. Oliveira, G.H. Lim, L.S. Lopes, A.M. Tomé, Towards lifelong assistive robotics: a tight coupling between object perception and manipulation, Neurocomputing 291 (2018) 151–166.
[27] L.S. Lopes, A. Chauhan, Scaling up category learning for language acquisition in human-robot interaction, in: Proceedings of the Symposium on Language and Robots, 2007, pp. 83–92.
[28] S. Valipour, C. Perez, M. Jagersand, Incremental learning for robot perception through HRI, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, (IROS), IEEE, 2017, pp. 2772–2777.
[29] H.B. Suay, S. Chernova, Effect of human guidance and state space size on interactive reinforcement learning, in: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, (RO-MAN), IEEE, 2011, pp. 1–6.
[30] A.L. Thomaz, C. Breazeal, Teachable robots: understanding human teaching behavior to build more effective robot learners, Artif. Intell. 172 (2008) 716–737.
[31] E. Senft, P. Baxter, J. Kennedy, S. Lemaignan, T. Belpaeme, Supervised autonomy for online learning in human-robot interaction, Pattern Recognit. Lett. 99 (2017) 77–86.
[32] H.V. Hasselt, M.A. Wiering, Reinforcement learning in continuous action spaces, in: Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, (ADPRL), IEEE, 2007, pp. 272–279.
[33] S.J.L. Knegt, M.M. Drugan, M.A. Wiering, Learning from Monte Carlo rollouts with opponent models for playing Tron, in: Proceedings of the 10th International Conference on Agents and Artificial Intelligence, (ICAART), Springer, 2018, pp. 105–129.
[34] K. Huang, X. Ma, R. Song, X. Rong, X. Tian, Y. Li, An autonomous developmental cognitive architecture based on incremental associative neural network with dynamic audiovisual fusion, IEEE Access 7 (2019) 8789–8807.
[35] D.L. Leottau, J. Ruiz-del-Solar, R. Babuška, Decentralized reinforcement learning of robot behaviors, Artif. Intell. 256 (2018) 130–159.
[36] M. Mozafari, S.R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, M. Ganjtabesh, First-spike-based visual categorization using reward-modulated STDP, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 6178–6190.
[37] T.-H. Teng, A.-H. Tan, J.M. Zurada, Self-organizing neural networks integrating domain knowledge and reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. 26 (5) (2015) 889–902.
[38] B. Chen, A. Zhang, L. Cao, Autonomous intelligent decision-making system based on Bayesian SOM neural network for robot soccer, Neurocomputing 128 (2014) 447–458.
[39] D.C.D.L. Vieira, P.J.L. Adeodato, P.M. Goncalves, A temporal difference GNG-based approach for the state space quantization in reinforcement learning environments,, in: Proceedings of the 25th International Conference on Tools with Artificial Intelligence, (ICTAI), IEEE, 2013, pp. 561–568.
[40] J.M.J. Murre, J. Dros, Replication and analysis of Ebbinghaus' forgetting curve, PLoS One 10 (7) (2015) e0120644.
[41] H. Ebbinghaus, Memory: A Contribution to Experimental Psychology, Teachers College Press, New York, NY, US, 1913.
[42] R.V. Mayer, Assimilation and forgetting of the educational information: results of imitating modelling, Eur. J. Contemp. Educ. 6 (4) (2017) 739–747.
[43] A.-H. Tan, FALCON: a fusion architecture for learning, cognition, and navigation, in: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, 2004, pp. 3297–3302.
[44] A.L. Thomaz, C. Breazeal, Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance,

in: Proceedings of the 21st National Conference on Artificial intelligence, (AAAI), 2006, pp. 1000–1005.
[45] F. Cruz, S. Magg, C. Weber, S. Wermter, Training agents with interactive reinforcement learning and contextual affordances, IEEE Trans. Cogn. Dev. Syst. 8 (4) (2016) 271–284.
[46] F. Cruz, G.I. Parisi, S. Wermter, Multi-modal feedback for affordance-driven interactive reinforcement learning, in: Proceedings of the International Joint Conference on Neural Networks, (IJCNN), 2018, pp. 1–8.
[47] S.K. Kim, E.A. Kirchner, A. Stefes, F. Kirchner, Intrinsic interactive reinforcement learning - using error-related potentials for real world human-robot interaction, Sci. Rep. 7 (2017) 17562.
[48] W.B. Knox, P. Stone, Framing reinforcement learning from human reward: reward positivity, temporal discounting, episodicity, and performance, Artif. Intell. 225 (2015) 24–50.
[49] G. Li, S. Whiteson, W.B. Knox, H. Hung, Social interaction for efficient agent learning from human reward, Auton. Agents Multiagent Syst. 32 (1) (2018) 1–25.
[50] L. Cohen, A. Billard, Social babbling: the emergence of symbolic gestures and words, Neural Netw. 106 (2018) 194–204.
[51] P.M. Yanik, J. Manganelli, J. Merino, A.L. Threatt, J.O. Brooks, K.E. Green, I.D. Walker, A gesture learning interface for simulated robot path shaping with a human teacher, IEEE Trans. Hum. Mach. Syst. 44 (1) (2014) 41–54.
[52] H. Montazeri, S. Moradi, R. Safabakhsh, Continuous state/action reinforcement learning: a growing self-organizing map approach, Neurocomputing 74 (2011) 1069–1082.
[53] A.J. Smith, Applications of the self-organising map to reinforcement learning, Neural Netw. 15 (2002) 1107–1124.
[54] F. Cruz, G.I. Parisi, J. Twiefel, S. Wermter, Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, (IROS), IEEE, 2016, pp. 759–766.
[55] H. Shao, Fading model of drivers' short-term memory of traffic signs, in: Proceedings of the 2010 International Conference on Machine Vision and Human–Machine Interface, (MVHI), IEEE, 2010, pp. 704–707.
[56] J.T. Wixted, E.B. Ebbesen, Genuine power curves in forgetting: a quantitative analysis of individual subject forgetting functions, Mem. Cogn. 25 (5) (1997) 731–739.
[57] S.G. Hu, Y. Liu, T.P. Chen, Z. Liu, Q. Yu, L.J. Deng, Y. Yin, S. Hosaka, Emulating the Ebbinghaus forgetting curve of the human brain with a NiO-based memristor, Appl. Phys. Lett. 103 (133701) (2013) 1–4.
[58] L. Li, Sentiment-enhanced learning model for online language learning system, Electron. Commer. Res. 18 (1) (2018) 23–64.
[59] N. Censor, D. Sagi, Benefits of efficient consolidation: short training enables long-term resistance to perceptual adaptation induced by intensive testing, Vis. Res. 48 (7) (2008) 970–977.
[60] B. Zhou, B. Zhang, Y. Liu, K. Xing, User model evolution algorithm: forgetting and reenergizing user preference, in: Proceedings of the 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, (iThings/CPSCom), IEEE, 2011, pp. 444–447.
[61] N. Cowan, S. Saults, L. Nugent, The ravages of absolute and relative amounts of time on memory, in: Proceedings of the Nature of Remembering: Essays in Honor of Robert G. Crowder, American Psychological Association, 2004, pp. 315–330.
[62] S.H. Kasaei, A.M. Tome, L.S. Lopes, Hierarchical object representation for open-ended object category learning and recognition, in: Proceedings of the 30th Annual Conference on Neural Information Processing Systems, (NIPS), 2016, pp. 1956–1964.
[63] A. Chauhan, L.S. Lopes, An experimental protocol for the evaluation of open-ended category learning algorithms, in: Proceedings of the 2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems, (EAIS), IEEE, 2015, pp. 1–8.

Ke Huang and received her B.S. degree from the school of Automation Engineering, Qingdao University, Qingdao, China, in 2015. She is currently working toward a Ph.D. with the School of Control Science and Engineering, Shandong University, Jinan, China. Her research interests include autonomous cognitive development of robots, self-organizing incremental neural network, multi-modalities integration and interactive reinforcement learning.

Xin Ma received the B.S. degree in industrial automation and the M.S. degree in automation from Shandong Polytech University (now Shandong University), Shandong, China, in 1991 and 1994, respectively. She received the Ph.D. degree in aircraft control, guidance, and simulation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1998. She is currently a Professor at Shandong University. Her current research interests include artificial intelligence, machine vision, human–robot interaction, and mobile robots.

Rui Song received his B.E. degree in industrial automation in 1998, M.S. degree in control theory and control engineering in 2001 from Shandong University of Science and Technology, and Ph.D. in control theory and control engineering from Shandong University in 2011. He is engaged in research on intelligent sensor networks, intelligent robot technology, and intelligent control systems. His research interests include Medical robots, and the quadruped robots. He is currently an Associate Professor at the School of Control Science and Engineering of Shandong University in Jinan China, and one of the directors of the Center of Robotics of Shandong University.

Xuewen Rong received his B.S. and M.S. degrees from Shandong University of Science and Technology, China, in1996 and 1999, respectively. He received his Ph.D. from Shandong University, China, in 2013. He is currently a senior engineer at the School of Control Science and Engineering, Shandong University, China. His research interests include robotics, mechatronics, and hydraulic servo driving technology.

Xincheng Tian received his B.S. degree in industrial automation and M.S. degree in automation from Shandong Polytech University (now Shandong University), Shandong, China, in 1988 and 1993, respectively. He received his Ph.D. in aircraft control, guidance, and simulation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2000. He is currently a Professor at Shandong University. His current research interests include robotics, mechatrotics, and CNC techniques.

Yibin Li received his B.S. degree in automation from Tianjin University, Tianjin, China, in 1982, M.S. degree in electrical automation from Shandong University of Science and Technology, Shandong, China, in 1990, and Ph.D. in automation from Tianjin University, China, in 2008.From 1982 to 2003. He worked with Shandong University of Science and Technology, China. Since 2003, he has been the Director of the Center for Robotics, Shandong University. His-research interests include robotics, intelligent control theories, and computer control system.