Unsupervised temporospatial neural architecture for sensorimotor map learning.

EZENKWU, C.P. and STARKEY, A.

2021

Unsupervised Temporospatial Neural Architecture for Sensorimotor Map Learning

Chinedu Pascal Ezenkwu*† and Andrew Starkey*
*School of Engineering, University of Aberdeen, Aberdeen, UK
{chineduezenkwu, a.starkey}@abdn.ac.uk
†Electrical, Electronic and Computer Engineering, University of Uyo, Nigeria
{chineduezenkwu}@uniuyo.edu.ng

Abstract—The ability to learn the sensorimotor maps of unknown environments without supervision is a vital capability of any autonomous agent, be it biological or artificial. An accurate sensorimotor map should be able to encode the agent's world and equip it with the capability to anticipate or predict the results of its actions. However, to design a robust autonomous learning technique for an unknown, dynamic, partially observable or noisy environment remains a daunting task. This paper proposes a Temporospatial Merge Grow When Required (TMGWR) network for continuous self-organisation of an agent's sensorimotor awareness in noisy environments. TMGWR is an adaptive neural algorithm that learns the sensorimotor map of an agent's world using a time series self-organising strategy and the Grow When Required (GWR) algorithm. The algorithm is compared with GNG, GWR and TGNG in terms of their disambiguation performance, sensorial representation accuracy and sensorimotor-link error, a new metric that is developed in this paper to evaluate how well a sensorimotor map represents causality in the agent's world. The outcomes of the experiments show that TMGWR is more efficient and suitable for sensorimotor map learning in noisy environments than the competing algorithms.

Index Terms—autonomous agent, sensorimotor awareness, dynamic environment, unsupervised learning, causality

I. INTRODUCTION

Learning in human infants, especially at the sensorimotor stage as presented in Piaget's constructivist epistemology [1], has become a source of inspiration in autonomous systems research. Through an unsupervised exploration of its environment, a human infant improves its innate primitive model of the world to a sophisticated sensorimotor representation of its environment, which can be later applied in non-trivial and complex skill-demanding tasks. This idea has been built upon in the context of machine autonomy in recent times. An autonomous agent must first explore its environment and build a sensorimotor model of the environment to enable it to predict the effects of its actions as well as to plan in order to succeed in this environment. The aforesaid has revived interests in unsupervised learning as a possible approach towards autonomous learning [2],[3]. Unsupervised learning has been useful in data compression and dimensionality reduction [4] due to its ability to extract principal features from unlabelled multidimensional sensory observations. Leveraging on this potential, unsupervised techniques have been used to confine continuous space observations and/or actions to a finite space [5],[6],[7],[8], hence simplifying an agent's world for real-time exploitation. Yet, traditional unsupervised learning methods are unable to provide an agent with the right behaviour information since the sensory information is not mapped to specific outputs or actions as the case may be [9], [10]. Due to this limitation, these methods are often combined, to act as vector quantisation techniques, with supervised learning [7] or reinforcement learning algorithms [11].

Self-organising Map (SOM) [12] is a very influential unsupervised learning technique. It is a neurophysiologically-motivated neural architecture that learns the topographic map that best represents a set of observations. A unique feature of SOM is the topological ordering of the neurons such that the spatial location of each neuron in the map represents a specific cluster of the input patterns and its neighbourhoods. The performance of SOM largely depends on its dimension, always predefined to suit the task at hand. This is a limitation in cases where there is no prior knowledge of the change in the dynamic distribution of the sensory inputs. Researchers have developed several variants of SOM that allow the variation of the dimension of the network based on certain criteria. Popular examples of this are Growing Neural Gas (GNG) [13], Grow When Required (GWR) network [14] and Grow Cell Structure (GCS) [15]. The advantage of these kinds of network over the standard SOM is that the correct number of nodes is not to be decided *a priori*; as such, they are suitable in situations where the distribution of observations is not known. The key difference amongst these algorithms is in their mechanisms for deciding when to add a new node. All the same, each of them create edges among nodes based on their sensory proximity; that is, if the nodes are close neighbours on the map. The edges created based on sensorial proximities do not enable the map to represent causality. Owing to this, Butz, Reif and Herbort have investigated the possibility of linking nodes on the map based on their sensorimotor proximities [16]. This implies that a directed edge is connected from one node to another if there is an action that can cause the agent's experience to change from the former to the later. With this, the agent is able to develop a model for causal reasoning and planning in the environment.

The technique proposed in this paper is among a few growing self-organising neural architectures that learn a topological map of neurons with linked edges representing possible transitions between neurons. Toussaint [17], [18] called such map a

sensorimotor map because it couples sensory prototypes with motor signals. Toussaint employed GNG in the design of his Connectionist World Model (CWM) [18]. The CWM is able to learn a growing neural model for different observations in the agent's environment as well as the implications of its actions by encoding motor knowledge in the lateral connectivity between nodes. In GNG, the base algorithm for CWM, edges are created between nodes on the basis of their sensory proximities. The implication of using the original GNG in developing the sensorimotor map of an agent is that it can create a lot of unnecessary nodes and links, which may not properly represent the dimensionality of the agent's environment. Butz, Reif and Herbort proposed the Time Growing Neural Gas (TGNG) [16]. TGNG, unlike in GNG, provides directed links between nodes based on their sensorimotor proximities. However, both GNG and TGNG are only suitable for Markovian environments and their performances are very much affected if the environment is noisy or partially observable. This informs our proposal of the Temporospatial Merge Grow When Required (TMGWR) network. TMGWR network is an adaptive neural architecture that learns the topological map and the sensorimotor links [16] between neurons using a time series self-organising strategy [19]. TMGWR network connects nodes based on their sensorimotor proximities, as is in TGNG, such that these edges can encode the transition possibilities as well as the motor signals that can cause transitions between nodes. Unlike in GNG and TGNG, the TMGWR network takes advantage of the GWR network to enable it to expand the size of the network following the dynamic distribution of the input vectors. The decision to add a node is made based on how well the map is able to represent each instance in the environment.

The remainder of the paper is organised as follows — section II presents a review of related work in sensorimotor map learning. The metrics for evaluating and comparing self-organising methods are discussed in section III. Section IV presents a detailed explanation of TMGWR network. Experiments conducted for this paper as well as the results are described in section V while section VI concludes the paper.

II. REVIEW OF RELATED WORKS

Chaput [20] proposed a sensorimotor schema mechanism called constructivist learning architecture (CLA). CLA is a neural network implementation of the Drescher's schema mechanism [21]. Schema, according to Piaget, is a framework for organising and representing some features of the world [1]. Drescher defined a schema as a triple comprising an agent's perception, action and expectations. CLA is realised using hierarchies of SOMs [12] to help a robot to develop schemas starting from sets of primitive items and actions to complex sensorimotor schemas and goal-directed composite actions. The author implemented a technique called harvesting, a very important step in CLA algorithm, for selecting relevant items for each schema before passing them on to the next layer of SOM. The implementation of harvesting in CLA limits it to a binary sensory environment and therefore CLA has a scale-up issue and cannot be implemented as a real-world autonomous agent.

Toussaint [17],[18] demonstrates that a self-organising neural network is able to couple sensory and motor signals together. Toussaint used GNG to avoid the problem associated with predefined fixed size SOM. The neural architecture is named sensorimotor map because it integrates sensory and motor signal in a map. Nodes in the sensorimotor map represent the sensory inputs while the lateral connections among the nodes encode motor signals that have regularly resulted in the same transitions. The lateral connections strengths are modulated by the motor activations in a Hebbian-like fashion. Unlike GNG used in Toussaint's approach, Butz et al [16] proposed the TGNG, a variant of GNG that connects nodes on the cognitive map based on their sensorimotor proximities instead of sensory proximities. This implies that two nodes are connected if an action can cause a change from one node to another.

Mohan, Morasso and Metta [22] built on Toussaint sensorimotor map [18] in developing a mechanism that allows a GNOSYS robot to learn 'when' to optimize 'what constraint' while realising a spatial goal and to be able to push a ball intelligently to the corners of a table while avoiding traps in arbitrary positions on the table. A key contribution of the paper is in the design of a bifurcation parameter, which measures the anticipatory ability of the model. Based on the value of the bifurcation parameter the agent decides whether to explore or exploit its environment. Furthermore, the paper proposes three simple rules for distribution of received end reward among contributing neurons in a high dimensional sensorimotor map.

Kubisch [23] also proposed a GNG-based algorithm, the Growing Multi-Expert Structure (GMES). Unlike in the traditional GNG, in GMES a winner node is the node with the minimum prediction error. In addition, a new node is added in GMES when the learning capacity of the winning node is exhausted unlike in GNG in which a new node is added at a constant number of intervals. However, in the paper GMES was not used to learn the sensorimotor map of the environment. The algorithm was employed for unsupervised learning of continuous sensory state spaces in open-ended reinforcement learning tasks. The experiment conducted in the paper is a physically simulated pendulum.

Mici, Parisi and Wermter [24] used hierarchy of GWR networks for sensorimotor learning and prediction of visually demonstrated arm movements in order to reproduce them in synchrony with the human demonstrator. However, this was applied in an imitative learning scenario, which is in contrast to our ideal environment for a completely autonomous agent. An interesting finding in the paper is that a learning architecture based on GWR network is able to learn in a noisy environment with faulty visual sensor when provided with incomplete data sequence.

Although these works have yielded good performances in the diverse experiments conducted in the papers, yet, they are not suitable for applications in high noise or ambiguous sensor space. Each of the preceding algorithms assumes that

the environment is Markovian; so the current state is enough for the agent to make decision about the next state. This is not always the case in some real-world applications especially when the agent's world is partially observable or has a lot of indistinguishable features. In such a case, some previous history is required to enable the agent to disambiguate similar stimuli since a sequence of the current and one or more of its preceding states are used to classify each sensory input. Variants of SOM have been developed to handle temporal sequence processing [25]. Some of the most used temporal processing unsupervised methods are Recursive SOM (Rec-SOM) [26], Temporal Kohonen Map (TKM) [27], Recurrent Self-Organising Map (RSOM) [27], SOM for structured data (SOM-SD) [28], Merge Grow Neural Gas [29], Merge SOM (MSOM) and Merge Neural Gas (MNG) [19]. In contrast to the basic SOM, these methods use a sequence of previous best matching units in deciding what the current best matching unit will be. This makes them suitable for unsupervised clustering of time series data such as speech signals [30], patterns from stock time-series [31], human behavioural patterns [32] and as well as robot's experiences [33][34]. Even though these temporal self-organising neural networks have proven suitable for time-series and noisy datasets, they suffer some of the limitations of their base algorithms. For example, RecSOM, TKM, RSOM, SOM-SD, MSOM and MNG are based on the standard SOM, as such they require that the number of nodes are correctly chosen beforehand, causing them to be less suitable for unknown environments. MGNG is based on GNG, hence its neurons grow at constant time rate, independent of the distribution of the instances in the dataset, making it likely for MGNG to underfit or overfit the environment depending on the maximum number of nodes specified during design. Furthermore, none of these techniques can represent causality since their nodes are linked by sensorial proximity and not sensorimotor proximity.

A new kind of temporal sequence unsupervised model, Topological Temporal Hebbian Self-Organizing Map (T2HSOM) has been applied as a gaze planner in word recognition [35]. T2HSOM is a modification of Temporal Hebbian Self-Organizing Map (THSOM) [25]. Unlike THSOM, which clusters input vectors based on their space similarity only, the T2HSOM tries to optimise topological clustering based on both space similarity and co-occurrence of input vectors in similar sequences. However, like the other temporal processing techniques, T2HSOM does not incorporate actions between observations. Graziano, Koutník and Schmidhuber [36] modified T2HSOM by incorporating a transition-map in the temporal activation component, hence making explicit the use of the action taken between observations. The algorithm is called Temporal Network for Transitions (TNT). Experiments show that TNT learns a good representation of different sizes of maze under conditions of high-noise and stochastic actions. However, THSOM, T2HSOM and TNT are built on SOM and thus require that the number of nodes is correctly selected to make them suitable for the environment of interest. Another limitation of

TNT is that its learning parameters decrease with the age of the network. This is a huge disadvantage for an autonomous learning agent, as it does not encourage continual and open-ended learning, which is a key attribute for autonomy.

Pierris and Dahl proposed the Hierarchical SOM-based Encoding (HSOME) [37],[38],[39]. HSOME builds on CLA [20] but unlike in CLA, there is no limit to the height of the hierarchy of SOMs. The algorithms has demonstrated the ability to encode and reproduce sequential data as well as identify hidden states. However, a major limitation of HSOME is that "it produces a lossless encoding instead of a principle component representation" [37], [39] of the observations in the agent's world. This requires that the map is large enough to fit the sequences of observations. Meanwhile, HSOME consists of SOMs, hence, the correct number of nodes is to be decided at the design time. This imposes limitation to the scalability of the method.

Recently, efforts have been made to develop a sensorimotor model of a partially observable environment using a recurrent deep neural network [9]. The authors presented an algorithm called a Recurrent Sensorimotor Encoder (Recurrent-SM-encoder) for compact representation of the agent world using sensorimotor prediction and memory. Results in the paper underscore the importance of memory in sensorimotor learning and prediction. Although proposed for their future work, the authors did not realise a sensorimotor map or a graph representation of the environment with nodes and edges representing observations and actions respectively. The preceding forms the thesis of our research.

We propose an unsupervised recurrent growing sensorimotor map learning technique for continuous self-organisation of an agent's sensorimotor awareness in partially observable environments. The algorithm is called Temporospatial Merge Grow When Required (TMGWR) Network. The key features of the algorithm are I. the nodes on the sensorimotor map are linked based on their sensorimotor proximity II. in similar way as Merge Neural Gas (MNG) [19] it employs the recursive temporal context approach as a way of keeping track of the sensorimotor history. III. it uses the GWR network architecture to enable the map to grow in respect to changes in the environment IV. all the learning parameters are kept constant so as to be able to handle learning in continuously changing environments. Section IV presents a detailed explanation of TMGWR.

III. METRICS FOR QUANTIFYING GOODNESS OF SENSORIMOTOR MAP

The base algorithm for sensorimotor map construction as presented in this paper is the self-organising neural network. Quantisation error is a basic metric for evaluating the goodness of SOM and its variants in representing a given set of data points. However, the quantisation error "can only be used to compare maps to each other" and not as a stand-alone assessment of quality because the reported value is usually in the scale of the input data [40]. Likewise, it does not seem plausible to compare the performance of non-recurrent

self-organising algorithms with the recurrent types using the quantisation error as a metric. This is because while the non-recurrent self-organising methods such as the SOM, GNG, TGNG and GWR optimise the quantisation error directly the time series techniques such as TMGWR, MSOM, MNG and so on have an additional component representing history in the cost function; as such, the non-recurrent techniques would have an unfair advantage over these recurrent techniques when they are compared using quantisation error. In this paper, the sensorial and the sensorimotor representation accuracies of the algorithms are evaluated using the purity and the sensorimotor-link error respectively.

A. Sensorial representation metric

Metrics such as the quantisation error, topographic error, purity and so on can be used in quantifying the goodness of a sensorimotor map in terms of how well it represents the spatial distribution of the sensory observations in the agent's environment. In this paper, purity is used for this purpose. Manning, Raghavan and Schutze defined purity as "a measure of the extent to which clusters contain a single state" [41]. In the context of this paper, purity measures the ability of each node in the sensorimotor map to represent a known state in the agent's world. This is given as equation 1:

$$purity = \frac{\sum_{n \in \mathcal{N}} max_{s \in \mathcal{S}} |n \cap s|}{M} X 100\% \quad (1)$$

$\mathcal{N}$ is the set of nodes that make up the sensorimotor map, $\mathcal{S}$ is the set of world states in the agent's environment, while M is the total number of observations made by the agent while learning in the environment.

B. Sensorimotor representation metric

To the best of our knowledge, there is no metric that is able to evaluate how well a sensorimotor map represents causality in the agent's world. We define a new metric, called sensorimotor-link error (SE), suitable for quantifying how well different maps represent causality. SE is the ratio of number of impossible transitions to the total number of transitions learnt by the Map. This is summarised in equation 2. E is a set of connections learnt by the map; $|E|$ is the number of transitions in E. $\mathcal{I}\{.\}$ is an indicator function that returns 1 when the argument evaluates to true, otherwise 0. $E^i \left[\vec{w}_{t-1}^{\{1\}}, \vec{w}_t^{\{1\}} \right]$ is the ith transition from node $\vec{w}_{t-1}^{\{1\}}$ to $\vec{w}_t^{\{1\}}$.

$$SE = \frac{\sum_{i=1}^{|E|} \mathcal{I}\{E^i[\vec{w}_{t-1}^{\{1\}}, \vec{w}_t^{\{1\}}] \notin \mathcal{H}\} + \tau}{|E| + \tau} \quad (2)$$

$\mathcal{H}$ is a set of all possible transitions while τ is a sufficiently small positive number. τ is useful for implementation as it prevents division by zero if no transition has been learnt. Throughout the paper τ is set to 0.0001.

IV. TEMPOROSPATIAL MERGE GROW WHEN REQUIRED NETWORK

TMGWR is a self-organising neural network that learns a set of spatial prototypes or neurons, $\mathcal{N}$ that best represent sequences of sensory observations. The neurons are connected by a set of edges E representing the change of state due to a given action or motor signal. Each neuron $n \in \mathcal{N}$ consists of a weight vector $\vec{w}_n$ for abstracting observation in the current time step and a context vector $\vec{c}_n$ representing all the past time steps of a sequence. Both $\vec{w}_n$ and $\vec{c}_n$ have the same dimension as the input space.

The sensorimotor map is initialised with two neurons. The agent acts randomly in the environment and generates a sequence of observations $\vec{x}_1, ..., \vec{x}_t$. At each time step t, the distance, $d_n(t)$ of each neuron n from the current observation $\vec{x}_t$ and the global context $\vec{C}_g$ [19] is computed as follows:

$$d_n(t) = \alpha \|\vec{x}_t - \vec{w}_n\|^2 + (1 - \alpha)\|\vec{C}_g - \vec{c}_n\|^2, \quad \forall n \in \mathcal{N} \quad (3)$$

The parameter $\alpha \in (0, 1)$ helps to decide the importance of current observation over history. $\vec{C}_g$ is a linear combination of the weight and context vectors of the winning node l in the previous time step $t - 1$.

$$\vec{C}_g = \beta \vec{w}_l + (1 - \beta)\vec{c}_l \quad (4)$$

where, the parameter $\beta \in (0, 1)$ controls the effect of the recent past over history.

If the similarity $\xi(t)$ of the weight of the winning node l at time t, and the current observation $\vec{x}_t$ is less than the activity threshold θ, then a new node r is created with the weight $\vec{w}_r$ initialised to $\vec{x}_t$ while the value of its context vector $\vec{c}_r$ is set to $\vec{C}_g$ calculated at the time t. The similarity function $\xi(t)$ is a Gaussian kernel.

$$\xi(t) = exp\left(-\frac{\|\vec{x}_t - \vec{w}_l(t)\|^2}{2\omega^2}\right) \quad (5)$$

where, ω determines the shape of the Gaussian.

Meanwhile, if $\xi(t) > \theta$ then the winning unit l is updated as follows:

$$\vec{w}_l(t) = \vec{w}_l(t) + \epsilon_b(\vec{x}_t - \vec{w}_l(t)) \quad (6)$$

$$\vec{c}_l(t) = \vec{c}_l(t) + \epsilon_b(\vec{C}_g(t) - \vec{c}_l(t)) \quad (7)$$

In order to maintain topological ordering of nodes on the map, each neuron i in the topological neighbourhood of l is updated as follows:

$$\vec{w}_i(t) = \vec{w}_i(t) + \Omega(l, i)\epsilon_n(\vec{x}_t - \vec{w}_i(t)) \quad (8)$$

$$\vec{c}_i(t) = \vec{c}_i(t) + \Omega(l, i)\epsilon_n(\vec{x}_t - \vec{c}_i(t)) \quad (9)$$

where, ϵ_b and ϵ_n are adaptation rates for the winning node and nodes in its topological neighbourhood respectively. $\Omega(l, i)$ is the neighbourhood function for computing the similarity of

each neuron i from the winning neuron in the topological space [36].

$$\Omega(l,i) = \begin{cases} exp\left(-\frac{\|\vec{x}_t - \vec{w}_i(t)\|^2}{2\omega^2}\right), & \|\vec{x}_t - \vec{w}_i(t)\| < \phi \\ 0, & otherwise \end{cases} \quad (10)$$

ϕ is the topological neighbourhood cut-off parameter. A directed edge is created from the immediate past winning node l_{t-1} to the current winning node l_t if such edge does not exist. The age of the new edge is initialised to 0. However, if the edge already exists, its age is reset to 0 while the ages of all edges that emanate from l_{t-1} are increased by 1. The graph is pruned by deleting old edges and removing nodes that do not have neighbours. The TMGWR algorithm is shown in Algorithm 1.

Algorithm 1 TMGWR

1: create a set $\mathcal{N}$ of two nodes
2: define an empty transition set E
3: set the weight vectors of the two nodes to the initial sensation $\vec{x}_{t-1}$ and the context vectors $\vec{c}_{t-1} = \vec{0}$
4: initialise the global temporal context $\vec{C}_g = \vec{0}$
5: **while** not terminated **do**
6:     select action a_t
7:     $\vec{x}_t = ACT(\vec{x}_{t-1}, a_t)$
8:     compute $d_n(t)$ $\forall n \in \mathcal{N}$ according to equation (3)
9:     identify the winning node $l = argmin_{n \in \mathcal{N}} d_n(t)$
10:     Compute $\xi(t)$ according to equation (5)
11:     **if** $\xi(t) <$ activity threshold, θ **then**
12:         add a new node r to $\mathcal{N}$
13:         set the weight vector $\vec{w}_r(t)$ to $\vec{x}_t$
14:         set the context vector $\vec{c}_r(t)$ to $\vec{C}_g$
15:         use r as the winning node i.e. set l to r
16:     **else**
17:         update $\vec{w}_l(t)$ and $\vec{c}_l(t)$ according to equations
18:           (6) and (7)
19:         update neighbours of l according to equations
20:           (9) and (10)
21:     **end if**
22:     update $\vec{C}_g$ according to equation (4)
23:     **if** the directed edge $(l_{t-1}, l_t) \notin E$ **then**
24:         create it and initialise its age to 0
25:     **else**
26:         reset its age to 0 and increase the age of every
27:           other edges starting from l_{t-1}
28:     **end if**
29:   remove old edges and lonely nodes
30: **end while**

V. EXPERIMENTS

A. Setup

Fig. 1 shows the environments used for the experiments in this paper. Both environments are two-dimensional mazes. The sensor values to the agent are 2D coordinates of locations within the maze. Solid areas represent walls, while empty

areas are passages in the mazes. There are four possible actions available to the agent: up, down, right and left. Each action moves the agent by 1 unit in the desired direction if there is no wall in the direction of the intended movement, otherwise the agent's state remains unchanged. Exploration of the environment is performed with a random walk that changes its direction with a probability of 10% if no wall is in the agent's current direction, otherwise, the agent chooses any of the other three directions with 100% probability. The length of the random walk in all cases is 20,000 time steps during which the agent learns the topology of the maze online. Experiment V-B1 uses the environment of Fig. 1a while experiments V-B2 and V-B3 are run in the environment of Fig. 1b. In each of the experiments, the performance of TMGWR is compared with those of GNG, TGNG and GWR both in noiseless and noisy scenarios. In noisy settings, the agent's actions in the environment result in noisy observations with 20% probability. The observations are corrupted with Gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma^2)$). The noise level ranges from $\sigma^2 = 0$ to $\sigma^2 = \frac{4}{3}$ at an interval of $\frac{1}{6}$. The hyperparameters used throughout the experiments are presented in Table I. The newly introduced parameters in the table, such as the ones specific to TGNG and GNG, are defined as follows – λ is the constant rate of adding nodes; μ is the error weighting parameter; δ is the error reducing factor for the two nodes with highest amount of errors; ρ is the error reducing factor for all nodes while T_{max} is the maximum age of each edge in the map . For each algorithm, the hyperparameters were tuned by Bayesian optimisation of purity measure with maximum number of evaluations set to 150. The algorithms were optimised independently in the environment of Fig. 1b following the exploration policy described above. The maximum number of nodes, N_{max} any algorithm can grow up to is set to 300. This is necessary to ensure that none of the algorithms overfits the environment too much. Furthermore, the motivation part of TGNG is ignored throughout the experiment so that the same exploration policy is employed for all the algorithms in different scenarios. Ten trials of each experiment were run, with the boxplots of the algorithms' performances reported.
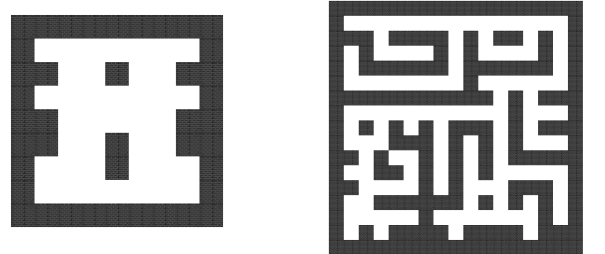
(a)   (b)

Fig. 1: Agent Environments. (a) 7 X 7 maze (b) 15 X 15 maze

TABLE I: Hyperparameters

Parameter	Value			
	GWR	*GNG*	*TGNG*	*TMGWR*
ϵ_b	0.5090	0.7923	0.8376	0.55194
ϵ_n	0.0944	0.1144	0	0.8999
λ		68		
α				0.9501
μ			0.6535	
δ		0.4832		
θ	0.6716		0.6001	0.4135
β				0.800
ρ		0.9185		
ω				1
T_{max}	25	21	51	17
ϕ				0.6002

B. Results

1) Sensorimotor Map Building: This experiment shows that an autonomous agent can use the TMGWR to learn the sensorimotor map of either noiseless or noisy environments with minimum number of nodes. The agent explores the environment of Fig. 1a by choosing actions for 20,000 time steps following the exploration policy that has been described in section V-A. Using the outcomes of these actions, the agent learns the sensorimotor map of the environment in the sensorimotor space. The map learnt with TMGWR is compared with those by GWR, GNG and TGNG in the same scenario. Fig. 2 shows the best maps obtained for each of the algorithms after five runs in each case. The experiments were run both in noiseless and noisy scenarios – at a noise level of $\sigma^2 = \frac{1}{6}$. GWR and GNG, nodes are linked by undirected edges indicating that the links are based on sensorial proximities and not sensorimotor proximities as is the case with TGNG and TMGWR. In the two scenarios, GWR shows a better spatial representation of the environment than the GNG but worse than TGNG and TMGWR. However, its sensorimotor representation is poor because nodes are linked based on their sensory proximity. GNG shows the worst representation of the environment in both scenarios. This is because it is least sensitive to the dynamic distribution of the sensory input when compared to other methods. Moreover, it has too many nodes causing it to overfit the environments. TGNG under-represented the environment in the noiseless scenario because it creates lesser number of nodes and sensorimotor links than it required to represent the environment properly. In the noisy case, TGNG add extra few nodes and it is able to show a better spatial representation of the environment than in the noiseless case. Meanwhile, TGNG established some incorrect sensorimotor links in both scenarios. TMGWR proved to represent the environment better than others in the two scenarios, then followed by TGNG. The sensorimotor maps due to TMGWR represent the environment with 100% accuracy both in the noiseless and noisy cases. Fig. 3 shows the number of nodes each of the algorithms use in building

(a) GWR ($\sigma^2 = 0$)
Number of nodes = 37

(b) GWR $\left(\sigma^2 = \frac{1}{6}\right)$
Number of nodes = 41

(c) GNG ($\sigma^2 = 0$)
Number of nodes = 54

(d) GNG $\left(\sigma^2 = \frac{1}{6}\right)$
Number of nodes = 216

(e) TGNG ($\sigma^2 = 0$)
Number of nodes = 37

(f) TGNG $\left(\sigma^2 = \frac{1}{6}\right)$
Number of nodes = 42

(g) TMGWR ($\sigma^2 = 0$)
Number of nodes = 40

(h) TMGWR $\left(\sigma^2 = \frac{1}{6}\right)$
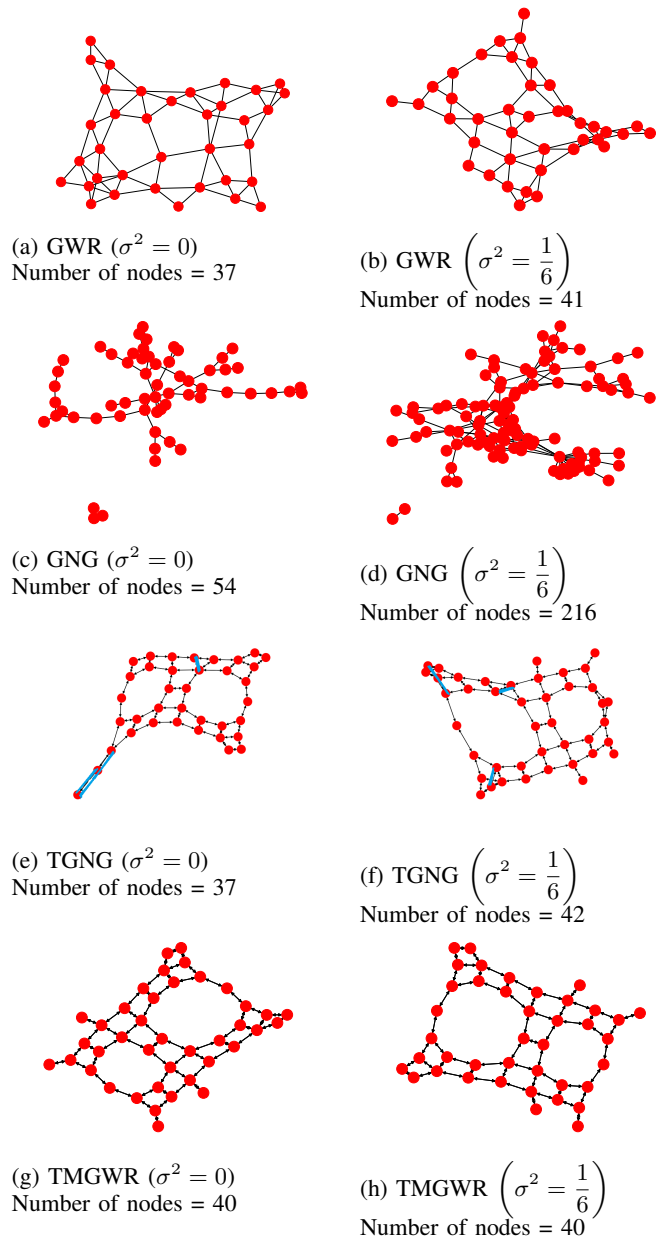Number of nodes = 40

Fig. 2: Sensorimotor maps for environment of Fig. 1a. Maps on the left side represent the performances of the algorithms in noiseless scenario while maps on the right side show the performances of the algorithms in noisy scenario. Blue connections in maps due to TGNG, show incorrect sensorimotor links

sensorimotor map at different noise levels. The result shows that algorithms that add nodes based on error such as GWR, TGNG and TMGWR, do better than ones that add nodes at constant time rate such as GNG. Moreover, TMGWR used the minimum number of nodes as the noise level increases. Unlike in GWR and TGNG, TMGWR uses history of observations and the observation at the current time step while building the sensorimotor map and this helps it to figure out the underlying

states of noisy observations and map them to the right nodes instead of creating an extra node for them due to noise.
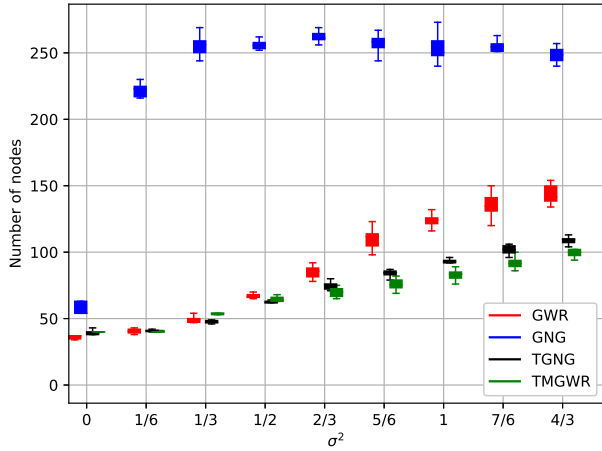

Fig. 3: Number of nodes created by each algorithm at different noise levels in the environment of Fig. 1a

Fig. 4: Percentage accuracies of the algorithms at different noise levels in the environment of Fig. 1b

2) Quantifying Goodness of the Map: The two metrics presented in section III were used to measure and compare the goodness of maps due to the algorithms. The performances of the algorithms at different noise levels are reported in Fig. 4 and Fig. 5. At each noise level, the agent builds a map, using each of the algorithms, while exploring the environment. After 20,000 time steps, the sensory and sensorimotor qualities of the map are evaluated. Fig. 4 presents the percentage accuracy in identifying correct nodes for the agent's observations while Fig. 5 shows the amount of error each algorithm made after creating edges between the nodes. From the result in Fig. 4, TGNG and TMGWR performed better than the other two algorithms with TMGWR leading TGNG. Meanwhile, in Fig. 5 TMGWR exhibits a better and more robust sensorimotor behaviour even at high noise scenarios. The result shows that the sensorimotor behaviour of TGNG is largely affected by noise. Despite its excellent performance in a noiseless scenario, TGNG performed worst than all the other algorithms as the noise level increases. It creates a number of incorrect paths or links at a high level of noise despite being able to abstract the sensory observations to a good extent. The better performance of TMGWR in this experiment can be attributed to its temporal nature which makes it suitable in non-Markovian environments.

3) Disambiguation of noisy observations: This experiment is to evaluate the ability of a pre-trained sensorimotor map in identifying the underlying-states from noisy observations due to newly generated random walks. The algorithms were first trained in the environment at $\sigma^2 = \frac{1}{3}$ noise level. The sensorimotor maps due to the algorithms are used after training to identify observations at different levels of Gaussian noise while the agent performs random walks. At each noise level, the purity of the map due to each of the algorithms
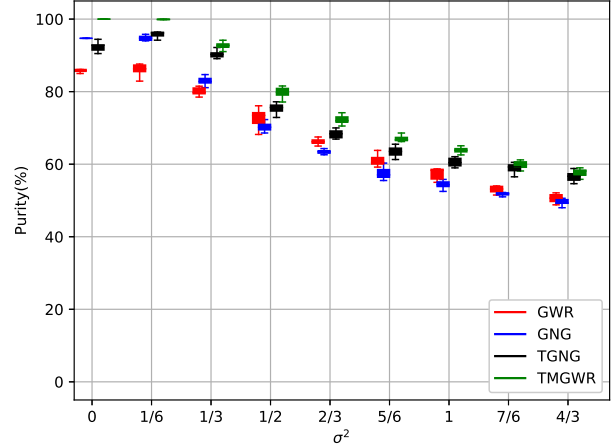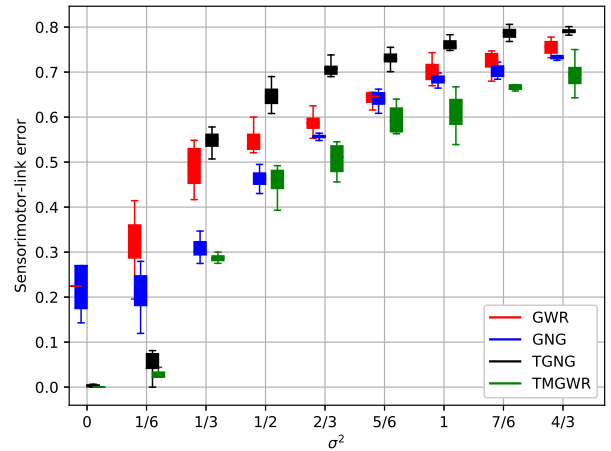

Fig. 5: Sensorimotor-link errors of the algorithms at different noise levels in the environment of Fig. 1b

is computed and reported in Fig. 6. TMGWR shows better disambiguation potential than other algorithms. This is because it uses both memory and its growing mechanism to keep track of the changes in the environment. TMGWR network is therefore more suitable for partially observable and changing environment than the other three algorithms. Moreover, the result shows that growing self-organising neural techniques that use how much error the map makes in representing observations, as a strategy for deciding when to add an extra node or not, show better resistance to noise than the ones that add nodes at constant time rate. For example, TMGWR, TGNG and GWR performed better than GNG as the level of noise is increased.
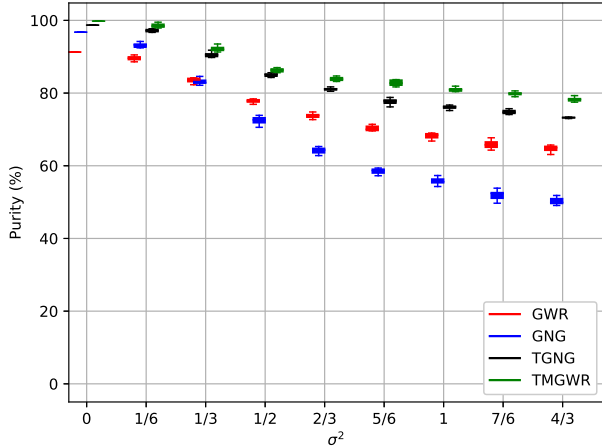
Fig. 6: Disambiguation performances of the algorithms at different noise levels in the environment of Fig. 1b

VI. CONCLUSION

The paper presents a Temporospatial Merge Grow When Required (TMGWR) network for sensorimotor map learning of an unknown, noisy or partially observable environment. In order to learn the sensorimotor map of any given environment, TMGWR develops the sensorial and sensorimotor representations of the environment during explorations. This map can later be used for autonomous decision-making or a goal-directed planning in the environment. The algorithm has been compared with GNG, GWR and TGNG on the basis of their disambiguation abilities, sensorial and sensorimotor representation accuracies. Sensorimotor-link error has been formulated in this paper for evaluating how well a sensorimotor map learning algorithm can represent causality in its environment. Based on the experiments presented in the paper, it has been shown that TMGWR is more efficient and suitable for sensorimotor map learning in noisy and partially observable environments than the competing algorithms.

TMGWR recruits additional nodes using the same mechanism that is applied in GWR algorithm. Moreover, TMGWR learns to represent the environment in a recurrent manner making it able to keep track of changes in the environment. In this paper we did not leverage the habituation mechanism in the traditional GWR algorithm. This could be useful for an informed exploration of a large environment. We plan to investigate how the use of the habituation mechanism or similar intrinsic motivations can help TMGWR learn efficiently in large environments. Instead of the agent engaging in random explorations of its environment, intrinsic motivation can enable it to pay more attention to interesting aspects of the environment. Besides, this paper has not examined goal-directed planning on the sensorimotor map learnt with the TMGWR network. Future work will investigate goal-directed planning based on TMGWR in a high-dimensional noisy environment such as in a vision-based environment.

The performance of the TMGWR-based approach in goal-directed tasks will be compared against those of traditional reinforcement learning algorithms.

REFERENCES

[1] J. Piaget, "The origins of intelligence in children new york: Int," 1952.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[3] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[4] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.

[5] I. Cos-Aguilera, G. Hayes, and L. Canamero, "Using a sofm to learn object affordances," in *Procs 5th Workshop of Physical Agents (WAF'04)*. University of Edinburgh, 2004.

[6] S. Kim and F. C. Park, "Fast robot motion generation using principal components: Framework and algorithms," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2506–2516, 2008.

[7] K. Ishii, S. Nishida, and T. Ura, "A self-organizing map based navigation system for an underwater robot," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4466–4471.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[9] T. Kulak and M. G. Ortiz, "Representation learning in partially observable environments using sensorimotor prediction," *arXiv preprint arXiv:1803.00268*, 2018.

[10] C. P. Ezenkwu and A. Starkey, "Machine autonomy: Definition, approaches, challenges and research gaps," in *Intelligent Computing-Proceedings of the Computing Conference*. Springer, 2019, pp. 335–358.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] T. Kohonen, "The self-organizing map. proceedings of the ieee, 78," 1990.

[13] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in neural information processing systems*, 1995, pp. 625–632.

[14] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural networks*, vol. 15, no. 8-9, pp. 1041–1058, 2002.

[15] B. Fritzke, "Let it grow-self-organizing feature maps with problem dependent cell structure," in *Artificial neural networks*. Citeseer, 1991.

[16] M. V. Butz, K. Reif, and O. Herbort, "Bridging the gap: Learning sensorimotor-linked population codes for planning and motor control," in *International Conference on Cognitive Systems, CogSys*, vol. 2008, 2008.

[17] M. Toussaint, "A sensorimotor map: Modulating lateral interactions for anticipation and planning," *Neural computation*, vol. 18, no. 5, pp. 1132–1155, 2006.

[18] ——, "Learning a world model and planning with a self-organizing, dynamic neural system," in *Advances in neural information processing systems*, 2004, pp. 926–936.

[19] M. Strickert and B. Hammer, "Merge som for temporal data," *Neurocomputing*, vol. 64, pp. 39–71, 2005.

[20] H. H. Chaput, "The constructivist learning architecture: A model of cognitive development for robust autonomous robots," Ph.D. dissertation, 2004.

[21] G. L. Drescher, *Made-up minds: a constructivist approach to artificial intelligence*. MIT press, 1991.

[22] V. Mohan, P. Morasso, G. Metta *et al.*, "The distribution of rewards in sensorimotor maps acquired by cognitive robots through exploration," *Neurocomputing*, vol. 74, no. 17, pp. 3440–3455, 2011.

[23] M. Kubisch, "A growing multi-expert structure for open-ended unsupervised learning of sensory state spaces," in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2017, pp. 190–195.

[24] L. Mici, G. I. Parisi, and S. Wermter, "An incremental self-organizing architecture for sensorimotor learning and prediction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 918–928, 2018.

[25] J. Koutnik, "Inductive modelling of temporal sequences by means of self-organization," in *Proceeding of International Workshop on Inductive Modelling*, 2007, pp. 269–277.

[26] T. Voegtlin, "Recursive self-organizing maps," *Neural networks*, vol. 15, no. 8-9, pp. 979–991, 2002.

[27] M. Varsta, J. Heikkonen, J. Lampinen, and J. D. R. Millán, "Temporal kohonen map and the recurrent self-organizing map: Analytical and experimental comparison," *Neural processing letters*, vol. 13, no. 3, pp. 237–251, 2001.

[28] M. Hagenbuchner, A. Sperduti, and A. C. Tsoi, "A self-organizing map for adaptive processing of structured data," *IEEE transactions on Neural Networks*, vol. 14, no. 3, pp. 491–505, 2003.

[29] A. Andreakis, N. v. Hoyningen-Huene, and M. Beetz, "Incremental unsupervised time series analysis using merge growing neural gas," in *International Workshop on Self-Organizing Maps*. Springer, 2009, pp. 10–18.

[30] J. G. Proakis, *Digital signal processing: principles algorithms and applications*. Pearson Education India, 2001.

[31] C. Guo, H. Jia, and N. Zhang, "Time series clustering based on ica for stock data analysis," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2008, pp. 1–4.

[32] V. Kurbalija, C. von Bernstorff, H.-D. Burkhard, J. Nachtwei, M. Ivanović, and L. Fodor, "Time-series mining in a psychological domain," in *Proceedings of the Fifth Balkan Conference in Informatics*. ACM, 2012, pp. 58–63.

[33] R. Gopalapillai, D. Gupta, and T. S. B. Sudarshan, "Experimentation and analysis of time series data for rescue robotics," in *Recent Advances in Intelligent Informatics*, S. M. Thampi, A. Abraham, S. K. Pal, and J. M. C. Rodriguez, Eds. Cham: Springer International Publishing, 2014, pp. 443–453.

[34] P. Sebastiani and M. Ramoni, "Bayesian clustering by dynamics of european school population," in *Proceedings of the ISBA2000 Conference*, 2000.

[35] M. Ferro, D. Ognibene, G. Pezzulo, and V. Pirrelli, "Reading as active sensing: a computational model of gaze planning during word recognition," *Frontiers in Neurorobotics*, vol. 4, p. 6, 2010.

[36] V. Graziano, J. Koutník, and J. Schmidhuber, "Unsupervised modeling of partially observable environments," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 503–515.

[37] G. Pierris and T. S. Dahl, "Humanoid tactile gesture production using a hierarchical som-based encoding," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 2, pp. 153–167, 2014.

[38] ——, "Learning robot control using a hierarchical som-based encoding," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 1, pp. 30–43, 2017.

[39] ——, "A developmental perspective on humanoid skill learning using a hierarchical som-based encoding," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 708–715.

[40] G. T. Breard, "Evaluating self-organizing map quality measures as convergence criteria," 2017.

[41] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.