Chapter

An Episodic-Procedural Semantic Memory Model for Continuous Topological Sensorimotor Map Building

Wei Hong Chin, Naoyuki Kubota and Chu Kiong Loo

Abstract

For humans to understand the world around them, learning and memory are two cognitive processes of the human brain that are deeply connected. Memory allows information to retain and forms an experiences reservoir. Computational models replicating those memory attributes can lead to the practical use of robots in everyday human living environments. However, constantly acquiring environmental information in real-world, dynamic environments has remained a challenge for many years. This article proposes an episodic-procedure semantic memory model to continuously generate topological sensorimotor maps for robot navigation. The proposed model consists of two memory networks: i) episodic-procedural memory network (EPMN) and ii) semantic memory network (SMN). The EPMN comprises an Incremental Recurrent Kernel Machines (I-RKM) that clusters incoming input vectors as nodes and learns the activation patterns of the nodes for spatiotemporal encoding. The SMN then takes neuronal activity trajectories from the EPMN and task-relevant signals to update the SMN and produce more compact representations of episodic experience. Thus, both memory networks prevent catastrophic forgetting by constantly generating nodes when the network meets new inputs or updating node weights when the incoming input is similar to previously learned knowledge. In addition, idle or outlier nodes will be removed to preserve memory space.

Keywords: episodic memory, semantic memory, sensorimotor map, topological map, robot navigation

1. Introduction

One of the essential features of common living locomotive organisms is their capability to traverse their daily environment with life-critical tasks. For example, rats can learn to visit or avoid places of food that they have visited, and squirrels are excellent at rediscovering places of food that they have previously hidden. Many animals escape to a previously visited shelter if they are undergoing an urgent threat, such as a bear that escapes to a cave for hibernation to preserve energy during the winter season. A specious hypothesis is that living organisms should have a cognitive mechanism to represent their environment as a collection of important regions, such as nest locations and food places. When necessary, they can recall these regions and utilize their relations to perform navigation tasks [1].

The capability of an autonomous mobile robot to represent its environment as a spatial map and to determine its position concurrently has been widely analyzed in the robotics society. The process is termed SLAM (Simultaneous Location and Mapping), and several state-of-the-art have been introduced that works remarkably well [2, 3]. Another research area is to generate a topological map that maps the environment's structure. Robots can plan trajectories and navigate to target locations using topological graphs. However, the sophistication of maps increases exponentially with the length of the robot's journey in most current graph-based approaches [4]. If new nodes and edges are added to the map continuously, the requirement for processing time and memory storage increases over time, stopping applications from long-term mapping. As a result, methods for controlling the scale of the topological map are critical in functional robotic applications that require continuous exploration in environments [5–7].

Biological methods do not appear to experience enormously from the deficiencies mentioned above in artificial navigation [1]. For example, rats can explore, search, and travel in large and dynamic environments for a long time. They can adapt to the environment changes quickly, for instance, searching new ways if a previously visited route is unavailable or choosing potential shortcuts when new access spots are available. Therefore, several computer goal-oriented navigation systems were introduced to partially emulate how the brain could represent space and apply these representations for navigation tasks. Memory is a fundamental perspective for the acquisition of experience. Memory is essential for the understanding, learning, and cognition of the interactions of robots in complex environments [8]. Episodic memory is a kind of memory that retains human experiences in a particular and conscious way.

This article proposes an episodic-procedural semantic memory model for topological sensorimotor map construction. The robot can use the generated topological sensorimotor map to perform indoor navigation. The following are our contributions to this study: i) The proposed model can learn multiple sensory information to generate the topological sensorimotor map incrementally; ii) Because of the nature of episodic memory attributes, the robot can perform goal navigation with path-planning algorithms; iii) The semantic memory layer can serve as a medium for humans to interact with a robot to perform navigation tasks, and iv) The proposed method continuously updates the generated topological map (can expand or shrink) to maintain the size of the map based on the environment without the need for human interference.

2. Related works

Many practical approaches to solving the SLAM problem have been introduced in robot mapping. Lu and Milios [9] were the first to use a pose graph to implement global map optimization. The robot's poses are represented as vertices in a graph, and the spatial boundaries between poses are represented as edges in a graph. The map's scale proliferates in this traditional graph-based approach as the robot discovers new regions. Consequently, there is a rapid rise in the need for storage and computing resources. When direct linear solvers are applied, the traditional graph-based solution has quadratic memory consumption growth with the number of variables in the worst case. Efforts to increase the performance of graph-based mapping algorithms are underway. In standard equations, the sparsity structure of the matrix is used to allow quick linear online solvers. Many SLAM libraries are available to deal with this dilemma with tens of thousands of variables in just a few seconds, such as g2o [10], and RTAB-Map [11]. Memory usage increases linearly with the number of variables, even using iterative linear solvers. Returning to the exact location many times complicates the case. This strategy becomes less effective as more vertices and edges are applied to the same spatial area. For the time being, there are only a few works that attempt to answer how to store a map for long-term exploration. Consequently, achieving a long-term mapping solution [5] that can control, or at the very least restrict, the size of the map is essential.

Vertex and edge sparsification, which trades map precision for memory and computational power, is one of the most effective techniques to reduce the map's complexity. To avoid redundant vertices and insert informative measurements to the map, an information-based compact pose SLAM algorithm was proposed in an information-theoretic fashion [12]. In pose global optimization, an information-based criterion was adopted to determine the laser scans should be marginalized, maintaining the sparsity of laser-based 2D pose maps. To obtain a light blanket based on the Markov blanket of a boundary vertex, the generic linear constraint criteria [13] and nonlinear graph sparsification were proposed [14].

Another approach was introduced that focused on solving the traditional pose graph's temporal scalability [15]. This approach eliminates the addition of redundant vertices and edges before the graph's global optimization. This approach has been demonstrated in indoor areas using a binocular visual SLAM framework, and it is an effective solution for medium-scale environments such as houses and factories. The idea of neighborhood area and scene integration is introduced [7] to achieve sparsification of the cognitive map without adding unnecessary vertices and edges to the cognitive map.

One of the biologically-inspired proposed methods is RatSLAM [16, 17]. The approach represents the environment as a set of pose cells, and each pose cell is linked to a view cell. RatSLAM was successfully implemented in small and large environments for spatial mapping, but the framework does not handle target-oriented navigation. Erdem and Hasselmo [18] proposed a biologically inspired computational model for goal-oriented navigation. In this model, the environment is represented as several grid cells with different scales and spacing and gradually converge into one place cell. The model gradually recruits new place cells to encode the autonomous agent's current location when the agent meets a notable location during exploration. Each place cell has a reward cell, and the lateral weight of the connection between two reward cells is equivalent to the time between the autonomous agent's successive visits to the reward cells. With the lateral connections, autonomous agents can navigate to the goal location from its starting location. However, the methods mentioned above focus on emulating place cells and grid cells for spatial map building.

Humans seem to accommodate themselves better in complex environments and recall past experiences to perform tasks simultaneously generate new experiences and skills. These significant behaviors usually develop from experiences that rely on learning. Likewise, the assumption is that experience also implies for robots [19]. Thus, the learned experiences can be integrated into a spatial map so that robots can freely observe and navigate in any environment. Current methods rely on the RatSLAM concept, such as BatSLAM [20] using sonar sensing, which has been developed. Tang et al. [21] included an episodic memory module in navigational tasks to process contextual information. The approach is designed for maze-controlled situations, but its effectiveness in open spaces such as corridors, offices, and homes is still unknown.

3. Proposed method

The proposed model consists of two hierarchical memory networks: i) episodic-procedural memory and ii) semantic memory. New nodes (experiences) are generated in each memory network as new sensory information is obtained. Topology links are generated to connect nodes and store robot behaviors. These connections provide the robot with procedural knowledge so that an action can be taken to proceed from one circumstance to another. The episodic-procedural network is an Incremental Recurrent Kernel Machines (I-RKM) which incrementally cluster incoming input data as nodes in an unsupervised fashion. The I-RKM is the Infinite Echo State Network extension [22, 23]. Each node in the network further encodes an activation value used for spatiotemporal learning. The semantic memory network is hierarchically connected to the episodic memory network. It is also another I-RKM that receives bottom-up inputs from the episodic memory network and top-down signals such as labels or signs for generating representations that contain semantic knowledge on a larger timescale. The mechanism of neural operation in the semantic memory network is similar to the episodic procedural memory network with an additional requirement to create a new node. In this network, node learning happens as the network correctly predicts the class label of the classified input sequence from the episodic memory network through the learning process. A new node will only be created if the incorrect network class label. This criterion is also the additional element that modulates nodes update. In particular, each semantic node preserves information over time sequences higher than episodic nodes due to the hierarchical learning of input data.

The episodic network serves as a novelty detector in the robot navigation mission. Each node in the network represents a group of related input features and creates new nodes if the incoming input features do not fit into any network nodes. Nodes in the episodic network also encode the robot's location for localization purposes. In addition, each link encodes a robot's action, such as turning angle and moving speed, to serve as procedural information that allows the robot to perform a sequence of actions and travel from one place to another. Each node encodes the semantic meaning of human operator cues in the semantic network. Semantic definition marks the explored space with various names, such as a hallway, room, or kitchen, to provide a medium for human-robot interaction. If no external sensory information is available, the episodic procedural memory network performs an action-oriented internal simulation through the playback of node sequences and actions encoded in their links to consolidate knowledge (memory) and mitigate catastrophic forgetting. Each node in the SMN represents a region of the environment. The robot utilizes this information to change its moving behaviors, such as wall following, obstacle avoidance, or fast travel. **Figure 1** shows the overview of the proposed method.

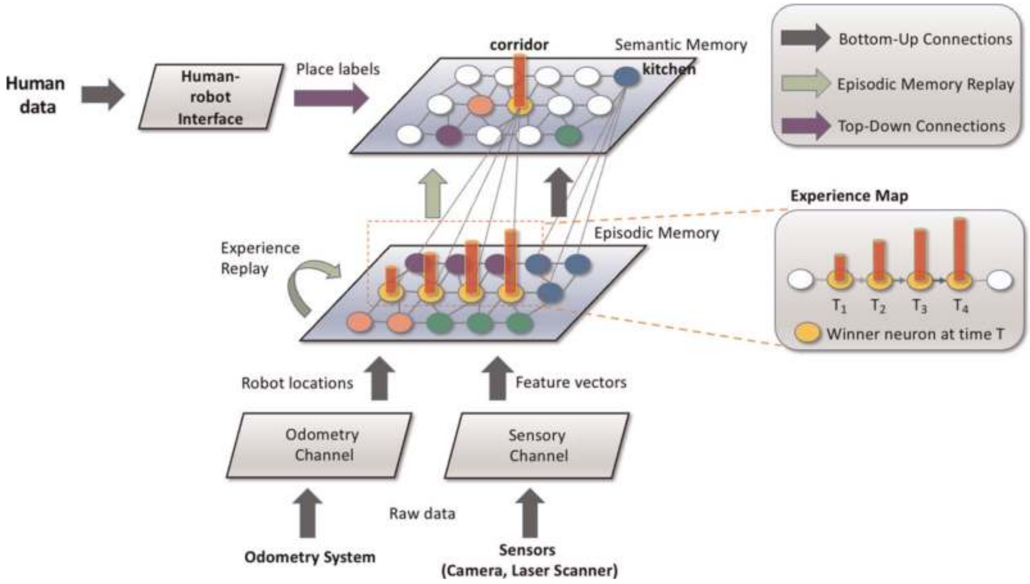

Figure 1.
The overview of the proposed method which consists of two memory networks: The episodic procedural memory network and semantic memory network. The episodic procedural memory network clusters incoming sensory input as nodes progressively and learns fine-grained spatiotemporal correlations between them. The semantic memory network adjusts the amount of architectural flexibility based on task-relevant inputs to build a topological semantic map with more compact episodic representations.

3.1 Echo state network

Echo State Networks [24] can be considered large, randomly recurring neural networks with a single sequential, trained readout layer. The network computes a wide range of non-linear, spatial–temporal mappings of input data. The reservoir can be seen as a spatial–temporal kernel in which the mapping of a high-dimensional space is explicitly computed. Hermans et al. [22] proposed a Recurrent Kernel Machines (RKM) that extends Echo State Networks' idea to infinite-sized recurrent neural networks (RNNs). The proposed method is regarded as recursive kernels. When a RNN with internal weights W, input weights V, and an internal state s receives an input x_t at time t, it produces the following output:

$$y_t = h(Vx_t + Ws_t) \tag{1}$$

where h is the product of the activation function (for example, the hyperbolic tangent) and the projection function. A recursive method's core idea is that Eq. (1) can be represented as follows:

$$h(Ws_t + Vx_t) = h\left([W|V]\begin{bmatrix} s_t \\ x_t \end{bmatrix}\right) \tag{2}$$

It's a function of the input's concatenation with the prior internal state. The same reasoning can be applied to kernel functions, with the base function inputs consisting of a concatenation of the current input and the prior recursive mapping:

$$\phi(x_t, \phi(x_{t-1}, \phi(\ldots))) = \phi([x_t|\phi(x_{t-1}|\phi(\ldots))]) \tag{3}$$

Hermans et al. [22] has shown that recursive variations of kernels with $k(x, x') = f\left(\|x - x'\|^2\right)$ and $k(x, x') = f(x \cdot x')$ form can be derived using this structure as a reference. For example, the recursive-SE kernel has the form:

$$\kappa_t^{SE}(x, x') = \exp\left(-\frac{\|x_t - x_t'\|^2}{2l^2}\right)\exp\left(\frac{\kappa_t^{SE}(x, x') - 1}{\sigma_p^2}\right) \tag{4}$$

We propose a computational model called Incremental Recurrent Kernel Machines (I-RKM) for continuously creating topological maps based on characteristics of RKM. The EPMN and the SMN are two hierarchical memory levels in the proposed method. The I-RKM is described in-depth in the following sections.

3.2 Episodic procedural memory network (EPMN)

An I-RKM constitutes the EPMN. In reaction to input vectors, the network dynamically grows or contracts by adding or removing nodes. To encode node relationships, edges will be created to connect nodes. The I-RKM notations are tabulated in **Table 1**.

Based on the sensory input, the network first generates two recurrent nodes. Each node in the network is comprised a weight vector w_j. For further learning, the network uses the Eqs. (5) and (6) to identify the node that best fits the current sensory input $x(t)$. Eq. (6) creates the Infinite Echo State which is identical to Eq. (4).

$$b = \arg\min\left(T_j(t)\right) \tag{5}$$

$$T_j(t) = \exp\left[-\frac{\|x^c(t) - w_j^c\|^2}{2\sigma_i^2}\right]\exp\left[\frac{T_j(t-1) - 1}{\sigma^2}\right] \tag{6}$$

Notation	Definition
$T_j(t)$	Activation value of node j at t
$\kappa(t)$	Recursive kernel at t
$w_b(t-1)$	Best matching node weights at $t-1$
r_j	Regularity counter of node j
γ_j	Contributing factor of node j
τ_j, λ	Decay factors for regularity counter
σ_i^2, σ^2	Kernel width
ρ	Learning threshold
$P_{(m,n)}$	Temporal connection between node m and n
V	Associative matrix for labeling
b	Index of best matching node

Table 1.
The notations of I-RKM.

Following that, the activation value of the best matching node (BMN) J is determined as follows:

$$a_b(t) = \exp\left(-T_b\right) \tag{7}$$

If the activation value $a_b(t)$ is smaller than a predefined threshold a_T, the condition is fulfilled. A new node N is added to the network with the following weights:

$$w_N = 0.5 \cdot (x(t) + w_b) \tag{8}$$

To connect the winning node b and the second BMN, a new link is established. If $a_b(t)$ is greater than a_T, the winning node b can represent the input $x(t)$. As a result, the winning node b and its neighbor nodes n are updated as follows in response to input $x(t)$:

$$w_{j(\mathrm{new})} = \gamma_j \cdot r_j \cdot \left(x(t) - w_{j(\mathrm{old})}\right) \tag{9}$$

If no connection exists between the BMN $a_b(t)$ and the second-best matching node, a new connection will be made to connect them. Each edge has an age counter that grows by one with each iteration. The age of the link between the best and second-best matching nodes is reset to zero. Nodes with no connections and a habituation counter larger than the preset value will be removed from the network, as will connections with an age greater than the preset threshold. In addition, each episodic node has a regularity counter $r_j \in [0, 1]$ that indicates the strength of its firing over time. The value of the newly formed episodic node is $r_j = 1$. Using the following equation, the regularity value of the BMN and its adjacent nodes decreases with each iteration:

$$\Delta r_j = \tau_j \cdot \lambda \cdot \left(1 - r_j\right) - \tau_j \tag{10}$$

As a result, the significance of the node's regularity can be associated with the relevance or importance of the information stored in the node. Regularity values for nodes that have been often activated in response to learning inputs are presented in the regularity Eq. (10). If the link exceeds the threshold, isolated nodes will be removed from its network. Due to the nature of the network, the topological network expands during the robot's journey in the robot navigation mission. However, nodes generated at the start of the journey are eliminated from its network. Thus, we have introduced a new criterion of node removal [25] with the following equation:

$$v = \mu(H) + \sigma(H) \tag{11}$$

where H is a vector representation of the network's regularity, μ is the mean function, and σ is the standard deviation. Nodes with regularity values more than the threshold will be removed.

Only if $b_J(t) < \rho_b$ and $r_J < \rho_r$ can a new episodic node be added to the network. If the activation and regularity thresholds are met, the episodic nodes will be updated via Eq. (9). In the EPMN, a set of events constitutes an episode, which retains distinct historical occurrences and episodes that are linked to one another. To learn recurrent node activation patterns in the network, we incorporate temporal connections. Temporal connections represent the sequence of activated nodes throughout the learning stage.

A temporal connection between the two consecutively activated nodes will be enhanced by 1 for each learning iteration. When the BMN b is activated at time t and then again at time $t-1$, the temporal relationship between them is reinforced as follows:

$$P^{\text{new}}_{(b(t),b(t-1))} = P^{\text{old}}_{(b(t),b(t-1))} + 1 \tag{12}$$

For each recurrent node m, the next node g from the encoded time series can be obtained by selecting the largest value of P as shown below:

$$g = \arg\max P_{(m,n)} \tag{13}$$

where n are the neighbors of m. As a result, the recurrent node activation sequence can be reestablished without the need for any further input data.

3.3 Semantic memory network (SMN)

The semantic memory layer is linked to the episodic memory layer hierarchically. It is made up of an I-RKM that obtains bottom-up inputs from the episodic memory layer and top-down inputs such as labels or tags to develop representations that incorporate semantic information over a more extended period. By delivering signals from the top-down signals, semantic information could be retrieved.

The mechanism of neural activity in the SMN is similar to that of the EPMN, with the requirement for the creation of new nodes. Node learning happens in this layer when the network accurately predicts the class label of the labeled input sequence from the EPMN during the learning process. If the class label is incorrect, a new node will be added. This additional criterion influences the rate at which the nodes update. Furthermore, due to the hierarchical learning of incoming data, each semantic node maintains knowledge through periods higher than episodic nodes. As a result, the SMN selects the winning node based on the BMN of the EPMN in the following manner:

$$b_s = \arg\min\left(T^{\text{SMN}}_j(t)\right) \tag{14}$$

$$T^{\text{SMN}}_j(t) = \exp\left[-\frac{\|x(t) - w_j\|^2}{2\sigma_i^2}\right]\exp\left[\frac{T^{\text{SMN}}_j(t-1) - 1}{\sigma^2}\right] \tag{15}$$

The selected node is either assumed to be the correct semantic node for the given sequence of episodic inputs, or it is more dominant than other semantic nodes, or both. The SMN receives input data from the EPMN, i.e., the EPMN's BMNs with regard to $x(t)$. The BMNs in the network are calculated with the Eqs. (14) and (15). Because the input is derived from bottom-up neural episodic weights, $x(t)$ is substituted by w^{em}_b for node learning.

Thus, a new semantic node is created only if the BMN b fails to satisfy three criteria: 1) $a^{\text{sm}}_b(t) < \rho_a$; 2) $r^{\text{sm}}_b < \rho_r$; and 3) BMN's label ζ^{sm}_b is not the same as the data input's label ζ (Eq. (21)). It should be noted that if the data input is not labeled, this label matching requirement in the semantic memory layer is ignored. If the winner of the semantic node b predicts the label ζ_b that is the same as the class label ζ of the input $x(t)$, the node learning process is started by the extra learning factor $\psi = 0.001$. As a result, Eq. (9) will become:

$$w_{j(\text{new})}^{\text{SMN}} = \psi \cdot \gamma_j \cdot r_j \cdot \left(w_b^{\text{EPMN}} - w_{j(\text{old})}^{\text{SMN}} \right) \tag{16}$$

The SMN learns to create more compact representations of the input labels. Data labels govern the network's stability and plasticity, with new semantic nodes addition only when the network is unable to estimate the correct data input class label.

3.4 Episodic procedural memory self-replay

To generate meaningful sequential data for memory playback, we exploit the spatiotemporal connections of nodes in the EPMN. When there is no input feed into the network, the EPMN uses its nodes as input for learning (self-replay). For example, if the winning episodic node b is activated by input data, the next temporal node can be selected by choosing the node with the largest activation value of P. For each node j, a set of nodes playback with length $K^{\text{EPMN}} + 1$ is calculated as follows:

$$U_j = \left\langle w_{u(0)}^{\text{EPMN}}, w_{u(1)}^{\text{EPMN}}, \cdots, w_{u\left(K^{\text{EPMN}}\right)}^{\text{EPMN}}, \right\rangle \tag{17}$$

$$u(i) = \arg\max P_{(j,u(i-1))} \tag{18}$$

where K^{EPMN} is the number of temporal nodes, $P(i,j)$ is the episodic temporal connection matrix, and $u(0) = j$. The temporal connection of episodic nodes stored in the network is capable of autonomously generating a series of events and replaying to the network without retaining the relations of previously received training data.

3.5 Data associative system

During the training phase, each node can be assigned a class label of l based on the input data. The L class label yields the l label. The frequency of each individual label in the network is stored in the $V(j,l)$ associative matrix for this labeling approach. This implies that each node j has a distribution counter that holds the frequency of a certain sample label. When a new node N is created and the label ζ associated with the input data $x(t)$ is specified, the matrix V is enlarged by one row and initialized with $V(N,\zeta) = 1$ and $V(N,l) = 0$. When an existing BMN b is chosen for updating, the V matrix is updated in the following manner:

$$V(b,\zeta)_{(\text{new})} = V(b,\zeta)_{(\text{old})} + \varphi^{+} \tag{19}$$

$$V(b,l)_{(\text{new})} = V(b,l)_{(\text{old})} + \varphi^{-} \tag{20}$$

Notice that φ^{+} must always less than φ^{-} and the label ζ is within the L class label. If the data label ζ does not exist in L, a new column in V is added and set to $V(b,\zeta) = 1$ and $V(b,l) = 0$. The matrix V will not be updated if there is no label associated with the given input gesture. The winning label ζ_j for a node j is calculated as follows:

$$\zeta_j = \text{label}(j) \equiv \arg\max V(j,l), \tag{21}$$

where l is label in class label L. The advantage of this labeling approach [26] is that no number of class labels must be specified in advance. Because the number of class label is uncertain, this is crucial when dealing with continuous learning in real world application.

4. Experimental setup and results

We first validate the proposed method using the COLD benchmark dataset [27, 28]. The COLD dataset is a large-scale, customizable testing environment for generally validating vision-based localization algorithms intended to perform on mobile platforms in realistic environments. A mobile robot gathers the dataset in three separate locations with different environmental conditions such as weather conditions, day or night time. It contains various formats, including RGB images, videos, and laser scans. RGB images and videos are gathered using a standard onboard camera and an omnidirectional camera. Instead of learning the image pixels, we use fixed random weights of Convolutional Neural Network (CNN) [29] for extracting visual features that sufficiently express the environment states. A simple CNN with fixed random weights, for example, can extract visual information with high classification accuracy in image classification tasks [30]. In this work, the extracted features from fixed random weights CNN and the robot's odometry data will be inputs to the EPMN, and the output of the EPMN will be the input of the SMN. Each data is fed into the memory networks sequentially without repetition for topological map building. Unlike batch learning, feeding the data sequentially to the memory networks fulfill the continuous learning criteria where data is only seen once. This criterion is crucial for robot navigation as the robot often traverses the environment continuously from one place to another. The hyperparameters for training the I-RKM in both memory networks are tabulated in **Table 2**.

Several metrics have been developed to assess the quality of a topological memory network. The total quantization error (TQE) is a popular metric, which quantifies the average distance between each data vector and its BMN. The BMU is the winning node in our case since it has the most significant match value and fulfills the vigilance parameter. The TQE measures the fitness of the generated topological map to the

Hyperparameter	Value
α^1, α^2	0.5
ρ_a	0.75
ρ_r	0.1
τ_j	0.5
λ	1.05
σ_i^2, σ^2	1.1, 1.4
γ_b	0.2
γ_n	0.001
r_e	0.001

Table 2.
Training parameters for the I-RKM of the memory networks.

robot's actual navigation route. As a result, the ideal topological map is expected to have the lowest TQE. The lower the TQE, the smaller the average distance between the BMNs and the robot's actual trajectory, indicating that the topological map is closer to the original route.

Furthermore, we evaluate the feasibility of the generated topological map using node localization accuracy. The pre-processed image dataset is transmitted to the I-RKM of both memory networks for each iteration to determine the BMN. The Euclidean distance between the BMN's encoded position and the robot's position from the dataset is used to compute the localization accuracy. Localization is accomplished if the Euclidean distance is smaller than a predefined value (0.1 m in these experiments). Because the purpose of SMN is to encode location label information, the localization accuracy is computed differently. Localization in SMN is fulfilled if the BMN's encoded location label is the same as the label from the dataset, similar to the standard classification accuracy.

4.1 Benchmark dataset results

The odometry and pre-processed image datasets were utilized as input to the I-RKM in the benchmark dataset experiment. To accomplish self-memory replay in EPMN, we continually feed the data in a mini-batch fashion (10 data per mini-batch).

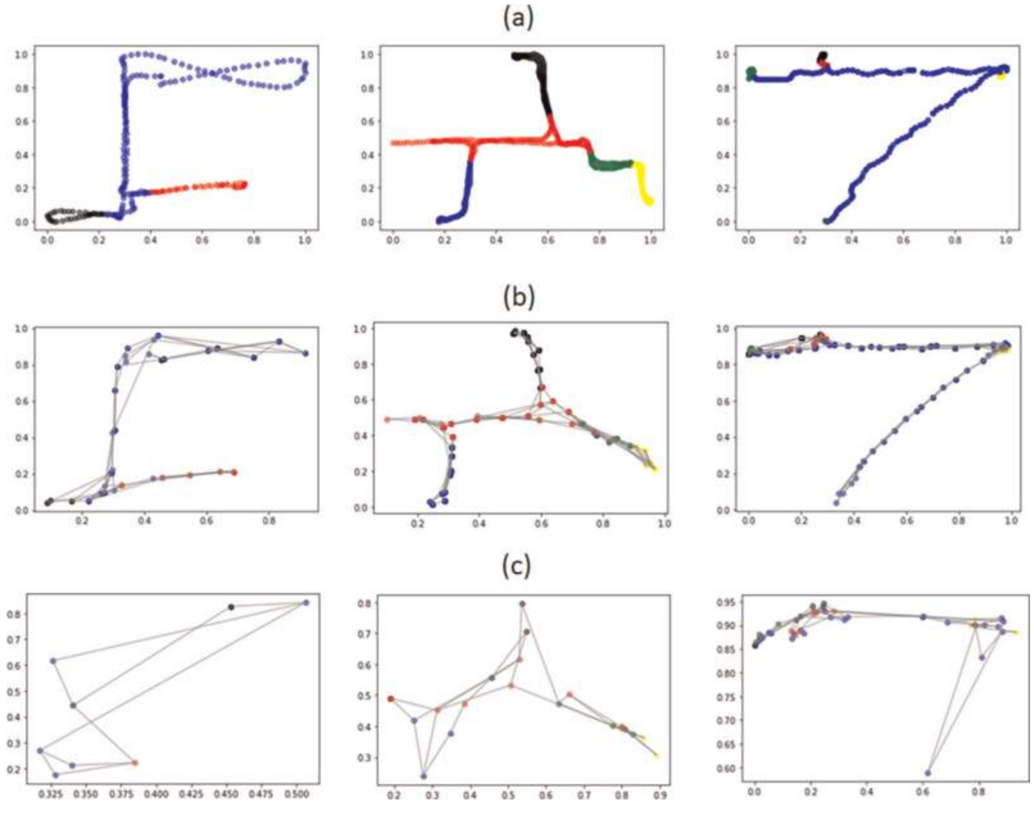

Figure 2.
Row (a) shows the robot's real path for collecting the COLD dataset: Saarbruken, Freiburg and Ljubljana (from left to right). Rows (b) and (c) illustrate the topological map of the episodic-procedural memory network and the semantic memory network, respectively. (a) robot navigation path, (b) episodic procedural memory network, (c) semantic memory network.

Then, after each mini-batch, memory self-replay was triggered. The topological map is made up of a series of nodes and edges. Different colored circles represent nodes, and each one holds the robot's coordinates (x, y), a place label, and a feature vector representing the surroundings. Links are black lines that link all nodes in the map to indicate node relationships. **Figure 2** shows the exact path taken by the robot in three different buildings with different environmental conditions and the topological maps generated by the proposed method. **Table 3** shows the TQE and localization accuracy of the topological map for each dataset. TQE and localization accuracy was found to be relatively constant across datasets. As a result, memory network learning is consistent across buildings with varying environmental conditions.

4.2 Physical robot experiment results

We validated our suggested technique further utilizing a mobile robot attached to an iPhone for image data acquisition and an Intel i5 CPU NUC PC for processing data and controlling the robot as shown in **Figure 3(a)**. The robot can traverse the surroundings autonomously, avoid obstacles, and follow walls. The robot's movement speed ranges from 0.05 to 0.5 m/s. The EPMN receives data from the iPhone and odometry to produce a topological map, whereas the SMN accepts EPMN output as input.

The experiments were carried out on the 7th floor of a university hallway, study area, and rest space that connected with one other, as shown in **Figure 3(b)**. The

Dataset	TQE (EPMN)	TQE (SMN)	Accuracy (EPMN) (%)	Accuracy (SMN) (%)
Freiburg cloudy 1	0.0283	0.2048	94.0	94.4
Freiburg cloudy 2	0.0322	0.2793	94.0	96.4
Freiburg cloudy 3	0.0204	0.2112	91.4	94.3
Freiburg sunny 1	0.0093	0.2122	94.0	95.0
Freiburg sunny 2	0.1179	0.2495	95.3	96.0
Freiburg sunny 3	0.0228	0.1466	91.5	92.0
Ljubljana cloudy 1	0.0143	0.2613	91.8	93.0
Ljubljana cloudy 2	0.0898	0.2743	84.9	85.0
Ljubljana cloudy 3	0.0047	0.3568	91.6	85.7
Ljubljana sunny 1	0.0118	0.2231	94.3	95.1
Ljubljana sunny 2	0.0480	0.2273	93.3	94.0
Ljubljana sunny 3	0.0798	0.3939	90.2	91.5
Saarbruken cloudy 1	0.0661	0.1544	91.5	92.0
Saarbruken cloudy 2	0.0020	0.1700	82.1	83.0
Saarbruken cloudy 3	0.0990	0.1410	91.7	93.1
Saarbruken night 1	0.0063	0.1292	92.1	93.6
Saarbruken night 2	0.0016	0.1678	89.5	90.2
Saarbruken night 3	0.1075	0.1899	85.0	86.9

Table 3.
The TQE and localization accuracy of the topological map that generated by the memory networks using COLD datasets.

purpose of experimenting with such environmental settings is to confirm that our proposed technique can work in a natural environment with moderately varying environmental factors. We instructed the robot to explore the experimental site, beginning in the study area and traveling to the rest area through the hallway, then returning to the start point. I-RKM continually learns from incoming sensory data in both memory networks and builds the topological map. After the first traverse, self-memory replay is triggered before the next traverse begins. The robot explored the surroundings with various movement behaviors depending on the location. For example, in the study area, the robot is set to obstacle avoidance mode since the environment is crowded with moving people and objects. Because the hallway is a straight path, the movement behavior is altered to the wall following and fast-speed mode when the robot enters the hallway. We repeated the experiment ten times. The metrics evaluation is identical to the benchmark dataset experiments (**Figure 4**). **Figure 5(a)** and **(b)** show the TQE and localization accuracy of the memory networks respectively.

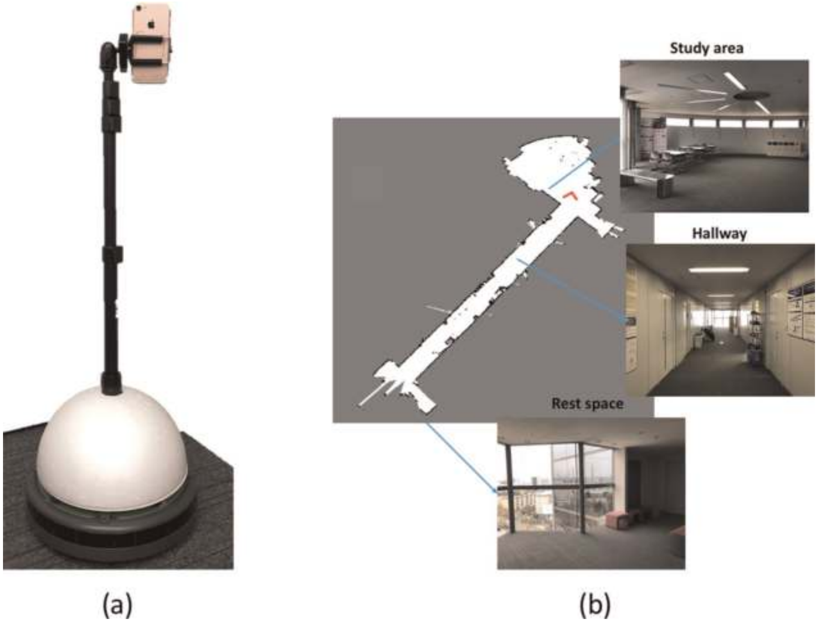

Figure 3.
(a) Physical robot equipped with an iPhone. (b) The experimental environment.

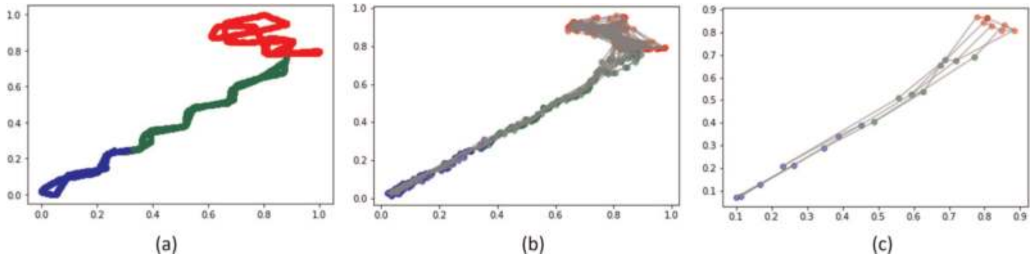

Figure 4.
(a) Robot navigation path; (b) topological map generated by the EPMN; (c) topological map generated by the SMN.

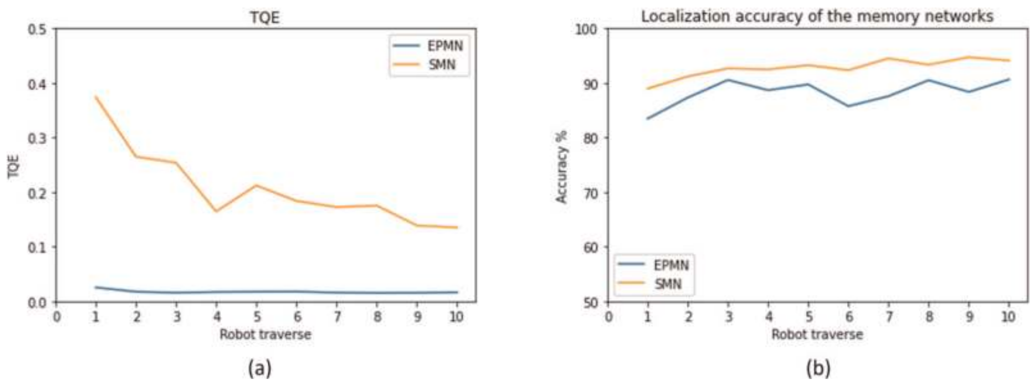

Figure 5.
(a) TQE of the topological map generated by the memory networks; (b) localization accuracy of the memory networks.

5. Discussion

We have shown that the memory networks can generate topological maps with benchmark datasets and physical robot experiments. Topological maps are built up from nodes that encode specific sensory information, providing flexibility and maintainability for robot navigation. New nodes are constantly added to the memory networks during environment learning, or existing nodes are updated. Edges link new nodes to existing nodes and can be used to guide navigation activities. Each node represents a region of the world, and it will be selected for learning if it corresponds to the robot's current sensory data. This property demonstrates that I-RKM retains previously learned knowledge and creates a topological map based on the robot's traverse path. According to the experiment results, all of the topological maps generated by I-RKM are almost identical to the actual robot path.

Because of the nature of memory network learning, the EPMN generates more nodes than the SMN. Because the SMN will use the EPMN output to generate the topological map, the SMN will learn the more sparse category representation. EPMN's topological map can be utilized for robot localization and navigation. The topological nodes connection allows the robot to navigate from one location to another. The topological map in SMN is sparser than in EPMN, and the TQE is higher than in the EPMN. However, the topological map of the SMN can be utilized for place classification tasks.

The proposed memory network training takes odometry data into account and visual measures. As a result, memory networks can distinguish areas with relatively similar visual sensory input, overcoming the difficulties of online detection and recognition of topological nodes. According to the node matching and localization findings, the robot failed to locate itself during navigation on several occasions because of a sudden change in the environment, resulting in no topological nodes matching with these sensor data. This issue can be solved by adjusting the vigilance parameter. The higher the value of the vigilance parameter, the more sensitive the memory networks are to changing environmental conditions and vice versa.

6. Conclusion

We presented Incremental-Recurrent Kernel Machines that mimic human episodic-procedural semantic memory and can progressively learn the spatiotemporal connection of sensory input from camera and odometry to build a topological map. I-RKM in both memory networks autonomously updates the topological map by expanding or shrinking its episodic memory structure. Furthermore, I-RKM consolidates the spatial map through self-episodic memory replay, eliminating the requirement for external sensory inputs. I-RKM has been validated through benchmark datasets and physical robot implementation. In the future, we will combine I-RKM with a path planning algorithm to use the topological map's structure for goal-directed navigation. In addition, we plan to leverage the edges connection between nodes by encoding traverse information on the edges. The robot can navigate from one place to another autonomously that solely depends on memory with little or no human intervention. Finally, we will improve and test I-RKM's performance in more challenging and larger environments.

Acknowledgements

This work was partially supported by JST [Moonshot R&D] [Grant Number JPMJMS2034].

Author details

Wei Hong Chin[1]*, Naoyuki Kubota[1] and Chu Kiong Loo[2]

1 Tokyo Metropolitan University, Tokyo, Japan

2 University of Malaya, Kuala Lumpur, Malaysia

*Address all correspondence to: weihong@tmu.ac.jp

References

[1] O'Keefe J, Nadel L. The hippocampus as a cognitive map. Behavioral and Brain Sciences. 1979;**2**(4):487-494. DOI: 10.1017/S0140525X00063949

[2] Brooks R. A robust layered control system for a mobile robot. IEEE Journal on Robotics and Automation. 1986;**2**(1): 14-21. DOI: 10.1109/JRA.1986.1087032

[3] Thrun S. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence. 1998;**99**(1):21-71. DOI: 10.1016/S0004-3702(97)00078-7

[4] Henrik K, Cyrill S. Information-theoretic compression of pose graphs for laser-based SLAM. The International Journal of Robotics Research. 2012;**31**(11): 1219-1230. DOI: 10.1177/02783 64912455072

[5] Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics. 2016;**32**(6): 1309-1332. DOI: 10.1109/TRO.2016. 2624754

[6] Chin WH, Loo CK, Toda Y, Kubota N. An Odometry-free approach for simultaneous localization and online hybrid map building. Frontiers in Robotics and AI. 2016;**3**:68-77. DOI: 10.3389/frobt.2016.00068

[7] Zeng T, Si B. A brain-inspired compact cognitive mapping system. Cognitive Neurodynamics. 2021;**15**:91-101. DOI: 10.1007/s11571-020-09621-6

[8] Buzsáki G, Moser EI. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. Nature Neuroscience. 2013;**16**(2):130-138. DOI: 10.1038/nn.3304

[9] Lu F, Milios E. Globally consistent range scan alignment for environment mapping. Autonomous Robots. 1997;**4**: 333-349. DOI: 10.1023/A:1008854305733

[10] Kümmerle R, Grisetti G, Strasdat H, Konolige K, Burgard W. G2o: A general framework for graph optimization. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2011); 9–13 May 2011; Shanghai, China. IEEE; 2011. pp. 3607-3613

[11] Labbé M, Michaud F. RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. Journal of Field Robotics. 2019;**35**:416-446. DOI: 10.1002/ rob.21831

[12] Ila V, Porta JM, Andrade-Cetto J. Information-based compact pose SLAM. IEEE Transactions on Robotics. 2010; **26**(1):78-93. DOI: 10.1109/TRO.2009. 2034435

[13] Carlevaris-Bianco N, Eustice RM. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2013); 6–10 May 2013; Karlsruhe, Germany. IEEE; 2013. pp. 5748-5755

[14] Mazuran M, Burgard W, Tipaldi GD. Nonlinear factor recovery for long-term SLAM. The International Journal of Robotics Research. 2016;**35**(1–3):50-72. DOI: 10.1177/0278364915581629

[15] Johannsson H, Kaess M, Fallon M, Leonard JJ. Temporally scalable visual SLAM using a reduced pose graph. In: Proceedings of the IEEE International Conference on Robotics and Automation

(ICRA 2013); 6–10 May 2013; Karlsruhe, Germany. IEEE; 2013. pp. 54-61

[16] David B, Scott H, Janet W, Gordon W, Peter C, Milford M. OpenRatSLAM: An open source brain-based SLAM system. Autonomous Robots. 2013; **34**(3):149-176. DOI: 10.1007/s10514-012-9317-9

[17] Milford M, Jacobson A, Chen Z, Wyeth G. RatSLAM: Using Models of Rodent Hippocampus for Robot Navigation and beyond. Robotics Research: The 16th International Symposium ISRR2016, Springer Tracts in Advanced Robotics. pp. 467-485. DOI: 10.1007/978-3-319-28872-727

[18] Erdem UM, Hasselmo ME. A biologically inspired hierarchical goal directed navigation model. Journal of Physiology Paris. 2014;**108**(1):28-37. DOI: 10.1016/j.jphysparis.2013.07.002

[19] Endo Y. Anticipatory robot control for a partially observable environment using episodic memories. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008); 19–23 May 2008; Pasadena, CA, USA. IEEE; 2008. pp. 2852-2859

[20] Steckel J, Peremans H. BatSLAM: Simultaneous localization and mapping using biomimetic sonar. PLoS One. 2013; **8**(1):e54076. DOI: 10.1371/journal.pone.0054076

[21] Tang H, Yan R, Tan KC. Cognitive navigation by neuro-inspired localization, mapping, and episodic memory. IEEE Transactions on Cognitive and Developmental Systems. 2018;**10**(3):751-761. DOI: 10.1109/TCDS.2017.2776965

[22] Hermans M, Schrauwen B. Recurrent kernel machines: Computing with infinite Echo state networks. Neural Computation. 2012;**24**(1):104-133. DOI: 10.1162/NECOa00200

[23] Soh H, Demiris Y. Spatio-temporal learning with the online finite and infinite Echo-state Gaussian processes. IEEE Transactions on Neural Networks and Learning Systems. 2015;**26**(3): 522-536. DOI: 10.1109/TNNLS.2014.2316291

[24] Jaeger H. The "Echo State" Approach to Analysing and Training Recurrent Neural Networks-with an Erratum Note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report; 2010. p. 148

[25] Liew WS, Loo CK, Gryshchuk V, Weber C, Wermter S. Effect of pruning on catastrophic forgetting in growing dual memory networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2019); 14–19 July 2019; Budapest, Hungary. IEEE; 2019. pp. 1-8

[26] Parisi GI, Tani J, Weber C, Wermter S. Lifelong learning of human actions with deep neural network self-organization. Neural Networks. 2017;**96**: 137-149. DOI: 10.1016/j.neunet.2017.09.001

[27] Wang X, Zhao Y, Pourpanah F. Recent advances in deep learning. International Journal of Machine Learning and Cybernetics. 2020;**1**:747-750. DOI: 10.1007/s13042-020-01096-5

[28] Pronobis A, Caputo B. COLD: The CoSy localization database. The International Journal of Robotics Research. 2009;**28**(5):588-594. DOI: 10.1177/0278364909103912

[29] Saxe AM, Koh PW, Chen Z, Bhand M, Suresh B, Ng AY. On random weights

and unsupervised feature learning. In: Proceedings of the International Conference on International Conference on Machine Learning (ICML 2011); June 28–2 July 2011; Madison, WI, USA. Omnipress; 2011. pp. 1089-1096

[30] Tong Z, Tanaka G. Reservoir computing with untrained convolutional neural networks for image recognition. In: Proceedings of the International Conference on Pattern Recognition (ICPR 2018); 20–24 August 2018; Beijing, China. IEEE; 2018. pp. 1289-1294