Metadata of the chapter that will be visualized in SpringerLink

Book Title	Artificial General Intelligence
Series Title	
Chapter Title	Analogical Problem Solving in the Causal Cognitive Architecture
Copyright Year	2022
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
	ORCID	http://orcid.org/0000-0001-8052-6448

Abstract

The Causal Cognitive Architecture 3 is a biologically inspired cognitive architecture based heavily on navigation maps—arrays holding spatial navigation information about the external environment but also coopted by the architecture for much of its data storage and representational requirements. Sensory information is stored in navigation maps and operated on in the architecture. Enhancement of feedback pathways in the architecture allows the intermediate results of operations on navigation maps to be re-processed in the next operating cycle and has been shown to allow the architecture to generate causal behavior. Here it is shown that this also can readily allow the emergence of analogical processing as a core mechanism in the architecture. If a navigation map cannot be processed to yield an actionable output, then it is compared to a similar navigation map and automatically an analogical result is produced which the architecture can possibly use as an output. Analogical processing as a core mechanism may be advantageous in creating more capable artificial general intelligence systems.

Keywords (separated by '-'): Analogies - Causality - Cognitive architecture - Artificial general intelligence

Analogical Problem Solving in the Causal Cognitive Architecture

Howard Schneider(✉)

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

Abstract. The Causal Cognitive Architecture 3 is a biologically inspired cognitive architecture based heavily on navigation maps—arrays holding spatial navigation information about the external environment but also coopted by the architecture for much of its data storage and representational requirements. Sensory information is stored in navigation maps and operated on in the architecture. Enhancement of feedback pathways in the architecture allows the intermediate results of operations on navigation maps to be re-processed in the next operating cycle and has been shown to allow the architecture to generate causal behavior. Here it is shown that this also can readily allow the emergence of analogical processing as a core mechanism in the architecture. If a navigation map cannot be processed to yield an actionable output, then it is compared to a similar navigation map and automatically an analogical result is produced which the architecture can possibly use as an output. Analogical processing as a core mechanism may be advantageous in creating more capable artificial general intelligence systems.

Keywords: Analogies · Causality · Cognitive architecture · Artificial general intelligence

1 Introduction

Analogies may lie at the heart of human cognition [1]. Analogical problem solving allows us to solve day to day problems, make sense of novel situations and to plan behavior, and thus it is of relevance to creating a working artificial general intelligence (AGI). We describe here how in the development of a brain-inspired cognitive architecture, analogical reasoning appears to readily emerge, not as some specialized skill (e.g., to be used when performing human intelligence tests) but rather as a ubiquitous, core mechanism of cognition of the architecture.

The Causal Cognitive Architecture 3 (CCA3) is a biologically inspired cognitive architecture loosely inspired by the mammalian brain, in particular the mammalian hippocampus, and based heavily on navigation maps [2, 3]. The navigation maps in the simulated architecture [3] are arbitrarily limited size $6 \times 6 \times 6$ arrays holding spatial navigation information about the external environment but also coopted by the architecture for much of its data storage and representational requirements, as well as for the various small algorithms, termed "primitives" it uses.

B. Goertzel et al. (Eds.): AGI 2022, LNAI 13539, pp. 1–13, 2022.
https://doi.org/10.1007/978-3-031-19907-3_10

The key components of the CCA3 are shown in Fig. 1. The architecture takes as an input the set of sensory features streaming in from different perceptual sensors. Objects detected in this stream of sensory features are segmented, and visual, auditory, and other sensory features of each segmented object are spatially mapped onto navigation maps dedicated to one sensory modality. These single-sensory navigation maps are then mapped onto a best matching multi-sensory navigation map taken from the Causal Memory Module and operated on in the Navigation Module. Instinctive and learned primitives, essentially small rules or productions, themselves using modified navigation maps, are then applied onto the navigation map in the Navigation Module, producing a signal to the Output Vector Association Module and then to the external embodiment.

There are extensive feedback pathways throughout the architecture—states of a downstream module can influence the recognition and processing of more upstream sensory inputs. In the Causal Cognitive Architecture 3, the feedback pathways between the Navigation Module/Object Segmentation Gateway Module and the Input Sensory Vectors Association Modules are enhanced allowing intermediate results from the Navigation Module to be stored in the Input Sensory Vectors Association Modules. If so, in the next cognitive cycle (i.e., cycles of passing input sensory vectors into and through the architecture), these intermediate results will automatically be considered as the input sensory information and propagated to the Navigation Module and operated on again. 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 [3].

Below we show that a consequence of this enhancement in feedback processing of the intermediate results of the architecture is not only the ability to generate causal (or pseudo-causal) behavior [3] but that the architecture now readily uses analogical reasoning as a central and core mechanism of action.

2 Functioning of the Causal Cognitive Architecture 3 (CCA3)

We will work through the operation of the Causal Cognitive Architecture 3 (CCA3) shown in an overview in Fig. 1. A series of equations presented below describes the operation of the key modules in Fig. 1. These equations effectively represent a pseudocode for the architecture. Named procedures in some equations represent blocks of pseudocode. For example, `Input_Sens_Vect_Shaping_Modules.normalize()` in (10) represents the code for transforming arrays of input sensory data into arrays with the same dimensions used by the other modules of the architecture. Additional details about specific equations/pseudocode can be found in reference Schneider [3].

2.1 Input Sensory Vectors Shaping Modules

Inputs for any sense modality are sensed (or simulated) as a 2D or 3D spatial array of inputs, which vary with time (2, 4, 6). We arbitrary assume visual, auditory, and olfactory inputs in our current model, but sensory modalities, of course, can be expanded. A vector *s(t)* holds the arrays representing the sensory inputs $\mathbf{S}_{\sigma,t}$ of different sensory systems (9).

It is transformed into a normalized $\boldsymbol{s'}(t)$ (10). Any input sensory system inputs $\mathbf{S'}_{\sigma,t}$ are now in an array of dimensions (m, n, o) (11).

$$S_1 \in R^{m\_1\mathrm{x}n\_1\mathrm{x}o\_1} \tag{1}$$

$$S_{1,t} := \text{visual\_inputs(t)} \tag{2}$$

$$S_2 \in R^{m\_2\mathrm{x}n\_2\mathrm{x}o\_2} \tag{3}$$

$$S_{2,t} := \text{auditory\_inputs(t)} \tag{4}$$

$$S_3 \in R^{m\_3\mathrm{x}n\_3\mathrm{x}o\_3} \tag{5}$$

$$S_{3,t} := \text{olfactory\_inputs(t)} \tag{6}$$

$$\sigma := \text{sensory system identification code} \in \mathrm{N} \tag{7}$$

$$\text{n\_}\sigma := \text{total number of sensory systems} \in \mathrm{N} \tag{8}$$

$$\boldsymbol{s}(t) = [\mathbf{S}_{1,\mathrm{t}}, \mathbf{S}_{2,\mathrm{t}}, \mathbf{S}_{3,\mathrm{t}}, \ldots, \mathbf{S}_{\mathrm{n}\_\sigma,\mathrm{t}}] \tag{9}$$

$$\boldsymbol{s'}(t) = \texttt{Input\_Sens\_Vect\_Shaping\_Modules.normalize}(\boldsymbol{s}(t)) = [\mathbf{S'}_{1,t}, \mathbf{S'}_{2,t}, \ldots, \mathbf{S'}_{n\_\sigma,t}] \tag{10}$$

$$\therefore \mathbf{S'}_{\boldsymbol{\sigma},\mathbf{t}} \in R^{mxnxo} \tag{11}$$

2.2 Input Sensory Vectors Associations Modules

There is a separate Input Sensory Vectors Association Module for each sensory modality. A "local" navigation map refers to a navigation map dedicated to one sensory modality. The first operation on an array $\mathbf{S'}_{\boldsymbol{\sigma},\mathbf{t}}$ is to match it against all the existing local navigation maps **LNM** in the Input Sensory Vectors Association Module $\boldsymbol{\sigma}$. For example, the visual processed inputs $\mathbf{S'}_{\mathbf{1},\mathbf{t}}$ are matched against $\boldsymbol{all\_maps}_{1,t}$ which represents all the local navigation maps in the visual Input Sensory Vectors Association Module (15, 16, 17). The next operation is to update the best-matched local navigation map $\mathbf{LNM}_{(\boldsymbol{\sigma},\,\boldsymbol{\Upsilon},\mathbf{t})}$ with the actual sensory input $\mathbf{S'}_{\boldsymbol{\sigma},\mathbf{t}}$ (20, 21). The best-matching and updated local navigation maps $\mathbf{LNM'}_{(\boldsymbol{\sigma},\,\boldsymbol{\Upsilon},\,\mathbf{t})}$ of all the different sensory systems of the CCA3 are then represented in vector $\boldsymbol{lnm}_t$ (22).

$$\mathbf{mapno} := \text{map identification code} \in \mathrm{N} \tag{12}$$

$$\boldsymbol{\theta} := \text{total number of local navigation maps in a sensory system } \sigma \in \mathrm{N} \tag{13}$$

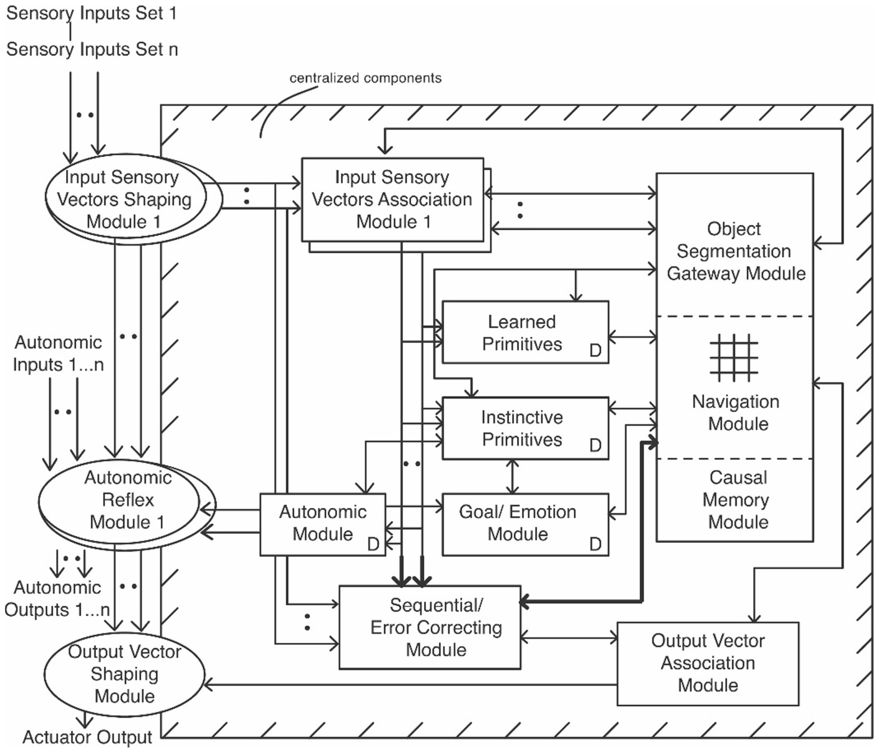

Fig. 1. Overview of the Causal Cognitive Architecture 3 (CCA3)

$$\mathbf{LNM}_{(\sigma,\mathbf{mapno})},\ \mathbf{WNM'}_{\mathrm{t}} \in \mathrm{R}^{\mathrm{mxnxo}} \tag{14}$$

$$\mathbf{all\_maps}_{\sigma,\mathbf{t}} = [\mathbf{LNM}_{(\sigma,\mathbf{1},\mathbf{t})}, \mathbf{LNM}_{(\sigma,\mathbf{2},t)}, \boldsymbol{LNM}_{(\sigma,\mathbf{3},t)}, \ldots, \mathbf{LNM}_{(\sigma,\theta,t)}] \tag{15}$$

$$\boldsymbol{\Upsilon} := \mathbf{mapno} \text{ of best matching map in a given set of navigation maps} \in \mathbf{mapno} \tag{16}$$

$$\mathbf{LNM}_{(\sigma,\Upsilon,t)} = \texttt{Input\_Assocn\_Module}_{\sigma}\texttt{.match\_best\_local\_navmap(}\ \mathbf{S'}_{\sigma,t},\ \mathbf{WNM'}_{t\text{-}1}) \tag{17}$$

$$\mathbf{h} = \text{number of differences allowed to be copied onto existing map} \in \mathrm{R} \tag{18}$$

$$\mathbf{new\_map} := \mathbf{mapno} \text{ of new local navmap added to sensory system } \boldsymbol{\sigma} \in \mathbf{mapno} \tag{19}$$

$$|\text{differences}\left(\mathbf{S'}_{\sigma,\mathbf{t}}, \mathbf{LNM}_{(\sigma,\,\Upsilon,\mathbf{t})}\right)| \leq \mathbf{h}, \Rightarrow \mathbf{LNM'}_{(\sigma,\,\Upsilon,\mathbf{t})} = \mathbf{LNM}_{(\sigma,\,\Upsilon,\mathbf{t})} \cup \mathbf{S'}_{\sigma,\mathbf{t}} \tag{20}$$

$$|\text{differences}\left(\mathbf{S'}_{\sigma,\mathbf{t}}, \mathbf{LNM}_{(\sigma,\,\Upsilon,\mathbf{t})}\right)| > \mathbf{h}, \Rightarrow \mathbf{LNM'}_{(\sigma,\,\Upsilon,\mathbf{t})} = \mathbf{LNM}_{(\sigma,\,\mathbf{new\_map},\mathbf{t})} \cup \mathbf{S'}_{\sigma,\mathbf{t}} \tag{21}$$

$$\boldsymbol{lnm}_t = [\mathbf{LNM'}_{(1,\,\Upsilon,t)}, \mathbf{LNM'}_{(2,\,\Upsilon,t)}, \mathbf{LNM'}_{(3,\,\Upsilon,t)}, \ldots, \mathbf{LNM'}_{(n\_\sigma,\,\Upsilon,t)}] \tag{22}$$

2.3 Navigation Maps

$\mathbf{NM_{mapno}}$ represents a multi-sensory navigation map stored in the Causal Memory Module. $\mathbf{IPM_{mapno}}$ and $\mathbf{LPM_{mapno}}$ represent navigation maps used to respectively store instinctive and learned primitives. Note below that we define $\boldsymbol{cubefeatures_{\chi}}$ to be ***feature*** values in a cube (i.e., x, y, z location) in a navigation map anywhere in the architecture at address $\boldsymbol{\chi}$ (35). At present, the CCA3 takes a simplistic approach to the grounding problem: every cube in a navigation map with contents must contain a grounded feature, or else at least contain a link somewhere, as shown by (41) and (42).

$$\mathbf{NM}_{\boldsymbol{mapno}} \in \mathrm{R}^{\mathrm{mxnxo}}, \mathbf{IPM}_{\boldsymbol{mapno}} \in \mathrm{R}^{\mathrm{mxnxo}}, \mathbf{LPM}_{\boldsymbol{mapno}} \in \mathrm{R}^{\mathrm{mxnxo}} \tag{23}$$

$$\boldsymbol{\theta}\_\mathbf{NM} := \text{total NM}'\text{s} \in \mathrm{N}, \boldsymbol{\theta}\_\mathbf{IPM} := \text{total IPM}'\text{s} \in \mathrm{N}, \boldsymbol{\theta}\_\mathbf{LPM} := \text{total LPM}'\text{s} \in \mathrm{N} \tag{24}$$

$$\boldsymbol{all\_LNMs}_t := [\boldsymbol{all\_maps}_{1,t}, \boldsymbol{all\_maps}_{2,t}, \boldsymbol{all\_maps}_{3,t}, \ldots, \boldsymbol{all\_maps}_{n\_\sigma,t}] \tag{25}$$

$$\boldsymbol{all\_NMs}_t := [\boldsymbol{NM}_{1,t}, \boldsymbol{NM}_{2,t}, \boldsymbol{NM}_{3,t}, \ldots, \boldsymbol{NM}_{\theta\_NM,t}] \tag{26}$$

$$\boldsymbol{all\_IPMs}_t := [\boldsymbol{IPM}_{1,t}, \boldsymbol{IPM}_{2,t}, \boldsymbol{IPM}_{3,t}, \ldots, \boldsymbol{IPM}_{\theta\_IPM,t}] \tag{27}$$

$$\boldsymbol{all\_LPMs}_t := [\boldsymbol{LPM}_{1,t}, \boldsymbol{LPM}_{2,t}, \boldsymbol{LPM}_{3,t}, \ldots, \mathbf{LPM}_{\theta\_\mathbf{LPM},t}] \tag{28}$$

$$\boldsymbol{all\_navmaps}_t := [\boldsymbol{all\_LNMs}_t, \boldsymbol{all\_NMs}_t, \boldsymbol{all\_IPMs}_t, \boldsymbol{all\_LPMs}_t] \tag{29}$$

$$\boldsymbol{modcode} := \text{module identification code} \in \mathrm{N} \tag{30}$$

$$\boldsymbol{mapcode} := [\boldsymbol{modcode}, \boldsymbol{mapno}] \tag{31}$$

$$\chi := [\boldsymbol{mapcode}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] \tag{32}$$

$$\mathbf{feature} \in \mathrm{R}, \mathbf{action} \in \mathrm{R} \tag{33}$$

$$\boldsymbol{\Phi}\_\mathbf{feature} := \text{last } \mathbf{feature} \text{ contained by a cube}, \boldsymbol{\Phi}\_\mathbf{action} := \text{last } \mathbf{action} \text{ contained by a cube}, \boldsymbol{\Phi}\_\chi := \text{last } \chi \text{ (i.e., address to link to) contained by a cube} \tag{34}$$

$$\boldsymbol{cubefeatures}_{\chi,\mathrm{t}} := [\boldsymbol{feature}_{1,\mathrm{t}}, \boldsymbol{feature}_{2,\mathrm{t}}, \boldsymbol{feature}_{3,\mathrm{t}}, \ldots, \boldsymbol{feature}_{\boldsymbol{\Phi}\_\mathrm{feature},\mathrm{t}}] \tag{35}$$

$$\boldsymbol{cubeactions}_{\chi,\mathrm{t}} := [\mathbf{action}_{1,\mathrm{t}}, \mathbf{action}_{2,\mathrm{t}}, \mathbf{action}_{3,\mathrm{t}}, \ldots, \mathbf{action}_{\boldsymbol{\Phi}\_\mathbf{action},\mathrm{t}}] \tag{36}$$

$$\boldsymbol{linkaddresses}_{\chi,t} := [\boldsymbol{\chi}_{1,\boldsymbol{t}}, \boldsymbol{\chi}_{2,\boldsymbol{t}}, \chi_{3,\boldsymbol{t}}, \ldots, \chi_{\boldsymbol{\Phi}\_\chi,\mathbf{t}}] \tag{37}$$

Author Proof

$$\boldsymbol{cubevalues}_{\chi,t} := [\boldsymbol{cubefeatures}_{\chi,t}, \boldsymbol{cubeactions}_{\chi,t}, \boldsymbol{linkaddresses}_{\chi,t}] \tag{38}$$

$$\boldsymbol{cubevalues}_{\chi,t} = \boldsymbol{all\_navmaps}_{\chi,t} \tag{39}$$

$$\boldsymbol{linkaddresses}_{\chi,t} = \mathrm{link}(\chi, \boldsymbol{t}) \tag{40}$$

$$\boldsymbol{grounded\_feature} := \forall_{\boldsymbol{feature}} : \boldsymbol{feature} \in \boldsymbol{all\_LNMs}_{\chi} \tag{41}$$

$$\begin{aligned}\forall_{\chi,t} : \boldsymbol{all\_navmaps}_{\chi,t} &= \boldsymbol{grounded\_feature}\ \mathrm{OR}\ \mathrm{link}(\boldsymbol{all\_navmaps}_{\boldsymbol{\chi},\mathbf{t}}) \neq [\,] \\ &\mathrm{OR}\ \boldsymbol{all\_navmaps}_{\chi,t} = [\,]\end{aligned} \tag{42}$$

2.4 Sequential/Error Correcting Module

Binding temporal features as spatial features in the navigation maps is described in more detail in [3] via directing the sensory inputs in a parallel path through the Sequential/Error Correcting Module, as depicted in Fig. 1. For example, Vector Navigation Map **VNM** binds the ***visual_motion****(t)* in similar navigation map coordinates as the other sensory inputs (49). The navigation map $\mathbf{VNM''_t}$ containing the visual motion and audio changes (50), and navigation map $\mathbf{AVNM_t}$ containing processed sound patterns (51), are then sent to the Object Segmentation Gateway Module/Navigation Module. Computation of **VSNM′** (55) requires **VSNM** which is discussed in the next module.

$$\boldsymbol{s'\_series}(\boldsymbol{t}) = \left[\boldsymbol{s'}(\boldsymbol{t}-\mathbf{3}), \boldsymbol{s'}(\boldsymbol{t}-\mathbf{2}), \boldsymbol{s'}(\boldsymbol{t}-\mathbf{1}), \boldsymbol{s'}(\boldsymbol{t})\right] \tag{43}$$

$$\boldsymbol{visual\_series(t)} = \texttt{SeqError\_Correct\_Mod.visual\_inputs}(\ \boldsymbol{s'\_series(t)}\ ) \tag{44}$$

$$\boldsymbol{auditory\_series(t)} = \texttt{SeqError\_Correct\_Mod.auditory\_inputs}(\ \boldsymbol{s'\_series(t)}\ ) \tag{45}$$

$$\boldsymbol{visual\_motion(t)} = \texttt{SeqError\_Correct\_Mod.visual\_match}(\ \boldsymbol{visual\_series(t)}\ ) \tag{46}$$

$$\boldsymbol{auditory\_motion(t)} = \texttt{SeqError\_Correct\_Mod.auditory\_match}(\ \boldsymbol{auditory\_series(t)}\ ) \tag{47}$$

$$\mathbf{VNM} \in \mathrm{R}^{mxnxo}, \mathbf{AVNM} \in \mathrm{R}^{mxnxo} \tag{48}$$

$$\mathbf{VNM'_t} = \mathbf{VNM_t} \cup \boldsymbol{visual\_motion}(\boldsymbol{t}) \tag{49}$$

$$\mathbf{VNM''_t} = \mathbf{VNM'_t} \cup \boldsymbol{auditory\_motion}(\boldsymbol{t}) \tag{50}$$

$$\mathbf{AVNM_t} = \texttt{SeqError\_Correct\_Mod.auditory\_match\_process}(\ \boldsymbol{auditory\_series(t)}\ ) \tag{51}$$

$$\mathbf{VSNM} \in \mathrm{R}^{mxnxo} \tag{52}$$

$$\boldsymbol{visual\_segment\_series}(\boldsymbol{t}) = [\mathbf{VSNM_{t-3}}, \mathbf{VSNM_{t-2}}, \mathbf{VSNM_{t-1}}, \text{and } \mathbf{VSNM_t}] \tag{53}$$

$$\boldsymbol{visseg\_motion(t)} = \texttt{SeqError\_Correct\_Mod.visual\_match}(\boldsymbol{visual\_segment\_series(t)}) \tag{54}$$

$$\mathbf{VSNM'_t} = \mathbf{VSNM_t} \cup \boldsymbol{visseg\_motion}(\boldsymbol{t}) \tag{55}$$

2.5 Object Segmentation Gateway Module

The Object Segmentation Gateway Module attempts to segment a sensory scene into objects of interest. In theory all sensory modalities can be segmented, but at present, only the visual local sensory map $\mathbf{LNM'_{(1,\,\Upsilon,t)}}$ is segmented (56–60). **WNM** is the "working navigation map" which is held in the Navigation Module and upon which operations of the instinctive and learned primitives can be applied. $\mathbf{VSNM_t}$(60) is transformed into $\mathbf{VSNM'_t}$ (52–55) and then contains visual sensory information segmented into different objects as well as binding information about the motion for each of these segments. **CONTEXT** is a contextual value which presently is assigned to the value of the previous **WNM**.

$$\mathbf{LNM'_{(1,\,\Upsilon,t)}} = \boldsymbol{lnm_t}[0] \tag{56}$$

$$\mathbf{CONTEXT} := \in \mathrm{R}^{mxnxo} \tag{57}$$

$$\mathbf{WNM} := \in \mathrm{R}^{mxnxo} \tag{58}$$

$$\mathbf{CONTEXT_t} = \mathbf{WNM_{t-1}} \tag{59}$$

$$\mathbf{VSNM_t} = \texttt{Object\_Seg\_Mod.visualsegment}(\mathbf{LNM'_{(1,\Upsilon,t)}}, \mathbf{VNM''_t}, \mathbf{CONTEXT_t}) \tag{60}$$

2.6 Causal Memory Module

The single sensory $\mathbf{LNM'}$s are then matched against previously stored multi-sensory navigation maps stored in the Causal Memory Module. The best matched map is used as the working navigation map **WNM** (61). $\mathbf{Actual_t}$ (63) is a representation of $\mathbf{VSNM'_t}$, containing objects and motion from the visual sensory inputs, $\mathbf{AVNM_t}$ containing audio information from the auditory sensory inputs, and $\mathbf{LNM'_{(3,\,\Upsilon,t)}}$ containing information from the olfactory sensory inputs. $\mathbf{WNM_t}$ is then updated with the current sensory input and transformed into $\mathbf{WNM'_t}$ (65, 66).

$$\mathbf{WNM_t} = \texttt{Causal\_Memory\_Module.match\_best\_multisensory\_navmap} \\ (\mathbf{VSNM'_t}, \mathbf{AVNM_t}, \mathbf{LNM'_{(3,\Upsilon,t)}}, \mathbf{LNM'_{(4,\Upsilon,t)}}, \ldots, \mathbf{LNM'_{(n\_\sigma,\Upsilon,t)}}) \tag{61}$$

$$\mathbf{h'} = \text{number of differences allowed to be copied onto existing navigation map} \in \mathrm{R} \tag{62}$$

$$\mathbf{actual_t} = [\mathbf{VSNM'_t}, \mathbf{AVNM_t}, \mathbf{LNM'_{(3,\Upsilon,t)}}, \mathbf{LNM'_{(4,\Upsilon,t)}}, \ldots, \mathbf{LNM'_{n\_\sigma,\Upsilon,t}}] \tag{63}$$

$$\mathbf{NewNM} \in \mathrm{R}^{mxnxo} \tag{64}$$

$$|\mathbf{differences(actual_t, WNM_t)}| \le \mathbf{h'}, \Rightarrow \mathbf{WNM'_t} = \mathbf{WNM_t} \cup \mathbf{actual_t} \tag{65}$$

$$|\mathbf{differences(actual_t, WNM_t)}| > \mathbf{h'}, \Rightarrow \mathbf{WNM'_t} = \mathbf{NewNM_t} \cup \mathbf{actual_t} \tag{66}$$

2.7 Navigation Module

Each cognitive cycle there is always a "working primitive" **WPR** (which is the best matching instinctive primitive (**WIP**) or learned primitive (**WLP**)) applied on the working navigation map **WNM'** in the Navigation Module, resulting in an **action** value (76–78). Normally the **action** value is then propagated to the output stages of the architecture and an action is taken in the real world (80–83). However, if the **action** value does not contain "move" (i.e., it is not actionable) then the output of the Navigation Module is instead fed back to the Input Sensory Vectors Association Modules (84). (Which from a biological evolutionary point of view, would have required only minor enhancements.) In the next cognitive cycle these intermediate results are returned to the Navigation Module (85) and operated on again. (The Input Sensory Vectors Association Modules automatically treat these intermediate results as if they are **LMN**'s of new sensory inputs, and automatically propagate them to the Navigation Module complex.)

$$\boldsymbol{emotion} \in \mathrm{R} \tag{67}$$

$$\mathbf{GOAL} \in \mathrm{R}^{mxnxo} \tag{68}$$

$$\boldsymbol{autonomic} \in \mathrm{R} \tag{69}$$

$$[\boldsymbol{emotion_t}, \mathbf{GOAL_t}] = \texttt{Goal/Emotion\_Mod.set\_emotion\_goal}(\boldsymbol{autonomic_t}, \mathbf{WNM'_t}) \tag{70}$$

$$\mathbf{WIP} \in \mathrm{R}^{mxnxo} \tag{71}$$

$$\mathbf{WIP_t} = \texttt{Instinctive\_Prims\_Mod.match\_primitive}(\boldsymbol{actual_t}, \boldsymbol{emotion_t}, \mathbf{GOAL_t}) \tag{72}$$

$$\mathbf{WLP} \in \mathrm{R}^{mxnxo} \tag{73}$$

$$\mathbf{WLP_t} = \texttt{Learned\_Prims\_Mod.match\_primitive}(\boldsymbol{actual_t}, \boldsymbol{emotion_t}, \mathbf{GOAL_t}) \tag{74}$$

$$\mathbf{WPR} \in \mathrm{R}^{mxnxo} \tag{75}$$

$$\mathbf{WLP_t} = [\,], \Rightarrow \mathbf{WPR_t} = \mathbf{WIP_t} \tag{76}$$

$$\mathbf{WLP_t} \neq [\,], \Rightarrow \mathbf{WPR_t} = \mathbf{WLP_t} \tag{77}$$

$$\mathbf{action_t} = \texttt{Navigation\_Module.apply\_primitive}(\mathbf{WPR_t}, \mathbf{WNM'_t}) \tag{78}$$

$$\boldsymbol{output\_vector} \in \mathrm{R}^{n'} \tag{79}$$

$$\mathbf{action}_t = \text{“move*”}, \Rightarrow \boldsymbol{output\_vector_t} = \texttt{OutVect\_Module.action\_to\_output}(\mathbf{action}_t, \mathbf{WNM'_t}) \tag{80}$$

$$\boldsymbol{motion\_correction} \in \mathrm{R}^2 \tag{81}$$

$$\mathbf{action_t} = \text{“move*”}, \Rightarrow \boldsymbol{motion\_correction_t} = \texttt{SeqError\_Correct\_Mod.motion\_correction}(\boldsymbol{action_t}, \mathbf{WNM'_t}, \boldsymbol{visual\_series(t)}) \tag{82}$$

$$\boldsymbol{output\_vector'_t} = \texttt{OutVector\_Module.apply\_motion\_correction}(\boldsymbol{output\_vector_t}, \boldsymbol{motion\_correction_t}) \tag{83}$$

$$(\mathbf{action_t} \neq \text{“move*” and } \mathbf{WPR_t} \neq [\text{“discard*”}]) \text{ or } \mathbf{WPR_t} = [\text{“feedback*”}], \texttt{Navigation\_Module.feedback\_intermediate}(\mathbf{WNM'_t}) \tag{84}$$

$$(\mathbf{action_{t-1}} \neq \text{“move*” and } \mathbf{WPR_{t-1}} \neq [\text{“discard*”}]) \text{ or } \mathbf{WPR_{t-1}} = [\text{“feedback*”}], \forall_\sigma : \mathbf{LNM_{(\sigma, \Upsilon, t)}} = \texttt{Input\_Assoc\_Module}_\sigma\texttt{.extract\_}\sigma(\mathbf{WNM'_{t-1}}) \tag{85}$$

3 Analogical Feedback

3.1 The Problem of Processing the Intermediate Results

As noted above, when an operation on a navigation map does not result in an actionable output, rather than wait for another sensory input to be processed in the next cognitive cycle, the Causal Cognitive Architecture will feed back these intermediate outputs of the Navigation Module and re-process them in the next cognitive cycle. While for certain combinations of sensory inputs and instinctive or learned primitives this may eventually give a useful output, even a causally related output [3], often it does not.

We describe here an algorithm which emerges readily from the architecture for processing of the intermediate results whereby analogical results are generated that may be more useful than simply feeding back and returning the intermediate results unchanged in the next cognitive cycle as occurs in the previous architecture [3]. As well, from the biologically inspired point of view, note that this algorithm requires only a small evolutionary step from the previous architecture.

In Eq. (86) we see the state where the working navigation map **WNM′** that was produced from the sensory inputs does not result in any actionable result in the Navigation Module, and so, there is the signal to feed back these results to the Input Sensory Vectors Association Modules, where they can be temporally stored. At the same time, in (87) the working navigation map that was produced **WNM′** is matched against the various navigation maps stored in the Causal Memory Module and the best matching navigation map becomes the working navigation map **WNM′**.

In (90) the navigation map which the new working navigation map **WNM′** linked to in the past, becomes the working navigation map **WNM′**. And in (91) the difference between these two navigation maps, i.e., what happened essentially in the past, is stored as the working navigation map **WNM′**. Then in the next cognitive cycle the original working navigation map that was fed back and stored in the Input Sensory Vectors Association Modules (in Eq. 86) is retrieved and added to (rather than overwriting) the Navigation Module (92). Thus, at this point, the working navigation map $\mathbf{WNM'_t}$ in the Navigation Module contains the action that occurred in the past of a similar working navigation map in a possible analogical situation. The demonstration example in the section below will illustrate this more clearly.

$$\begin{aligned}&(\mathbf{action_t} \neq \text{“ move*”} \text{ and } \mathbf{WPR_t} \neq [\text{“discard*”}] \text{ and } \mathbf{WPR_t} \neq [\text{“feedback*”}])\\&\text{or } \mathbf{WPR_t} = [\text{“analogical*”}],\\&\Rightarrow \texttt{Navigation\_Module.feedback\_intermediate}(\mathbf{WNM'_t})\end{aligned} \tag{86}$$

$$\Rightarrow \mathbf{WNM'_t} = \texttt{Causal\_Memory\_Module.match\_best\_multisensory\_navmap}(\mathbf{WNM'_t}) \tag{87}$$

$$\Rightarrow \mathbf{short\_term\_memory} \in \mathrm{R}^{mxnxo} \tag{88}$$

$$\Rightarrow \mathbf{short\_term\_memory} = \mathbf{WNM'_t} \tag{89}$$

$$\Rightarrow \mathbf{WNM'_t} = \texttt{Navigation\_Module.next\_map1}\ (\mathbf{WNM'_t}) \tag{90}$$

$$\Rightarrow \mathbf{WNM'_t} = \mathbf{WNM'_t} - \mathbf{short\_term\_memory} \tag{91}$$

$$\begin{aligned}&(\mathbf{action_{t\text{-}1}} \neq \text{“move*”} \text{ and } \mathbf{WPR_{t\text{-}1}} \neq [\text{“discard*”}]) \text{ or } \mathbf{WPR_{t\text{-}1}} = [\text{“analogical*”}],\\&\Rightarrow \mathbf{WNM'_t} = \texttt{Navigation\_Module.retrieve\_and\_add\_intermediates}\end{aligned} \tag{92}$$

The procedure `feedback_intermediate` in (86) takes the navigation map $\mathbf{WNM'_t}$ and breaks it up into local navigation maps **LNMs** representing its sensory components and stores the **LNMs** in their respective sensory modules in the Input Sensory Vectors Association Modules. In the procedure in (92) `retrieve_and_add_intermediates` these **LNMs** holding the intermediate results from (86) are transmitted to the Navigation Module where they are added to (rather than replacing) the working navigation map creating the new working navigation map $\mathbf{WNM'_t}$.

The procedure `match_best_multisensory_navmap` in (87) is the same as the one in (61) and is described in more detail in [2, 3].

The procedure `next_map1` in (90) looks at the link addresses (37) of the working navigation map **WNM'$_t$** and then retrieves the last (i.e., most recent) navigation map which this **WNM'$_t$** linked to, which now becomes the new **WNM'$_t$**. The procedure `next_map1` is the one simulated in the demonstration example below. (Other similar procedures are available. For example, `next_map2` will examine every one of the link addresses (37) of the working navigation map **WNM'$_t$** and then execute Eqs. (91) and (92) for all these link addresses and then attempt to choose the best analogical result.)

3.2 Analogical Feedback Demonstration Example

A simple demonstration of above equations via a Python computer simulation (with sensory inputs simulated as well) is shown below. This example shows the advantageous nature of the inductive analogic abilities created by the inclusion of Eqs. (86) to (92).

Figure 2 shows a working navigation map **WNM'$_t$** in the Navigation Module of the CCA3 (using 6x6x0 maps). Visual lines in the environment were sensed by an agent using the CCA3 and are propagated to the **WNM'$_t$** as shown in Fig. 2. What action should the Navigation Module take now? How to make sense of this environment?

No particular primitives are triggered, so an instinctive primitive is used as the working primitive **WPR** which contains "analogical". Thus, instead of producing an output action, the Navigation Module will feed this working navigation map back to the Input Sensory Vectors Association Modules where it can be temporarily stored, and the analogical algorithm occurs (86–92). Figure 3 shows the best match from the Causal Memory Module of **WNM'$_t$** which then becomes the new working navigation map (87). Then the navigation map which occurred after the navigation map in Fig. 3 (i.e., in the past when the map in Fig. 3 was stored in the Causal Memory Module) which is represented as a link in the map in Fig. 3 (not shown as it is in a non-spatial dimension of the navigation map), is activated and becomes the new **WNM'$_t$**, via (90) and as shown in Fig. 4. The difference between the navigation maps in Fig. 4 and Fig. 3 represents what happened in the past (91). Then in the next cognitive cycle, as described in (92), what happened in the past is added to the original Working navigation map (Fig. 2) resulting in a new **WNM'$_t$**, shown in Fig. 5.

Thus, if a straightforward resolution of a navigation map is not immediately possible (i.e., an instinctive or learned primitive is applied to a navigation map resulting from various sensory inputs, and there is not an actionable output), the architecture will automatically produce an analogical result. Note that other instinctive or learned primitives can then further process, as well as reject or output, the analogical result that is produced.

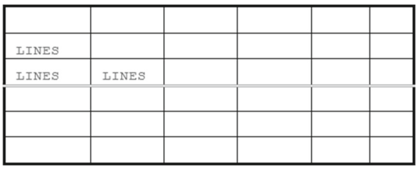

Fig. 2. (top left) Working Navigation Map **WNM′** – what action should occur?

LINES					
LINES	LINES				
LINES	LINES				
LINES					

Fig. 3. (top right) Best match from Causal Memory Module of previous **WNM′**.

LINES					
LINES	LINES	LINES	LINES	LINES	
LINES	LINES				
LINES					

Fig. 4. (bottom left) This is the Working Navigation Map **WNM′** that occurred after the best match **WNM′**.

LINES					
LINES	LINES	LINES	LINES	LINES	

Fig. 5. (bottom right) Retrieve the starting **WNM′** and apply the difference to it

4 Discussion

Above we reviewed how in developing a cognitive architecture loosely modeled on the mammalian brain, by enhancing pre-existing feedback pathways we can obtain causal abilities [2–4], and then with another small enhancement we show the emergence of inductive analogical abilities. There is a long history of analogical problem solving in the field of artificial intelligence [5]. The purpose of this work is not to show a better means of analogical abilities (although in conjunction with the overall causal architecture they may one day in fact prove to be quite advantageous) but to show how in a mammalian brain inspired cognitive architecture, causal and inductive analogical abilities effectively can emerge from the architecture.

Most earlier approaches to analogic problem solving were symbolic, e.g., Gentner's Structure-Mapping Engine (SME) [5], Hofstadter and Mitchell's Copycat program [6], and required human structuring and knowledge. In the last decade more connectionist approaches to analogy-making have been proposed. Wu and colleagues describe the Scattering Compositional Learner that puts neural networks in a sequence to elucidate the compositional structure of a problem and allows analogical reasoning [7].

While the deep learning approach to analogical reasoning overcomes the need for much of the human prior knowledge that symbolic systems required, a huge training set is still nonetheless required, something humans do not require, and issues such as performance via biases rather than understanding, remain [8]. Mitchell notes that while in the last decade there have been tremendous improvements in the ability of AI systems to recognize images and generate natural language, the ability of artificial intelligence systems to handle analogies, concepts, and abstractions, still remains an open problem.

The CCA3 demonstration example above, represents a very basic example of the use of analogical problem solving in the architecture. The example was a simplified analogy taken from Chollet's Abstraction and Reasoning Corpus [9]. To solve more complex problems in Chollet's corpus, additional primitives could be added to the CCA3 and used in conjunction with the architecture's intrinsic analogical problem solving. For example, if the next navigation map as shown in Fig. 4 was a rotation of 90 degrees plus the addition of the contiguous 'LINE' squares, then a set of intrinsic primitives for detecting and effecting basic geometric transformations is required.

As noted above, when enhanced feedback processing of intermediate results occurs in the Causal Cognitive Architecture, there is the possibility for analogies as a core mechanism in cognition. Given that many aspects of the architecture are brain inspired, that suggests that indeed analogies may be central to human cognition. Chen and colleagues have shown that one year old infants are capable of analogical problem solving [10]. Hofstadter presents evidence arguing for the role of constant analogy making in the human mind [1]. Similarly, analogical mechanisms may prove to be an important ability in allowing more capable future artificial general intelligence systems.

References

1. Hofstadter, D.R.: Analogy as the core of cognition. In: Gentner, D., Holyoak, K.J., Kokinov, B.N. (eds.) The Analogical Mind, pp. 499–538. MIT Press (2001)
2. 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
3. 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
4. Schneider, H.: The meaningful-based cognitive architecture model of schizophrenia. Cogn. Syst. Res. **59**, 73–90 (2020). https://doi.org/10.1016/j.cogsys.2019.09.019
5. Falkenhainer, B., Forbus, K.D., Gentner, D.: The structure-mapping engine. In: Proceedings of the National Conference on Artificial Intelligence, AAAI, pp 272–277 (1986)
6. Hofstadter, D.R., Mitchell, M.: The copycat project. In: Holyoak, K.J., Barnden, J.A. (eds.) Advances in Connectionist and Neural Computation Theory, vol. 2, pp. 31–112 (1994). Ablex
7. Wu, Y., Dong, H., Grosse, R., Ba, J.: The Scattering Compositional Learner. arXiv:2007.04212 (2020). https://doi.org/10.48550/arXiv.2007.04212
8. Mitchell, M.: Abstraction and analogy-making in artificial intelligence. Ann. N Y Acad. Sci. **1505**(1), 79–101 (2021). https://doi.org/10.1111/nyas.14619
9. Chollet, F.: On the Measure of Intelligence. arXiv:1911.01547 (2019)
10. Chen, Z., Sanchez, R., Campbell, T.: From beyond to within their grasp: analogical problem solving in 10- and 13-month-olds. Dev. Psychol. **33**, 790–801 (1997). https://doi.org/10.1037//0012-1649.33.5.790