



SOAR Improved Artificial Neural Network for Multistep Decision-making Tasks

Guoyu Zuo¹ · Tingting Pan¹ · Tielin Zhang² · Yang Yang³

Received: 6 June 2019 / Accepted: 5 February 2020 / Published online: 6 March 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Recently, artificial neural networks (ANNs) have been applied to various robot-related research areas due to their powerful spatial feature abstraction and temporal information prediction abilities. Decision-making has also played a fundamental role in the research area of robotics. How to improve ANNs with the characteristics of decision-making is a challenging research issue. ANNs are connectionist models, which means they are naturally weak in long-term planning, logical reasoning, and multistep decision-making. Considering that a small refinement of the inner network structures of ANNs will usually lead to exponentially growing data costs, an additional planning module seems necessary for the further improvement of ANNs, especially for small data learning. In this paper, we propose a state operator and result (SOAR) improved ANN (SANN) model, which takes advantage of both the long-term cognitive planning ability of SOAR and the powerful feature detection ability of ANNs. It mimics the cognitive mechanism of the human brain to improve the traditional ANN with an additional logical planning module. In addition, a data fusion module is constructed to combine the probability vector obtained by SOAR planning and the original data feature array. A data fusion module is constructed to convert the information from the logical sequences in SOAR to the probabilistic vector in ANNs. The proposed architecture is validated in two types of robot multistep decision-making experiments for a grasping task: a multiblock simulated experiment and a multicup experiment in a real scenario. The experimental results show the efficiency and high accuracy of our proposed architecture. The integration of SOAR and ANN is a good compromise between logical planning with small data and probabilistic classification with big data. It also has strong potential for more complicated tasks that require robust classification, long-term planning, and fast learning. Some potential applications include recognition of grasping order in multiobject environment and cooperative grasping of multiagents.

Keywords Cognitive planning · SOAR · Deep neural network · Decision-making · Robot grasping task

Introduction

Various kinds of computational cognitive architectures have been proposed and successfully applied to many cognitive tasks in the last 40 years [1]. A cognitive architecture should be capable of processing information related to specific cognitive functions, such as perception, memory, attention, or decision-making via interactive learning with humans

or the outside environment. Any cognitive computational efforts for these cognitive functions will contribute greatly to opening the black box of the biological cognitive system. Three basic types of cognitive architecture have been proposed and have contributed significantly to the development of robot intelligence, i.e., the symbolic (cognitivist) type, emergent (connectionist) type, and hybrid type, as shown in Fig. 1.

- **The symbolic architectures** have the characteristics of hand-designed symbolic “if-then” production rules, which are logically concluded on the basis of the outside world. These architectures are powerful in logical inference, planning, reasoning, and other symbol-related tasks. However, they inevitably have some weaknesses, such as poor network flexibility and inadequate extensibility, especially in a changing

Guoyu Zuo, Tingting Pan have equal contribution to this work and should be regarded as co-first authors. Tielin Zhang and Yang Yang contributed to this paper equally.

✉ Guoyu Zuo
zuoguoyu@bjut.edu.cn

Extended author information available on the last page of the article.

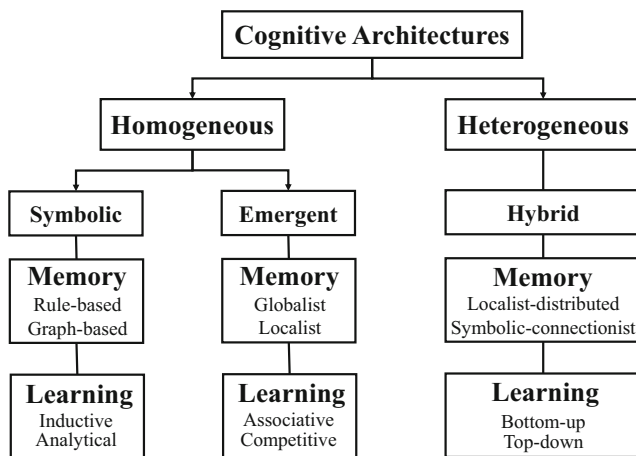


Fig. 1 The three main types of cognitive architectures

environment. ACT-R [2, 3] and SOAR (state, operator, and result) [4, 5] are the two commonly used logically oriented architectures. ACT-R aims to define (or explain) the basic cognitive and perceptual procedures in the brain, which makes it more psychology-related. In contrast, SOAR focuses more on symbolic cognitive processes and usually takes advantage of different types of memory knowledge for better planning or reasoning; thus, it is more broadly used on robot-related cognitive tasks.

- **The emergent architectures** are parallel-computation-type architectures, which are usually based on a large number of nonlinear computational nodes and distributed synaptic weights. They are powerful in input-output mapping and short-term decision-making but weak on the explanation of transparency, slow in learning efficiency, and easily affected by the catastrophic forgetting phenomenon in the subsequent learning of new behaviors [6]. Some of these architectures are deep neural networks (DNNs), which are mainly inspired by the structure of the biological brain, while others, such as SPAUN [7] and HTM [8], are seen as deeper inspirations from the perspectives of both structures and functions.
- **The hybrid architectures** attempt to take advantage of both symbolic and emergent architectures for the better representation of information, long-term planning, and reasoning. Considering that implicit knowledge can be captured by distributed subsymbolic structures such as neural networks, while explicit knowledge has a comparatively transparent symbolic representation, the learning model CLARION [9] uses symbolic and subsymbolic representations for explicit factual knowledge and implicit procedural

knowledge respectively. The model named Leabra [10] uses localist representations for labels and distributed representations of features in its learning procedure.

DNNs are important emergent architectures that have good performance on both spatial information abstraction and temporal information prediction [11]. To date, human-level classification performance on the ImageNet dataset (with millions of natural images) has been achieved by DNNs [12, 13]. Similar progress has been made in the research areas of image recognition and classification [14], object identification [15, 16], sequential frame prediction [17], one-step decision-making [18], memory strengthened efficient learning [19], and so on. In addition, with the development of deep reinforcement learning, DNNs have also been successfully applied to the robot-related tasks, such as motion planning [20, 21], pose estimation [22, 23], 3D environment sensation [16, 24], robot-human interaction [21, 25], and related games, such as Atari 2600 games [26] and DeepMind Go games [27].

However, long-term planning, or even dynamic multistep planning, is the basic request for intelligent robot control. ANNs perform poorly in continuous planning and logical decision making; hence, ANNs cannot handle these kinds of tasks well. Recurrent neural networks (RNNs), which have shown advantages in sequential information processing, are actually designed for short-term temporal prediction and still cannot handle long-term planning tasks. The multistep decision-making task has the challenge of both high-accuracy one-step identification (or classification) and long-term planning, which requires a hybrid architecture to integrate these two special kinds of cognitive abilities well.

In this paper, a SOAR improved ANN (SANN) architecture is proposed, which takes advantage of both the long-term cognitive planning capability of SOAR and the powerful feature detection capability of DNNs. The proposed SANN architecture contains three main modules: the SOAR module for perceptual description, logical reasoning, memory, and long-term planning; the multilayer DNN module for feature selection and decision-making; and the intelligent data fusion module, which is constructed for better information conversion from logic to probabilistic representations and vice versa. In addition, the SANN architecture will intelligently change the inner loops to map different inputs to different shallow or deep ANN modules, which makes possible the integration of architectures with different levels of complexity.

This paper is organized as follows. The “[Related Work](#)” section introduces related work. “[The SANN Architecture](#)” section introduces the SANN architecture and the three main modules. The “[Experiment](#)” section verifies the proposed algorithm in the two types of robot multistep

decision-making tasks. Finally, a conclusion and future outlook are provided in the “**Conclusion**” section.

Related Work

Most DNN architectures try to handle planning or reasoning problems by updating their inner structural connections for better information processing. Yang et al. [28] establish a learning system for robot motion planning in which two different seven-layer CNNs are constructed for pattern recognition and graspability identification. In [15], a real-time CNN approach is proposed for robotic grasp detection that can make a direct regression from the raw RGB-D image to the pose coordinates. To realize a multistep grasp, the input image is changed to an $N \times N$ matrix, and the output is a 7-dimensional vector. In the input matrix, the first channel is a heat map, which represents the graspability probability of the specific region, and the other six channels represent the predicted grasp coordinates for that region. In [29], an end-to-end deep Q-network (DQN) is set to learn a successful strategy directly from high-dimensional sensory inputs by using end-to-end reinforcement learning. A visual manipulation relationship network (VMRN) based on convolutional DNNs is proposed and applied to infer the relationship between objects and operations in [30].

Other alternative methods attempt to strengthen DNN-based cognitive architectures by integrating the additional symbolic modules. A hybrid architecture that contains a perception module, grasping module, and throwing module is proposed by Google; TossingBot is then equipped with the architecture and performs satisfactorily on both the picking up and throwing tasks in the real environment. This architecture innovatively integrates symbolic physics knowledge and DNN architecture and obtains a pickup time that is twice as fast as that of previous cognitive architectures [31]. The model based on selective attention

is constructed by adding the cognitive reasoning module to the networks in the task of smart-phone scenario recognition [32]. DNNs are strengthened by the visual reasoning module based on the SOAR architecture and obtain better performance in human-robot interaction [33] and service robot controlling tasks [34, 35].

Some methods do not use DNNs to deal with planning and decision-making problems. Some attempts are inspired by the human brain mechanism. Zhou et al. [36] design the principle of long-term and short-term hierarchical asynchronous learning based on an updating and storage mechanism that imitates human knowledge. To express the subordinate and nonsubordinate functions in fuzzy information, Liu et al. [37] propose interval-valued linguistic intuitionistic fuzzy numbers (IVLIFNs), which consider the subjectivity of human cognition in decision-making and the difficulty in using numbers to describe intricate and fuzzy details.

The SANN Architecture

As shown in Fig. 2, the SANN model contains three submodules: the SOAR module, the multilayer ANN module, and the data fusion module. First of all, in the SOAR module, the original information is described as long and short program knowledge, and the internal operators are then used to plan and infer the logic sequence. Second, in the multilayer ANN module for decision-making, a shallow-deep network structure is designed specifically to address different difficulties in the real task. A part of the network structure is shared between the modules to improve the utilization of the network. Finally, the data fusion module in the SANN model establishes a connection between the SOAR and the multilayer ANN module. Here, the logical sequences obtained by SOAR are converted to probabilities, and data fusion is realized by combining the probability vector and the original data feature array.

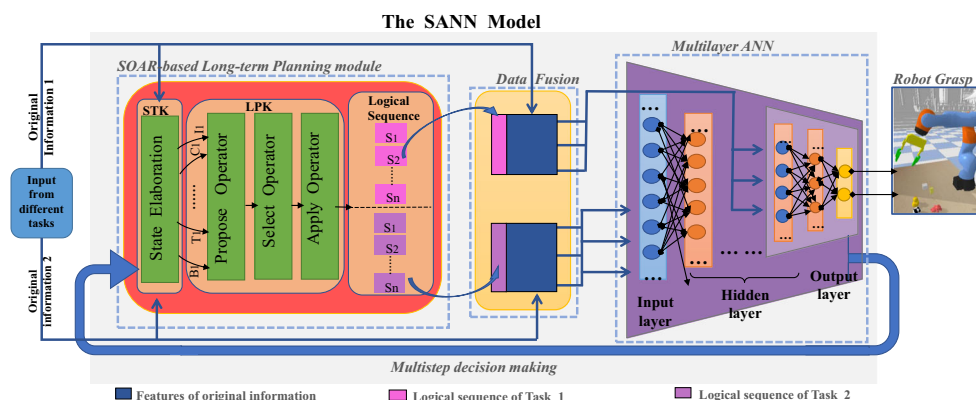


Fig. 2 The SANN architecture

The SOAR Module for Long-term Planning

The cognitive theory underlying SOAR is the problem of space hypothesis (PSH), which contends that nearly all goal-oriented behaviors can be cast as a search procedure through a space of possible states and attempt to achieve a goal. At each step of the PSH, a single operator is selected and then applied to the current state, which leads to the internal updates of the state and the request for a new operator. Complex activities such as planning can also be seen as decomposable procedures of PSH, which contains a sequence of selections or operators. Here, the role of the SOAR module that we introduced is to provide long-term logical planning and to provide logical sequences for robotic behavior decisions in different environments.

Figure 3 shows the functional compositions of SOAR, where S_i represents the current problem-solving state; the operator, represented by O_i , is the specific transition of the state; and G_i refers to the desired goal of the problem-solving activity or the goal of the logical reasoning tasks.

In the SOAR module, there are two different types of working memory for describing and storing various kinds of knowledge: short-term memory knowledge (SMK) for the state set $\{S_i, i \in N\}$ and symbolized long-term procedural knowledge (LPK) for the operator set $\{O_i, i \in N\}$. The two memory types will be integrated as a symbolic graph structure of SOAR. The SMKs and LPKs not only influence but also depend on each other. On the one hand, the state elaborations can indirectly affect the selection and application of the operators by creating the knowledge that matches the application rules. On the other hand, the operators will further update the predefined state conditions

with regulations. When the designed state of the WME is satisfied with the “if-then” production rules, then the LPK will be matched and updated by the execution operators, showing the logical programming and long-term memory characteristics of SOAR.

The logical planning process for SOAR solving problems is equivalent to the process for updating and changing the current state S_i until it reaches the target G_i , in which various operations of the operator O_i are utilized. We refer to the above process as a planning cycle of SOAR, shown as the left part of Fig. 3. The SOAR planning proceeds through several logic cycles, and each cycle has five phases. However, only four planning steps are taken in our model. Figure 4 shows a simple SOAR planning algorithm.

- **Input:** The mechanism called “input functions” is provided in SOAR to receive information from the real or simulated environments. All inputs are represented as substructures of the “I/O” attribute that is in the working memory’s top-level state. We use an attribute to “input-link” from the “I/O” object of SOAR, and the values of the “input-link” are identifiers whose augmentations are the complete set of input working memory elements (WMEs), such as vision-input-link, text-input-link, and other input-links related to the external environment.
- **State elaboration:** In the long-term planning cycle of the SOAR module, this step changes the perceptual inputs obtained from the environment to the SOAR state; that is, SOAR’s internal representation is used to symbolize all the input information. All knowledge in the “state elaboration” step is stored in the WM’s SMK. The WME is constructed as the basic unit of

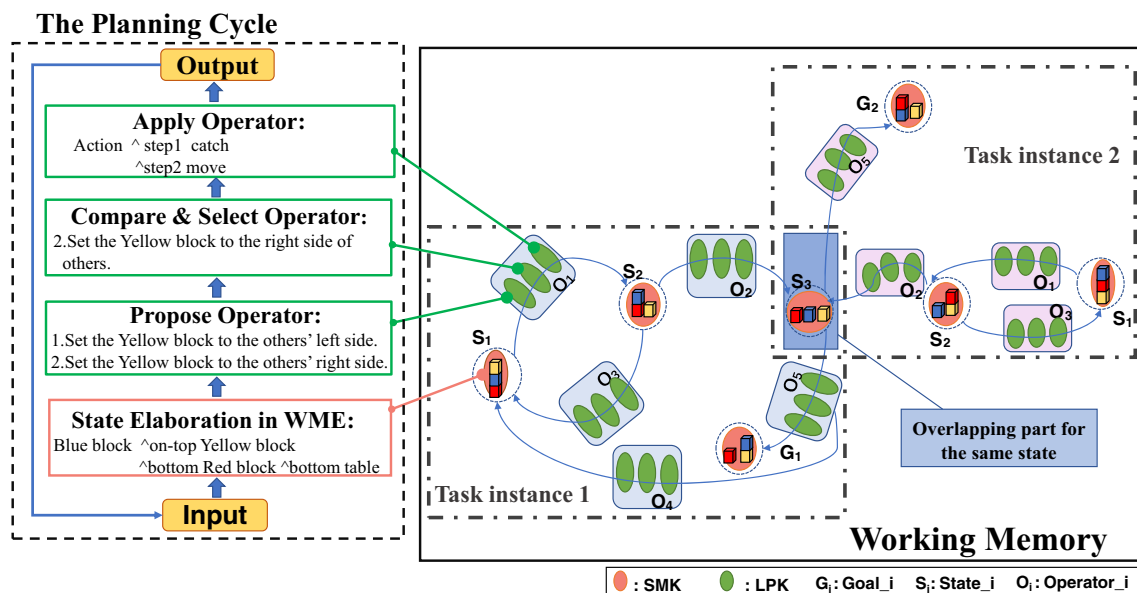


Fig. 3 The functional compositions of SOAR

Fig. 4 A simple SOAR planning process

<p>SOAR while (Halt is false) Cycle; Cycle Input; State Elaboration; Operation in Operator Proposal Phase ; Compare Phase ; Application Phase ; Output; Input while (attempt to receive data from external environment) Create Input Functions; Create Input-Links; State Elaboration while (some external data are input) Create New State; Create New WM-element; Operation in Operator</p>	<p>Proposal Phase while (some i-supported productions are waiting to fire or retract) Fire Newly Matched Productions; Retract Newly Unmatched Productions; Compare Phase for (each state in the stack, starting with the top-level state) until (a new decision is reached) Evaluate Operator Preferences; /* for the state being considered */ if (one operator preferred after preference evaluation) Select New Operator; else /* could be no operator available or */ Create New State; /* unable to decide between more than one */ Application Phase while (some productions are waiting to fire or retract) Fire Newly Matched Productions; Retract Newly Unmatched Productions; Output while (attempt to effect changes in the external environment) Create Output Functions; Create Output-Links;</p>
--	--

the working memory to save different SMKs or LPKs based on different specific subprocedures in the tasks. The WME has the following form:

$$\{identifier, attribute, value\} \rightarrow identifier \wedge attribute = value \quad (1)$$

An object related to the task can be represented as a set of WMEs with the same first identifier. In addition, similar knowledge in different tasks will share the same WME subgraph module for better information representation.

- **The operation of operators:** For a task with the goal G_i , the transition between different states S_i is achieved by a three-step action on the operator O_i , namely, operator proposal, operator comparison and selection, and operator application. As the first step, one or more candidate operators are proposed. All proposed operators are parallel, and they are triggered by matched “productions” in parallel. The second step of the SOAR planning cycle is to compare the proposed candidate operators to select one or more of them. This selection can be completed via the production rules to test the proposed operators and the current state and then to create some preferences that are stored in the preference memory. The preferences are used to declare the relative or absolute merits of the candidate operators. The production rules are similar to the “if-then” statement in conventional programming languages. The “if” part of the production is its condition, and the “then” is its output action. When the conditions are met in the current situation, as defined by the working memory, the production is matched and will fire, which means that its actions will be executed, and changes will be made to the working memory. When SOAR solves the

internal problem, it updates and changes the current state by applying the selected operators.

- **Output:** As mentioned above, the “output functions” mechanism is also provided in SOAR for reacting to the external environment. All outputs are represented as the substructure of the “I/O” attribute that is in the working memory’s top-level state. An “output-link” attribute is used for the “I/O” object in SOAR. The values of the “output-link” are the identifiers whose augmentations are the complete set of WMEs, such as logical-output-link, reasoning-output-link, and other output-links related to the decision order.

The Multilayer ANN Module for Robotic Grasping Decision-making

The cognitive process of the human brain generally includes three subprocedures: perceptual recognition, logical reasoning, and decision-making. Among them, decision-making is the cognitive externalization that can be seen as the final output of the whole cognitive process. Generally, human decisions can be divided into two parts: logical decisions in the brain and execution decisions for action. Here, the SANN model established a multilayer ANN module as an imitation of the decision-making procedure of the human brain. We did not address decisions related to behavioral execution.

When the SOAR module is absent from our SANN model, the ANN module can make preliminary decisions by itself. However, the results often seem to be inaccurate and inconsistent with real situations. When the SOAR module is introduced into the SANN model, its long-term planning ability leads to better decision-making performance by the ANN module. The ANN module in SANN provides a judgment decision on the graspability of objects in different

environments. It makes the final decision according to the task target and the SOAR reasoning results.

The multilayer ANN module includes one input layer, some hidden layers, and one output layer. The input information comes from the fusion module. The number of neurons is closely related to the dimension of the feature vectors in the input information. The output layer has two neurons that show the decisions, that is, whether the object can be grasped or is not graspable. As shown in Fig. 5, the multilayer ANN module has a shallow-deep network structure: one is the shallow ANN, and the other is the deep ANN. The shallow network is used to receive the results from the data fusion module to perform the relatively simple classification task. The deep network is used to make decisions about complex tasks. All the components of the shallow structure are part of the deep network structure. The purpose is to save time in designing the network structure and to integrate the components into the same module for various decision tasks.

The Data Fusion Module for Feature Conversion and Combination

The data fusion module serves as a bridge between the SOAR module and the multilayer ANN module, and it also plays an important role in feature conversion and combination. From Fig. 2, we can see that the fusion module has two input sources: the original information from different tasks and the logical sequences obtained by SOAR’s long-term planning. The probability vector, which is the logical expression of the rational planning of decision results, is calculated according to the logical sequence obtained from the planning module. Then, the vector is combined with the feature vectors of the original data to complete the fusion.

The raw data can be regarded as a feature array of $M \times N$, which is composed of M N -dimensional samples.

The SOAR module can obtain several logical planning sequences related to the decision results of the target. The fusion module calculates the corresponding logical probability. The same samples may result in different logical sequences for the target decision. The probability of the target is as follows:

$$P_{\text{target}} = \sum_{i=1}^R p(i) \tag{2}$$

where R represents the number of sequences corresponding to the sample obtained by the SOAR reasoning. And the probability of each logical sequence is as follows:

$$p = \left(\frac{1}{N_{dr}}\right)^{T_o-1} \tag{3}$$

where N_{dr} is the number of categories of the target decision results, and T_o is the logical execution order of the target object in a logical sequence.

The calculated logical probability vector of $M \times 1$ is directly combined with the feature array of $M \times N$. The fused results of $M \times (N + 1)$ are input into the multilayer ANN module for decision-making.

Experiment

Two robotic grasping experiments as shown in Fig. 6 were conducted to verify the proposed SANN model. The first experiment aimed to evaluate the robotic graspability in the simulated multiblock environment. The second one was performed in an updated version of the first experiment as we shifted the task scenario to the actual situation. Then, the SANN model was used to determine the graspability of the target coffee cup by the robot. Similar to the psychological judgment and thinking of human beings before performing a certain behavior, the expected behavior is logically reasoned

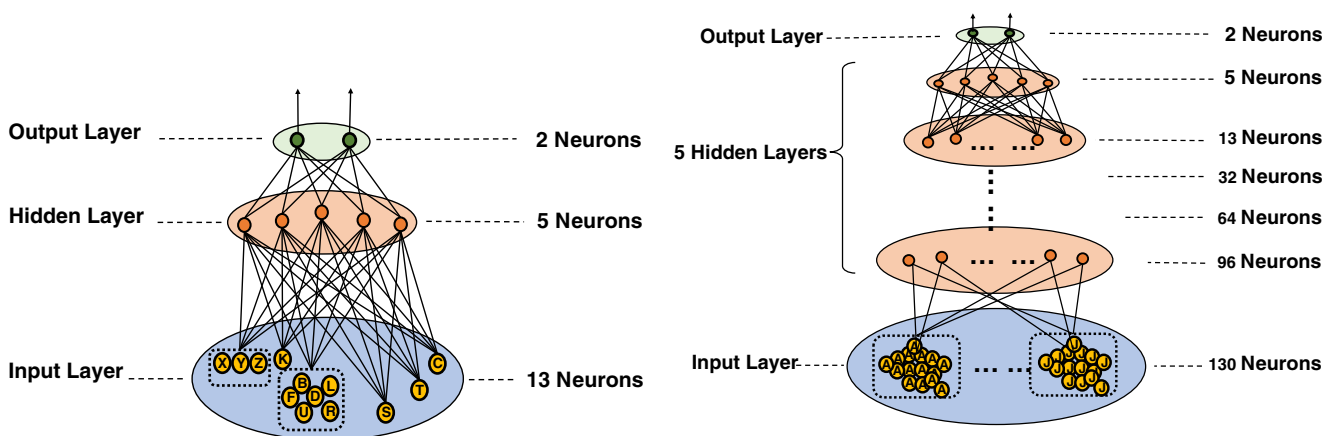


Fig. 5 The schematic diagrams of the shallow ANN for the simulated task (left) and the DNN for the real-scenario task (right)

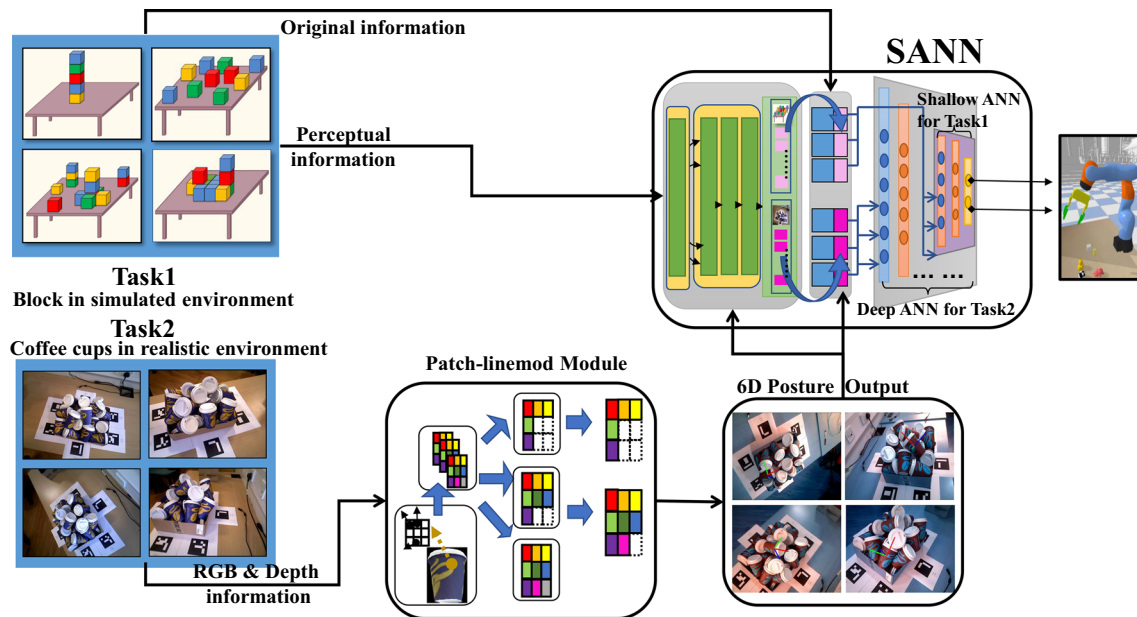


Fig. 6 Both simulated and realistic experiments were designed for the SANN architecture

in relation to the current state of the target to realize the appropriate cognition before the behavior is output.¹

For the robot, the graspability of the object is also a state prejudgment, which is common in humans' actual grasping operations. Humans carry out the analysis of the environment, objects, and even tasks before the grasping action is purposefully performed. Especially when grabbing a specific object in a multiobject environment, logical reasoning and cognition of the relationship between the objects are necessary. Only in this way can humans perform reasonable and effective behavior planning and decision-making. If the robot is expected to have a human-like thinking process and cognitive psychology, then we need to add cognitive planning capability to the robot before decision-making. The SANN can help the robot obtain this ability.

Graspability Identification for the Multiblock Task in the Simulated Environment

In the simulated experiment, the decision and judgment of SANN were tested on the robotic graspability of blocks. Cube blocks of sizes $5 \times 5 \times 5$ and $10 \times 10 \times 10$ were used as the task objects in this experimental scenario. Several cubes (up to 26) that were randomly selected from 26 cubes were placed on the table and arranged in three different ways. The two datasets D1 and D2 are constructed based on the individual block and the whole image scenario, respectively.

- **Scattered mode (C1):** The blocks were placed in any position on the table randomly and discretely. There is no mutual stacking relationship between them.
- **Single-column mode (C2):** The blocks on the table were all arranged in a column (one on top of the other). There was a single and repeated stacking relationship between them.
- **Complex mode (C3):** There was a more complicated positional relationship of the blocks than in the C1 and C2 modes. The relationship between the blocks and the stacking situations of different blocks were often complex and diverse.
- **Block dataset (D1):** In this dataset, each sample contained the features of a specific block in an arbitrary arrangement, including the characteristics of the block and the relationship between the different blocks.
- **Scenario dataset (D2):** In this dataset, each sample was a scenario image, including all the features of the blocks in an arbitrary arrangement.

The block scenario of any one of the arrangement modes can be regarded as a set of input data for the SANN model. Meanwhile, to show the relationship between the object and its features in each scenario, we used the selected 13 features of the object block to describe its feature attributes. Table 1 shows different feature attributes and their values.

Figure 7 is the visualization graph for the cognitive reasoning procedure in the SOAR module. The descriptors are divided into two branches: input information and output information. In the figure, S is the root node of the overall

¹<https://github.com/thomasaimondy/SoarImprovedANN>

Table 1 Descriptor information in the multiblock task

13 Descriptor Features in WM (WME: identifier, attribute, value)

Attribute	X	Y	Z	Distance	Size	Color	Target	Left, right, front, backward, up/down
Value	0–130	0–130	0–130	0–130	5	1 Red	1 Block_A	<1 None/table
						2 Yellow	2 Block_B	1 Block_A
						3 Blue	3 Block_C	2 Block_B
						4 Green	4 Block_D	3 Block_C
					10
						24	Block_V	24 Block_V
						25	Block_W	25 Block_W
						26	Block_X	26 Block_X

state description; I/O includes the output information O_1 and the input information I_1 ; C_1 represents various objects; R_1 shows the task targets; B_1 and T_1 represent the objects and tables, respectively; and L_1 is used to express the locations of objects.

The shallow multilayer network in SANN is constructed for this simulated experiment. The input layer contains 15 neurons, which can be seen as 15 features: one is a logical

item of long-term planning, one is the image ID to which the block belongs, and the remaining features correspond to 13 features of the block. The hidden layer has five neurons. The output layer has two neurons, the same as the number of categories of tasks, i.e., graspability.

We selected 10,000 data points as the training set to train the ANN and selected 2000 data points as the test set to conduct multiple iterative experiments both with and

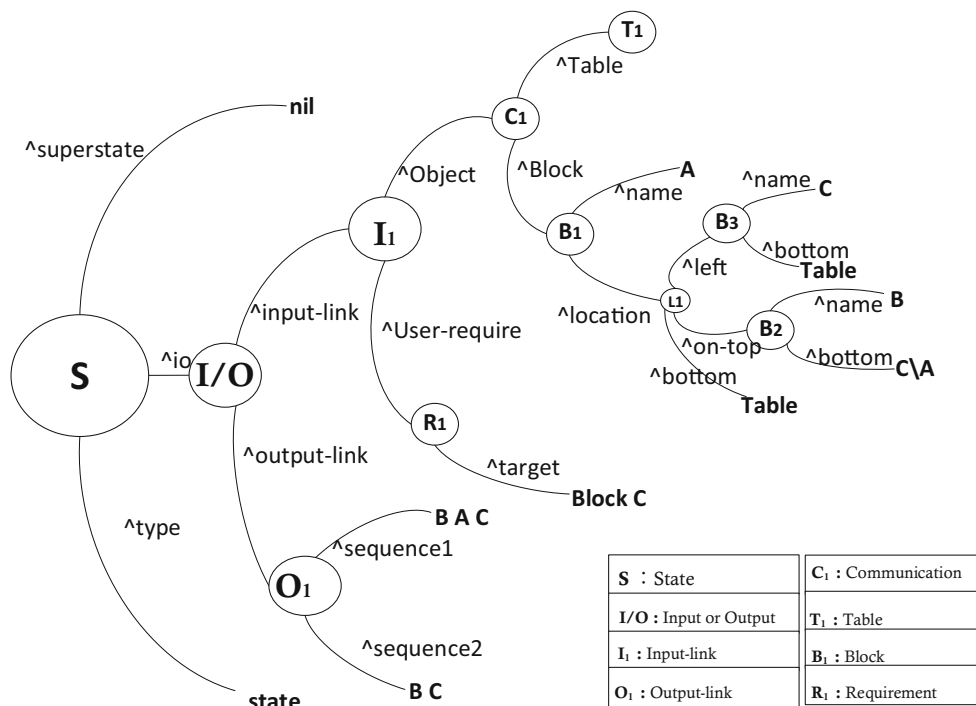


Fig. 7 Descriptors of cognitive reasoning in SANN

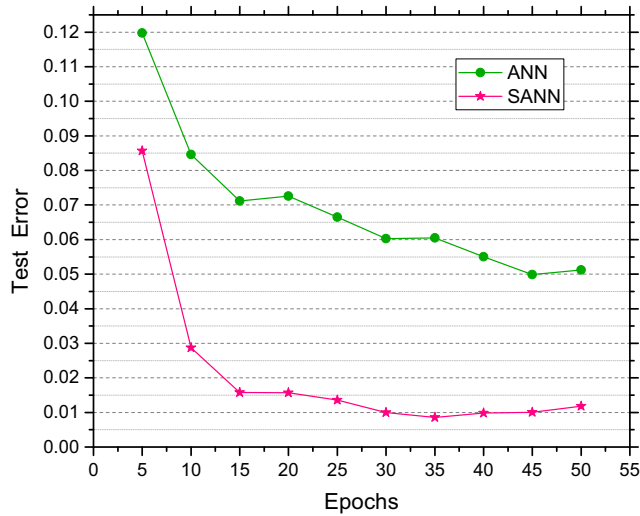


Fig. 8 Test errors of D1 in C3. The overall error rates of our SANN model represented by the pink line are significantly lower than those of the standard ANN represented by the green line. This shows that the combination of long-term planning results based on SOAR with traditional ANN is helpful to improve the accuracy of multistep decision-making

without SOAR reasoning. Figure 8 shows the difference (error rate) between the predicted results and the actual labeled results. In the figure, the D1 dataset is used in the simulated environment. The C1 and C2 modes are relatively simple, and it is easy to make their logical judgments, so we show the experimental results of D1 only in the complex mode of C3. The test error can be predicted by the neural network, which is represented as the proportion of the

wrong predicted samples over the actual labeled samples of the test set.

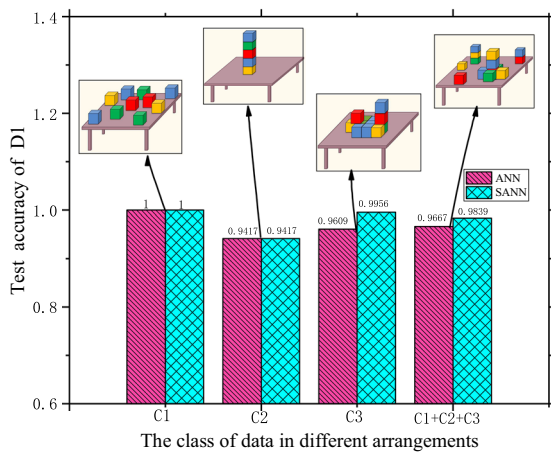
$$Test_{error} = \frac{Number(Test\ Sample_{wrong})}{Number(Test\ Sample)} \tag{4}$$

As shown in Fig. 9a, the experimental results on D1 show that the SANN model has higher decision-making accuracy than the standard ANN. For multitarget scenarios in the complex mode, the accuracy of our SANN model reaches 99.56%, 3% higher than the standard ANN without long-term planning.

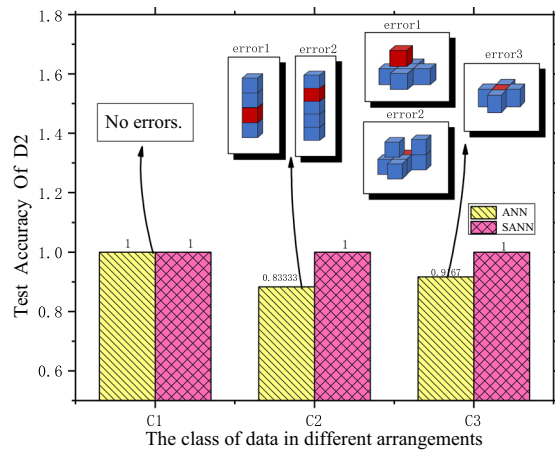
Figure 9 b shows the experimental results on D2. With the support of long-term planning-based SOAR, the judgment accuracy significantly improves. The performance of SANN improves more in the more complicated conditions of C2 and C3 than in simple conditions such as C1.

Graspability Identification for the Multicup Task in the Real Scenario

To verify the SANN model in the real scenario, a class of samples was selected from the Doumanoglou dataset [38], which is the dataset commonly used in the SIXD Challenge [39]. The Doumanoglou dataset contains two types of items, takeaway coffee-cups and juice boxes, and the training set contains 2376 RGB images of a single object and 2376 depth images. For the test set, different quantities of coffee cups were randomly placed in a cardboard box. The same placement scenario contains multiple RGB images and depth images from different



(a) Test accuracy of D1



(b) Test accuracy of D2

Fig. 9 Test accuracy of datasets in different conditions

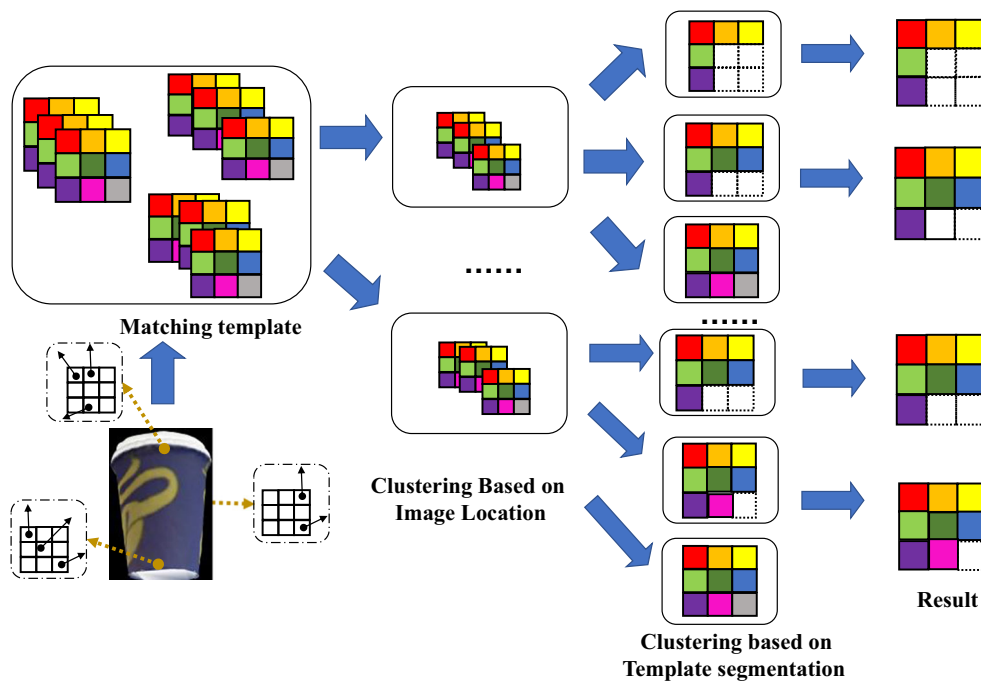


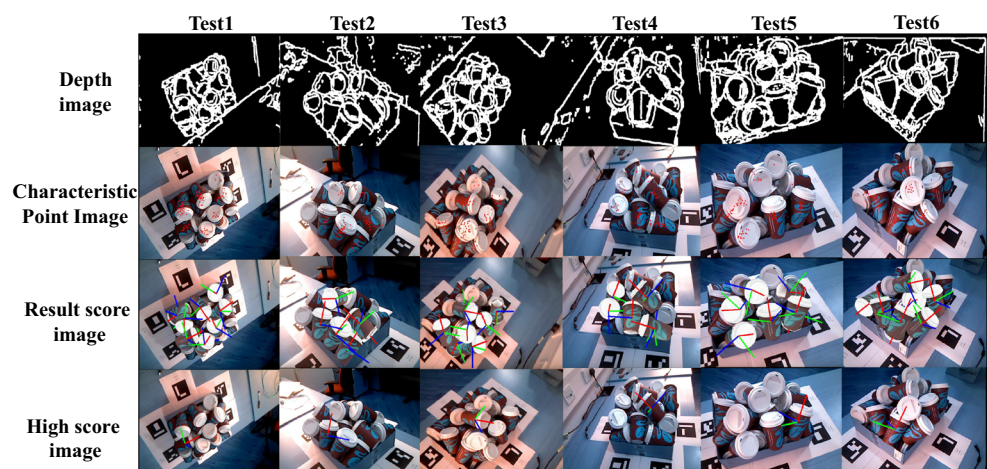
Fig. 10 The clustering process of Patch-LineMod

angles, from which 56 images were selected as our test set. Images with low-quality ground-truth poses were removed from the dataset, and the ground-truth poses for the remaining images were refined.

In the experiment, the pose estimation method for practical application was used to enable the SANN model to be applied to the real scenario. LineMod [40, 41] is a classical 6D pose estimation algorithm that can solve the problem of real-time detection and location of 3D objects

against complex backgrounds. However, as a template-based algorithm, LineMod requires a large number of templates and cannot recognize multiple targets in complex scenarios. Therefore, in view of the multiobject application background of the SANN model, we used the updated template clustering algorithm Patch-LineMod to eliminate the mismatching results according to the size of the clustering. The clustering process of Patch-LineMod is shown in Fig. 10.

Fig. 11 The output of the 6D posture estimation. The first row is the depth map of the test image. After processing, our method labels each object with the feature points shown in the second row. The third row shows the results of all the estimated poses of detected objects. The fourth row is the results of the highest pose estimation score



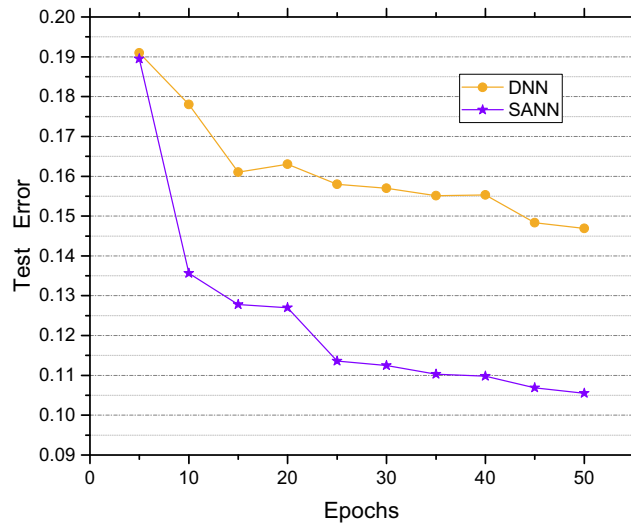


Fig. 12 Test errors of the real scenario with the standard DNN and SANN architectures

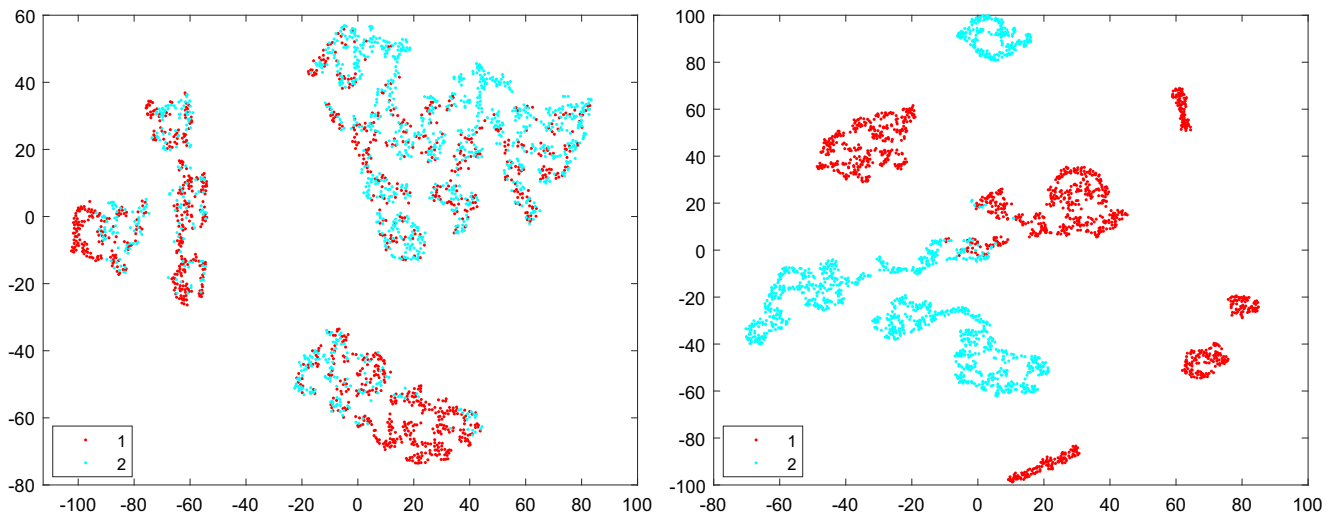


Fig. 13 Visualization of the input layer of standard ANN (left) and SANN (right) with t-SNE

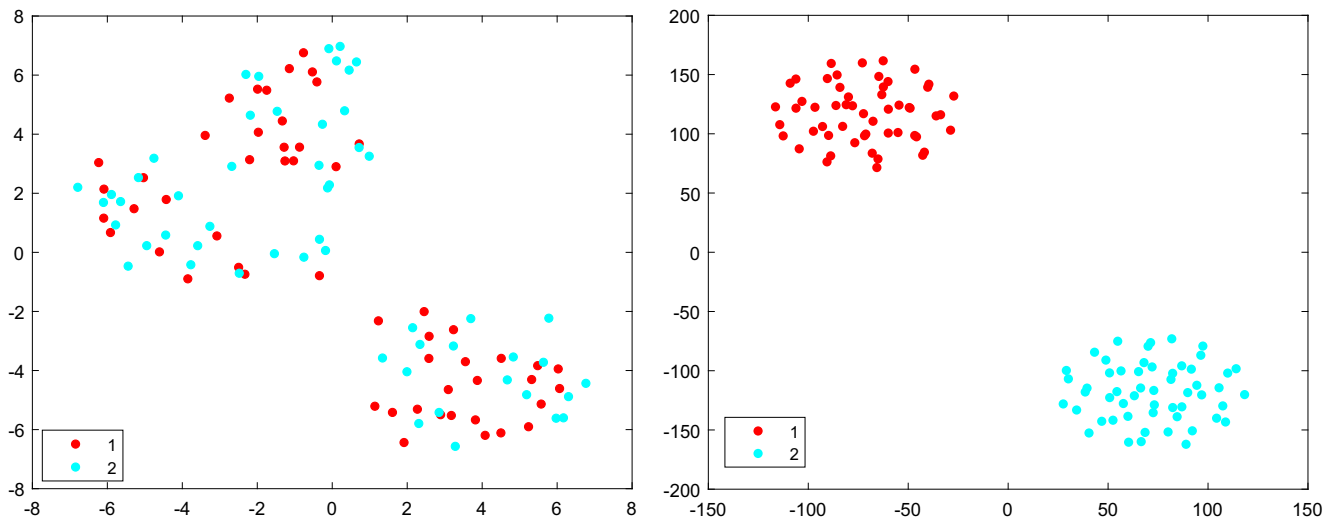


Fig. 14 Visualization of the hidden layer of standard ANN (left) and SANN (right) with t-SNE

A preprocessing module using the Patch-LineMod method² was used to estimate the pose of the object and to identify each object in the image. Figure 11 shows the identification results for the coffee cups in the Doumanoglou dataset using the Patch-LineMod module. After the pose estimation process, a two-dimensional 13*10 image corresponding to the image was generated. Each row of the array contained 13 pose estimation features: three position features, nine rotation features, and one fractional feature. The number 10 indicates that up to 10 objects were selected from a single sample image for calculation and judgment. Figure 12 shows the results of the SANN model judgment of the real image after pose calculation. It can be seen from the figure that the test error of the decision result (purple line) with our SANN model is significantly lower than that of the decision result (yellow line) with the standard ANN without long-term planning.

Analyses of the Performance of SANN

The performance of SANN is further analyzed with the contribution of the SOAR module to the whole architecture. Here, we use the “t-distributed Stochastic Neighbor Embedding” (t-SNE) [42, 43], which is a nonlinear dimensionality reduction algorithm for mining high-dimensional data to map multidimensional data even to two or three dimensions, to analyze the information in different ANN layers.

Figure 13 shows the comparative analyses of the input layers with and without an additional planning module.

The right side is the results of SANN, while the left side is the results of the standard ANN. It is evident that the clustering performance of SANN is far better, and most samples are well classified. In contrast, the standard ANN architecture cannot separate the graspable and nongrasable objects from each other. This results show that the simple input-output mapping classifier cannot effectively handle multistep decision-making tasks, while the SOAR planning module is powerful in performing logical analysis. Figure 14 shows the t-SNE results of the hidden layers in SANN and standard ANN, from which we also observe the clustering power of the SOAR module.

Conclusion

The multistep decision-making task for a robot is a major challenge for most symbolic or emergent cognitive architectures. Hence, there is a great need for integrative

architecture with the characteristics of high-dimensional feature abstraction, memory storage, long-term reasoning, and planning. We propose a SOAR improved artificial neural network (SANN) architecture to handle this kind of task. The SANN contains three parts: the SOAR module for long-term planning, the data fusion module for feature conversion and combination, and the multilayer ANN module for decision-making. The SOAR module is used for perceptual description, logical reasoning, memory, and long-term planning. The data fusion module calculates the probability of the vector according to the logical sequence and then combines the probability vector with the feature vectors of the original data. The multilayer ANN module is established as an imitation of the decision-making procedure of the human brain.

Multistep decision-making tasks were conducted in both simulation and realistic environments, and the results show the power of the SANN architecture. Our model considers only the decision-making process and not the execution part. Logical planning through environmental information has completed decision-making in the simulated brain. The implementation of behaviors is attempted only through traditional means. In addition, our model can be applied to multiple decision tasks in a complex scenario, such as the judgment of grasping order in a multiobject environment, cooperative grasping, and recognition of multiple agents.

The SANN architecture can be seen as a standard hybrid type cognitive architecture that has successfully integrated both the symbolic (cognitivist) type and emergent (connectionist) type of architecture, and the data fusion module of SANN attempts to make possible the conversion of information from these two sides. How the biological brain integrates these two different types of information is still a mystery. However, a deeper analysis of these two cognitive procedures will provide more hints or inspirations. Our next research will focus on how the logical information could be internally represented in a connectionist network, which may help the robot approach human-level intelligence.

Funding Information This study is funded by the Beijing Natural Science Foundation (No. 4182008), the National Natural Science Foundation of China (No. 61873008), the National Natural Science Foundation of China (No. 61806195), and the Beijing Academy of Artificial Intelligence (BAAI).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

²<https://github.com/thomasaimondy/patchlinemod>


References

- Kotscheruba I, Tsotsos JK. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artif Intell Rev*;40:1–78.
- Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y. An integrated theory of the mind. *Psychol Rev*. 2004;111(4):1036.
- Anderson JR. Human symbol manipulation within an integrated cognitive architecture. *Cogn Sci*. 2005;29(3):313–341.
- Laird JE, Newell A, Rosenbloom PS. Soar: an architecture for general intelligence. *Artif Intell*. 1987;33(1):1–64.
- Laird JE. *The Soar cognitive architecture*. MIT Press, Cambridge. 2012.
- French RM. Catastrophic forgetting in connectionist networks. *Trends Cogn Sci*. 1999;3(4):128–135.
- Eliasmith C, Trujillo O. The use and abuse of large-scale brain models. *Curr Opin Neurobiol*. 2014;25:1–6.
- Hawkins J, Ahmad S. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Front Neural Circ*. 2016;10:23.
- Sun R, Peterson T, Merrill E. A hybrid architecture for situated learning of reactive sequential decision making. *Appl Intell*. 1999;11(1):109–127.
- O'Reilly RC, Wyatte D, Herd S, Mingus B, Jilk DJ. Recurrent processing during object recognition. *Front Psychol*. 2013;4:124.
- Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. 2015;61:85–117.
- He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*; 2015. p. 1026–1034.
- Wang Z, Wang X, Wang G. Learning fine-grained features via a cnn tree for large-scale classification. *Neurocomputing*. 2018;275:1231–1240.
- Dahl GE, Yu D, Li D, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans Audio Speech Lang Process*. 2012;20(1):30–42.
- Redmon J, Divvala S, Girshick R, Farhadi A. You only once: look Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016. p. 779–788.
- Maturana D, Scherer S. Voxnet: a 3d convolutional neural network for real-time object recognition. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE; 2015. p. 922–928.
- Oh J, Guo X, Lee H, Lewis RL, Singh S. Action-conditional video prediction using deep networks in atari games. In: *Advances in neural information processing systems*; 2015. p. 2863–2871.
- Weisz G, Budzianowski P, Su P-H, Gasic M. Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Trans Audio Speech Lang Process (TASLP)*. 2018;26(11):2083–2097.
- Zen H, Sak H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE; 2015. P. 4470–4474. 2015.
- Finn C, Levine S. Deep visual foresight for planning robot motion. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2017. p. 2786–2793. 2017.
- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves Ax, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540):529.
- Ge L, Ren Z, Li Y, Xue Z, Wang Y, Cai J, Yuan J. 3d hand shape and pose estimation from a single rgb image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2019. p. 10833–10842.
- Dong J, Jiang W, Huang Q, Bao H, Zhou X. Fast and robust multi-person 3d pose estimation from multiple views. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2019. p. 7792–7801.
- Huajun Z, Jin Z, Rui W, Tan M. Multi-objective reinforcement learning algorithm and its application in drive system. In *2008 34th Annual Conference of IEEE Industrial Electronics*. IEEE; 2008. p. 274–279.
- Hester T, Vecerik M, Pietquin O, Lanctot M, Piot B, Horgan D, Quan J, Sendonaris A, Osband I, et al. Deep q-learning from demonstrations. In: *Thirty-Second AAAI Conference on Artificial Intelligence*; 2018.
- Ellefsen KO, Torresen J. Self-adapting goals allow transfer of predictive models to new tasks; 2019.
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van denDriesscheG, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M. Mastering the game of go with deep neural networks and tree search. *Nature*. 2016;529(7587):484–489.
- Yang Y, Yi L, Fermuller C, Aloimonos Y. Robot learning manipulation action plans by “watching” unconstrained videos from the world wide web. In: *Twenty-ninth Aaai Conference on Artificial Intelligence*; 2015.
- Volodymyr M, Koray K, David S, Rusu AA, Joel Vx, Bellemare MG, Alex G, Martin R, Fidjeland AK, Georg O. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540):529.
- Zhang H, Lan X, Zhou X, Tian Z, Zhang Y, Zheng N. Visual manipulation relationship network for autonomous robotics. In: *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE; 2018. p. 118–125. 2018.
- Zeng A, Song S, Lee J, Rodriguez A, Funkhouser T. Tossingbot: learning to throw arbitrary objects with residual physics. 2019.
- Chen H-Z, Tian G-H, Liu G-L. A selective attention guided initiative semantic cognition algorithm for service robot. *Int J Autom Comput*. 2018;15(5):559–569.
- Van Dang C, Pham TX, Gil K-J, Shin Y-B, Kim J-W, et al. Implementation of a reusable human-robot interaction task with humanoid robot by connecting soar and ros. *J Korea Robot Soc*. 2017;12(1):55–64.
- Puigbo J-Y, Pumarola A, Angulo C, Tellez R. Using a cognitive architecture for general purpose service robot control. *Connect Sci*. 2015;27(2):105–117.
- Zheng J, Cai F, Chen W, Feng C, Chen H. Hierarchical neural representation for document classification. *Cogn Comput*. 2019;11(2):317–327.
- Zhou K, Wei R, Xu Z, Zhang Q, Lu H, Zhang G. An air combat decision learning system based on a brain-like cognitive mechanism. *Cognitive Computation*. 2019.
- Liu P, Qin X. A new decision-making method based on interval-valued linguistic intuitionistic fuzzy information. *Cogn Comput*. 2019;11(1):125–144.
- Doumanoglou A, Kouskouridas R, Malassiotis S, Kim T-K. Recovering 6d object pose and predicting next-best-view in the crowd. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2016. p. 3583–3592.
- Hodan T, Michel F, Brachmann E, Kehl W, GlentBuch A, Kraft D, Drost B, Vidal J, Ihrke S, Zabulis X, et al. Bop: benchmark for 6d object pose estimation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*; 2018. p. 19–34.
- Hinterstoisser S, Lepetit V, Ilic S, Holzer S, Bradski G, Konolige K, Navab N. Model based training, detection and pose

- estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. Springer; 2012. p. 548–562.
41. Hinterstoisser S, Holzer S, Cagniart C, Ilic S, Konolige K, Navab N, Lepetit V. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: IEEE International Conference on Computer Vision; 2012.
42. Van Der Maaten L. Accelerating t-sne using tree-based algorithms. *J Mach Learn Res.* 2014;15(1):3221–3245.
43. van der Maaten L, Hinton G. Visualizing data using t-sne. *J Mach Learn Res.* 2008;9:2579–2605.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Guoyu Zuo¹  · Tingting Pan¹ · Tielin Zhang² · Yang Yang³

Tingting Pan
pantingting@emails.bjut.edu.cn

Tielin Zhang
tielin.zhang@ia.ac.cn

Yang Yang
1701210376@pku.edu.cn

¹ Faculty of Information Technology, Beijing University of Technology, Beijing, China

² Institute of Automation, Chinese Academy of Sciences, Beijing, China

³ School of Software and Microelectronics, Peking University, Beijing, China