A Neurobiological Schema Model for Contextual Awareness in Robotics

Tiffany Hwu*†, Hirak J. Kashyap‡, Jeffrey L. Krichmar†‡

*HRL Laboratories, LLC
Malibu, California, USA, 90265

†Department of Cognitive Sciences
‡Department of Computer Science
University of California, Irvine
Irvine, California, USA, 92697

Email: tjhwu@hrl.com, {kashyaph, jkrichma}@uci.edu

***Abstract*—A robot operating in multiple settings must develop stable as well as flexible representations of the tasks and contexts associated with their environments. Taking inspiration from neurobiology, we apply a neural network model of schemas and memory consolidation to train the Toyota Human Support Robot to find and retrieve objects in indoor settings. We define schemas to be collections of objects bound together by a common context. In this case, the robot must learn schemas associated with rooms found in a school based on objects typically found in those rooms. Because the model develops schema representations for each room, the robot can rapidly perform object retrieval tasks associated with familiar schemas and disambiguate the tasks by context. Our experiment explores the effects of the model in an embodied setting and shows the benefits of applying research in memory consolidation to contextual awareness in robotics.**

***Index Terms*—Memory consolidation, Learning contexts, Cognitive robotics, Neuromodulation, Neurorobotics, Schemas**

I. Introduction

When operating in varied environments, robots must learn the appropriate tasks to perform within a context. This requires mental representations that are flexible enough to learn tasks in new contexts and yet stable enough to retrieve and maintain tasks in old contexts. Similarly in neuroscience, the stability-plasticity dilemma asks how the brain is plastic enough to acquire new memories quickly and yet stable enough to recall memories over a lifetime [1], [2]. Using ideas and theories from memory consolidation in neuroscience, we aim to improve contextual awareness in robotics.

One theory of how the brain balances stability and plasticity is that information is stored in schemas, which are defined as items bound together by common contexts [3]. Tse and colleagues demonstrated this by training rats on different schemas, which were collections of associations between different foods and their locations in a square arena [4]. They found that the rats were able to learn new information quickly if it fit within a familiar schema. Additionally, the rats were able to learn new schemas without forgetting previous ones. The hippocampus (HPC) was necessary for learning schemas and any new information matching a schema. In a followup study, Tse and colleagues showed increased plasticity in the medial prefrontal cortex (mPFC) when information was consistent with a familiar schema [5]. Recently, a neural network model showed how the mPFC develops representations of schemas and modulates indexing patterns in the hippocampus to form schema-specific task representations [6]. With the addition of neuromodulatory areas, the model learned rapidly when information was consistent with a familiar schema.

In this paper, we show how a model of schemas and memory consolidation could be applied to a robot task to explore its effects in an embodied setting and test how these concepts might improve contextual awareness in real-world settings. The goals of the present work are to: 1) investigate if concepts from rodent neuroscience might have useful applications for robot behavior, and 2) examine whether the original model [6], which artificially replicated highly controlled rodent experiments, can be used with noisy sensory inputs and behavior in a real-world setting. Specifically, the setting was a school in which there was a classroom and a breakroom, each with objects typically found in these rooms. Our results suggest that the interaction between the HPC and mPFC can allow the robot to learn multiple memory episodes without the need for retraining. Furthermore, the present robotic schema model may have applications to resolve catastrophic forgetting and task switching in artificial neural networks.

II. Methods

A. Neural Network Model

A neural network was created to associate visual objects and places with particular schemas. The robot learns these associations to build schemas, which facilitate object retrieval. The neural network was created to model areas corresponding to the HPC and mPFC and simulated the known connections between these areas. Memories are formed through neurobiologically plausible learning with neuromodulation known to occur in these brain areas.

The proposed neural network model follows ideas from the theory of complementary learning systems [7], which suggests that the HPC rapidly forms indices of activity in the neocortex.

This work was funded in part by Toyota Motor North America.

As both the HPC and neocortex use hierarchical representations [8], [9], our model shows how the indexing behavior and neocortical learning occur in hierarchical streams. The model introduces an Indexing Stream to indicate contexts and stimuli, and a Representation Stream that learns tasks associated with the contexts (see Figure 1). The Indexing Stream is inspired by the interactions between HPC and mPFC, and the Representation Stream has similarities to the sensory and association cortex. The use of two separate streams prevents catastrophic forgetting when training the network on multiple tasks in succession [6]. This is because the Indexing Stream learns indexes of contexts, which in turn gates the Representation Stream to separate task representations by context. Additionally, the model contains neuromodulators for detecting novelty and familiarity. For example, if an object is novel, but the context is unfamiliar, a new schema must be learned. However, if an object is novel, and the context is familiar, the object can be added to an existing schema. Each component reflects an important brain function for context-aware learning in an embodied environment. Figure 1 depicts an overview of the network, with details of the modules and learning discussed below.

1) Indexing Stream: The Indexing Stream models the functionality of mPFC-HPC circuits in the brain by clustering inputs into different contexts and indexing information at increasing levels of detail along the dorsal-ventral axis of the HPC [8]. All layers in the Indexing Stream use the following equation to calculate activity at each time step:

$$\mathbf{x}_k(t) = f(W_k \mathbf{x}_{k-1}(t-1)), \tag{1}$$

where layer k is the postsynaptic layer, layer k-1 is the presynaptic layer, W_k is the weight matrix between layer k and k-1, and f is the Rectified Linear Unit (ReLU) transfer function:

$$f(\mathbf{x}) = max(\mathbf{x}, 0). \tag{2}$$

A contextual pattern first projects to the mPFC, where each individual neuron encodes a different schema. The mPFC neuron activity vector $\mathbf{x}_k$ at time t is calculated using Equation 1. A winner-take-all scheme sets all activations in the mPFC to zero except for the one with the maximum value. The Hebbian learning rule [12] is applied to train weights from context pattern to mPFC as

$$\Delta W_k = \eta_{pattern} \mathbf{x}_k \mathbf{x}_{k-1}^T, \tag{3}$$

where $\mathbf{x}_k$ is the mPFC layer, $\eta_{pattern}$ is the learning rate, and $\mathbf{x}_{k-1}$ is the context pattern layer. The weights are normalized such that the norm of weight vectors going to each postsynaptic neuron i is 1:

$$w_i = \frac{w_i}{||\mathbf{w}||}. \tag{4}$$

$\mathbf{w}$ is the vector of weights sharing a postsynaptic neuron, and w_i is a single weight in $\mathbf{w}$. Equations 1-4 enforce a stable clustering by ensuring that no single neuron has much higher activity than the others. The ventral HPC (vHPC) indexes mPFC activity, also using Equations 1-4. The dorsal HPC (dHPC) then indexes triplets of vHPC, context pattern, and action selection layer activities in the same way, but with a learning rate of $\eta_{indexing}$. The faster learning rate compared to $\eta_{pattern}$ reflects the rapid indexing of the HPC. The mPFC, vHPC, and dHPC therefore form a hierarchical stream of indices that increase in specificity.

2) Representation Stream: Just as the neocortex performs multimodal associations for perception and cognition [13], the Representation Stream learns tasks by associating actions to cues. Contextual information from the Indexing Stream disambiguates tasks by context, similar to mPFC functionality in animals. To maintain biological plausibility, we use Contrastive Hebbian Learning (CHL) [14], a local learning rule, to learn tasks. In a network with layers 0 through L, activity of the kth layer is $\mathbf{y}_k$ and the weight matrix from layer k-1 to k is W_k. Each weight matrix W_k has a feedback matrix of γW_k^T, such that each feedback weight value is a scaled down value of the feedforward weight. In free phase, the input layer $\mathbf{y_0}$ is fixed and the neuron activations of the other layers are given by

$$y_k(t) = f(W_k y_{k-1}(t-1) + \gamma W_{k+1}^T y_{k+1}(t-1)) \tag{5}$$

where f is the ReLu function.

Equation 5 is applied for T_s time steps for convergence, consistent with standard CHL [14]. The resulting neuron activations for free phase is $\check{y}_k$. Then in clamped phase, the output is fixed to the desired value to associate with the input, while the input remains fixed. In CHL, the desired value for clamping the output can occur in a supervised fashion. In the case of this experiment, the value is determined via connections from the dHPC of the Indexing Stream. Again Equation 5 is applied for T_s time steps for convergence to make $\hat{y}_k$, for the clamped phase. The update rule is then applied:

$$\Delta W_k = \eta_{CHL}(\hat{\mathbf{y}}_k \hat{\mathbf{y}}_{k-1}^T - \check{\mathbf{y}}_k \check{\mathbf{y}}_{k-1}^T), k = 1, ..., L. \tag{6}$$

The input layer of the representation stream represents a cued input, such as a stimulus, and the output layer represents a learned action to the stimulus. Intermediate layers represent the association cortex (AC), which can be designed as one or more layers to perform multimodal associations between cues and actions. While CHL is able to learn by clamping and unclamping the output layer alone and without clamping of the intermediate layers, the intermediate layer in our implementation takes additional inputs from the vHPC during clamping. These vHPC connections to AC are randomly sampled and held static, which result in unique activations in the intermediate layer depending on which vHPC neuron is active.

3) Novelty and Schema Familiarity: Neuromodulatory mechanisms speed up learning of novel stimuli when consistent with a previously learned schema. To calculate novelty, each neuron in the dHPC projects to the novelty submodule with a starting weight that represents the baseline level of

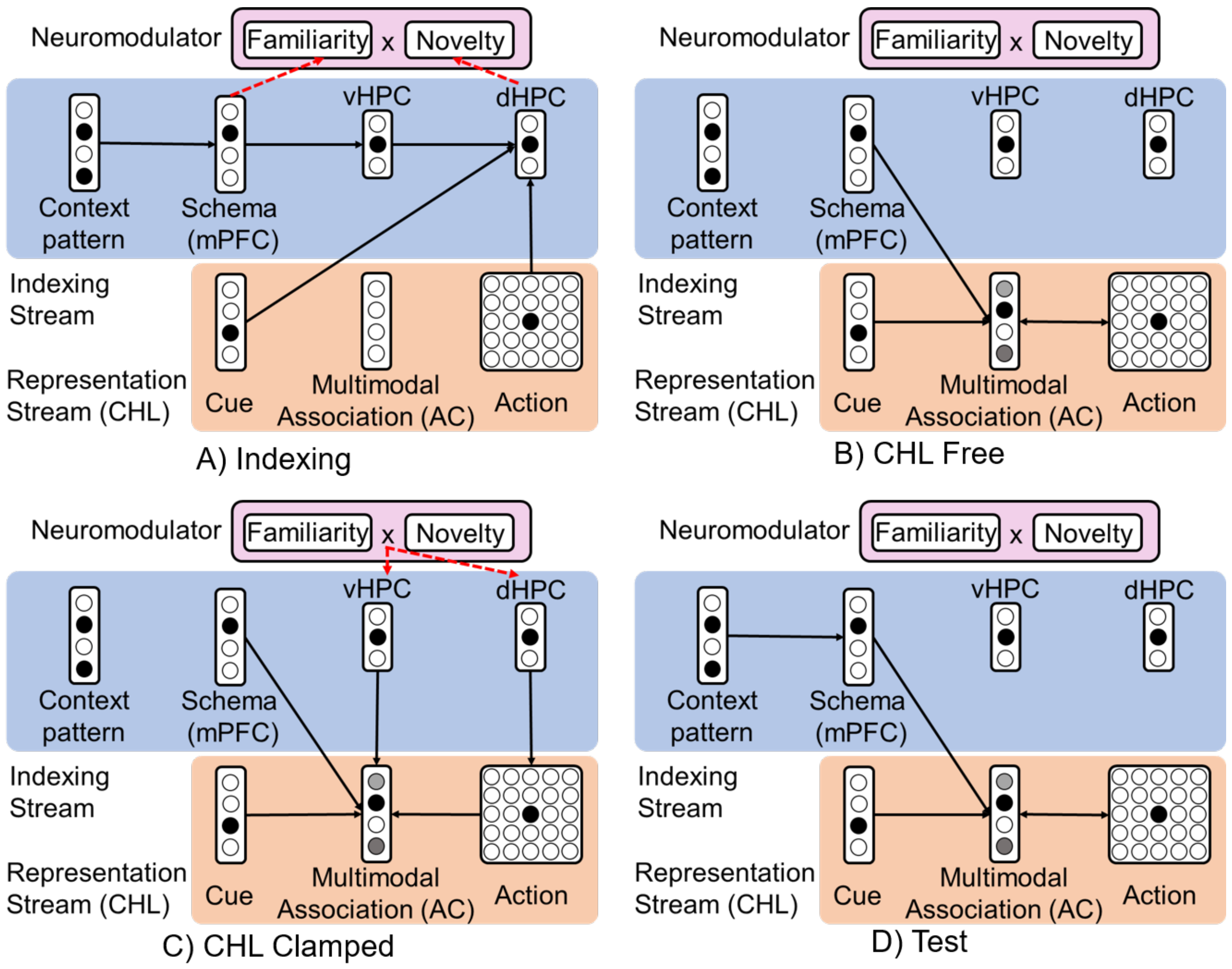


Fig. 1. Overview of network and learning phases. The blue box contains the Indexing Stream, the orange box contains the Representation Stream and the pink box contains the neuromodulators. The size of each layer in both streams is specified in Table I. There are four phases in an epoch of training. A) Indexing phase: the medial prefrontal cortex (mPFC) indexes the context pattern, the ventral hippocampus (vHPC) indexes the mPFC, and the dorsal hippocampus (dHPC) indexes triplets of vHPC, cue, and action. The activity of the neuromodulator is the product of activity from the novelty and familiarity areas. see Equations 1-4 for details. B) Free Phase of Contrastive Hebbian Learning (CHL): the cue and schema layers are clamped to an input value while the Representation Stream runs freely, as in Equation 5. C) Clamped Phase of CHL: the cue and schema remain clamped and the output is clamped via connections from dHPC. During training, results from the clamped and free phase are combined in Equation 6. D) Test Phase: this is used to query the network when performing tasks.

Fig. 2. The Toyota Human Support Robot [10], [11]. A scanning lidar at the base allows for SLAM mapping of the environment. A combination of height, arm, and gripper controls allows for objects to be picked up and put down on various surfaces. An RGBD camera allows for object segmentation, identification, and localization.

surprise when a new stimulus is presented. Whenever the activity of the dHPC is updated, the activity of the novelty submodule is found in the same way as neurons in the Indexing Stream. The weights from the dHPC to the novelty submodule are then updated with an anti-Hebbian learning rule:

$$\Delta W = -\eta_{indexing}\mathbf{x}_{novelty}\mathbf{x}_{dHPC}^{T}. \tag{7}$$

where W is the weight matrix of weights from the dHPC to novelty submodule, $\eta_{indexing}$ is the learning rate, $x_{novelty}$ is the activity of the novelty submodule, and x_{dHPC} is the activity of neurons in the dHPC. Since the dHPC uses winner-take-all, each weight from dHPC represents an individual novelty score for the corresponding dHPC neuron. The activity of the familiarity module, $x_{familiarity}$, is the weighted sum of inputs from the mPFC after winner-take-all, as in Equation 1. However, rather than a ReLU function, we use a shifted sigmoidal function:

$$f(\mathbf{x}) = \frac{1}{1 + e^{-s(\mathbf{x} - x_{shift})}}. \tag{8}$$

where s is the sigmoidal gain and x_{shift} is the amount of input shift. These weights are updated after mPFC activity is updated, using the Hebbian learning rule from Equation 3 with a learning rate of $\eta_{pattern}$. The activity of the neuromodulator is

$$neuromodulator = \mathbf{x}_{novelty} * \mathbf{x}_{familiarity} \tag{9}$$

This value determines the number of times the vHPC and dHPC will clamp and unclamp the representation layer in a single trial, emulating the resonance of brain areas in schema-consistent cases [3]. The number of extra epochs in a trial that are added to a default number of epochs, $e_{default}$ is calculated as the following:

$$epochs = e_{default} + neuromodulator * e_{boost} \tag{10}$$

4) Training Sequence: The neural network was trained to associate objects with contexts. All network weights are first initialized randomly along the range w_{min} and w_{max}, with fully connected weights between layers, with the following exceptions: 1) weights between vHPC and AC are first fully connected with a weight of w_{inh}. Then, the weights are set to 0 with probability P such that each neuron in vHPC would provoke a sparse pattern of activity in the AC, 2) weights going to the novelty module are all set to $w_{novelty}$, and 3) weights going to the familiarity module are all set to $w_{familiarity}$. As explored in [6], the level of sparsity provided by these weight values balances between catastrophic forgetting and ability to learn single tasks. The parameter values used for training can be found in Table I.

In the terminology of our experiment, one trial of training a schema consists of multiple epochs, with each epoch training a random paired association of an object and location within the schema. In one training epoch, the network runs through the first three phases in Figure 1. The number of training epochs in a trial is determined by the level of neuromodulation (see Equation 10). During CHL, the action layer is clamped to the value driven from the dHPC input. AC receives external input from the vHPC using Equation 1, which inhibits most of the AC neurons except for those that have a weight of 0 from the winning vHPC neuron. Since most of the neurons are inhibited, this effectively gates neuronal activity, as only sparse groups of non-inhibited neurons are activated for each vHPC neuron.

vHPC and dHPC therefore serve the roles of contextually gating intermediate layers of the Representation Stream and driving CHL activity. At the beginning of a trial, the number of training epochs is undetermined, but tentatively set at e_{settle} epochs (Figure 1A). During this time, activity levels of the neuromodulator are tracked, and the maximum neuromodulator activity found within this period is used to determine the ultimate number of training epochs within the trial. After each trial, the performance of the network is measured during the Test Phase in Figure 1D by presenting a cue to the network and allowing the network to settle on an action.

Population Sizes		Learning Parameters	
N_{cue}	18	T_s	5
N_{mPFC}	10	$\eta_{indexing}$	.1
$N_{multimodal}$	40	$\eta_{pattern}$	.0001
$N_{context}$	25	η_{CHL}	.001
N_{vHPC}	5	γ	.001
N_{dHPC}	40	e_{boost}	1000
		$e_{default}$	600
		e_{settle}	20
		w_{min}	0.3
		w_{max}	0.8
		w_{inh}	-10
		$w_{familiarity}$	.0001
		$w_{novelty}$	1
		s	200
		x_{shift}	.03
		P	0.3

TABLE I
PARAMETERS USED IN EXPERIMENT.

B. *Robot Inputs, Actions, and Graphical User Interface*

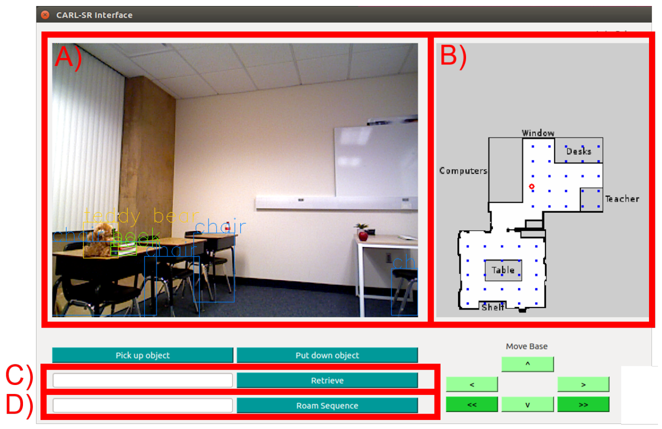


Fig. 3. The graphical user interface for controlling the HSR includes buttons for manual control of the robot as well as the following components for automated behavior: A) The head camera input of the HSR with object segmentation and detection. B) A map of the roamable area is displayed here. The red circle shows the current location of the HSR. Grid dots represent the locations of corresponding neurons in the action layer of the neural network. C) Retrieval commands are sent to the robot by entering the name of the object to be retrieved. D) A roam sequence is initiated by clicking this button, causing the HSR to explore the current room.

We test our model by training it on the task of finding and retrieving objects in an indoor space. Before running experiments, the robot mapped the two rooms using an existing SLAM algorithm [15]. During experiments, the robot uses its camera and lidar to recognize and locate objects in the room. The object/location paired associations are used to create or update schemas based on the neural network described in Section II-A. In a trial, the robot is prompted to retrieve an object, which requires prior knowledge of the schema in which the object belongs, and the location of the object. This causes the robot to navigate towards a location, recognize the object, grasp the object, and then return the object to its starting location. This is achieved by using ROS packages provided by Toyota Motor Corporation for mapping, navigation, and movement, as well as open-source libraries for computer vision. More implementation details are in the ensuing paragraphs.

We use the Toyota Human Support Robot (HSR) (Figure 2) and its associated Robot Operating System (ROS) packages to carry out the tasks [10], [11], [16], [17]. A scanning lidar at the base of the HSR allows for SLAM mapping of the environment. A combination of height, arm, and gripper controls allows for objects to be picked up and put down on various surfaces. An RGBD camera allows for object segmentation, identification, and localization. To relay commands to the robot, we create a graphical user interface in Python. Images from the camera are input to the object segmentation and recognition package, YOLOv3 [18], which is implemented in PyTorch [19]. The YOLOv3 network uses pretrained weights trained on the COCO image dataset [20]. The code for our experiment can be found at https://github.com/fitany/SchemaHSR.

The input context pattern layer contains one neuron for each detectable object, and the cue layer of our model contains one neuron for each object graspable by the robot. The activation of these neurons is set to 1 if detected by the robot in the last 60 seconds. The action layer output consists of a 2D grid of neurons representing possible locations for the objects.

For one trial of training, the robot is provided with a set of 5 locations in the room, which are hand-selected for maximum coverage. The robot visits each of these locations in random order, stopping at each point to perform a full head rotation, scanning to identify objects. Using the infrared depth sensor on the HSR, the locations and identities of all objects seen during the scan are recorded. The locations of the objects are set to the closest grid coordinate. After visiting each location, the robot then trains on the set of objects and locations, with the set of all objects seen in the past 60 seconds input to the context pattern layer. This information feeds into the mPFC layer, which indexes these objects as a single neuron representation through winner-take-all dynamics. This effectively clusters groups of objects seen in each individual room, with one mPFC neuron representing each room seen by the robot. The room in which the robot was located was thus known by the identity of the most active mPFC neuron. For all graspable objects, the RGBD camera detects their location and trains the representation stream for a single epoch with the object as the cue and the location as the action.

The performance of the robot is tested by typing the name of a graspable object into the graphical user interface and requesting the robot to retrieve the item. The neural network then receives input of all objects seen in the last 60 seconds into the context pattern, and the requested object as the cue. Running the network in Test Phase, the output layer produces different levels of activation corresponding with the belief of where the cued object was located. The activities of the 5 output neurons with highest activation are normalized into a probability distribution. The robot then visits each of these 5 points by sampling from the probability distribution, removing that point from the list, and re-normalizing the probabilities. At each point, the robot performs a full head scan. If the cued object is seen, the robot then aborts the searching process and picks up the object. The movement sequence for picking up the object is set to navigate to a specific distance and angle from the object, raising the height of the robot to match the object, lowering the arm, and closing the gripper in a pre-programmed fashion. The graspable objects in our environment are small and generally spherical, such that the same basic grasping routine is applied to all objects.

C. Experimental Design for Learning and Maintaining Schemas

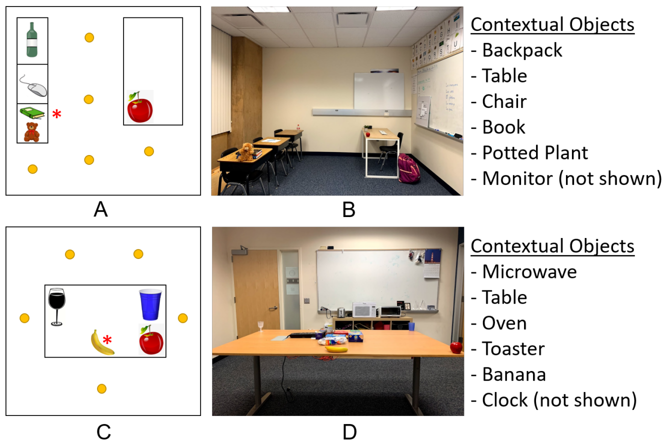


Fig. 4. Experimental setup for the classroom and breakroom schemas. Yellow dots represent the destinations for the robot to roam and scan during training. A) Room layout and paired associations trained in the classroom. The bottle, teddy bear, and apple were laid out in the specified locations. After training on this layout, the teddy bear was exchanged for the mouse, introducing novelty. The robot then trained on this environment. The book (starred) was not trained as a paired association, but was included as a contextual item during training. B) Physical setup of classroom and contextual items. C) Room layout and paired associations trained in the breakroom, which is adjacent to the classroom. The wine glass, cup, and apple were laid out on the central table. The banana (starred) was not trained as a paired association, but was included as a contextual item during training. D) Physical setup of breakroom and contextual items.

In a series of experiments, we trained the robot to retrieve objects in two adjacent rooms, a classroom and a breakroom (Figure 4).

1) Experiment 1 - Learning and updating a single schema: The first experiment was to train the robot on a single schema (Experiment 1a). The robot was placed in a classroom setup, with typical classroom items as seen in Figures 4A-B. The graspable items were an apple, a bottle, and a teddy bear, placed in different locations around the room. After training and testing for 5 trials on the classroom, the teddy bear was replaced by a computer mouse (Experiment 1b). The robot then went through one trial of training, and was tested on its ability to retrieve this novel object. In all testing trials, the robot started at the same location in front of the computers, and brought the retrieved items to a neutral location near the door separating the two rooms.

2) Experiment 2 - Maintaining multiple schemas: The second experiment was to train the robot on a second schema consisting of a breakroom as seen in Figures 4C-D. The graspable items were an apple, a cup, and a wine glass. After training and testing for 5 trials, the classroom schema was tested again to see if the robot was able to maintain performance of prior tasks. As the apple was present in both contexts, we wanted to test whether the neural network could place this overlapping object in both schemas. For all testing trials, the robot started by facing the table and microwave, and brought objects to the same neutral location between the two rooms. When choosing locations to explore, only neurons corresponding to the breakroom were used.

3) Experiment 3 - Schema prompting: The third experiment was to test whether schemas could help the robot retrieve items it was never explicitly trained to retrieve. Starting from the neutral drop-off location, the robot was shown a banana for context, and nothing else. Then, it was cued to retrieve the banana. Since the banana was part of the breakroom schema, we predicted that the robot would search for the banana in the breakroom first as opposed to the classroom. Figure 7 shows the sequence of actions when retrieving the banana. We repeated the same procedure with a small graspable book to ensure that schema prompting would work for the classroom schema as well.

III. Results

A video of our main results can be seen at https://youtu.be/_lPV0BuhSBI. We repeated all experiments 5 times. For each repetition, the weights were randomly initialized at the beginning and used throughout all experiments. Figure 5 shows the performance in Experiments 1 and 2. Performance was calculated analytically by cueing each graspable item in the room to the network, running it in Test Phase, and calculating the percentage of activation level of the action neuron corresponding to the location of the object. Behavioral performance in terms of retrieval time was also tested on Trial 0 (prior to any training), and Trial 4. The performance of an individual in any particular trial was averaged over each of the graspable objects in the room. In Experiment 1a, performance improved over the trials. In Experiment 1b, the retrieval time was quick despite having only one training trial, which showed that a novel object could be rapidly incorporated into an existing schema. Figure 6 shows example trajectories taken by the robot after training in each experiment. The robot goes directly to the location of the item, picks it up, and returns to drop off the item at a neutral location.

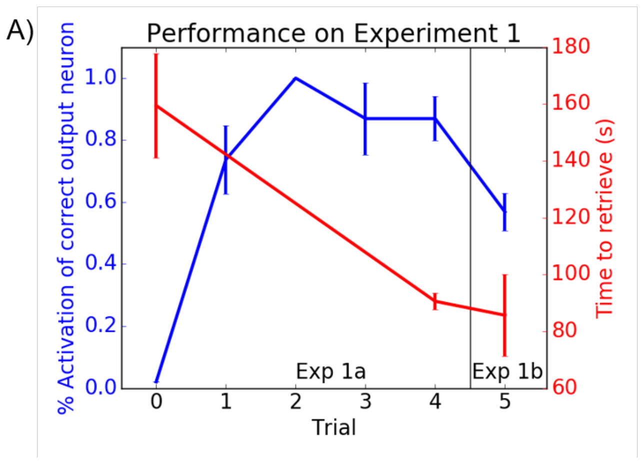
In Experiment 2, performance improved over time with this new context or schema (Figure 5). As in Experiment 1, a novel object that fit within the context was quickly learned. Furthermore, when the individual was prompted to retrieve an object in the previously learned schema (CR in Figure 5), the performance was not degraded, demonstrating the ability to hold two separate schemas in memory without forgetting either one.

In Experiment 3, the robot explored the room corresponding with the schema containing the prompted object, dropping the object in the appropriate location (Figure 7). Figures 8A-B show that the action layer activation is high for object locations in the same room as the prompted item. Figure 9 shows selected sets of weights in the network after training on all experiments. Weights from the context pattern to mPFC show two rows containing distinct patterns of high and low weight values. This indicates that there is one mPFC neuron encoding each schema. Weights from the vHPC, cue and action to dHPC show 6 rows, each with 3 high values. This means that some dHPC neurons are encoding triplets of cue, action, and schema. Weights to the novelty and familiarity areas show that novelty has decreased for some dHPC inputs and familiarity has increased for the two schema neurons. Weights from mPFC to AC show two columns representing the two schema neurons, each containing a different pattern of high weight values. This shows the gating effect of the vHPC on the AC, which separates task representations by schema. More of the gating is seen in weights from AC to action, as the action neurons corresponding to the locations of items have strong weights from different sets of AC neurons.

IV. Discussion

Neuroscience research on schemas inspired a memory model, which allowed the Toyota Human Support Robot to separate and maintain objects by context, quickly learn the locations of novel objects, and retrieve new objects by its knowledge of a schema. This relates to current approaches in improving contextual awareness when carrying out commands [21]. For instance, as the HSR is designed to aid humans in household tasks, context is important when determining how to carry out a task. Maintaining schemas of different rooms allows the robot to be aware of contexts. When asked to retrieve items at an unknown location, the robot explored the rooms associated with the appropriate schema. This led to faster retrieval times. The same principle can be applied to other tasks, such as tidying up, seeking individuals, or behaving appropriately given a context (e.g., quiet in the classroom, raising one's voice in a noisy breakroom).

There have been various approaches to conceptual understanding of space in robotics. For instance, [22] combines SLAM systems and object recognition with clustering and semantic knowledge for better navigation. Hierarchical conceptual representations of space have been explored by [23]. Probabilistic models of episodic memory have also been used to store and recall experiences of robots, improving control and human-robot interaction [24], [25]. Compared to these works, our model ties in knowledge of brain connectivity and functionality, and studies neurobiological mechanisms for higher order reasoning and planning.



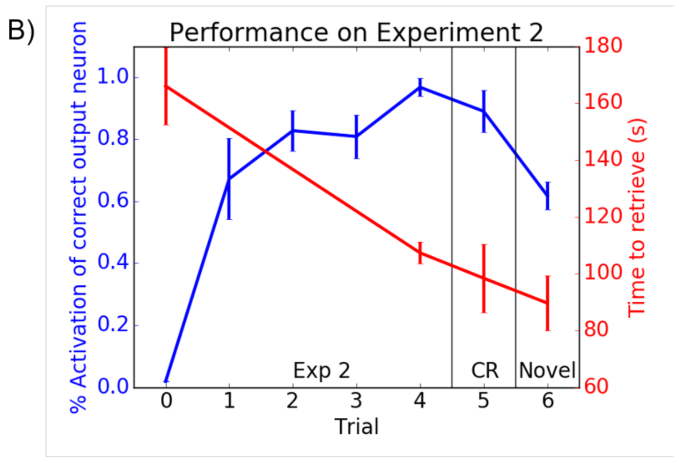


Fig. 5. Performance on Experiments 1 and 2 on a population of n = 5. A) For Experiment 1, the activation of the correct location neuron of a cued object increased with training (blue line) and the retrieval time decreased (red line). When a novel object was introduced (Exp 1b), the location activation was still high and retrieval time was low, despite only having had one trial of training. B) Performance also improved over time when a novel schema was introduced in Experiment 2. When returning to the classroom schema (CR), performance of original objects and novel objects was retained. Points denote the mean performance and error bars denote the standard deviation of the population.

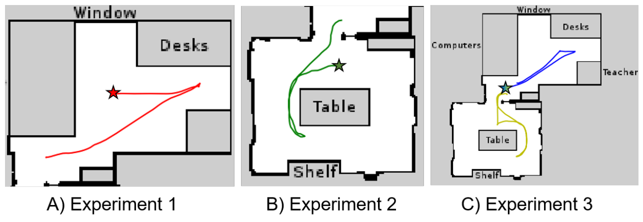


Fig. 6. Trajectories of an individual robot at various stages of the experiments. Stars indicate the start of the trajectory. A) The red line shows the trajectory of the robot when retrieving the bottle after Trial 4 of training in Experiment 1. B) The green line shows the the retrieval of an apple in Trial 4 of Experiment 2. C) The yellow line shows the trajectory of the robot retrieving the banana in Experiment 3. The blue line shows retrieval of the book. Although the entire area is accessible during retrieval of both objects, each object is associated with a prior schema, prompting the robot to search in the room corresponding to that schema.

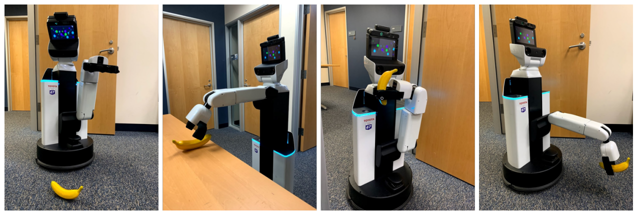


Fig. 7. Sequence of actions in Experiment 3. The HSR was shown a banana, with nothing else in its field of view. The experimenter then placed the banana on the breakroom table, outside of the HSR's view. The robot went to the breakroom to pick up the banana and navigated to the drop off location to deposit it.

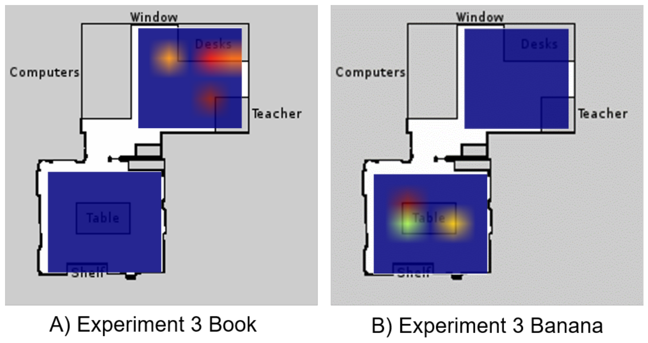


Fig. 8. Heatmap of action layer during Experiment 3. A) When the robot was presented with a book and asked to retrieve it, the action layer showed high activity for objects in the classroom schema. B) When the robot was presented with a banana and asked to retrieve it, the action layer showed high activity for objects in the breakroom.

To meet our objectives of examining whether the neurobiological model could be usefully applied to machine learning and robotics, we designed experiments similar to those performed in the original rat experiments [4]. Our robot experiments showed basic concepts of: 1) learning a schema, 2) gradually incorporating novel information into a schema, 3) rapidly consolidating information into an existing schema if it fits within the schema context, and 4) maintaining multiple schemas in memory. The robot experiments, which were intentionally simple in order to reduce environmental confounds and isolate different brain functionalities, constitute an important first step in designing memory systems for machine learning and robotics.

In the future, we hope to combine multiple levels of abstraction in the Indexing Stream by adding multiple layers in the HPC. By representing these levels in a fully connectionist way, our model could extend to end-to-end deep learning techniques for task learning. With the addition of the Indexing Stream to a basic task-learning network like the Representation Stream, a deep neural network can use contextual information to maintain separate representations of tasks. This may aid in the prevention of catastrophic forgetting if the robot must learn tasks in multiple settings.

Furthermore, the neuromodulator model introduced here can detect novelty and familiarity in the physical environment, prompting an increase in training epochs to learn new task-relevant information. By only increasing learning when a

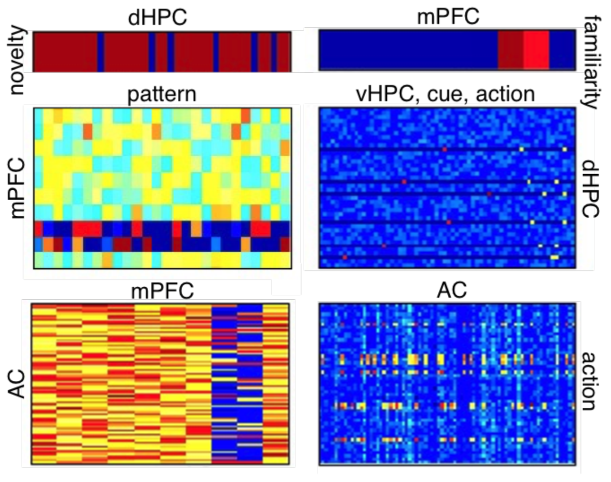


Fig. 9. Selected weights on one individual after the experiment. Each set of weights is depicted as a heatmap, with warmer colors representing higher values, rows representing the post-synaptic layer neurons, and columns representing the pre-synaptic layer neurons.

schema is familiar and a paired association is novel, the robot avoids learning irrelevant information. This has potential applications in one-shot learning [26], as the use of a neuromodulatory area is a bio-inspired approach to rapid learning.

The use of robotics to explore hippocampal function has revealed insights into how spatial representation in the brain arises from navigation. Our experiment shows how context is tied to these spatial representations via interaction with the mPFC. By studying the model in an embodied setting, we learned of various sources of uncertainty that must be addressed by the brain. Initial tests showed that inaccurate object detection and depth readings led to large errors in the network. Furthermore, the model was unaware of errors in grasping, attempting to complete the retrieval sequence even if the object had not been grasped. While it is possible to decrease these sources of uncertainty, the brain still must process remaining uncertainty throughout the network. For instance, rather than having activation inputs of 1 or 0, the activations could relate to the certainty of object detection from perceptual errors. Likewise, the strength of activations in the output layer may prompt the robot to allocate more resources to motion planning when the action to be performed is uncertain.

V. Conclusion

The present work demonstrates how ideas from memory models in the brain may improve robotic applications and issues in artificial intelligence, such as catastrophic forgetting and lifelong learning.

References

[1] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in psychology*, vol. 4, p. 504, 2013.

[2] W. C. Abraham and A. Robins, "Memory retention–the synaptic stability versus plasticity dilemma," *Trends in neurosciences*, vol. 28, no. 2, pp. 73–78, 2005.

[3] M. T. van Kesteren, D. J. Ruiter, G. Fernández, and R. N. Henson, "How schema and novelty augment memory formation," *Trends in neurosciences*, vol. 35, no. 4, pp. 211–219, 2012.

[4] D. Tse, R. F. Langston, M. Kakeyama, I. Bethus, P. A. Spooner, E. R. Wood, M. P. Witter, and R. G. Morris, "Schemas and memory consolidation," *Science*, vol. 316, no. 5821, pp. 76–82, 2007.

[5] D. Tse, T. Takeuchi, M. Kakeyama, Y. Kajii, H. Okuno, C. Tohyama, H. Bito, and R. G. Morris, "Schema-dependent gene activation and memory encoding in neocortex," *Science*, vol. 333, no. 6044, pp. 891–895, 2011.

[6] T. Hwu and J. L. Krichmar, "A neural model of schemas and memory encoding," *Biological Cybernetics*, pp. 1–18, 2019.

[7] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[8] H. Eichenbaum, "Prefrontal–hippocampal interactions in episodic memory," *Nature Reviews Neuroscience*, vol. 18, no. 9, p. 547, 2017.

[9] P. Lavenex and D. G. Amaral, "Hippocampal-neocortical interaction: A hierarchy of associativity," *Hippocampus*, vol. 10, no. 4, pp. 420–430, 2000.

[10] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of the research platform of a domestic mobile manipulator utilized for international competition and field test," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7675–7682.

[11] ——, "Development of human support robot as the research platform of a domestic mobile manipulator," *ROBOMECH Journal*, vol. 6, no. 1, p. 4, 2019.

[12] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[13] A. A. Ghazanfar and C. E. Schroeder, "Is neocortex essentially multisensory?" *Trends in cognitive sciences*, vol. 10, no. 6, pp. 278–285, 2006.

[14] J. R. Movellan, "Contrastive hebbian learning in the continuous hopfield model," in *Connectionist models*. Elsevier, 1991, pp. 10–17.

[15] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[16] T. Yamamoto, T. Nishino, H. Kajima, M. Ohta, and K. Ikeda, "Human support robot (hsr)," in *ACM SIGGRAPH 2018 Emerging Technologies*. ACM, 2018, p. 11.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*. Kobe, Japan, 2009.

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[19] "pytorch-yolo3," https://github.com/marvis/pytorch-yolo3/, 2015.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] R. Paul, J. Arkin, N. Roy, and T. M Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," *The International Journal of Robotics Research*, 2016.

[22] I. Kostavelis and A. Gasteratos, "Semantic maps from multiple visual cues," *Expert Systems with Applications*, vol. 68, pp. 45–57, 2017.

[23] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.

[24] M. Sigalas, M. Maniadakis, and P. Trahanias, "Episodic memory formulation and its application in long-term hri," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2017, pp. 599–606.

[25] Y. Endo, "Anticipatory robot control for a partially observable environment using episodic memories," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2852–2859.

[26] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.