Metadata of the chapter that will be visualized in SpringerLink

Book Title	Biologically Inspired Cognitive Architectures 2023
Series Title	
Chapter Title	A Brain-Inspired Cognitive Architecture (BICA) Approach to the Neurosymbolic Gap
Copyright Year	2024
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Switzerland AG

Corresponding Author	Family Name	**Schneider**
	Particle	
	Given Name	**Howard**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Sheppard Clinic North
	Address	Vaughan, ON, Canada
	Email	hschneidermd@alum.mit.edu
		howard.schneider@gmail.com
	ORCID	http://orcid.org/0000-0001-8052-6448

Abstract	In this paper we consider a brain-inspired cognitive architecture approach to the neurosymbolic gap. The difference in the abilities of artificial neural networks (e.g., excellent perception) and symbolic systems (e.g., excellent logic) can be referred to as the neurosymbolic gap. Most attempts to combine properties of neural networks and symbolic systems are hybrid combinations of these different systems. A brain-inspired cognitive architecture (BICA), the Causal Cognitive Architecture 5 (CCA5), has both connectionist and symbolic properties. This architecture uses spatial navigation maps as the common data structure and requires spatial and temporal binding of inputs, predictive coding, innate knowledge procedures, and the ability to feed back and re-operate on intermediate results. We show how this BICA approach closes the neurosymbolic gap without the need to overtly combine separate symbolic systems and neural networks. As well, given that the BICA model presented is inspired by the mammalian and in particular the human brain, it provides insight into the mechanisms at work in cognition.

Keywords (separated by '-')	Cognitive architecture - Artificial intelligence (AI) - Neurosymbolic

A Brain-Inspired Cognitive Architecture (BICA) Approach to the Neurosymbolic Gap

Howard Schneider(✉)

Sheppard Clinic North, Vaughan, ON, Canada
hschneidermd@alum.mit.edu

Abstract. In this paper we consider a brain-inspired cognitive architecture approach to the neurosymbolic gap. The difference in the abilities of artificial neural networks (e.g., excellent perception) and symbolic systems (e.g., excellent logic) can be referred to as the neurosymbolic gap. Most attempts to combine properties of neural networks and symbolic systems are hybrid combinations of these different systems. A brain-inspired cognitive architecture (BICA), the Causal Cognitive Architecture 5 (CCA5), has both connectionist and symbolic properties. This architecture uses spatial navigation maps as the common data structure and requires spatial and temporal binding of inputs, predictive coding, innate knowledge procedures, and the ability to feed back and re-operate on intermediate results. We show how this BICA approach closes the neurosymbolic gap without the need to overtly combine separate symbolic systems and neural networks. As well, given that the BICA model presented is inspired by the mammalian and in particular the human brain, it provides insight into the mechanisms at work in cognition.

Keywords: Cognitive architecture · Artificial intelligence (AI) · Neurosymbolic

1 The Neurosymbolic Gap and Hybrid Solutions

In this paper we explore a brain-inspired cognitive architecture (BICA) approach to the neurosymbolic gap. We consider how this BICA approach closes the neurosymbolic gap without the need to overtly and often awkwardly combine a symbolic system to an artificial neural network. Given that the BICA approach presented below is inspired by the mammalian and in particular the human brain, it provides insight into the mechanisms at work in cognition and in discussions of consciousness.

In terms of perception, artificial neural networks (ANN's) perform on the level of humans [1]. However, in terms of logically making sense of a problem at hand, especially if training examples are limited, their performance barely compares to the level of a four-year old child [2]. Enhanced deep learning generative transformer models are able to write human prose or generate images in response to text. However, as Leivada, Murphy and Marcus [3] point out, such systems are not able to understand language actually logically at the level of a small child. There is an effective lack of logical abilities in neural network-based artificial intelligence (AI) systems. This difference in the abilities

A. V. Samsonovich and T. Liu (Eds.): BICA 2023, SCI 1130, pp. 1–12, 2024.
https://doi.org/10.1007/978-3-031-50381-8_84

of neural network AI systems and symbolic, logical AI systems is sometimes referred to as the "neurosymbolic gap" [4, 5].

Neurosymbolic AI attempts to combine properties of ANN's and symbolic systems, i.e., neural network learning properties with the ability for symbolic reasoning. Garcez and Lamb [6] review research that attempts to integrate artificial neural networks with logical symbolic systems. Kautz [7] summarizes the field of neurosymbolic AI systems and divides it into six major designs. All of these designs are hybrid systems, i.e., part neural network and part symbolic system. As an example, one of Kautz's groups are "Symbolic[Neuro]" systems which use a neural network recognition method within a symbolic AI system. Cingillioglu [8] classifies neurosymbolic AI from more neural to more symbolic, but again all these are hybrid systems.

Garcez and Lamb [6] note that neurosymbolic AI systems attempt to provide a bridge between localist and distributed representations. Below we describe a different type of non-hybrid neurosymbolic AI system, a brain-inspired cognitive architecture, that does not require or use any such bridge.

2 A Non-hybrid Solution to the Neurosymbolic Gap

In this paper we explore a non-hybrid (i.e., non-overtly hybrid) solution to the neurosymbolic gap that does not combine a typical artificial neural network with a typical symbolic AI system, as the previous solutions described above all use to some extent. The solution which we present is the Causal Cognitive Architecture. This architecture is a brain-inspired cognitive architecture (BICA) [9–16]. Given the existence of spatial maps in the hippocampi in mammals, as well as navigational abilities throughout the vertebrates and many invertebrates, the architecture postulates and requires the use of navigation maps not just for navigation but in the core mechanisms of the architecture. As well, the architecture requires spatial and temporal binding of sensory inputs, predictive coding (errors between what the architecture thought it would sense and actually senses are propagated forward), innate knowledge procedures concerning objects, physics, agents, numbers, and social group members, the ability to feed back and re-operate on intermediate results and optionally core analogical processing of sensory inputs and stored data.

The current version of the architecture, the Causal Cognitive Architecture 5 (CCA5), is shown in Fig. 1. The principal, recurring data element of the architecture is the "navigation map," an example is shown in Fig. 2. The navigation map is not a typical neural network, although it is connectionist in operation. There is not one navigation map in the architecture but millions or billions of them [15].

3 Operation of the Causal Cognitive Architecture 5 (CCA5)

3.1 Cognitive Cycles

Each cognitive cycle, sensory features streaming in from different perceptual sensors are processed by the Input Sensory Vectors Shaping Modules (Fig. 1) and made compatible with the navigation map format (Fig. 2) used internally by the architecture. Each sensory system's inputs are propagated to the Input Sensory Vectors Association Modules

(Fig. 1) where they are spatially bound, i.e., mapped onto a best-matching local (i.e., local to a particular sensory system) navigation map. Then in the Object Segmentation Gateway Module (Fig. 1), objects detected in the sensory inputs are segmented, and visual, auditory, and other sensory features of each segmented object are spatially mapped onto additional navigation maps dedicated to one sensory modality. This represents the first step in spatial object binding [13–16].

As well, a parallel sensory stream has gone through the Sequential/Error Correcting Module (Fig. 1) which converts changes with time into a vector value which is also bound along with the spatial features onto the same navigation maps, effectively representing temporal binding (not shown in Figs. 3 or 4) [14–16].

These single-sensory navigation maps are then mapped onto a best-matching multi-sensory navigation map taken from the Causal Memory Module (Figs. 1, 3 and 4). This represents the second step in spatial and temporal object binding [13–16].

The best-matching local (i.e., for each sensory system) navigation maps and the best-matching multi-sensory navigation maps effectively represent what the architecture expects to see (Fig. 3). The information mapped onto these best-matching local and multisensory navigation maps (Fig. 4) represents effectively changes or errors in what the actual sensory inputs actually were, i.e., an effective predictive coding is occurring rather than memorization of input sensory values. (If there are large differences then new navigation maps will actually be created [14, 15].)

Instinctive primitives and learned primitives are actions to perform on navigation maps, essentially acting as small rules, and also are stored within modified navigation maps. Instinctive primitives are innate knowledge procedures concerning objects, agents, numbers, and social group members. Learned primitives are procedures which are learned by the architecture. Instinctive primitives and learned primitives are selected by a process similar to Fig. 3's best-matching navigation map process matching the sensory inputs as well as results of previous intermediate results in the Navigation Module (Fig. 1). A best-matching instinctive primitive or learned primitive (termed the Working Primitive or WPR) is then applied against the best-matching multisensory navigation map (which has been updated with the changes of the sensory inputs, termed the Working Navigation Map or WNM) in the Navigation Module.

The application of WPR on WNM via the mechanisms (which can actually be quite straightforward) of the Navigation Module (Fig. 1), produces a signal to the Output Vector Association Module (Fig. 1) and then to the external world.

The instinctive primitives are based on the work of Spelke and others [17, 18] showing similar innate procedures in mammalian (mainly human) infants.

Imagine that a robot controlled by a CCA5 architecture is in a forest and has a goal (the Goal/Emotion Module in Fig. 1 will influence this [14–16]) of moving to a certain point. There is a river in front of it. The sensory inputs are processed as described above. In the Navigation Module there is a Working Navigation Map (WNM) shown in Fig. 2, representing water in front of the CCA5 controlled robot. An instinctive primitive related to water is the best-matching instinctive or learned primitive to the sensory inputs, and becomes the Working Primitive (WPR). The operation of WPR on WNM is to avoid water and as a result instead of continuing straight the robot turns left. Then a new cognitive cycle starts again—new sensory inputs are processed through the architecture, the Navigation Module (Fig. 1) produces an output (or not, as described in the next sections), and an output action occurs, and so on.

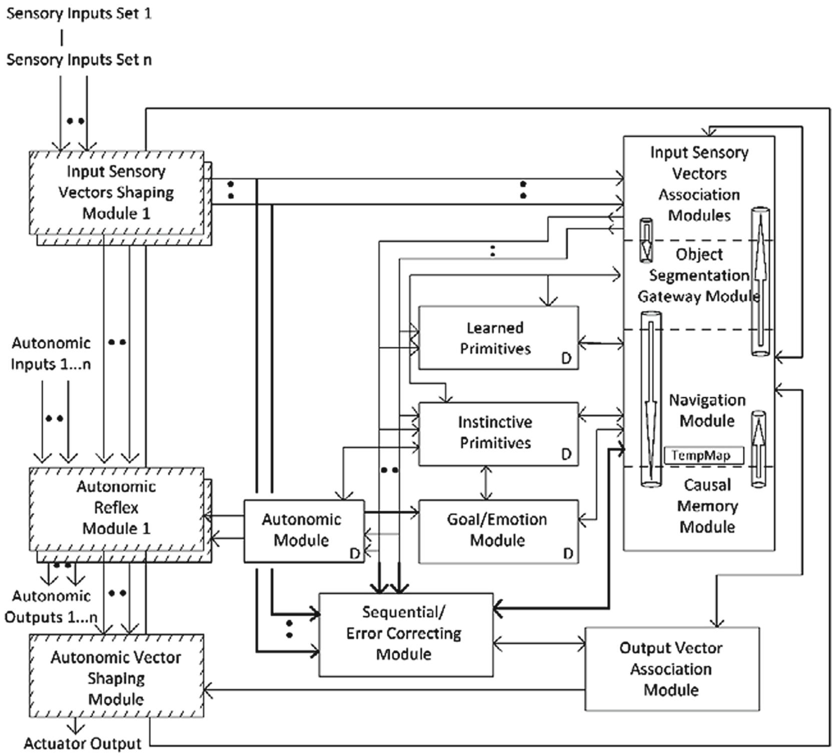

Fig. 1. Causal Cognitive Architecture 5 (CCA5). ("D" in some modules signifies that its properties change as the architecture develops, i.e., with experience and usage.)

3.2 Feedback

Feedback pathways are ubiquitous throughout the CCA5 architecture—states of a downstream module can influence the recognition and processing of more upstream sensory inputs. The differences between the expected sensory input and the actual sensory input are computed and fed forward, and influence the binding of the sensory inputs onto local sensory navigation maps in the Input Sensory Vectors Association Modules as well as the final binding of the local navigation maps onto a multisensory navigation map (Figs. 3 and 4) which becomes the working navigation map in the Navigation Module. This is described above, and is explored in more formal detail in [16].

In the CCA5 the feedback pathway between the Navigation Module and the Input Sensory Vectors Association Modules is enhanced. Normally, if the result of an operation of the Working Primitive (WPR) on the Working Navigation Map (WNM) in the Navigation Module does not produce an actionable output, then no action occurs. Perhaps

solid	solid	solid	water	solid	solid
solid	solid	solid	water	solid	solid
solid	solid	water	water, sound23	solid	solid
solid	solid	water	water	solid	solid
solid	solid	water, link{4574}	solid	solid	solid
solid, iprimitive{8974}	solid	water	solid	solid	solid

Fig. 2. Example of a Navigation Map–the 6 × 6 × 0 spatial dimensions are shown containing sensory features (most visual but "sound23" is auditory) and some links to other navigation maps and to instinctive primitives (i.e., instructions for operations). Solid arrows represent links within the navigation maps. Dashed arrows represent links to cells in other navigation maps.

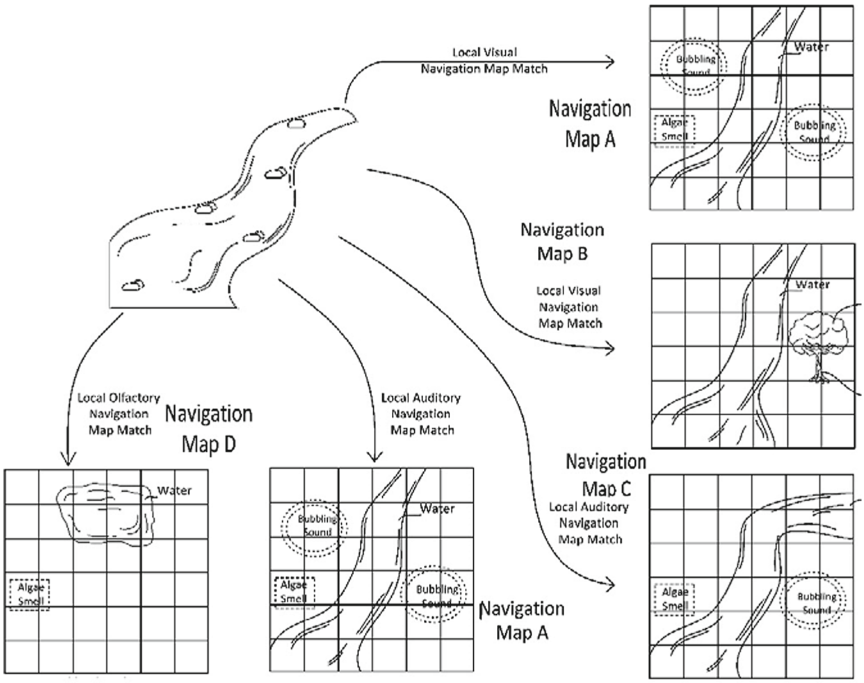

Fig. 3. Matching best-matching local input sensory navigation maps to Causal Memory Module previously stored best-matching multi-sensory maps. The best match is Navigation Map A.

in the next cognitive cycle with different sensory inputs and possibly a different instinctive primitive chosen as the best-matching instinctive primitive (WPR), the Navigation Module will produce an actionable output. However, now with the enhanced feedback pathway between the Navigation Module and the Input Sensory Vectors Association Modules, the intermediate results of the Navigation Module which did not produce any actionable output, can be fed back and stored in the Input Sensory Vectors Association

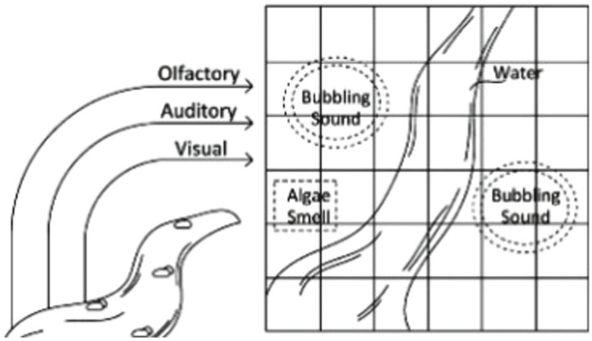

Fig. 4. Updating the Causal Memory Module best-matching multi-sensory navigation map A with features from the sensory scene different than what has been previously stored on the map. (A similar process occurs for each of best-matching local input sensory data navigation maps.)

Modules. In the next cognitive cycle these intermediate results will automatically be considered as the input sensory information and propagated to the Navigation Module and operated on again.

As shown in [14–16], by feeding back and re-operating on the intermediate results, the Causal Cognitive Architecture is able to formulate and explore possible cause and effect of actions, i.e., generate causal behavior. This is not surprising as the instinctive primitives (or the learned primitives, i.e., rules and procedures which the architecture learns) can essentially perform small logical operations and by being able to re-operate on the intermediate results the architecture can explore the effect of various actions. An example of such causal behavior is given by [14] where a robot controlled by a similar Causal Cognitive Architecture working as a patient aide has never seen this particular patient or hospital room before. The patient is using a walker to walk and asks the robot for a glass of water. As the patient's hand releases from the walker to accept the glass of water the patient starts falling down. The robot has not been preprogrammed how to stop a patient from falling but has a goal not to allow a patient to fall. The motion of the patient's arm and hand are bound onto a navigation map with what is called a motion prediction vector indicating the patient's body going towards the direction of the ground at a certain angle. The activation of a learned primitive in the Causal Cognitive Architecture robot that the patient should not fall, then triggers a physics instinctive primitive to push back against something falling or moving, in order to stop the movement. Thus, the Causal Cognitive Architecture robot pushes back against the falling patient opposite to the angle of the motion prediction vector, and stops the patient's fall [14].

More sophisticated learned primitives can allow more extensive exploration of effects. As well, a more sophisticated simulation whereby the architecture is attempting to repair a broken machine with many turning gears [14] can demonstrate cause and effect in the architecture better. However, in all cases, the same mechanism is being used—instinctive and learned primitives are being applied to a navigation map and then the intermediate results are re-operated on as necessary until a desired result is obtained. (Note that time-outs will occur after a certain number of cognitive cycles.)

This mechanism of feeding back results and then re-operating on them (instead of the new sensory inputs) may seem somewhat awkward to the computer scientist—why not instead create some temporary memory registers and a small algorithm directing results to and between these registers? The reason is that the Causal Cognitive Architecture is biologically inspired, and from an evolutionary perspective, it seems more reasonable that by enhancing feedback pathways from the Navigation Modules, intermediate results could be stored in the Input Sensory Vectors Association Modules and reprocessed in the next cognitive cycle, with few other evolutionary changes required. As well, this may be consistent with the evolutionary emergence of causal abilities and psychotic disorders [10, 11, 15].

3.3 Analogical Reasoning

Above we saw that if there is no actionable output [i.e., an output that can be propagated to the Output Vector Association Module (Fig. 1) from the Navigation Module (Fig. 1)], then we can feed back the Working Navigation Map (WNM) to the Input Sensory Vectors Association Modules. Then in the next cognitive cycle these intermediate results will automatically be considered as the input sensory information and propagated to the Navigation Module and operated on again.

Experimentation revealed that re-operating on different combinations of sensory inputs which are processed into a different Working Navigation Map (WNM) causing possibly a different selected Working Primitive (WPR), unfortunately often still does not give a causally related output or may give an output which may be actionable but not that useful. However, this experimentation [16] revealed that with a very small additional modification to the architecture [the appropriation of part of the Navigation Module to use as a temporary memory register, TempMap (Fig. 1)] and to the feedback algorithm, then more advantageous analogical processing of the intermediate results can occur.

Equations/pseudo-code (1)–(5) are adapted from [16]. If after the operation of the Working Primitive (WPR, which is a navigation map, i.e., an array) on the Working Navigation Map (WNM, also an array) there is no actionable output [i.e., no output which can be sent to the Output Vector Association Module (Fig. 1)], then as represented (i.e., *action* $\neq$ " move*") in (1), the results in the Navigation Module are treated as intermediate results and are fed back to be temporarily stored in the Input Vectors Association Modules, i.e., `Nav_Mod.feedback_to_assocn_mod(WNM)` (1). These intermediate results are also sent to the Causal Memory Module (Fig. 1) where they are matched to the best navigation map in the Causal Memory Module which becomes the new Working Navigation Map (WNM) (2). The most recently used link (or other scheme [16]) of this new WNM points to a navigation map which is then stored in the TempMap memory location within the Navigation Module (Fig. 1), and which we term TEMPMAP (3). Then in (4), TEMPMAP (also an array) is automatically propagated to the Navigation Module where the value of the current Working Navigation Map (WNM) is subtracted from TEMPMAP with the difference now forming the current value of the Working Navigation Map (WNM).

$$action \neq \text{“ move*”},$$

$$\Rightarrow Nav\_Mod.feedback\_to\_assocn\_mod(WNM) \qquad (1)$$

$$\Rightarrow WNM = Causal\_Mem\_Mod.match\_best\_map(WNM) \tag{2}$$

$$\Rightarrow TEMPMAP = Nav\_Mod.use\_linkaddress1\_map(WNM) \tag{3}$$

$$\Rightarrow WNM = Nav\_Mod.subtract(WNM, TEMPMAP) \tag{4}$$

$action_{t-1} \neq$ "move*",

$$\Rightarrow WNM = Nav\_Mod.retrieve\_and\_add\_vector\_assocn() \tag{5}$$

In the next cognitive cycle (thus, $t-1$ represents the preceding cycle) as (5) shows, the previous intermediate results stored in the Input Vectors Association Module are propagated forward to the Navigation Module, where the pseudocode in (5) specifies they are added to the current Working Navigation Map (WNM), forming a new Working Navigation Map (WNM).

Induction by analogy effectively occurs by this process of moving, comparing, subtracting, and adding navigation maps. Consider Eqs. (6)–(9). There are two variables **x** and **y**. Variable **x** has properties P_1, P_2, P_3, P_4, … P_n (6). Variable **y** also has properties $P_1, P_2, P_3, P_4, \ldots P_n$ (7). We now see that variable **y** has another property N (8). Therefore in (9) we can conclude by induction by analogy that variable **x** also has property N.

In (1) we can refer to WNM as variable **x**, or perhaps as navigation map x. We want to know what this navigation map x will do next, i.e., which navigation map will it call. Consider variable **y**, or perhaps named as navigation map **y**, as referring to WNM in (2). It is the best-matching navigation map to navigation map x and thus we assume it will share many properties. We explore what navigation map **y** does next (i.e., what navigation map does the *linkaddress* we chose link to). We see that navigation map **y** links to the navigation map which we then store in TEMPMAP (3) and that the difference between navigation map **y** and TEMPMAP is WNM (4). We will consider this difference, i.e., current WNM in (4) to be property N (8). Since navigation map **y** has property N, therefore by induction by analogy, we can say that navigation map x also has property N (9). Thus, we add property N, which is actually the difference, i.e., current WNM in (4), to navigation map x, which is actually the original WNM, producing the result of navigation map x with property N as being the Working Navigation Map WNM represented in (5).

$$P_1\boldsymbol{x} \text{ and } P_2\boldsymbol{x} \text{ and } \cdots P_n\boldsymbol{x} \tag{6}$$

$$P_1\boldsymbol{y} \text{ and } P_2\boldsymbol{y} \text{ and } \cdots P_n\boldsymbol{y} \tag{7}$$

$$N\boldsymbol{y} \tag{8}$$

$$\therefore N\boldsymbol{x} \quad \blacksquare \tag{9}$$

3.4 An Example of Analogical Reasoning in the CCA5

Induction by analogy in the CCA5, described in the previous section, proved advantageous for the utility of the architecture, not for human IQ-like tests (which the CCA5 is not developed enough to test on in any case) but rather, in the routine day to day decisions an agent operating in a relatively complex environment is required to make. Given that the CCA5 is mammalian brain inspired and that analogical reasoning emerged with relatively few changes, it is not surprising that there exists much psychological evidence that the core of human cognition relies on analogies [19].

Consider a simple example. There is a large hole in the ground filled with leaves. An embodiment controlled by a CCA5 architecture (which we will simply call "CCA5") has the current goal of going across a field when it comes to this large hole. Normally, if there is a large, empty hole, an instinctive primitive will be triggered (via the mechanisms discussed in the preceding sections) which operates on the Working Navigation Map resulting in a decision to avoid the hole. The Navigation Module will make a decision to turn right or turn left rather than go straight. However, if there is a solid path (e.g., a bridge, although the current CCA5 does not actually know what a "bridge" is) across the hole then the CCA5 will continue along the solid path to the other side of the hole and continue going across the field.

In a new example, the CCA5 sees a large hole filled with leaves (which it knows from an internal catalog as "solid08"). The CCA5 does not know anything about leaves other than they are solids. It may not know what to do, but the feedback mechanisms will not provide much help, and eventually an actionable decision results—the CCA5 will continue in paths across solids otherwise. In such a case, the CCA5 would go forward, the leaves would not support its weight, and it falls down into the hole and becomes damaged.

Consider another example, however, where the CCA5 uses analogical reasoning. The CCA5 still knows nothing about leaves (other than being able to recognize them as "solid08" from a preprogrammed catalog), but it had the previous experience of stepping in a small hole filled with crumpled newspaper (which it recognized as "solid22", knowing nothing about newspaper) and its foot falling into the hole. Thus, in this new case, what happens is when the CCA5 sees the hole with leaves ("solid08") the analogy mechanism described above causes the navigation map representing stepping on "solid22" (which shares properties as being in sheets with "solid08") to link to the navigation map of falling into the hole with newspapers, i.e., falling in. This is the result (i.e., navigation map) which is fed forward in the next cognitive cycle to the Navigation Module. This causes another instinctive primitive to be triggered about falling into a hole, which is applied against the Working Navigation Map in the Navigation Module. The result of the operation of this intrinsic primitive is not to go forward. Thus, the Navigation Module makes the decision to go left or right, and does not fall into the hole filled with leaves. Thus, even though our very simple and immature CCA5 (i.e., it has not been well developed with instinctive primitives, it has no learned primitives, and so on) had little particular knowledge ahead of time about leaves, it was able to automatically (via the core mechanism of induction by analogy) make the correct decision not to continue straight across the leaves in the hole.

4 Discussion

As noted above, the difference in the abilities of neural network AI systems and symbolic, logical AI systems is sometimes referred to as the "neurosymbolic gap" [4, 5]. As noted above, neurosymbolic AI systems which combine properties of ANN's and symbolic systems, are almost always overtly hybrid systems [6–8]. However, in this paper we have explored the ability of a brain-inspired cognitive architecture, the Causal Cognitive Architecture 5 (CCA5), to effectively represent a more integrated solution to the neurosymbolic gap.

Above we have reviewed the operation of the CCA5, particularly with regard to the topic of the neurosymbolic gap. The reader is directed to [10–16] for a more detailed description and analysis of the architecture of the CCA5. Schneider [16] gives a detailed and more formal description of the function and data flow of the components making up the architecture. The operation of the architecture can be summarized via Fig. 1 in terms of cognitive cycles of sensory features streaming in, being formatted, propagated to the Input Sensory Vectors Association Modules and spatial binding to local (single sensory) navigation maps and then to a multi-sensory best-matching navigation map from the Causal Memory Module (Figs. 3 and 4), as well as being propagated to the Sequential/Error Correcting Module and then temporal binding onto the multi-sensory best-matching navigation map. The processed sensory inputs as well as previous results from the Navigation Module, trigger the selection of a best-matching "Working Primitive", which is applied against the updated multi-sensory best-matching navigation map ("Working Navigation Map"), often producing an actionable output from the Navigation Module. The output from the Navigation Module is further processed at the Output Vector Association Module and sent to actuators of the embodiment. Then another cognitive cycle occurs.

If no actionable output occurs from the Navigation Module, then as discussed above, the current Working Navigation Map can be fed back to the Input Sensory Vectors Association Modules and operated on again in the next cognitive cycle. As discussed above, causal abilities as well as inductive analogical reasoning readily emerge from these feedback cycles.

The architecture can match sensory inputs in a connectionist fashion and via the predictive coding of matching against stored navigation maps and spatial and temporal binding, can perceive well even with noisy, imperfect sensory inputs. The architecture can perform varied logical reasoning through the instinctive and learned primitives, as well having causal and inductive analogical reasoning as its core mechanisms. Thus, in addition to its connectionist properties it also has symbolic ones.

It could be argued that the CCA5 non-hybrid solution, does in fact have hybrid components, hence the term "non-overtly hybrid" used above. The navigation maps containing instinctive and learned primitives indirectly represent collections of logical operations. The operations on the navigation maps to find best-matching navigation maps and other such operations, do occur in parallel, and links are strengthened and weakened between different navigation maps and different cells even in the same map. So, in this manner, there are symbolic and connectionist aspects to the architecture. However, these components are very different than what is seen in typical symbolic systems or in artificial neural networks. As well, these components cannot function on their own—they are critical parts working together to make up the architecture. The components presented in Fig. 1 are essential to the successful workings of the CCA5. The architecture of the CCA5 is essentially that of a non-hybrid system, which as demonstrated above has neurosymbolic properties. Creating systems that effectively close the neurosymbolic gap is an active area of research [4–7]. As noted above, these are largely hybrid approaches which combine a neural network system with a symbolic AI system. There are challenges in doing so—the representations in the neural networks are very different than the representations in the symbolic AI system combined with the latter, depending on how the combination is achieved. An advantage of a non-overtly-hybrid system such as the CCA5 is that there is no such awkward combination of representations required.

Future work on the CCA5 and its succeeding architectures is to automate the acquisition of sensory inputs. Currently, simulated sensory inputs have to be manually hand-fed to the model in order for it to build up its navigation maps. Doing so will allow more experimental examples to better understand the properties of the system and possible practical uses for it. As well, the collection of instinctive primitives is very small at present. Enhancing this collection of primitives will assist with the collection of sensory inputs as well as processing of navigation maps.

As demonstrated above, one of the approaches that should also be considered with regard to better neurosymbolic integration, is a brain-inspired cognitive architecture approach, such as the example of the CCA5. As well, in discussions of cognition, artificial cognition, or consciousness [20], given that the CCA5 is inspired by the human brain and thus may reflect its mechanisms, the neurosymbolic properties of cognition should be considered.

References

1. Phillips, P.J., Yates, A.N., et al.: Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. PNAS **115**(24), 6171–6176 (2018). https://doi.org/10.1073/pnas.172135511
2. Waismeyer, A., Meltzoff, A.N., Gopnik, A.: Causal learning from probabilistic events in 24-month-olds. Dev. Sci. **18**(1), 175–182 (2015). https://doi.org/10.1111/desc.12208
3. Leivada, E., Murphy, E., Marcus, G.: DALL-E 2 Fails to Reliably Capture Common Syntactic Processes. arXiv (2022). https://doi.org/10.48550/arXiv.2210.12889
4. Besold, T.R., d'Avila Garcez, A., Bader, S., et al.: Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. arXiv (2017). https://doi.org/10.48550/arXiv.1711.03902
5. Ebrahimi, M., Eberhart, A., Bianchi, F., Hitzler, P.: Towards bridging the neuro-symbolic gap. Appl. Intell. **51**, 6326–6348 (2021). https://doi.org/10.1007/s10489-020-02165-6
6. Garcez, A.A., Lamb, L.C.: Neurosymbolic AI: The 3rd Wave. arXiv (2020). https://doi.org/10.48550/arXiv:2012.05876
7. Kautz, H.A.: The third AI summer: AAAI Robert S. Engelmore memorial lecture. AI Mag. **43**(1), 93–104 (2022). https://doi.org/10.1002/aaai.12036
8. Cingillioglu, N., Russo, A.: DeepLogic: Towards End-to-End Differentiable Logic Reasoning. arXiv (2019). https://doi.org/10.48550/arXiv.1805.07433v3
9. Schneider, H.: Meaningful-based cognitive architecture. In: Samsonovich, A.V. (ed.) Procedia Computer Science, vol. 145, pp. 471–480 (2018). https://doi.org/10.1016/j.procs.2018.11.109

10. Schneider, H.: The meaningful-based cognitive architecture model of Schizophrenia. Cogn. Syst. Res. **59**, 73–90 (2020). https://doi.org/10.1016/j.cogsys.2019.09.01
11. Schneider, H.: Subsymbolic versus symbolic data flow in the meaningful-based cognitive architecture. In: Samsonovich, A. (ed.) BICA 2019. Advances in Intelligent Systems and Computing, vol. 948 (2020). https://doi.org/10.1007/978-3-030-25719-4_61
12. Schneider, H.: Causal cognitive architecture 1: integration of connectionist elements into a navigation-based framework. Cogn. Syst. Res. **66**, 67–81 (2021). https://doi.org/10.1016/j.cogsys.2020.10.021
13. Schneider, H.: Causal cognitive architecture 2: a solution to the binding problem. In: Klimov, V.V., Kelley, D.J. (eds.) Studies in Computational Intelligence, vol. 1032 (2022). https://doi.org/10.1007/978-3-030-96993-6_52
14. Schneider, H.: Causal cognitive architecture 3: a Solution to the binding problem. Cogn. Syst. Res. **72**, 88–115 (2022). https://doi.org/10.1016/j.cogsys.2021.10.004
15. Schneider, H.: Navigation map-based artificial intelligence. AI **3**(2), 434–464 (2022). https://doi.org/10.3390/ai3020026
16. Schneider, H.: An inductive analogical solution to the grounding problem. Cogn. Syst. Res. **77**, 74–216 (2023). https://doi.org/10.1016/j.cogsys.2022.10.005
17. Spelke, E.S.: Initial knowledge. Cognition **50**, 431–445 (1994). https://doi.org/10.1016/0010-0277(4)90039-6
18. Kinzler, K.D., Spelke, E.S.: Core systems in human cognition. In: von Hofsten, C., Rosander, K. (eds.) Progress in Brain Research, vol. 164 (2007) (Chapter 14)
19. Hofstadter, D.R.: Analogy as the core of cognition. In: Gentner, D., et al. (eds.) The Analogical Mind: Perspectives from Cognitive Science, pp. 499–538. MIT Press (2001)
20. Bołtuć, P.: Consciousness for AGI. In: Samsonovich, A.V., Klimov, V.V. (eds.) Procedia Computer Science, vol. 169, pp. 365–372 (2020)