Title
CogNGen: Building the Kernel for a Hyperdimensional Predictive Processing Cognitive Architecture

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 44(44)

Authors
Ororbia, Alexander
Kelly, M. Alex

Publication Date
2022

Peer reviewed

CogNGen: Constructing the Kernel of a Hyperdimensional Predictive Processing Cognitive Architecture

Alexander Ororbia (ago@cs.rit.edu)
Rochester Institute of Technology, New York, United States

M. Alex Kelly (alex.kelly@carleton.ca)
Carleton University, Ottawa, ON, Canada

Abstract

We present a new cognitive architecture that combines two neurobiologically-plausible computational elements: (1) a variant of predictive processing known as neural generative coding (NGC) and (2) hyperdimensional / vector-symbolic models of human memory. We draw inspiration from well-known cognitive architectures such as ACT-R, Soar, Leabra, and Spaun/Nengo. Our cognitive architecture, the COGnitive Neural GENerative system (CogNGen), is in broad agreement with these architectures, but provides a level of detail between ACT-R's high-level, symbolic description of human cognition and Spaun's low-level neurobiological description. CogNGen creates the groundwork for developing agents that learn continually from diverse tasks and model human performance at larger scales than what is possible with existent cognitive architectures. We aim to develop a cognitive architecture that has the power of modern machine learning techniques while retaining long-term memory, single-trial learning, transfer-learning, planning, and other capacities associated with high-level cognition. We test CogNGen on a set of maze-learning tasks, including mazes that test short-term memory and planning, and find that the synergy between its predictive processing and vector-symbolic components allow it to master the maze tasks. Future work includes testing CogNGen on more tasks and exploring methods for efficiently scaling hyperdimensional memory models to lifetime learning.

Keywords: Artificial Intelligence; Cognitive Architectures; Predictive Processing; Memory; Reinforcement Learning; Active Inference; Neural Generative Coding; Vector-Symbolic Architectures; MINERVA

Introduction

Machine learning methods based on artificial neural networks (ANNs) are implemented through algebraic manipulations of vectors, matrices, and tensors in high-dimensional spaces. While ANNs have an impressive ability to process data to find patterns, they do not typically model high-level cognition and are usually models of only a single task. Otherwise, when an ANN is trained to learn a series of tasks, catastrophic interference occurs, with each new task causing the ANN to forget all previous tasks (French, 1999; Mannering & Jones, 2021; McCloskey & Cohen, 1989). Conversely, symbolic cognitive architectures, such as the widely used ACT-R (Anderson, 2009; Ritter, Tehranchi, & Oury, 2019), can capture the complexities of high-level cognition but scale poorly to the naturalistic, non-symbolic data of sensory perception (e.g., images) or to big datasets necessary for modelling learning over a lifetime (e.g., corpora with billions of words).

Are symbolic and ANN approaches compatible? Symbolic and neural models can be understood as theories of cognition operating at different levels of description (Kersten, West, & Brook, 2016). Is it possible to provide a theory that bridges these two levels, a reduction of the symbolic to the neural, while retaining the strengths and capabilities of each?

We propose a cognitive architecture that is built on two neurobiologically and cognitively plausible models, namely a variant of predictive processing known as neural generative coding (NGC) (Ororbia, Mali, Giles, & Kifer, 2020) and vector-symbolic (a.k.a. hyperdimensional) models of memory (Hintzman, 1986; Jamieson & Mewhort, 2011; Kelly, Mewhort, & West, 2017). Desirably, the use of these particular building blocks yields naturally scalable, local update rules, based on variants of Hebbian learning (Hebb, 1949), to adjust the overall system's synaptic weight parameters while facilitating robustness in acquiring, storing, and composing distributed representations of tasks that the system encounters. Our intent is to advance towards a cognitive architecture capable of modeling human performance at all scales of learning, from the half-hour lab experiment to skills acquired gradually over a lifetime. By combining predictive processing with vector-symbolic models, we aim to create a model of cognition that has the power of modern machine learning techniques while retaining long-term memory, single-trial and transfer learning, planning, and high-level cognition.

While our ultimate aims are lofty, in this paper we demonstrate proof of concept. We show that our architecture, CogNGen (the COGnitive Neural GENerative system), is able to learn variants of a maze-learning task, including mazes that require planning (getting a key to open a locked door) and short-term memory (picking the correct path based on an earlier cue). In the context of reinforcement learning, our results further demonstrate that the synergy between predictive processing circuits and vector-symbolic models of short and long-term memory is competitive with several powerful intrinsic curiosity deep learning approaches, offering promising performance when the problem-specific reward is sparse.

The Common Model of Cognition

Since Newell (Newell, 1973) first argued that good empirical work and piecemeal theoretical work are insufficient to understand the mind, researchers in cognitive science have sought to develop functional, testable theories of cognition as a whole. Cognitive architectures serve as both unified theories of cognition and as computational frameworks for imple-

In J. Culbertson, A. Perfors, H. Rabagliati & V. Ramenzoni (Eds.), *Proceedings of the 44th Annual Conference of the Cognitive Science Society.*

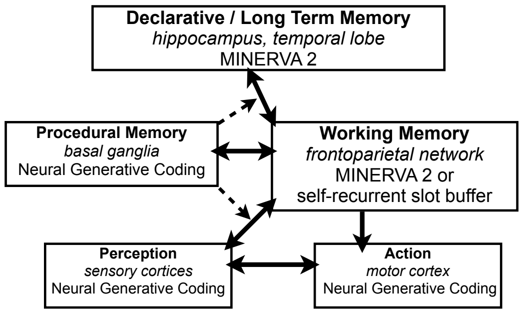


Figure 1: Common Model of Cognition (Laird et al., 2017), associated brain areas (Stocco et al., 2021; Steine-Hanson et al., 2018; Stocco et al., 2018) and our approach to modelling each module. Solid arrows are data passing. Dashed arrows are modulation of data passing.

menting models of specific experimental tasks. Forty years of research has developed hundreds of cognitive architectures, many with strong similarities to each other (Kotseruba & Tsotsos, 2018), suggesting an emerging consensus on the basic principles of cognition. The *Common Model of Cognition* (Laird et al., 2017) is a high-level theory of the modules of the mind and how they interact (see Fig. 1), proposed on the basis of commonalities between three cognitive architectures: ACT-R (Anderson & Lebiere, 1998), Soar (Laird, 2012), and SIGMA (Rosenbloom, Demski, & Ustun, 2016).

The Common Model of Cognition consists of perceptual and motor modules that interact with the agent's environment, short-term or working memory buffers which hold the active data in the agent's mind, a declarative or long-term memory module that holds the agent's world knowledge, and a procedural memory module that controls the flow of information and evaluates possible actions (Laird et al., 2017). An evaluation of fMRI data from 200 participants across diverse tasks found correlations in patterns of activity across brain areas consistent with the Common Model of Cognition's modules and their interactions (Stocco et al., 2021).

Neural Building Blocks

We start by first formally describing the fundamental neural circuits that are used to construct the modules of the CogNGen kernel instantiation of the Common Model of Cognition. In this work, $\leftarrow$ is assignment, $\odot$ is a Hadamard product, $\cdot$ is a matrix/vector multiplication (or dot product if operators are vectors of the same shape), and $\mathbf{v}^{\mathrm{T}}$ is the transpose.

Neural Generative Coding (NGC)

Neural generative coding (NGC) is an instantiation of the predictive processing brain theory (Rao & Ballard, 1999; Friston, 2005; Clark, 2015), yielding an efficient, robust form of predict-then-correct learning and inference. An NGC circuit in CogNGen receives two sensory vectors, an input $\mathbf{x}^i \in \mathcal{R}^{I\times 1}$ (I is the input dimensionality) and an output $\mathbf{x}^o \in \mathcal{R}^{O\times 1}$ (O is the output or target dimensionality). Compactly, an NGC circuit is composed of L layers of feedforward neuronal units, i.e., layer ℓ is represented by the state vector $\mathbf{z}^\ell \in \mathcal{R}^{H_\ell\times 1}$ containing H_ℓ total units. Given the input–output pair of sensory vectors $\mathbf{x}^i$ and $\mathbf{x}^o$, the circuit clamps the last layer $\mathbf{z}^L$ to the input, i.e, $\mathbf{z}^L = \mathbf{x}^i$, and clamps the first layer $\mathbf{z}^0$ to the output, i.e., $\mathbf{z}^0 = \mathbf{x}^o$. Once clamped, the NGC circuit will undergo a settling cycle where it processes the input and output vectors for K steps in time, i.e., it process sensory signals over a stimulus window of K discrete time steps. The activities of the internal neurons (all neurons in between the clamped layers, i.e., $\ell = L-1\dots 1$) are updated as follows:

$$\mathbf{z}^\ell \leftarrow \mathbf{z}^\ell + \beta\Big(-\gamma\mathbf{z}^\ell + (\mathbf{E}^\ell\cdot\mathbf{e}^{\ell-1})\otimes\partial\phi^\ell(\mathbf{z}^\ell) - \mathbf{e}^\ell\Big) \qquad (1)$$

where $\mathbf{E}^\ell$ is a matrix containing error synapses that pass along mismatch signals from layer $\ell-1$ to ℓ (this can be learnable or set to the scaled transpose of the predictive synaptic matrix, i.e., $\mathbf{E}^\ell = \lambda_e(\mathbf{W}^\ell)^T$). β is the neural state update coefficient and set according to $\beta = \frac{1}{\tau}$, where τ is the integration time constant in the order of milliseconds. This update equation indicates that a vector of neural activity changes, at each step within a settling cycle, according to (from left to right in Equation 1), a leak term (modulated by the factor γ), the bottom-up pressure from mismatch signals in lower level neural regions, and a top-down pressure from the neural region above. $\mathbf{e}^\ell \in \mathcal{R}^{H_\ell\times 1}$ are an additional population of special neurons that are tasked entirely with calculating mismatch signals at a layer ℓ, i.e., $\mathbf{e}^\ell = \mathbf{z}^\ell - \bar{\mathbf{z}}^\ell$, the difference between a layer's current activity (or clamped value) and an expectation/prediction produced from another layer. Specifically, the layer-wise prediction $\bar{\mathbf{z}}^\ell$ is computed as: $\bar{\mathbf{z}}^\ell = g^\ell(\mathbf{W}^{\ell+1}\cdot\phi^{\ell+1}(\mathbf{z}^{\ell+1}))$ ($\mathbf{W}^\ell$ denotes the matrix of predictive synapses). $\phi^{\ell+1}$ is the activation function (and $\partial\phi^{\ell+1}$ is its derivative) for state variables and g^ℓ (set to the identity) is applied to predictive outputs.

After processing the input–output pair for K steps (repeatedly applying Equation 1 K times), the synaptic weight matrices are adjusted with a Hebbian-like update rule as follows:

$$\Delta\mathbf{W} = \mathbf{e}^\ell\cdot(\phi^{\ell+1}(\mathbf{z}^{\ell+1}))^T\odot\mathbf{M}_W \qquad (2)$$

$$\Delta\mathbf{E} = \gamma_e(\Delta\mathbf{W})^T\odot\mathbf{M}_E \qquad (3)$$

where γ_e is a factor (less than one) to control the time-scale of the error synaptic evolution (ensuring they change more slowly than the predictive ones). $\mathbf{M}_W$ and $\mathbf{M}_E$ are modulation matrices that perform synaptic scaling to ensure additional stability in the learning process (Ororbia & Mali, 2021). Note that all NGC circuits in CogNGen are implemented according to the mechanistic process described above.

Another important functionality of an NGC circuit is its ability to ancestrally project a vector (akin to a feedforward pass, since no settling process is required) through the underlying directed generative model – we represent this process as $f_{proj}(\mathbf{x}^i;\Theta)$. Formally, ancestrally projecting a vector $\mathbf{x}^i$ through an NGC circuit proceeds as follows:

$$\mathbf{z}^\ell = \bar{\mathbf{z}}^\ell = g^\ell(\mathbf{W}^{\ell+1}\cdot\phi^{\ell+1}(\mathbf{z}^{\ell+1})),\ \forall\ell = (L-1),\dots,0 \qquad (4)$$

where $\mathbf{z}^L = \mathbf{x}^i$ (only the input/top-most layer is clamped to a specific vector, such as current input $\mathbf{x}^i$).

Memory

We simulate both declarative (long-term) and working (short-term) memory using the MINERVA 2 model of human memory (Hintzman, 1984). We choose MINERVA 2 since it captures a wide variety of human memory phenomena across different settings and, as such, seems a good candidate for integration into a cognitive architecture. MINERVA 2 has been applied to many experimental paradigms, including judgement of frequency and recognition (Hintzman, 1984), category learning (Hintzman, 1986), implicit learning (Jamieson & Mewhort, 2009, 2011), associative and reinforcement learning phenomena from both the animal and human learning literature (Collins, Milliken, & Jamieson, 2020; Jamieson, Crump, & Hannah, 2012), heuristics and biases in decision-making (Dougherty, Gettys, & Ogden, 1999), hypothesis-generation (Thomas, Dougherty, Sprenger, & Harbison, 2008), learning word meanings (Jamieson, Avery, Johns, & Jones, 2018; Kwantes, 2005), and production of grammatical sentences (Johns, Jamieson, Crump, Jones, & Mewhort, 2016a; Kelly, Ghafurian, West, & Reitter, 2020).

Adding to Memory MINERVA 2 stores memory traces: observations or sequence of observations of the state of the world. Each memory trace is represented as an array of real numbers (i.e., a vector). MINERVA 2 stores all memory traces separately as rows in a continuously growing table.

The continuous growth of the memory table results in a scaling problem for CogNGen, with significant slow downs even in the small maze learning tasks under consideration in this paper. Most MINERVA 2 models store only a small number of memory traces, though a few MINERVA 2 models used for language processing have stored up to between 20000 (Jamieson et al., 2018) and 500000s traces (Johns, Jamieson, Crump, Jones, & Mewhort, 2016b; Kelly, Ghafurian, et al., 2020). With a persistent long-term memory store across learning the maze task, in the worst case, as many as millions of traces might be stored into CogNGen's memory.

We use MINERVA 2's forgetting mechanism (Hintzman, 1986) to randomly delete values from memory each time memory is updated at a sufficiently high probability to impose a computationally tractable limit on MINERVA 2's memory size. Other possible solutions to the unbounded growth of memory are to use a compressed, scale-invariant approximation to MINERVA 2 (Kelly et al., 2017) or to adopt a different memory system that grows only with the number of unique stimuli, rather than with each new sequence of observations (Kelly, Arora, West, & Reitter, 2020). However, exploring these alternatives for improving the CogNGen memory's ability to scale to larger problems is a matter for future work.

Retrieval from Memory In MINERVA 2, memory retrieval is not a look-up process, it is a reconstruction process. When a retrieval cue or *probe* is presented, each vector in the memory table is activated in proportion to its similarity to the cue (Hintzman, 1986). Similarity is computed as a normalized dot-product of the cue's vector with the stored vector. Each stored vector is activated by its similarity to the probe raised to a power $p \geq 3$. By raising the similarity to some power, the contribution of the most similar vectors is emphasized. An *echo* is retrieved from memory as a weighted sum:

$$\mathbf{m_e} = \sum_{i=1}^{m} \left(\frac{\mathbf{m_p} \cdot \mathbf{m}_i}{\sqrt{\mathbf{m_p} \cdot \mathbf{m_p}}\sqrt{\mathbf{m}_i \cdot \mathbf{m}_i}}\right)^p \mathbf{m}_i \tag{5}$$

where $\mathbf{m_e}$ is the *echo* or output from memory, $\mathbf{m}_i$ is the i-th trace in memory, $\mathbf{m_p}$ is the probe or input to memory, and m is the number of traces in the memory table.

MINERVA 2 is equivalent to a type of Hebbian associative memory with a fixed number of neurons and a limited storage capacity (Kelly et al., 2017). The Hebbian network is so large that it is more efficient to simulate the network's behaviour as a growing table of memory traces. However, MINERVA 2 can be noisily approximated using smaller, more tractable Hebbian associative memory models (Kelly et al., 2017), which we will explore in future versions of CogNGen.

Discrepancy Encoding In CogNGen, we combine insights from several versions of MINERVA 2. We make an architecture-wide commitment to predict-then-correct learning. Predict-then-correct can be implemented in MINERVA 2 as *discrepancy encoding* (Collins et al., 2020; Jamieson et al., 2012). Given a sequence of observations, MINERVA 2 can predict the next observation based on past experience. After the prediction, a new observation is made. We update memory with the *difference* between observation $\mathbf{m_x}$ and prediction $\mathbf{m_e}$, $\mathbf{m_x} - \mathbf{m_e}$, such that if the prediction is correct, no change is made to memory, whereas if the prediction is wrong, that prediction is inhibited in similar future contexts.

Short-Term vs. Long-Term Memory For CogNGen, we adopt (Collins et al., 2020)'s approach and model both working and declarative memory using MINERVA 2. Our working MINERVA 2 is cleared after a task is completed (i.e., a maze is solved), whereas the contents of the declarative MINERVA 2 (which serves as the episodic memory in this paper's instantiation of CogNGen) persist across tasks.

The CogNGen Kernel

Perception

In this study, since the environment that we investigate, Gym-Minigrid, provides a fixed, problem-specific encoder f_e and decoder f_g, we use a fixed encoding and decoding scheme to simplify our simulations. Future work will investigate learning the perception modules as NGC circuits instead.

Neural Generative Procedural Memory

Neuro-behavioral studies find that reward signals are used (by the brain) to evaluate whether or not an action (motor activity) is desirable/undesirable (Rangel, Camerer, & Montague, 2008). Action selection is driven by changes in the neural activity of the basal ganglia which estimate the value of the expected reward (Hikosaka, Nakamura, & Nakahara, 2006). Motivated by the finding of expected value estimation in the brain, the CogNGen's procedural module implements a neu-

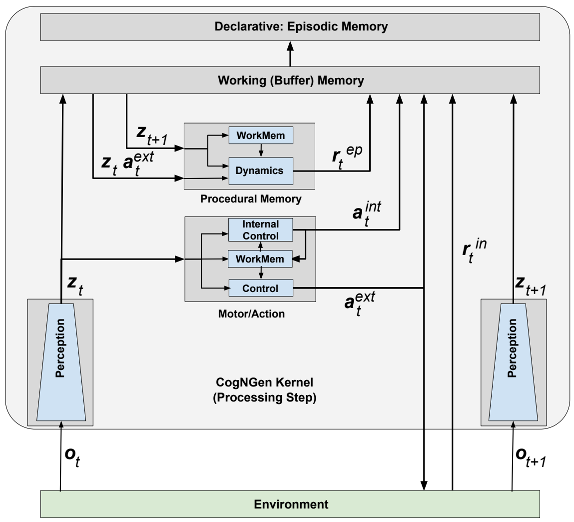


Figure 2: The CogNGen kernel architecture, depicted in its information processing mode (see Appendix for depiction of its learning mode). Black solid arrows indicate passing of data (which could be one or more vectors of information). Note that episodic memory is used to instantiate/drive CogNGen's learning step (not shown) by forcing it to recollect samples of transition sequences, transferring these to a working buffer that transfers relevant portions of its contents to the procedural memory and the motor-action modules.

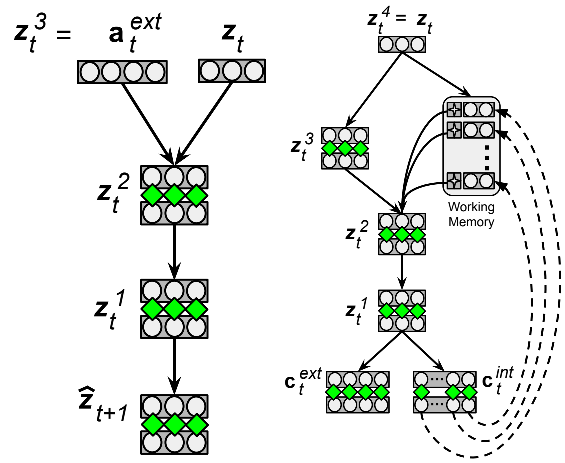


Figure 3: (Left) The CogNGen dynamics model. (Right) The CogNGen motor-action model. Arrows transform data across synapses. Gray circles represent stateful neurons and green diamonds represent error neurons. In the motor model, the internal control portion produces signals $\mathbf{c}_t^{int}$ that manipulate working memory and the external control portion transmits signals $\mathbf{c}_t^{ext}$ that affect the environment. Note that both internal and external control share the same working memory.

ral circuit that produces intrinsic reward signals. At a high level, this neural machinery facilitates some of the functionality offered by the basal ganglia and procedural memory, simulating an internal reward-creation process (Schultz, 2016).

We implement an NGC dynamics model (see Appendix[1] for details) from which a reward signal is calculated as a function of its own error neurons. We couple the dynamics model with a short-term memory module, based on MINERVA 2, which adjusts the reward value produced by the dynamics circuit by determining if an observed state is familiar or not.

In Figure 3 (Left), we graphically depict the design of the NGC dynamics model used to generate epistemic rewards (or intrinsic reward values meant to encourage exploration). This circuit takes in as input the current latent state $\mathbf{z}_t$ and the current external action $\mathbf{a}_t^{ext}$[2] to be taken by CogNGen and predicts the value of the future next step, $\mathbf{z}_{t+1}$, leveraging Equation 1 to compute its internal state layer values, i.e., $\mathbf{z}_t^3, \mathbf{z}_t^2, \mathbf{z}_t^1$.Upon receiving $\mathbf{z}_{t+1}$, the MINERVA 2 model coupled to the dynamics circuit stores the state vector, updating its current knowledge about the episode, and outputs a similarity score s^{recall}. Crucially, the contents of this MINERVA 2 are cleared upon termination of a task/maze.

To generate the value of the epistemic reward (Ororbia & Mali, 2021), the dynamics model first settles to a prediction $\hat{\mathbf{z}}_{t+1}$ (as per the process described in Section) given the value of CogNGen's next latent state $\mathbf{z}_{t+1}$. After its settling process has finished, the activity signals of its (squared) error neurons are summed to obtain the circuit's *total discrepancy* (Ororbia & Mali, 2021). This signal is next modified by the short-term MINERVA 2 memory (filter) module as follows:

$$r^{ep} = \begin{cases} \eta_e r^{ep} & s^{recall} \leq s_\theta \\ -0.1 & \text{otherwise} \end{cases} \tag{6}$$

where s_θ is an adjustable threshold that s^{recall} is compared against and $0 \leq \eta_e \leq 1$ is meant to weight the epistemic signal. In essence, if $s^{recall} \leq s^\theta$, then $\mathbf{z}_{t+1}$ is deemed unfamiliar/surprising and the agent is positively rewarded with the epistemic reward for uncovering a new (latent) state representation of its environment. Whereas if the opposite is true ($s^{recall} > s^\theta$), then the latent state is deemed familiar and the agent is provided with a negative penalty. Finally, the ultimate reward signal is computed by combining the epistemic signal with the problem-specific, instrumental (or extrinsic) reward value r_t^{in}, i.e., $r_t = r_t^{in} + r_t^{ep}$.

Neural Generative Motor Control

An agent must not only react to its environment but must also manipulate it. To do so, the agent needs circuits to drive its actuators. Building upon the notion of planning-as-inference (Botvinick & Toussaint, 2012), as in (Ororbia & Mali, 2021), we generalize NGC to the case of action-driven tasks, which we call *active neural generative coding* (ANGC).

Specifically, we design a motor-action model $f_a \colon \mathbf{z}_t \mapsto (\mathbf{c}_t^{int}, \mathbf{c}_t^{ext})$ (which offers some of the functionality provided by the motor cortex) that outputs two control signals at each time step, i.e., internal control signal $\mathbf{c}_t^{int} \in \mathcal{R}^{A_int \times 1}$ and external control signal $\mathbf{c}_t^{ext} \in \mathcal{R}^{A_ext \times 1}$. Note that a discrete internal

[1] https://www.cs.rit.edu/~ago/cogngen_cogsci2022_append.pdf
[2] $\mathbf{a}_t^{ext} \in \{0,1\}^{A_ext \times 1}$ is a one-hot encoding of discrete action a_t^{ext} and A_ext is the number of possible actions, as defined by the task.

action $a_t^{int} \in \{1, 2, ..., A_{int}\}$ is extracted via $a_t^{int} = \arg\max_i \mathbf{c}_t^{int}$ and external action $a_t^{ext} \in \{1, 2, ..., A_{ext}\}$ is extracted via $a_t^{ext} = \arg\max_j \mathbf{c}_t^{ext}$ where A_{int} is the number of discrete internal actions and A_{ext} is the number of discrete external, allowable actions. Action a_t^{ext} impacts the environment that the CogNGen system is currently interacting with while action a_t^{int} manipulates the action model's working memory.

Notably, within the NGC action-motor model is a modifiable working memory that allows the model to store a finite quantity M_w of projected latent state vectors into a set of self-recurrent memory vector slots. This particular working memory module, which we call the *self-recurrent slot buffer* (see Figures 1 & 2 and Appendix for details) serves as the glue that joins the modules of CogNGen together. Working memory in the Common Model of Cognition can be implemented in a variety of ways (Laird et al., 2017). In ACT-R (Anderson & Lebiere, 1998; Anderson, 2009), for example, the mind/brain is understood as consisting of modules connected by buffers, each storing data in a small, finite number of slots. Collectively, the buffers serve as ACT-R's working memory. The recurrent slot buffers in CogNGen serve the same purpose as ACT-R's buffers and are inspired by the memory model of Kruijne, Bohte, Roelfsema, and Olivers (2021).

Explicit Declarative Memory: Episodic Memory

In reinforcement learning, in order to improve the stability and convergence of ANNs trained over many episodes, each episode containing many transitions, experience replay is typically used (Mnih et al., 2015). In early studies of rats, neural replay sequences were detected in the hippocampus (Skaggs & McNaughton, 1996) during rest, where "place" cells spontaneously and rapidly fired in such a way so as to represent the previous paths traversed by the animals while awake. These "replay" sequences would only last nearly a fraction of a second but covered several seconds of real-world experience. Similar replay effects have been been detected in human subjects (Kurth-Nelson, Economides, Dolan, & Dayan, 2016), providing further biological justification of the replay buffer used in modern-day RL neural systems.

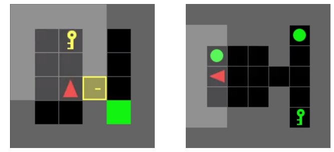
CogNGen also implements a replay mechanism in the form an episodic memory constructed with MINERVA. Information is transferred to this memory through an intermediate working buffer, where pieces of a transition (partial experience) are progressively stored as they are encountered throughout the agent-environment interaction process.

We do not update the motor-action and dynamics models online but instead update their parameters only when episodic memory is sampled. Thus, the CogNGen computational process has two phases at each time step of simulation: the processing step (Figure 2) and the learning step (see Appendix).

Experimental Results

The Mini GridWorld Problem

To evaluate CogNGen, we adapt a simulated environment from the OpenAI Gym extension, Mini-GridWorld (Chevalier-Boisvert, Willems, & Pal, 2018). We investigate two problems/tasks within its collection to evaluate the agents constructed using CogNGen, namely the *Door-Key* task and the *Memory* task. The format of each task's observation space (which is fundamentally an $N \times M$ tile grid) is a partially observable view of the agent's environment, which is created via a compact, efficient encoding of the original pixel space to a $7 \times 7 \times 3$ tensor (a 3-channel object that is created by mapping each visible grid cell to 3 integer values). Each tile contains either nothing (represented as zero) or one object (which has an associated discrete color and a discrete object type). Ultimately, each tile is encoded to an object index ($0 =$ unseen, $1 =$ empty, $2 =$ wall, etc.), a color index ($0 =$ red, $1 =$ green, etc.), and a state index ($0 =$ open, $1 =$ closed, $2 =$ locked).



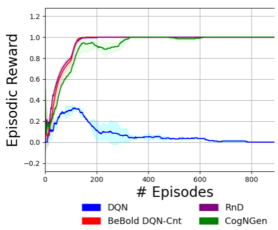
	Avg. Success Rate		Avg. Episode Length	
	DK	Mem	DK	Mem
DQN	0.00	40.0	100.0	41.14
RnD	100.0	48.5	3.71	2.78
BeBold	100.0	48.0	3.93	2.92
CogNGen	100.0	98.5	5.48	2.96

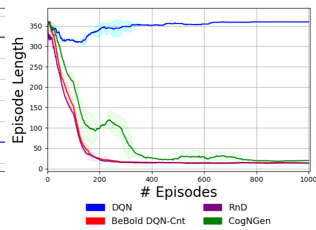
Table 1: In the top row, examples of the two tasks we experimented with are presented graphically – from left to right, the door-key task (DK) and the memory task (Mem). In the bottom row, we present results for the: (Left) Average success rate (%) over the last 100 episodes. (Right) Average episode length (% of maximum/worst-case episode length - closer to 0 is better/more efficient) over the last 100 episodes.

The agent itself is restricted to picking up one single object, such as a key, and may open a locked door if it carries a key that matches the door's color. The discrete action space for our agent can be summarized as a set of six actions: 1) turn left, 2) turn right, 3) move forward, 4) pick up an object, 5) drop the object that the agent is currently carrying, and 6) toggle/activate (such as opening a door/interacting with an object).[3] The reward signal provided by all tasks in the MiniGridWorld environment is sparse – the agent is only given a positive 1 extrinsic reward if it reaches the green goal tile and 0 otherwise, making all problems difficult from a reinforcement learning perspective. Each problem has a specific time step limit allotted to allow the agent to complete the task.

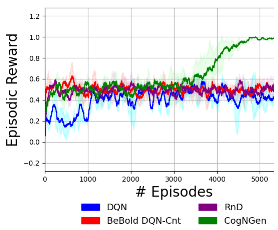
Problem Tasks For the "door-key" problem, the agent must find a key (the position of which changes across episodes) inside a locked room and then pick up the key in order to unlock the door so that it may leave the first room. Upon entering the second room, the agent must find the green tile in order to successfully exit, terminating the episode by reaching

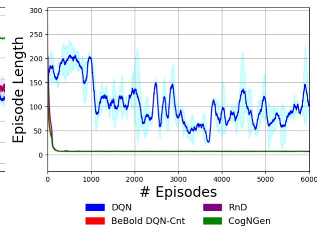
[3]Note: we omitted the optional action of raising a "done" signal.





(a) DK task - ep. reward. (b) DK task - ep. length.





(c) Mem task - ep. reward. (d) Mem task - ep. length.

Figure 4: (Left) Average reward and (Right) episode length (values are smoothed with an averaging window of 100) for (top-to-bottom): the door-key (DK) and memory (Mem) task (Right). Curves depict the mean (solid colored line) and standard deviation (light colored envelope) over five trials.

the goal state to receive a positive reward. For this study, we focus on the 6×6 room variant of the problem.

The memory task is, in contrast, posed as a memory test. The agent starts in a small room where it sees an object (a key or a ball) – note that we slightly modified the problem to ensure the agent starts the episode looking in the direction of the object. After perceiving the object, the agent must then turn around, exit the room and go through a narrow hallow that ends in a split. At the end of this split, the agent can either go up or go down, and at the end of each of these splits is a different object (either a key or a ball). To successfully complete the episode (and receive a positive reward), the agent must remember the initial object that it saw and go to the split that contains the correct matching object. For this study, we focus on the size 7 problem (7×7 room variant).

Baseline Models

We compare the CogNGen to several baseline models: a standard deep Q-network (DQN; Mnih et al., 2015), a DQN that leverages an intrinsic reward generated through random network distillation (RnD; a powerful intrinsic curiosity model; Burda, Edwards, Storkey, & Klimov, 2018), and a DQN that learns through a count-based formulation of the BeBold exploration framework (BeBold DQN-CNT; Zhang et al., 2020). In order to obtain robust and stable performance, we had to modify the RnD and BeBold intrinsic bonus calculations in order to learn in the above tasks by imposing a small negative penalty on discrete states that were visited more than once within an episode (meaning that a hash table had to be used to track global state coordinates and visitation counts of each previously seen state, which was reset at an episode's end). RnD and BeBold have access to problem-specific/global information from the Mini GridWorld environments whereas CogNGen and the DQN do not.

For details related to the settings as well as the specific values chosen for the hyper-parameters of the baselines and CogNGen's various modules, please see the Appendix.

Results and Discussion

In Table 1, we report the average success rate (at solving the task/reaching the goal state) as well as the average episode length (average measurements were computed over the last 100 simulated episodes for all models). In Figure 4, we present the reward curves, computed as the mean and standard deviation across five simulation runs.

Based on the results of our simulations, we find that (1) CogNGen is able to learn the grid-world tasks, (2) the performance is comparable to / on par with powerful deep intrinsic curiosity RL methods that have access to problem-specific, global information, and (3) that CogNGen can successfully solve and outperform all baselines on the memory task. Given that CogNGen approximates much of the functionality of modern-day RL tricks and mechanisms with large auto-associative Hebbian memory modules and predictive processing circuits, the results uncovered are promising. When comparing the baselines to CogNGen, we notice that there are some instances where the powerful BeBold DQN-CNT and RnD baselines yield shorter episodes or yield higher episodic rewards earlier (after converging to an optimal policy). We reason that this small gap/difference is likely due to: 1) BeBold DQN/RnD have access to global, problem-specific information (the agent's x-y coordinates within the world in order to calculate state visitation counts) whereas CogNGen only operates with local information/observations, 2) CogNGen's mechanism to update synapses relies on an episodic memory system that is imperfect (which is arguably more human-like yet introduces error in the recollections as compared to a standard experience replay buffer), and 3) CogNGen's motor-action model must also learn how to manipulate its coupled working memory (via internal actions) in addition to how to interact with its environment (via external actions), which requires learning a more complex policy.

Conclusions and Future Research

In this work, we presented CogNGen (the COGnitive Neural GENerative system), a novel cognitive architecture, or rather, its "kernel" (or core) composed of circuits based on neural generative coding (i.e., predictive processing) and auto-associative Hebbian memory (MINERVA 2). CogNGen lays down the groundwork for designing intelligent agents, composed of neurobiologically-plausible building blocks, that learn across diverse tasks as well as potentially model human performance at larger scales. Our results, on a challenging set of sparse reward reinforcement learning problems, show that the synergy between a classic model of human memory and predictive processing neural circuits can yield viable adaptive systems that elicit goal-directed behavior.

References

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Botvinick, M., & Toussaint, M. (2012). Planning as inference. *Trends in Cognitive Sciences*, *16*(10), 485–488.

Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2018). Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Chevalier-Boisvert, M., Willems, L., & Pal, S. (2018). *Minimalistic gridworld environment for openai gym.* `https://github.com/maximecb/gym-minigrid`. GitHub.

Clark, A. (2015). *Surfing uncertainty: Prediction, action, and the embodied mind.* Oxford University Press.

Collins, R. N., Milliken, B., & Jamieson, R. K. (2020). MINERVA-DE: An instance model of the deficient processing theory. *Journal of Memory and Language*, *115*, 104151. doi: 10.1016/j.jml.2020.104151

Dougherty, M. R. P., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review*, *106*, 180-209. doi: 10.1037/0033-295X.106.1.180

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, *3*(4), 128–135.

Friston, K. (2005). A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, *360*(1456), 815–836.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory.* New York: Wiley.

Hikosaka, O., Nakamura, K., & Nakahara, H. (2006). Basal ganglia orient eyes to reward. *Journal of Neurophysiology*, *95*(2), 567–584.

Hintzman, D. L. (1984). Minerva 2: A simulation model of human memory. *Behavior Research Methods, Instruments, and Computers*, *16*, 96-101. doi: 10.3758/BF03202365

Hintzman, D. L. (1986). "Schema abstraction" in multiple-trace memory models. *Psychological Review*, *93*, 411-428. doi: 10.1037/0033-295X.95.4.528

Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018, Jun 01). An instance theory of semantic memory. *Computational Brain & Behavior*, *1*(2), 119–136. Retrieved from `https://doi.org/10.1007/s42113-018-0008-2` doi: 10.1007/s42113-018-0008-2

Jamieson, R. K., Crump, M. J. C., & Hannah, S. D. (2012). An instance theory of associative learning. *Learning & Behavior*, *40*, 61-82. doi: 10.3758/s13420-011-0046-2

Jamieson, R. K., & Mewhort, D. J. K. (2009). Applying an exemplar model to the artificial-grammar task: Inferring grammaticality from similarity. *Quarterly Journal of Experimental Psychology*, *62*, 550-575. doi: 10.1080/17470210802055749

Jamieson, R. K., & Mewhort, D. J. K. (2011). Grammaticality is inferred from global similarity: A reply to kinder (2010). *The Quarterly Journal of Experimental Psychology*, *64*, 209-216. doi: 10.1080/17470218.2010.537932

Johns, B. T., Jamieson, R. K., Crump, M. J. C., Jones, M. N., & Mewhort, D. J. K. (2016a). The combinatorial power of experience. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the cognitive science society* (p. 1325-1330). Austin, TX: Cognitive Science Society. Retrieved from `https://mindmodeling.org/cogsci2016/papers/0236/`

Johns, B. T., Jamieson, R. K., Crump, M. J. C., Jones, M. N., & Mewhort, D. J. K. (2016b). The combinatorial power of experience. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the cognitive science society* (p. 1325-1330). Austin, TX: Cognitive Science Society.

Kelly, M. A., Arora, N., West, R. L., & Reitter, D. (2020). Holographic declarative memory: Distributional semantics as the architecture of memory. *Cognitive Science*, *44*(11), e12904. doi: 10.1111/cogs.12904

Kelly, M. A., Ghafurian, M., West, R. L., & Reitter, D. (2020). Indirect associations in learning semantic and syntactic lexical relationships. *Journal of Memory and Language*, *115*, 104153. doi: 10.1016/j.jml.2020.104153

Kelly, M. A., Mewhort, D. J. K., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal of Mathematical Psychology*, *77*, 142-155. doi: 10.1016/j.jmp.2016.10.006

Kersten, L., West, R. L., & Brook, A. (2016). Leveling the field: Talking levels in cognitive science. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the cognitive science society* (p. 2399-2404). Austin, TX: Cognitive Science Society. Retrieved from `https://cogsci.mindmodeling.org/2016/papers/0415/paper0415.pdf`

Kotseruba, I., & Tsotsos, J. K. (2018, Jul 28). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*. doi: 10.1007/s10462-018-9646-y

Kruijne, W., Bohte, S. M., Roelfsema, P. R., & Olivers, C. N. (2021). Flexible working memory through selective gating and attentional tagging. *Neural Computation*, *33*(1), 1–40.

Kurth-Nelson, Z., Economides, M., Dolan, R. J., & Dayan, P. (2016). Fast sequences of non-spatial state representations in humans. *Neuron*, *91*(1), 194–204.

Kwantes, P. J. (2005). Using context to build semantics. *Psychonomic Bulletin & Review*, *12*, 703-710. doi: 10.3758/BF03196761

Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, MA: MIT Press.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, *38*(4), 13–26. doi: 10.1609/aimag.v38i4.2744

Mannering, W. M., & Jones, M. N. (2021). Catastrophic interference in predictive neural network models of distributional semantics. *Computational Brain & Behavior*, *4*(1), 18–33.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, *24*(109), 92.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. doi: 10.1038/nature14236

Newell, A. (1973). You can't play 20 questions with nature and win. In W. G. Chase (Ed.), *Visual information processing* (p. 283-308). New York: Academic Press.

Ororbia, A. G., & Mali, A. (2021). Backprop-free reinforcement learning with active neural generative coding. *arXiv preprint arXiv:2107.07046*.

Ororbia, A. G., Mali, A., Giles, C. L., & Kifer, D. (2020). Continual learning of recurrent neural networks by locally aligning distributed representations. *IEEE Transactions on Neural Networks and Learning Systems*.

Rangel, A., Camerer, C., & Montague, P. R. (2008). A framework for studying the neurobiology of value-based decision making. *Nature reviews neuroscience*, *9*(7), 545–556.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1).

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). Act-r: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, *10*(3), e1488. doi: 10.1002/wcs.1488

Rosenbloom, P. S., Demski, A., & Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*(1), 1-103. doi: 10.1515/jagi-2016-0001

Schultz, W. (2016). Reward functions of the basal ganglia. *Journal of neural transmission*, *123*(7), 679–693.

Skaggs, W. E., & McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, *271*(5257), 1870–1873.

Steine-Hanson, Z., Koh, N., & Stocco, A. (2018, 12). Refining the common model of cognition through large neuroscience data. *Procedia Computer Science*, *145*, 813-820. (Papers on the Common Model of Cognition) doi: 10.1016/j.procs.2018.11.026

Stocco, A., Laird, J., Lebiere, C., & Rosenbloom, P. (2018). Empirical evidence from neuroimaging data for a standard model of the mind. In T. T. Rogers, M. Rau, X. Zhu, & C. W. Kalish (Eds.), *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 1094–1099). Austin, TX: Cognitive Science Society. Retrieved from `https://mindmodeling.org/cogsci2018/papers/0216/`

Stocco, A., Sibert, C., Steine-Hanson, Z., Koh, N., Laird, J. E., Lebiere, C. J., & Rosenbloom, P. (2021). Analysis of the human connectome data supports the notion of a "common model of cognition" for human and human-like intelligence across domains. *NeuroImage*, *235*, 118035. doi: 10.1016/j.neuroimage.2021.118035

Thomas, R. P., Dougherty, M. R., Sprenger, A. M., & Harbison, J. I. (2008). Diagnostic hypothesis generation and human judgment. *Psychological Review*, *115*, 155-185. doi: 10.1037/0033-295X.115.1.155

Zhang, T., Xu, H., Wang, X., Wu, Y., Keutzer, K., Gonzalez, J. E., & Tian, Y. (2020). Bebold: Exploration beyond the boundary of explored regions. *arXiv preprint arXiv:2012.08621*.