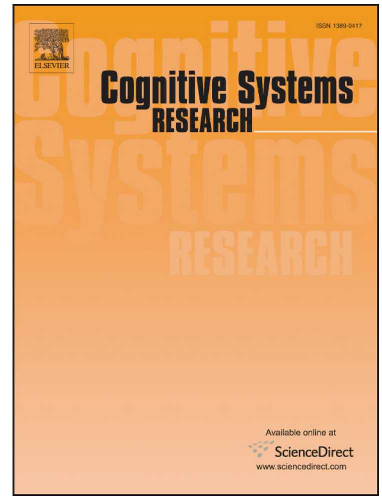
Journal Pre-proofs



Causal Cognitive Architecture 1: Integration of Connectionist Elements into a Navigation-Based Framework

Howard Schneider

PII: S1389-0417(20)30095-4
DOI: https://doi.org/10.1016/j.cogsys.2020.10.021
Reference: COGSYS 980

To appear in: *Cognitive Systems Research*

Received Date: 26 July 2020
Accepted Date: 28 October 2020

Please cite this article as: Schneider, H., Causal Cognitive Architecture 1: Integration of Connectionist Elements into a Navigation-Based Framework, *Cognitive Systems Research* (2020), doi: https://doi.org/10.1016/j.cogsys.2020.10.021

Causal Cognitive Architecture 1: Integration of Connectionist Elements into a Navigation-Based Framework

Howard Schneider[1]

[1]Sheppard Clinic North, Toronto, ON, Canada
hschneidermd@alum.mit.edu

Abstract

The brain-inspired Causal Cognitive Architecture 1 (CCA1) tightly integrates the sensory processing capabilities found in neural networks with many of the causal abilities found in human cognition. Causality emerges not from a central controlling stored program but directly from the architecture. Sensory input vectors are processed by robust association circuitry and then propagated to a navigational temporary map. Instinctive and learned objects and procedures are applied to the same temporary map, with a resultant navigation signal obtained. Navigation can similarly be for the physical world as well as for a landscape of higher cognitive concepts. There is good explainability for causal decisions. A simulation of the CCA1 controlling a search and rescue robot is presented with the goal of finding and rescuing a lost hiker within a grid world. A simulation of the CCA1 controlling a repair robot is presented that can predict the movement of a series of gears.

Keywords: Cognitive Architecture; Causality; Spatial Navigation; Artificial General Intelligence; Explainability

---

1. Introduction

1.1 Achieving Human-like Causal Behavior

Artificial neural networks (ANNs) can recognize patterns and perform reinforcement learning at a human-like proficiency (Goodfellow, Bengio & Courville, 2016; Mnih, Kavukcuoglu, Silver et al., 2015). However, compared to a four-year old child, in terms of logically and causally making sense of their environment or a problem at hand, especially if training examples are limited, they perform poorly (Ullman, 2019; Waismeyer, Meltzoff & Gopnik, 2015).

A number of cognitive architectures integrate subsymbolic and symbolic processing to varying degrees (Anderson, Bothell, Byrne et al., 2004; Bach, 2008; Rosenbloom, Demski, and Ustun, 2016). A review of the field by Langley (2017) notes that while early models were mainly symbolic, many of the modern architectures are more hybrid. Lake and colleagues (2017) propose that thinking machines should build causal models of the world, and discuss intuitive physics and psychology present in infants. Taatgen (2017) discusses the need to incorporate learning at multiple levels of abstraction, while Laird and Mohan (2018) discuss combining more innate architectural learning mechanisms.

Work by Graves and colleagues (2016) uses an ANN which can read and write to an external memory, i.e., a hybrid system. Huyck (2017) describes a neuromorphic-like cognitive architecture with a fast implicit subconscious system and a slow explicit conscious system. Epstein (Epstein, 2017; Epstein and Korpan, 2019) discusses cognitive and robotic modeling of spatial navigation. Hawkins and others (Hawkins, Lewis, Klukas et al., 2019; Schafer and Schiller, 2018) discuss how abstract concepts can be represented in a spatial framework.

Despite many of the above designs and implementations combining ANNs and symbolic elements, they do not approach the causal abilities seen in human children.

The Meaningful-Based Cognitive Architecture (MBCA) (Schneider, 2018; Schneider, 2020a; Schneider, 2020b) combines connectionist and symbolic elements in a biologically-plausible manner in which sensory input vectors are processed causally. A collection of intuitive and learned logic, physics, psychology and goal planning procedural vectors (essentially acting as small algorithms) are applied against inputs, and intermediate causal results can be fed back to the sensory input stages and processed over and over again. However, it must clumsily move data around to make a decision about choosing its next action. An overview of the MBCA is shown in Figure 1.

1.2 Overview of the Causal Cognitive Architecture 1 (CCA1)

This paper presents the Causal Cognitive Architecture 1 (CCA1). The CCA1 improves upon the MBCA with a more straightforward navigation-based logic system that works well with streams of vectors rather than needing more classical-like symbolic representations and processing that the MBCA required. The brain-inspired CCA1 tightly integrates the sensory processing capabilities found in neural networks with many of the causal abilities found in human cognition. Sensory input vectors are processed by robust association circuitry and then propagated to a navigational temporary map. Instinctive and learned objects and procedures are applied to the same temporary map, with a resultant navigation signal obtained. An overview of the architecture is shown in Figure 2.

As will be shown below, within the CCA1, causality emerges from the architecture, without needing any central controlling stored program, other than the built in operating cycles of the architecture. Of interest, navigation can similarly be for the physical world as well as for a landscape of higher cognitive concepts, and this is briefly explored below.

1.3 Hypotheses

In this paper the following hypotheses are considered:

1. The work below considers qualitatively the hypothesis that the CCA1 allows causality to emerge from a system without any central controlling stored program, other than the architecture's inherent repeating cycles of processing sensory inputs.

2. The work below considers qualitatively the hypothesis that with relatively small changes, an animal brain-inspired associative system such as the CCA1 operating in a pre-causal mode, can transition to a system capable of producing causal behavior.

1.4 Outline

This work will start with an overview of the Causal Cognitive Architecture 1. The operation and the use of this architecture in the pre-causal and the causal processing of inputs is then presented. It is discussed how very basic modifications in the architecture allow the CCA1 to achieve causal behavior.

A computer simulation of the CCA1 is then presented where the CCA1 controls a search and rescue robot that must rescue a lost hiker in the forest. First, examples with the CCA1 operating in more limited associative and pre-causal modes are given. Then an example is shown of the CCA1 utilizing its full architecture to allow causal processing of sensory inputs. Another simulation example is then given where the CCA1 controls a repair robot and is looking at a piece of machinery it has never seen before. In this example, it is shown that the CCA1 can predict the movement of a series of gears in the machinery.

In the final sections of this paper, the advantages of the CCA1 compared to its predecessor, the above mentioned MBCA, are discussed. As well the two hypotheses raised above are considered. It is reviewed how the navigation-based framework of the CCA1 allows causality to emerge from a system without any central controlling program other than the architecture's inherent and relatively straightforward repeating cycles of processing sensory inputs. It is then reviewed how relatively small changes can allow the transition of an associative system into a system with causal abilities.

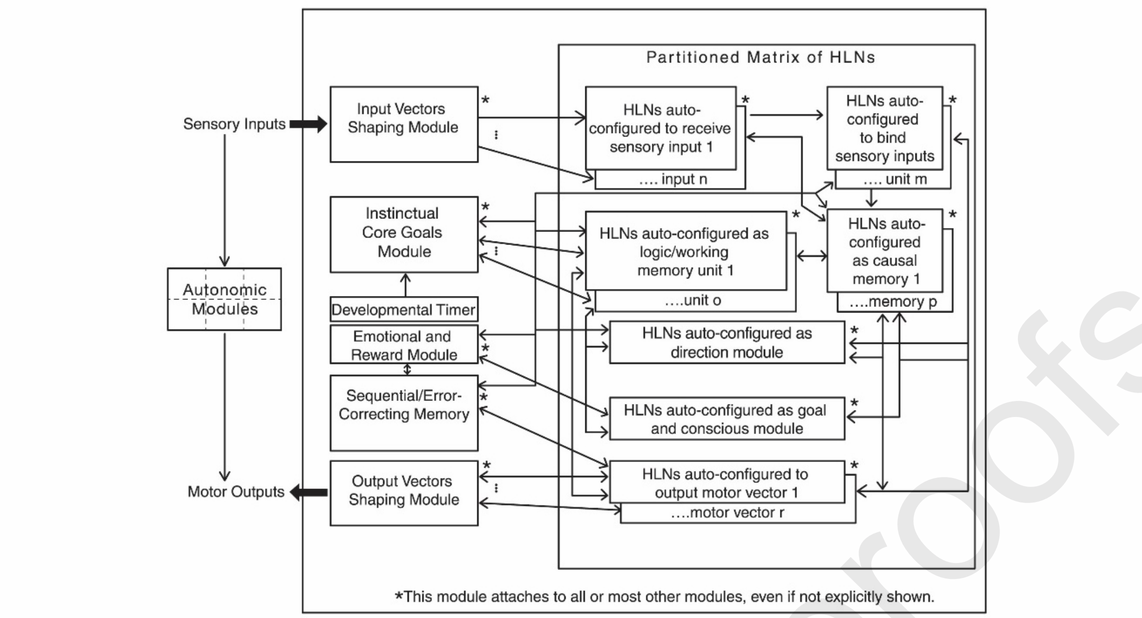


Figure 1. Meaningful-Based Cognitive Architecture (MBCA)

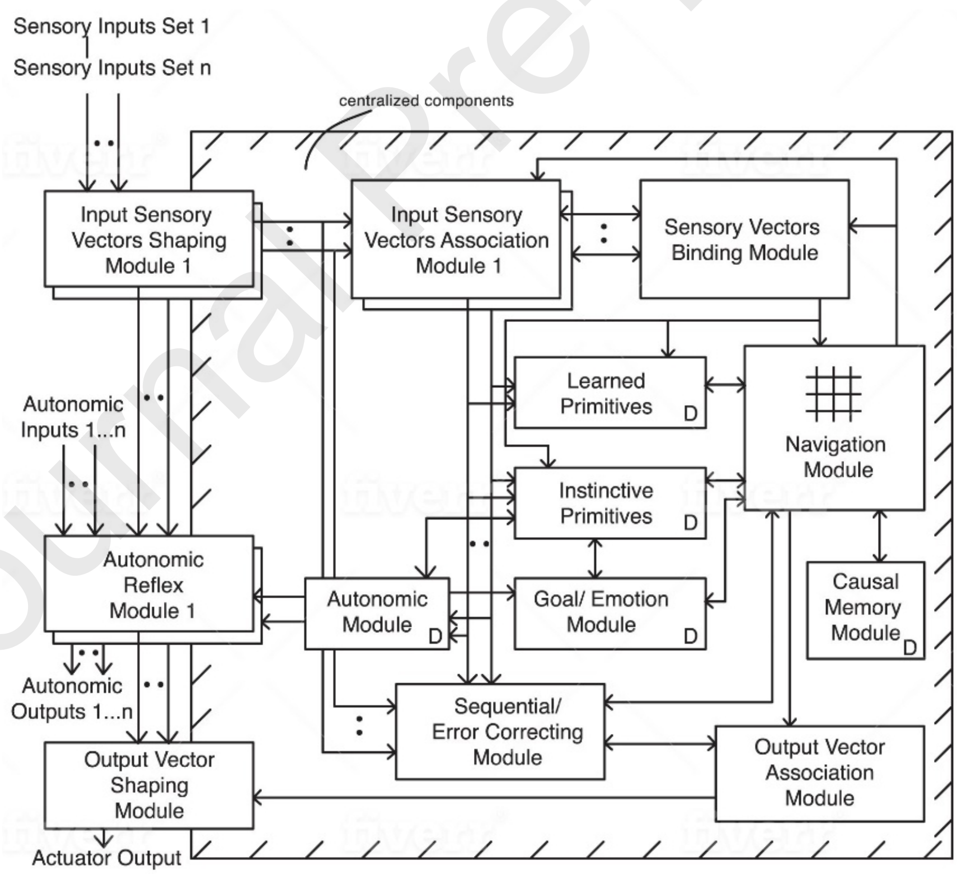


Figure 2. Causal Cognitive Architecture 1 (CCA1)
(Not all connections shown. D – Internal Developmental Timer)

2. Architecture and Operation of the CCA1

2.1 Sensory and Autonomic Inputs and Processing

An overview of the architecture of the CCA1 is shown in Figure 2. Sensory Inputs 1..n from different sensory systems 1..n, propagate to the Input Sensory Vectors Association Modules 1..n, with a module dedicated for each sensory system. Each such module contains a conventional neural network (Goodfellow, Bengio & Courville, 2016) or a hierarchy of Hopfield-like Network units (HLNs) (Schneider, 2018), or other roughly similar technology that can robustly associate an input sensory vector with other vectors (previously learned ones, instinctive pre-programmed ones, as well as other recent sensory input vectors) within the CCA1. Further binding of the processed input sensory vectors, via straightforward temporal mechanisms, or more complex global feedback mechanisms, occurs within the Sensory Vectors Binding Module.

Autonomic inputs, as in the physiological sense coming from the CCA1 and its embodiment, e.g., low energy reserves or an abnormal temperature rise in a component, and so on, are processed in the Autonomic Reflex Modules as well as the Autonomic Module (within the centralized components of the CCA1) which bidirectionally communicates with the former. The autonomic modules can offload a variety of routine, lower-level internal signal processing from the more complex CCA1 centralized components.

2.2 Pre-Causal Cognitive Processing and Output

Cognition in the CCA1 is at its core movement-based. At the simplest level, the CCA1's embodiment navigates through physical space, although at higher cognitive levels navigation occurs through a space of concepts and analogies. This is touched upon briefly further below.

The Navigation Module holds a temporary map of a small part of the inferred physical world, and on this small map are objects from the Sensory Vectors Binding Module, an object representing the CCA1 embodiment itself, and possible objects from the Instinctive Primitives Module and the Learned Primitives Module.

The Instinctive Primitives Module and the Learned Primitives Module are triggered by processed vectors from the Input Sensory Vectors Association Module, the Sensory Vectors Binding Module, as well as by the Goal/Emotion Module and the Autonomic Module. The Instinctive Primitives Module and the Learned Primitives Module can link possible objects onto the temporary map in the Navigation Module, as well as manipulate these objects in the Navigation Module.

The Navigation Module propagates an output that effectively causes the Output Vector Association Module to produce the desired movement of the embodiment of the CCA1.

The Sequential/Error Correcting Module recognizes and learns temporal sequences in input sensory vectors as occur in the physical world, and provides a signal to the Navigation Module.

The Output Vector Association Module does not just look up in a table what actuators to actuate for a desired movement by the Navigation Module but in a robust way learns a variety of different patterns of Actuator Outputs. As well, proprioception (the position of the CCA1's embodiment's actuators), along with other processed sensory inputs, are propagated to the Sequential/Error Correcting Module which produces an error signal between the desired and actual outputs, further improving output accuracy.

2.3 Internal Developmental Timer and Learning

Note that in Figure 2 some modules have in the corner the letter D, which stands for an Internal Developmental Timer. The properties of these modules are strongly influenced by the developmental stage of the CCA1.

A newly created CCA1 is not a tabula rasa. For example, when the CCA1 starts operating for the first time there are certain patterns that the Input Sensory Vectors Association Modules will recognize. As well the Instinctive Primitives Module, as noted above, contains a variety of objects it can map onto the temporary map of the Navigation Module and a variety of operations it can influence the Navigation Module to perform on these objects. It is advantageous for the CCA1 to learn and produce relevant actions at different stages of development. Thus, different Instinctive Primitives will be triggered at different stages, and different Learned Primitives will be learned or triggered at different stages.

Learning occurs throughout the CCA1. Feedback pathways are omnipresent in the architecture. At present, if ANNs are being used for association modules, then a separate learning period is required. If HLNs are being used instead, then continuous learning becomes more feasible (Schneider, 2018).

2.4 Causal Cognitive Processing and Output

In pre-causal operation of the CCA1, as described above, associations occur in the Input Sensory Vectors Association Modules and other modules described above. The Navigation Module effectively allows a sort of pre-causal processing with objects and rules being applied by the Instinctive and Learned Primitives Modules, and the Navigation Module making a navigation decision.

In causal operation, there is a more substantial feedback pathway from the Navigation Module to the Input Sensory Vectors Association Modules and other modules. As such, intermediate results of a problem or a causal situation can be fed back to the sensory input stages, and processed again in the next processing cycle. In the next section, an example is provided to illustrate this mechanism.

In a processing cycle, sensory inputs are processed through the CCA1, and the Navigation Module makes a decision about an output vector, and then the next processing cycle starts again. However, in causal operation if an intermediate result from the Navigation Module is fed back to the sensory input stages, then in the next processing cycle, the CCA1will take as the input the intermediate result which resides now in (or in the pathway of) the Input Sensory Vectors Association Modules. As such, intermediate results can be fed back to the sensory input stages and then processed by the CCA1 and Navigation Module, over and over again. As shown in the simulation example below, in this manner, causal processing of the inputs often results.

The Causal Memory Module shown in Figure 2 stores memories of operations of the Navigation Module. If similar events occur again, the Causal Memory Module can feed back into the Navigation Module which actions were taken in the past. The Causal Memory Module also gives the CCA1 a rough sense of time – what state and operation of the Navigation Module occurred in the past before or after another operation of the Navigation Module. Note that this also allows the CCA1 reasonable explainability of the causal decisions it makes, even with the handicap of associative processing of sensory data. The Causal Memory Module is functionally part of the Navigation Module, and access to the Causal Memory Module occurs via the Navigation Module.

3. CCA1 Simulation and Examples

3.1 CCA1 Simulation Overview

In the simulation of the CCA1, the embodiment of the CCA1 plus the CCA1 itself (together which are simply called here "the CCA1") must enter a grid world forest, and find and rescue a lost hiker. This is intentionally similar to the environment used by the MBCA simulation (Schneider, 2020b) although the environment used by the CCA1 includes a waterfall, allowing a more realistic demonstration of pre-causal versus causal behavior.

The simulation is written in the Python language, and via a menu driven user interface allows various modules of the architecture to be used, to be partially used or not to be used, enabling a spectrum of associative to pre-causal to fully causal models to be simulated. The selection menu is shown in Figure 3 and is discussed below. The small size of the grid world technically allows a simulation of an ANN or a simulation of an equivalent network of HLNs to be selected by menu for use in the various association modules.

Due to the proof of concept stage of the work, a measure of a CCA1 model's future expected value is simply taken as the reciprocal of the number of moves required to find a lost hiker over a number of simulation runs (which involve some randomness in perception of the environment).

Figure 4 shows the starting position of the CCA1 in the grid world. The lost hiker is in another square, and must be found (i.e., the CCA1 must move to this square) to be considered rescued by the CCA1. Please note that this is a bird's-eye view for the reader—the CCA1 does not see this information but must construct its own internal map of the world around it.

The CCA1 is able to safely navigate through the "forest" squares of the grid world. Edges of the grid world can be detected and simply do not allow movement. The CCA1 is able to cross shallow rivers. However, if it walks into what may seem like a river but is part of a waterfall, it will be considered to have fallen off the cliff portion of the waterfall and to be damaged, and thus the mission (i.e., the simulation) ends. If the CCA1 enters a "lake" square, then it will be considered to have become damaged, and thus the mission (i.e., the simulation) will end.

As noted above, in the simulation there is some randomness in the perception of the environment, as indeed occurs in the real world. Each processing cycle the CCA1 propagates the sensory inputs through its architecture, and over four cycles (i.e., one cycle facing north, another cycle facing east, another cycle facing south, and another cycle facing west) attempts to recognize the squares north, east, south, and west adjacent to its current position, storing this information in a temporary map it creates within the Navigation Module.

3.2 Utilization of CCA1 Components for Associative, Pre-Causal and Causal Processing

As noted above, at the start of a simulation, the user can specify combinations of associative, pre-causal and causal features of the architecture to be used. If only association components of the CCA1 are being used, then a sensory input vector can be recognized by the Autonomic Reflex Modules or by the Input Sensory Association Modules and directly trigger an output or indirectly trigger an output action via the Sequential/Error Correcting Module and Output Vector Association Module. On the other hand, if causal features are chosen then effectively all the components of the CCA1 shown in Figure 2 are used. If pre-causal features are chosen, then many of the components of the CCA1 shown in Figure 2 are used, but there is not the extensive feedback re-processing of partial results from the Navigation Module, a less extensive collection of Learned and Instinctive Primitives is used, and the Causal Memory Module is not as readily accessible by partial results from the Navigation Module.

These features of the architecture are specified, as shown in Figure 3, via menu selection of a lamprey-like brain, a fish-like brain, a reptile-like brain, a mammalian-like brain, a human-like brain or an augmented human-like brain. The actual mapping of the CCA1 components to animal neurophysiological structures is beyond the scope of this paper and is in any case only loosely inspired, at a level far above the spiking neuronal tier. As well, in comparing vertebrate neuroanatomy of different species, Butler and Hodos (1996) note that the concept of *scala naturae* arranging vertebrates in a sequence from "fish to frog to reptile to rat to cat to monkey to human" may seem intuitively correct, but so called lower vertebrates which are extant are in fact evolutionary distant from humans at this time, and should be considered as successful in terms of adapting to their environments. Thus, the CCA1 computer simulation menu which allows the selection of a particular class of animal-like brain, should only be taken as a user convenience, and not a rigorous mapping of that class's neurophysiological structures onto the CCA1 version chosen.

As well, not to detract from the core material of this paper, the selection of what is termed an augmented human-like brain was disabled. No claims of artificial general intelligence or superintelligence are made here, although in exploring pathways to these fields, it can be useful to consider and then simulate enhancements of certain features. For example, what is the effect of allowing two or two million Navigation Modules to run in parallel rather than the single Navigation Module in the CCA1 shown in Figure 2? Many other such questions can be explored in the augmented human-like brain selections.

```
Command Prompt - cca1_2020

Please choose type of "hippocampus"/"brain" which, of course,
 only loosely approximates the biological equivalent:
1. Lamprey hippocampal/brain analogue
2. Fish hippocampal/telencephalon analogue
3. Reptile hippocampal/pallium analogue
4. Mammalian hippocampus - note: meaningfulness, precausal
5. Human hippocampus - note: meaningfulness plus full causal features
6. Augmented Human level 1 - simultaneous multiple navigational threads
7. Augmented Human level 2 - algorithm center in each navigational module
Please make a selection:
```

Figure 3. User can select associative, pre-causal or causal features of the architecture at the start of the simulation

3.3 Pre-Causal Mode CCA1 Simulation Examples

In the ‘pre-causal’ simulation examples, the ‘reptile’ brain simulation is selected, as shown in Figure 3. If the ‘lamprey’ or ‘fish’ brain simulations were selected, somewhat similar actions would occur but behavior would be more associative, with less use of learned internal maps, and generally not as efficient in rescuing the lost hiker.

A new simulation starts (in ‘reptile hippocampal/ pallium analogue’ mode, i.e., pre-causal) with the CCA1 and the lost hiker in the positions as shown in Figure 4. The CCA1 recognizes to the south features which the Instinctive Primitives Module considers a lake object (name is arbitrary) with a signal to the Navigation Module to not navigate there. Thus there is no output from the Navigation Module in this processing cycle. In the next two processing cycles the CCA1 recognizes to the west and north features which the Instinctive Primitives Module considers edge objects with a signal to the Navigation Module to not navigate there. All these recognized objects are stored in the temporary map within the Navigation Module, i.e., the map is built up as the CCA1 navigates through the world. This internal map is shown in Figure 5.

In the following processing cycle the CCA1 recognizes to the east features which the Instinctive Primitives Module considers (i.e., links properties and functions onto the temporary map of the Navigation Module) a forest object (name is arbitrary). The other parts of the temporary map do not allow movement, and thus the Navigation Module (in conjunction with a procedural vector from the Goal/Emotion Module) makes a decision to move east. This movement of the CCA1 is shown in the bird’s-eye view in Figure 6.

Command Prompt - cca1_2020

```
hiker position set to:  4 2

Bird's-Eye View of Forest (CCA1 does not have this view)
```

EDGE	EDGE	EDGE	EDGE	EDGE	EDGE
EDGE	CCA1 *	forest	sh_rvr	forest	EDGE
EDGE	lake	forest	forest	forest	EDGE
EDGE	forest	wtrfall	forest	forest	EDGE
EDGE	forest	hiker	forest	forest	EDGE
EDGE	EDGE	EDGE	EDGE	EDGE	EDGE

Figure 4. Birds-Eye View of the Starting Positions of the CCA1 and the Lost Hiker
(Note: The CCA1 does not have this information but must build up its own internal temporary map in the Navigation Module)
(sh_rvr – ‘shallow river’; wtrfall – ‘waterfall’)

Command Prompt - cca1_2020

EDGE	EDGE				EDGE
EDGE	explored*	forest			
	lake				
EDGE					EDGE

Figure 5. Internal Map of the CCA1

```
Command Prompt - cca1_2020

CCA1 moved from (1, 1)  1,2

Bird's-Eye View of Forest (CCA1 does not have this view)
-----------------------------------------------------------------------------------------
EDGE       | EDGE       | EDGE       | EDGE       | EDGE       | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | CCA1  *    | sh_rvr     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | lake       | forest     | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | wtrfall    | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | hiker      | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | EDGE       | EDGE       | EDGE       | EDGE       | EDGE       |
-----------------------------------------------------------------------------------------
```

Figure 6. CCA1 Moves to the East into a Forest Square
(Note: The CCA1 does not have this information but must build up its own internal temporary map in the Navigation Module)
(sh_rvr – 'shallow river'; wtrfall – 'waterfall')

In this simulation example, the CCA1 continues to navigate through the landscape, and after a number of moves, the CCA1 arrives at the square "forest" east of the lost hiker. In the next few processing cycles its internal map will become further updated, and the Navigation Module will make the decision to move west to the square "lost hiker." The CCA1 is considered to have rescued the lost hiker and the simulation ends.

Learning occurs throughout the CCA1. Temporary maps the Navigation Module builds up, such as the one for this grid world, are stored, and can be associated and recalled in the future. Many thousands or millions of such maps can be stored. In a complex environment, even if a non-exact map is associated and recalled, it can be useful in guiding behavior to repeat successful moves and avoiding unsuccessful ones.

Consider a brand new CCA1, starting off again as shown in Figure 4. In this new simulation example, the CCA1 eventually navigates to the square "forest" just north of the square "waterfall" and this is shown in Figure 7. (Note that "waterfall" is labeled in the map in Figure 7 for the convenience of the reader. The CCA1 does not know this square is a waterfall nor have access to this map's information—it must try to build up its own internal map.) The CCA1 has already spent a number of moves in the north and east areas of the grid world without success in finding the lost hiker. As a result, a vector from the Goal/Emotion Module now favors navigation moves west and south.

```
Command Prompt - cca1_2020

Bird's-Eye View of Forest (CCA1 does not have this view)
-----------------------------------------------------------------------------------------
EDGE       | EDGE       | EDGE       | EDGE       | EDGE       | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | forest     | sh_rvr     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | lake       | CCA1  *    | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | wtrfall    | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | forest     | hiker      | forest     | forest     | EDGE       |
-----------------------------------------------------------------------------------------
EDGE       | EDGE       | EDGE       | EDGE       | EDGE       | EDGE       |
-----------------------------------------------------------------------------------------
```

Figure 7. New Simulation Example: CCA1 Moves to Square North of Waterfall
(Note: The CCA1 does not have this information but must build up its own internal temporary map in the Navigation Module)
(sh_rvr – 'shallow river'; wtrfall – 'waterfall')

In the next few processing cycles the CCA1 recognizes a lake to the west (which the Instinctive Primitive Module signals to avoid movement to) and to the south a shallow river with fast flowing noisy water (the cliff part of the waterfall is not visible). A shallow river does not trigger any prohibitions in the Instinctive Primitives Module since the CCA1 is able to cross shallow rivers. As such, the CCA1 moves south to the square labeled “waterfall” in Figure 7, where it is swept over the waterfall’s cliff, and is damaged. The CCA1’s Goal/Emotion Module makes a strong memory of the damage which occurred with the move towards such a fast flowing, noisy river.

The CCA1 is then repaired, and the next day it goes on another rescue mission. On this mission, if it recognizes a fast flowing river with much noise, then this will trigger in the Goal/Emotion Module and the Learned Primitives Module a signal to the Navigation Module not to navigate to this square.

3.4 Causal Mode CCA1 Simulation Examples

Consider a new simulation run with a brand new CCA1, again starting off as shown in Figure 4. The ‘human hippocampus’ simulation is selected. This CCA1 takes full advantage of the architecture, including extensive use of learned internal maps and more extensive feedback pathways allowing the re-processing of partial results of more complex problems.

In the new simulation, the CCA1 so happens to navigate again to the square “forest” just north of the square which in Figure 7 (intended only for the reader) is labelled a “waterfall.” The CCA1 has already meandered in the north and east areas of the grid world forest without success, and navigation south and west is as result now favored by the CCA1’s Goal/Emotion Module.

In the next few processing cycles the CCA1 recognizes a lake to the west (which the Instinctive Primitive Module signals to avoid movement to) and to the south a shallow river with fast flowing noisy water (the cliff part of the waterfall is not visible). This CCA1 has never seen a waterfall before. However, {“water”} + {“fast flow” + “noise”} triggers in the Instinctive Primitives Module {“water”} + {“push”}.

The Navigation Module cannot process a vector representing {“water” + “push”} and so feeds it back to the Input Sensory Vectors Association Module. In the next processing cycle, the Input Sensory Vectors Association Module ignores the external sensory inputs, but instead propagates the intermediate result {“water” + “push”}, as if it is the new sensory input.

{“water” + “push”} triggers in the Instinctive Primitives a vector causing the Navigation Module to bring up a new temporary map. On this new map is an object representing the CCA1 and objects representing water on much of the map. The Instinctive Primitives module procedural vector causes the object representing the CCA1 moved into the water in the temporary map, with water on top of it. This is shown in Figure 8.

Command Prompt - cca1_2020

Internal Map From Stack

air	air	air	air	air	air
water	water	water	water	water	water
water	water	water	water	water	water
water	water	water	CCA1 *	water	water
water	water	water	water	water	water
water	water	water	water	water	water
water	water	water	water	water	water

Figure 8. {“water” + “push”} triggers in the Instinctive Primitives module a vector causing the Navigation Module to bring up a new temporary map – the CCA1 is pushed into water

The Navigation Module cannot process this temporary map any further and so it feeds {"CCA1 under water"} back to the Input Sensory Vectors Association Module. In the next processing cycle, the Input Sensory Vectors Association Module ignores the external sensory inputs, but instead propagates the intermediate result {"CCA1 under water"} as if it is the new sensory input. This triggers in the Instinctive Primitives module a signal "do not go" which is sent to the Navigation Module. This then triggers the Navigation Module to retrieve the previous temporary map of the grid world forest, and "do not go" applies to the square south on that temporary map (and which in Figure 7 is labelled as a "waterfall").

In the next processing cycle the CCA1 recognizes the square to the east as forest and the Navigation Module instructs the CCA1 to navigate to the east, regardless of the biases of the Goal/Emotion Module for the west and south directions, as there are no possibilities to the west and south. Even though the CCA1 had never seen a waterfall before nor knew about its properties, it causally avoided this danger. This CCA1 continues to navigate, and a few moves later it finds and rescues the lost hiker.

While a number of processing cycles took place in deciding to avoid the waterfall, note that all these cycles were essentially triggered from one to another, with no central controlling stored program, other than the basic architecture of the CCA1.

3.5 Causal Mode CCA1 Non-Forest Simulation Example

Consider a new simulation run with a brand new CCA1. However, this time we have completely changed the environment from a forest, or actually from navigating any physical environment such as a forest or a city, to functioning as a robot that must repair broken machinery.

Figure 9 shows three of the many gears and pieces inside a broken machine the CCA1 is to repair. The CCA1 has not seen this particular type of machine before. In considering how the machine works, the CCA1 sees that if it moves, which is a turning movement, Gear A, then Gear B turns. There are thousands of parts in this machine. In order to understand how the machine works so that the CCA1 can repair it, the CCA1 does not have to examine the effect of moving or altering some combinations of all of the thousands of parts on the action of the other parts—doing so, even of only some of the parts, would involve millions and millions of possible combinations, and take the CCA1 years to build up the associations of the effect of moving or altering this part or parts on the other parts.

Looking at the three of the many gears inside the broken machine in Figure 9, it is easy for the CCA1 to predict that if it turns Gear C then Gear B will (or should) turn, without having to actually experience this, i.e., without the CCA1 having to physically do this experiment. Let's consider how it arrives at this prediction.

When the CCA1 turned Gear A and saw Gear B turning, as noted above, it constructed a temporary map in the Navigation Module as shown in Figure 10. Gear A moves and its neighbor square Gear B moves. This temporary map is stored in the Navigation Module which includes the Causal Memory Module.

A CCA1 acting as a repair robot would then have to ask itself what happens if it turns Gear C. However, asking questions to itself involves a whole set of additions to the Learned Primitives and a variety of temporary maps stored in the Navigation Module from previous experiences. To keep our example more straightforward, instead we ask the CCA1 a direct question—what happens to Gear B when Gear C is turned?

Gear C is beside Gear A and there is a Gear B still there, so the CCA1 retrieves the previous temporary map, and adds Gear C onto the map. This is shown in Figure 11.

The CCA1 then feeds back to the sensory inputs of 'push Gear C.' In the next processing cycle {"C"} + {"push"} is propagated through the CCA1 including to the Navigation Module's temporary maps and to the Instinctive Primitives. An Instinctive Primitive of {"C"} + {"moves"} is triggered and is sent to the Navigation Module's temporary map, and modifies Figure 11 with the addition of a push being applied to Gear C and Gear C moving (turning).

{"C"} + {"moves"} is then fed back to the sensory inputs. In the next processing cycle {"C"} + {"moves"} is propagated through the CCA1 and triggers in the Instinctive Primitives {"neighbor"} + {"moves"} which is sent to the Navigation Module's temporary map. The existing temporary map is modified to create a new temporary map as shown in Figure 12.

The external question (as noted above, a more sophisticated CCA1 would have the Learned Primitives to be able to ask itself such questions in repairing a broken piece of machinery) "What happens to Gear B when Gear C is turned?" is then given again as the Sensory Inputs to the CCA1, albeit in simple, machine understandable form. This is propagated to the Navigation Module where a temporary map containing Gear C lower on the stack-like buffer of recent temporary maps is retrieved, and combined with the current temporary map to produce the temporary map shown in Figure 13. There is information about Gear B on this map.

The internal state of the map, i.e. about Gear B, is used as output navigation descriptive information, rather than actual navigation signals as in the previous examples of the CCA1 navigating through the forest. This is a conceptual change from operating the CCA1 in non-causal modes of operation. There is a requirement of the Navigation Module to produce output which gives navigation outputs which does not actually navigate the CCA1 but instructs it to move actuators simply to communicate information about an internal state.

The Navigation Module sends the output "B moves" to the Output Vector Association Module (Figure 2) which then propagates to the Output Vector Shaping Module to produce an Actuator Output which communicates "B moves."

Note that causality readily emerges from the architecture of the CCA1 and its instinctual contents and learned contents (although in this example it was mainly instinctual primitives). There is no need for any special central controlling stored program, other than the inherent processing cycles of the architecture.

Many temporary maps may be used in allowing a causal solution to a problem. These temporary maps are stored in stack-like buffers and eventually in the Causal Memory Module, where they actually become quite permanent despite that the name "temporary" is used to describe them when they are initially created. It is possible for the CCA1 to play back the temporary maps, and thus provide explainability for the actions it took, if need be.

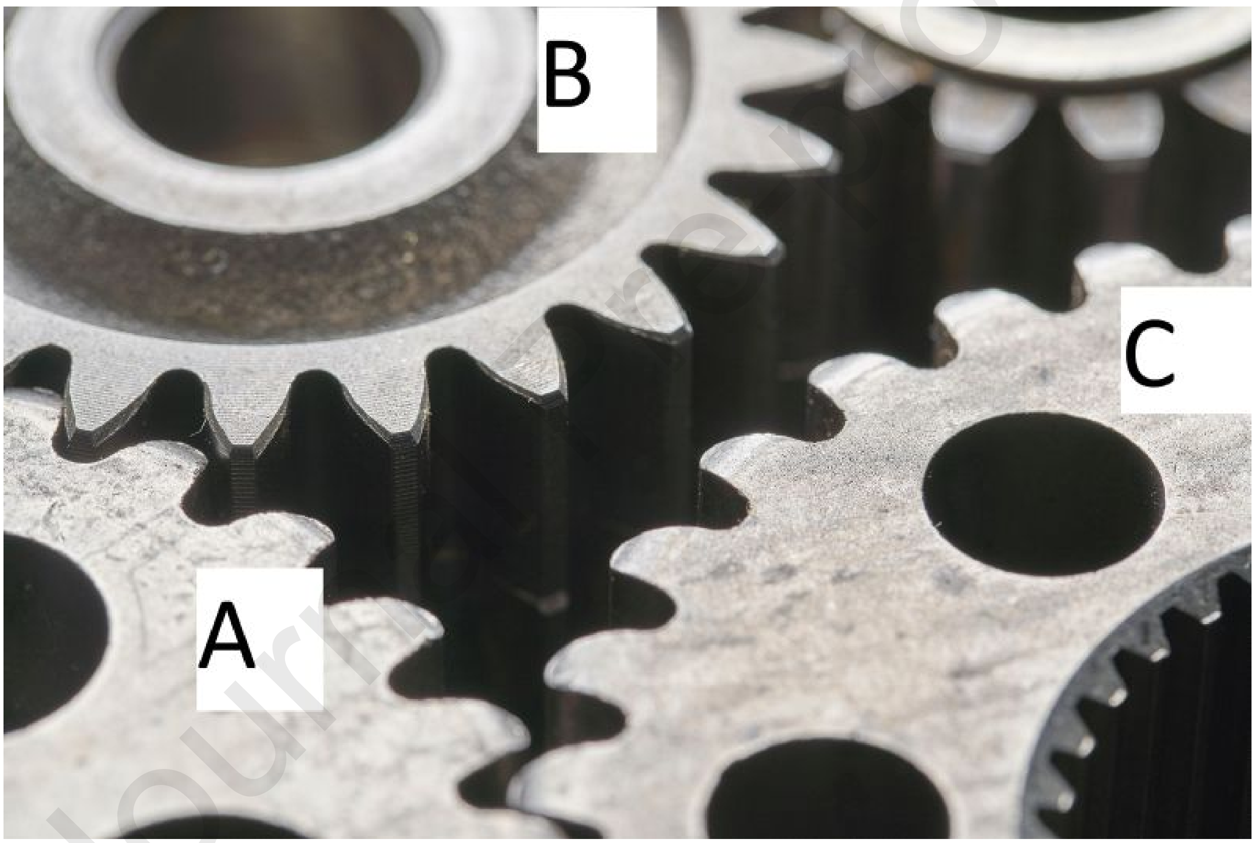


Figure 9. Three of the many gears and pieces inside a broken machine. The CCA1 has not seen this machine or gear set up before. If Gear C turns what happens to Gear B?

Command Prompt - cca1_2020

Internal Map From Stack

air*	air	air	air	air	air
air	*push	air	air	air	air
air	A;moves	B;moves	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air

Figure 10. Push on Gear A and Gear B moves (turns)

Command Prompt - cca1_2020

Internal Map From Stack

air*	air	air	air	air	air
air	*push	air	air	air	air
C	A;moves	B;moves	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air
air	air	air	air	air	air

Figure 11. Gear C is recognized and added to create a new temporary map

```
Command Prompt - cca1_2020

Internal Map From Stack
------------------------------------------------------------------------------------
air*        | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
*push       | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
C;moves     | A;moves     | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
```

Figure 12. If push (turn) Gear C then Gear C moves (turns) and therefore its neighbor will also move (turn)

```
Command Prompt - cca1_2020

Internal Map From Stack
------------------------------------------------------------------------------------
air*        | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
*push       | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
C;moves     | A;moves     | B;moves     | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
air         | air         | air         | air         | air         | air         |
 -----------------------------------------------------------------------------------
```

Figure 13. Update with previous temporary map. New temporary map shows that if Gear C is turned, then Gear B will move.

4. Discussion

4.1 Hopfield-like Networks (HLNs)

For pragmatic reasons, the CCA1 can be built with ANNs for associative functions and modified logic gates for enabling the processing cycles. The CCA1 can alternatively use hierarchies of HLNs as well as the logic-like structures of HLNs, similar to use in the MBCA (Schneider, 2018; Lázaro-Gredilla, Phoenix, D.S. and George, 2017). Using HLNs in the MBCA model gives three putative advantages which also apply to the CCA1 so constructed:

Better biological and evolutionary plausibility. The HLNs are inspired by mammalian cortical minicolumns (Buxhoeveden and Casanova, 2002; Schwalger, Deger and Gerstner, 2017). Evolutionary precursors to the mammalian cortex appear to go back to the earliest vertebrates (Suryanarayana, Robertson, Wallén, P., et al., 2017). Small changes in the arrangement of the HLNs and increased feedback pathways would seem to allow a transition in the mammalian brain from pre-causal behavior to full causal behavior (Schneider, 2020b).

A generative aspect to processing information. In the MBCA model (Schneider, 2018) the HLNs can be dynamically reconfigured a number of times each processing cycle to extract what is described as maximal "meaningfulness" from the input sensory vectors as well as in other aspects of the architecture, where the measure of meaningfulness is the reciprocal of the Shannon entropy, favoring activation of the maximal number of HLNs further downstream. Empirically this often gives better recognition of input vectors (Schneider 2018), although a proof of its usefulness is lacking.

An ability to quickly increase/decrease weights between different HLNs. The Goal/Emotion Module can reduce the class imbalance problem by large changes of weights between HLNs associated with infrequent but significant events, e.g., damage to the CCA1's embodiment.

4.2 Advantages of the CCA1 compared to the MBCA

More Efficient Movement of Data. The Meaningful-Based Cognitive Architecture (MBCA) (Schneider, 2018) contains Logic/Working Memory units which must request data from different parts of the architecture at different times. The CCA1 instead contains a Navigation Module which receives feeds of vectors and may trigger by its outputs other vectors in other parts of the architecture, but there are no discrete requests for data, and its operation is less complex than the operation of the MBCA Logic/Working Memory unit. As mentioned above, the CCA1 does not need any central controlling program, other than the inherent and relatively simple processing cycles of the architecture.

Enough Complexity for the CCA1 to Support the Psychosis Hypothesis. Schneider (2020b) hypothesizes that the complexity required to go from pre-causal to full causal operation of the MBCA results in imperfect functioning where the intermediate results sometimes are interpreted as actual external sensory inputs (i.e., hallucinations), which models the emergence of psychosis in *Homo sapiens*. This is supported by greater than a 10% prevalence of psychosis-like symptoms in humans (van Os, Hanssen, Bijl, et al., 2001) while psychosis is not naturally found in other animals, although versions of almost all other psychiatric disorders are (Xu, Schadt, Pollard et al., 2015). However, while not possessing as much added complexity as the MBCA model in going from pre-causal to causal operation, the CCA1 still has added enough features, and it can be argued the right amount of increased complexity from an evolutionary point of view (too many malfunctions and the species will not survive into the future despite the advantages of causal capabilities) particularly feeding back signals as intermediate results, to support the psychosis hypothesis.

More Biologically and Evolutionarily Plausible. Both the MBCA and CCA1 are biologically inspired cognitive architectures, although neither attempts to duplicate brain function down to the level of spiking neurons. A common feature of almost all animals with brains is the ability to perform some navigation, and in mammals the hippocampal-entorhinal system encodes physical as well as more abstract relations (Schafer and Schiller, 2018). Thus, it is biologically and evolutionarily more plausible for the CCA1 to have a dedicated Navigation Module, which is absent in the MBCA.

Simpler Transition to Higher Level Cognitive Processes. Consider the ability to make analogies. While in theory the Turing-complete nature of the Logic/Working Memory unit in the MBCA will allow just about any desired cognitive process, doing so requires complex routines and just the right data. On the other hand, the CCA1's architecture and temporary maps, readily form and use analogies.

Consider the case where the same forest search and rescue CCA1 above has returned from a mission in the forest and now is asked whether if it should spend more of its free time with person A or with person B. The persons are similar but person B is more smiley but also very noisy. At first glance, this question appears quite abstract, certainly far removed from a simple AI or AGI system, and better suited for the cognitive capabilities of a human poet or philosopher.

Object A (i.e., person A) and object B (i.e., person B) are put onto a temporary map in the CCA1's Navigation Module. The CCA1 decides whether it should navigate to object B—its Instinctive Primitives like smiling people. However, object B is noisy, and it results in pulling up the previous temporary map it had—the river seemed safe but made much noise also, and was considered a danger. Intermediate results are fed back to the sensory input stages and processed again, temporary maps are switched back again, and the noisy object B now is now associated with possible danger. Thus in this example, there is a navigation output to navigate to object A (i.e., person A). Without any specialized algorithm or any special central controlling stored program, other than the inherent and relatively simple operational cycles of the architecture, the CCA1 has made what would seem like a cognitively advanced decision.

4.3 Navigation-Based Framework of the CCA1

As noted above, a common feature in much of the animal world, including both the vertebrates and invertebrates (e.g., Kheradmand and Nieh, 2019; Boles and Lohmann, 2003) is the ability to move and the ability to navigate. The navigation-based framework of the CCA1 allows the obvious ability of navigation. Improvements and additions to the architecture (e.g., as per the simulation choices shown in Figure 3) can allow more advantageous navigation, without any special central controlling stored program, other than the inherent and relatively basic processing cycles of the architecture. This extends to full causal reasoning in navigation as shown in the example of the search and rescue in the forest.

However, the emergent properties of the navigation-based framework of the full CCA1, i.e., the full causal architecture, go beyond navigating in a physical world. As shown above, the same architecture readily allows predicting the effect of a pushing on one part in a machine it has seen for the first time, on some other distantly removed part in the machine. While it can be argued that the physical motion of the parts in the machine constitutes a navigation similar to navigation in a physical forest, the example above where the CCA1 has to decide to spend time with person A or person B, is very far removed from any physical navigation. Both the physical navigation and the navigation of higher cognitive concepts by the CCA1 represent a mechanism very different than the way a navigation problem would be solved by a conventional program written for a stored program computer, or solved by a hybrid neural network-stored program computer architecture.

As shown in the examples above, the Navigation Module can construct and swap in and out a number of temporary maps. While a given navigational decision, whether concrete (e.g., which direction in the forest to turn) or abstract (e.g., spend time with person A or person B), may only require a small number of these temporary maps, note that the CCA1's Navigation Module can have thousands or millions of temporary maps learned and stored, and can call up one map from another, as was shown in the examples above.

In the architecture described in this paper and shown in Figure 2, there is a single Navigation Module (which, as just mentioned can store millions of temporary maps) in each CCA1. However, as mentioned above in the discussion of Figure 3 with regard to the mode of operation of the CCA1 simulation, it is possible to choose 'Augmented Human-like' simulations where there can be multiple Navigation Modules operating at the same time. The exploration of such augmented architectures are discussed below in the section on future work.

4.4 Explainability

Gilpin and colleagues (2019) describe the property of interpretability in a model as allowing one to understand on some basis why the model performed this action and then the next action. They describe the property of explainability as a model being able to give the reasons for its behavior. Gilpin and colleagues note that an explainable model should be interpretable by default, although a model which is interpretable does not necessarily need to be explainable.

The Causal Memory Module of the CCA1, shown in Figure 2, stores so called temporary maps (which are created for temporary use in processing sensory inputs, but actually can be and are stored permanently in the Causal Memory Module). These temporary maps effectively give a history of the operations of the Navigation Module, i.e., an explanation of why the CCA1 arrived at a certain output given certain input values for a certain problem.

The temporary maps shown in Figures 10 – 13 provide an explanation of why the CCA1 predicted that Gear B will turn if Gear C is moved (turned). Actually, not all temporary maps used in this problem are shown in the figures, and there would be additional maps going from step to step in obtaining the solution the CCA1 did.

The etiology behind the storage of temporary maps in the Causal Memory was not to provide the property of explainability, but because similar events occur in the real world over again, albeit at varying time intervals. If similar events occur again at some point in the future, then similar temporary maps are triggered in the Causal Memory and fed back into the Navigation Module, thus providing a learned representation of which actions were taken in the past and which can advantageously be used for the problem at hand, possibly with some modifications as needed.

An emergent property of this architecture is having a rough sense of time. Meta-maps of temporary maps are made as they are stored. The CCA1 can readily reason that this event came before and after that event, thus providing a rough sense of time. However, it is noted that it is trivial to date and time stamp temporary maps using a technology-based architecture, something that is not as readily possible in brain-based architectures.

Another emergent property of this architecture, as noted above, is, of course, explainability. The storage of temporary maps gives a reasonable explanation of why a particular answer was given for a particular problem. However, it should be noted that this property may not be fully complete in the CCA1, just as it may not be so in a human. While operations which occur in other parts of the CCA1 do generate changes in parameters, i.e., synapses, throughout the CCA1, this learning does not generate stored maps as occurs in the Navigation Module. The CCA1 can indirectly, e.g., by self-experiment, generate maps stored in the Causal Memory in response to various sensory inputs, and thus have a rough intuition of why a non-Navigation Module event happened, but otherwise cannot really give reasonably complete explanations for its actions in processing inputs, in pattern matching and selecting a particular Instinctive Primitive, Learned Primitive or previously stored Temporary Map, or in processing outputs, all which can be affected by other modules in the architecture including the Goal/Emotion Module and Autonomic Module. Nonetheless, for most real world problems it anecdotally appears that the CCA1 should be able to provide, directly or indirectly, reasonable explainability as to the steps it took to arrive at a decision.

While the toy implementations of the CCA1 should be interpretable, especially coupled with the internal explanations the CCA1 can give (i.e., the stored temporary maps), a larger implementation of the CCA1 may be reasonably interpretable to a sufficiently capable machine but too complex to be readily understandable by a human observer.

4.5 Hypothesis of CCA1 Model of Allowing Causality to Emerge from a System without any Central Controlling Stored Program

At the start of this paper two hypotheses were raised. The first one was whether, from a qualitative point of view, the CCA1 model allows causal behavior to emerge from a system without any special central controlling stored program, other than the architecture's inherent and relatively straightforward repeating cycles of processing sensory inputs.

The proof of concept Python computer simulation of a CCA1 controlling a search and rescue robot in a forest that must locate and rescue a lost hiker, showed, albeit qualitatively and albeit in a toy grid world, that this architecture is indeed capable of producing advantageous reflex behavior, associative behavior, pre-causal behavior and if most elements of the architecture are utilized fully, causal behavior. It is important to note that there is no special centralized controlling stored program operating in these models, save for the straightforward repeating processing cycles. The CCA1, using only relatively basic processing cycles, moving streams of sensory input vectors forward and backwards in feedback loops through the architecture, is able to produce this spectrum of behavioral abilities, including bona fide causal behavior.

The CCA1 does not have any computer-like clock circuitry that centrally controls the timing of its functions. Vectors are propagated from circuit to circuit, and then cycles are repeated. There are minimal delays in propagation that results in some synchronization of activity in different parts of the architecture, but this is an emergent property.

It should be noted that the Causal Cognitive Architecture 1 does not represent the minimal components required for causal behavior. The CCA1 is brain-inspired and contains components that facilitate not only causal behavior but generally adaptive behavior under a variety of environments. However, a number of components can be disabled or removed, e.g., the Sequential/Error Correcting Module, the Autonomic Module, and others, and causal behavior can still occur.

4.6 Hypothesis of CCA1 Model of Showing that Relatively Small Changes can allow the Transition of an Associative System into a System with Causal Abilities

At the start of this paper two hypotheses were raised. The second one asked, in a qualitative fashion, whether with relatively small changes, an animal brain-inspired associative system such as the CCA1 operating in an associative or pre-causal mode, can transition to a system capable of producing causal behavior. In the proof of concept Python computer simulation of a CCA1 controlling a search and rescue robot in a forest that must locate and rescue a lost hiker, various components of the architecture could be turned on or off via menu selection of, figuratively speaking, a lamprey-like brain, a fish-like brain, a reptile-like brain, a mammalian-like brain, or a human-like brain.

As was shown above in the computer simulations, small transitions in features and pathways added to the CCA1 readily demonstrated a transition from a straightforward associative behavior, to a more powerful pre-causal behavior to a full causal behavior, and thus support the second hypothesis raised at the start of the paper that relatively small changes can allow the transition of an associative system into a system with causal abilities.

While, as noted above, caution must be taken extrapolating this result to the evolution of brains of so called lower and higher vertebrates and humans, in fact the CCA1 model provides a biologically, i.e., evolutionary, plausible pathway for the natural evolution of vertebral brains.

4.7 Conclusion and Future Work

This paper introduces the Causal Cognitive Architecture 1 (CCA1). The Python-language simulation of the CCA1 demonstrates a proof of concept of the architecture.

An overview of the CCA1 was given. It is brain-inspired and tightly integrates the sensory processing capabilities found in neural networks with many of the causal abilities found in human cognition. Sensory input vectors are processed by robust association circuitry and then propagated to a temporary map. Instinctive and learned objects and procedures are also applied to this same map, resulting in a navigation signal output. Intermediate results can be fed back to the input stages and processed again in the next cycle, the whole process resulting in causal processing of the original sensory inputs. Navigation as such can be used not only for the physical world, but also for a landscape of higher cognitive concepts. Causal memory allows good explainability for causal decisions.

Future work to further develop the CCA1 includes:

1. Characterization and more formalization of the range of models with the basic properties of the CCA1;
2. Enlarging and enhancing the CCA1 simulation, including larger content in the Instinctive Primitives Module and fuller simulation of the HLN networks used in the association modules;
3. Quantitatively comparing the performance of different CCA1 models with each other, as well with non-CCA1 models. Given the compact grid world it is possible to create a corpus of relevant environments, somewhat similar to Chollet's dataset (Chollet, 2019), and as such Legg and Hutter's (2007) universal intelligence (1) can actually be numerically approximated for different CCA1 and non-CCA1 models to allow their comparison:

$$\gamma(\pi) := \sum_{\mu \textit{ for all environments}} 2^{-K(\mu)} V_{\mu} \qquad (1)$$

where γ is the expected performance of agent π over *all* μ environments, K is the Kolmogorov complexity of μ, V is the future expected value;

4. Scaling up the architecture to perform a meaningful task;
5. Exploration of augmented architectures where, for example and as raised briefly above, there are multiple Navigational Modules operating simultaneously on the same input sensory vectors. A wide spectrum of possible architectural enhancements need to be considered to advantageously coordinate the operation and output of multiple simultaneously operating Navigational Modules.

References

Anderson, J.R., Bothell, D., Byrne, M.D., et al. (2004). An Integrated Theory of Mind. *Psychol. Rev.* 111(4), 1036-1060.

Bach, J. (2008). Seven Principles of Synthetic Intelligence. *Proceedings of 1st Conference on Artificial General Intelligence*, Memphis: 63-74.

Boles, L.C., and Lohmann, K.J. (2003). True navigation and magnetic maps in spiny lobsters. *Nature* 421(Jan. 2):60–63.

Butler, A.B. and Hodos, W. (1996). *Comparative Vertebrate Neuroanatomy*. Wiley-Liss, NY, NY.

Buxhoeveden, D.P., and Casanova, M.F. (2002). The minicolumn hypothesis in neuroscience. *Brain* May:125 (Pt 5):935-51.

Chollet, F. (2019). On the Measure of Intelligence. *arXiv:*1911.01547(cs.AI).

Epstein, S.L. (2017). Navigation, Cognitive Spatial Models, and the Mind. *AAAI 2017 Fall Symposium: Technical Report FS-17-05*.

Epstein, S.L., Korpan, R. (2019). Planning and Explanations with a Learned Spatial Model. *COSIT 2019*. Eds Timpf, S., et al., pp. 22:1–22:20, Leibniz International.

Gilpin, L.H., Bau, D., Yuan, B.Z. et al. (2018). Explaining Explanations: An Overview of Interpretability of Machine Learning. *Proceedings of the IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA 2018) ArXiv*: 1806.00069 [cs.AI].

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA.

Graves, A., Wayne, G., Reynolds, M., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature* 538: 471-476.

Hawkins, J., Lewis, M., Klukas, M., et al. (2019). A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. *Frontiers in Neural Circuits* 12:121.

Huyck, C.R. (2017). The Neural Cognitive Architecture. *AAAI 2017 Fall Symposium: Technical Report FS-17-05*.

Kheradmand, B. and Nieh, J.C. (2019). The Role of Landscapes and Landmarks in Bee Navigation: A Review. *Insects* 10(10):342.

Laird, J., and Mohan, S. (2018). Learning fast and slow: Levels of learning in general autonomous intelligent agents. *Proceedings of AAAI-2018*.

Lake, B.M., Ullman, T.D., Tenenbaum, J.B., and Gershman, S.J. (2017). Building machines that learn and think like people. *Behav Brain Sci.* 40, E253.

Langley, P. (2017). Progress and Challenges in Research on Cognitive Architectures. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.

Lázaro-Gredilla, M., Liu, Y., Phoenix, D.S. and George, D. (2017). Hierarchical compositional feature learning. *arXiv:*1611.02252v2.

Legg, S., Hutter, M. (2007). Universal Intelligence: *arXiv*:0712.3329 (cs.AI).

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature* Feb 26;518(7540):529-33.

Rosenbloom, P., Demski, A., and Ustun, V. (2016). The Sigma Cognitive Architecture and System. *Journal of Artificial General Intelligence*, *7*(1), 1-103.

Schafer, M., Schiller, D.(2018). Navigating Social Space. *Neuron*, 100(2):476-489.

Schneider, H. (2018). Meaningful-Based Cognitive Architecture. *Procedia Computer Science BICA 2018*, ed Samsonovich, A.V., 145:471-480.

Schneider, H. (2020). Subsymbolic Versus Symbolic Data Flow in the MBCA. In: Samsonovich A.(eds)BICA 2019. *Advances Intelligent Systems and Computing*, 948.

Schneider, H. (2020b). The Meaningful-Based Cognitive Architecture Model of Schizophrenia. *Cognitive Systems Research*

59:73-90.

Schwalger, T., Deger, M., and Gerstner, W. (2017). Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size. *PLoS Comput. Biol*. 13(4).

Suryanarayana, S.M., Robertson, B., Wallén, P., et al. (2017). The Lamprey Pallium Provides a Blueprint of the Mammalian Layered Cortex. *Curr Biol.* 27(21):3264-3277.

Taatgen, N.A (2017). Cognitive Architectures: Innate or Learned? *AAAI 2017 Fall Symposium: Technical Report FS-17-05* (2017).

Ullman, S. (2019). Using neuroscience to develop artificial intelligence. *Science* 363 (6428), 692-693.

van Os, J., Hanssen, M., Bijl, R.V. , et al. (2001). Prevalence of psychotic disorder and community level psychotic symptoms: an urban-rural comparison. *Arch. Gen. Psychiatry* Jul;58(7):663-8.

Waismeyer, A., Meltzoff, A.N., and Gopnik, A. (2015). Causal learning from probabilistic events in 24-month-olds: an action measure. Developmental *Science* 18(1):175-182.

Xu, K., Schadt, E.E., Pollard, K.S. et al (2015). Genomic and Network Patterns of Schizophrenia Genetic Variation in Human Evolutionary Accelerated Regions. *Molecular Biology and Evolution*. 32(5):1148-1160.