# Towards Open-Ended Learning of Action Sequences with Hierarchical Predictive Self-Organization

German I. Parisi and Stefan Wermter

*Abstract*— **Open-ended learning is fundamental in autonomous robotics for the incremental acquisition of knowledge through experience. However, most of the proposed computational models for action recognition do not account for incremental learning, but rather learn a batch of training actions without adapting to new inputs presented after training sessions. Therefore, this is the need to provide robots with the ability to incrementally process a set of available perceptual cues and to adapt their behavioural responses over time. In this work, we propose a neural network architecture with multilayer-predictive processing for incrementally learning action sequences. Our architecture comprises a hierarchy of self-organizing networks that progressively learn the spatiotemporal structure of the input using Hebbian-like plasticity. Along the hierarchical flow with increasingly larger temporal receptive fields, feedback connections from higher-order networks carry predictions of lower-level neural activation patterns, whereas feedforward connections convey residual errors between the predictions and the lower-level activity. This mechanism is used to modulate the amount of learning necessary to adapt to the dynamic input distribution and develop robust action representations. We present a simplified hierarchical architecture with two layers and describe a number of planned experiments for classifying human actions in an open-ended learning scenario.**

## I. INTRODUCTION

The robust processing of dynamic input patterns from video and audio streams plays a crucial role for learning robots engaged in tasks such as action recognition, detection of dangerous events (e.g., falling) and socially-aware communication through flexible human-robot interaction (e.g., multimodal interaction from audiovisual input). In this context, the design of learning methods that account for open-ended adaptive learning has been shown to be very challenging.

Computational models inspired by the hierarchical organization of the cerebral cortex have become increasingly popular for learning complex visual patterns such as action sequences from video. In particular, neural network approaches with deep architectures have shown very good results on a set of benchmark datasets containing daily actions [1, 2]. The terminology *deep architecture* generally refers to models that use a number of hierarchically-arranged layers (generally more than two) for learning latent structures in the input at different spatiotemporal scales. This processing scheme is in agreement with neurophysiological evidence that supports the presence of increasingly larger spatial and temporal receptive fields along the visual and auditory cortical pathways [3, 4]. Despite the number of neuroanatomical studies on the organization and connectivity of cortical areas responsible for the processing of dynamic stimuli [5, 6, 7], so far no common computational framework has been introduced that parsimoniously integrates well-established biological facts in terms of architecture and learning procedures [8, 9]. Different solutions have been proposed based on task-oriented neural network modelling, typically relying on a trade-off between biological findings and simplifications aimed to yield good performance and computational feasibility.

In the realm of action recognition, a large number of computational models have been proposed to learn a set of training actions with the use of hierarchically-arranged network layers [10, 11, 1, 2]. For instance, in Parisi *et al.* [2] we proposed a hierarchical neural approach for the self-organizing integration of pose-motion features from action videos. The model consists of growing self-organizing networks arranged in a hierarchical fashion to obtain progressively generalized representations of visual inputs with increasingly larger temporal windows. Each network in the hierarchical flow is fed with a set of neural activation trajectories from the previous layer. A variant of unsupervised learning with two labelling functions was proposed to extend the model for classification. Experiments have shown state-of-the-art results on two benchmarks of daily actions captured with depth sensors, i.e. KT full-body action dataset [12] and CAD-60 [13].

These and other similar approaches have been designed for learning a batch of training actions, thus implicitly assuming that a training set is available. Ideally, this training set contains all necessary knowledge that can be readily used to predict novel samples in a given domain. However, such a training scheme is not suitable in more natural scenarios where an artificial agent should incrementally process a set of perceptual cues that become available over time. This kind of learning paradigm, referred to as open-ended or incremental learning, is considered to be essential for cognitive development and plays a key role in autonomous robotics for the progressive acquisition of knowledge through experience and the development of meaningful internal representations during training sessions [14, 15].

It has been argued that hierarchical predictive models with interactions between top-down predictions and bottom-up regression may provide a computational mechanism to account for the learning of dynamic input distributions in an unsupervised fashion [1]. Predictive coding [16, 17] has been widely studied for understanding many aspects of brain organization and, in particular, it has been proposed that the

German I. Parisi and Stefan Wermter are with the Department of Informatics, University of Hamburg, Germany. {parisi,wermter}@informatik.uni-hamburg.de

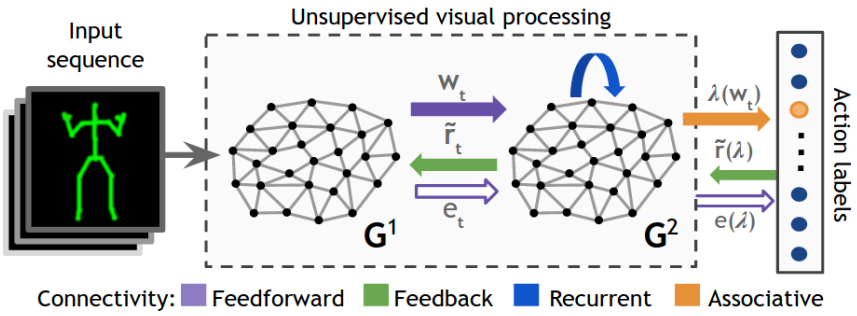


Fig. 1. Diagram of our two-layer architecture with GWR networks. Inter-layer connectivity is implemented with feedforward (purple) and feedback (green) connections modulating the amount of learning required at each hierarchical stage. In $G^2$, recurrent connectivity is used to learn temporal dependencies of neural activation trajectories from $G^1$. Furthermore, associative connections (orange) between visual representations and action labels are learned for classification purposes. Feedback from the "Action labels" layer is used in $G^1$ to modulate the learning of the set of prototype neurons necessary to correctly classify action sequences in $G^2$.

visual cortex can be modelled as a hierarchical network with reciprocal connections where top-down feedback connections from higher-order cortical areas convey predictions of lower-order neural activity and bottom-up connections carry the residual prediction errors. Tani *et al.* [18, 19] proposed that the generation and recognition of sensory-motor patterns for on-line planning in a robot learning scenario can be obtained by using recurrent neural network models extended with prediction error minimization. However, neural network models that implement a predictive learning scheme to achieve incremental, open-ended learning have not been yet fully investigated.

In this work, we propose a neural architecture for the incremental learning of action sequences. Our architecture comprises a two-stage hierarchy of growing self-organizing networks for processing action features with increasing temporal receptive windows. The network in the first layer learns a dictionary of time-independent action features, while the second network is equipped with recurrent connectivity to learn neural activation patterns from the first layer. We introduce a novel type of recurrent self-organizing learning algorithm (the Gamma-GWR) and an architecture with reciprocal connectivity based on a hierarchical predictive processing scheme for modulating the amount of learning required to learn action representations in an incremental fashion. In an open-ended learning scenario, the networks will adapt to the dynamic input distribution, whereas they will remain stable if the distribution becomes stationary. For simplicity, we present a neural architecture with two layers (while the model could be extended to more layers) and describe a number of experiments that we are planning to conduct for the task of open-ended learning of action sequences from multimodal streams (audiovisual input) using an extended version of the architecture.

## II. LEARNING ARCHITECTURE

Our learning model consists of two hierarchically-arranged Growing When Required (GWR) networks [20]. The first network layer $G^1$ learns a dictionary of prototype, time-independent action features using the standard mechanism of GWR learning. The second network layer $G^2$ is equipped with recurrent connectivity to learn temporal dependencies of the input in terms of neural activation trajectories from $G^1$. This hierarchical flow yields specialized neurons encoding information accumulated over larger temporal windows. We implement feedforward (bottom-up) and feedback (top-down) connectivity following the predictive coding principle for modulating the amount of learning required to adapt to dynamic input distributions and developing stable action representations. For classification purposes, the second network layer is equipped with associative connections between unsupervised visual representations and action labels. The feedback from the Action labels layer is used in $G^1$ to modulate the learning of the set of prototype neurons necessary to correctly classify action sequences in $G^2$. The overall architecture is illustrated by Fig. 1. For our learning scenario, we assume that a set of visual features describing relevant spatiotemporal properties of the input becomes available over time, e.g. 3D body joints from depth map video sequences for human action recognition.

### A. Incremental Learning of Topographic Maps

Topographic maps exhibiting experience-driven development are a common feature of the cortex for processing sensory inputs [21, 22]. Different models of neural self-organization have been proposed to resemble the dynamics of basic biological findings on Hebbian-like learning and map plasticity (e.g, [23, 24]). In this paper, we focus on a particular type of self-organizing network for incremental learning – the *Growing When Required* (GWR) network [20], composed of a set of neurons with their associated weight vectors linked by edges. The activity of a neuron is computed as a function of the distance between the input and its weight vector. During the training, the network dynamically changes its topological structure to better match the input space using competitive Hebbian learning [25].

Different from other incremental models of self-organization that create new neurons at a fixed growth rate (e.g., Growing Neural Gas (GNG) [24]), the GWR-based learning process creates new nodes whenever the activity of trained neurons is smaller than a given threshold. The amount of activation at time $t$ is computed as a function

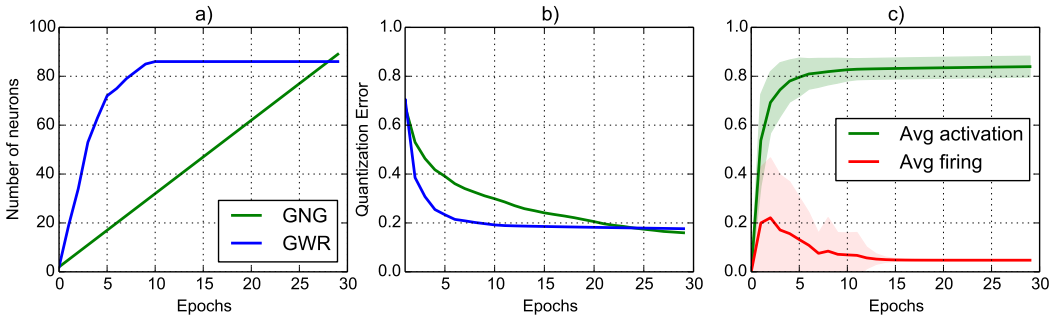


Fig. 2. Comparison of GNG and GWR: a) number of neurons, b) quantization error, and c) average activation and firing counter (GWR only) through 30 training epochs for the Iris dataset (150 four-dimensional samples).

of the distance between the current input $\mathbf{x}_t$ and its best-matching neuron $\mathbf{w}_{b_t}$:

$$a_t = exp(-\|\mathbf{x}_t - \mathbf{w}_{b_t}\|). \quad (1)$$

Additionally, the network implements a firing counter $\eta \in [0, 1]$ to express how frequently a neuron $\mathbf{w}_i$ has fired so that existing neurons are sufficiently trained before new ones are created. This mechanism creates a larger number of neurons at early stages of the training and then tunes the weights through subsequent training iterations (epochs). A comparison between GNG and GWR learning in terms of the number of neurons, quantization error (average discrepancy between the input and representative neurons in the network), and parameters modulating network growth (average network activation and firing rate) is shown in Fig. 2 over 30 training epochs for the Iris dataset[1]. Such a learning behaviour is particularly convenient for incremental learning scenarios since neurons will be created to promptly distribute in the input space, thereby allowing a faster convergence through iterative fine-tuning of the topological map. It has been shown that GWR-based learning is particularly suitable for novelty detection and cumulative learning in robot scenarios [26].

The standard formulation of the GWR algorithm does not account for temporal sequence processing as required for action recognition. Therefore, it is necessary to extend the algorithm with recurrent connectivity while preserving desirable GWR learning properties such as computational efficiency and network convergence.

### B. The Gamma-GWR Network

In recent work [27] we presented an extension of the standard GWR network for context learning using recursive connectivity as introduced in [28], and showed that our recursive GWR model outperforms other self-organizing networks that implement similar context learning schemes (e.g. [29]). This approach for temporal processing equips each neuron with a context descriptor so that the activation function is defined by the linear combination of activations driven by the current input and the previous timesteps. Both the weights and context descriptors of the neurons lie in the same feature space as the input.

[1] http://archive.ics.uci.edu/ml/datasets/Iris

In this work, we introduce a recursive GWR network that equips each neuron with an arbitrary number of context descriptors to increase the memory depth and temporal resolution following the idea of a Gamma memory model [30]. A similar approach has been previously applied to Growing Neural Gas learning showing good results in nonlinear time series analysis [31]. Following previous formulations of context learning, the activation of the network with a K-order Gamma memory becomes

$$d_i(t) = \alpha_w \cdot \|\mathbf{x}_t - \mathbf{w}_i\|^2 + \sum_{k=1}^{K} \alpha_k \cdot \|\mathbf{C}_k(t) - \mathbf{c}_k^i\|^2, \quad (2)$$

$$\mathbf{C}_k(t) = \beta \cdot \mathbf{c}_k^{I_{t-1}} + (1-\beta) \cdot \mathbf{c}_{k-1}^{I_{t-1}} \quad \forall K = 1, ..., K, \quad (3)$$

where $\alpha, \beta \in (0; 1)$ are constant values that modulate the influence of the current input and the past, and $\mathbf{c}_0^{I_{t-1}} \equiv \mathbf{w}^{I_{t-1}}$ with random $\mathbf{c}_k^{I_0}$ at $t = 0$. It has been shown that the mean memory depth is $D = K/(1-\beta)$ and its temporal resolution is $R = 1 - \beta$. Therefore, both depth and resolution are modulated by the value of $\beta$ [31]. To be noted is that in this recursive version of the GWR algorithm, the activation function (Eq. 1) is replaced with $a_t = exp(-d_i(t))$.

The training procedure of the proposed Gamma-GWR is illustrated by Algorithm 1. This training algorithm does not impose a specific criterion to stop the training of the network. Typically, a maximum number of training epochs can represent a convenient choice if a batch of inputs is available. However, if we assume that the distribution of the inputs is dynamic, then this criterion is no longer valid. Thus, it is necessary to adopt a mechanism that allows to keep learning novel input in open-ended learning scenarios while guaranteeing an acceptable degree of stability when the distribution becomes stationary.

### C. Predictive Coding and Open-Ended Learning

It has been argued that predictive coding [16, 17] provides a framework for explaining the hierarchical reciprocally connected organization of the cortex. Thus, the question arises on how this scheme may be used to provide a mechanism that learns dynamic stimuli distributions in a hierarchical fashion. More specifically, we are interested in a mechanism for achieving open-ended learning along a hierarchy of adaptive

**Algorithm 1** Gamma-GWR.

1: Start with a set of two random nodes, $A = \{\mathbf{w}_1, \mathbf{w}_2\}$ with context vectors $\mathbf{c}_k^i$ for $k = 1, ..., K, i = 1, 2$.
2: Initialize an empty set of connections $E = \emptyset$.
3: Initialize $K$ empty global contexts $\mathbf{C}_k = 0$.
4: At each iteration, generate an input sample $\mathbf{x}_t$.
5: Select the best and second-best matching neurons (Eq. 2):
$b = \arg\min_{i \in A} d_i(t)$, $s = \arg\min_{i \in A/\{b\}} d_i(t)$.
6: Update contexts $\mathbf{C}_k$ for next time step (Eq. 3).
7: Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
8: If $(exp(-d_b(t)) < a_T)$ and $(\eta_b < f_T)$ then:
Add a new node $r$ $(A = A \cup \{r\})$:
$\mathbf{w}_r = 0.5 \cdot (\mathbf{w}_b + \mathbf{x}_t)$, $\mathbf{c}_k^r = 0.5 \cdot (\mathbf{C}_k(t) + \mathbf{c}_k^i), \eta_r = 1$,
Update edges between neurons:
$E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$.
9: If no new node is added, update weight and context of the winning node and its neighbours $i$:
$\Delta\mathbf{w}_b = \epsilon_b \cdot \eta(b) \cdot (\mathbf{x}_t - \mathbf{w}_s)$,
$\Delta\mathbf{w}_i = \epsilon_n \cdot \eta(i) \cdot (\mathbf{x}_t - \mathbf{w}_i)$,
$\Delta\mathbf{c}_k^b = \epsilon_b \cdot \eta(b) \cdot (\mathbf{C}_k(t) - \mathbf{c}_k^s)$,
$\Delta\mathbf{c}_k^i = \epsilon_n \cdot \eta(i) \cdot (\mathbf{C}_k(t) - \mathbf{c}_k^i)$, with $0 < \epsilon_n < \epsilon_b < 1$.
10: Increment the age of all edges connected to $b$ of 1.
11: Reduce the firing counters of the best-matching neuron and its neighbours $i$:
$\eta_b = \eta_b + (\tau_b \cdot \kappa \cdot (1 - \eta_b) - \tau_b)$,
$\eta_i = \eta_i + (\tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_b)$,
with $\tau$, $\kappa$ constants controlling the curve behaviour.
12: Remove all edges with ages larger than $\mu_{max}$ and remove nodes without edges.
13: If the stop criterion is not met, repeat from step 4.

networks and modulating the amount of learning required at each stage so that robust representations of visual inputs can develop in an unsupervised fashion.

In the predictive coding model of the visual cortex [16], higher-level neurons attempt to predict responses of lower-level neurons through feedback (top-down) connections, while lower-level neurons send forward the prediction error and the actual neural activity via bottom-up connections. As one ascends the hierarchy, neurons predict and estimate signal properties at a larger spatiotemporal scale, as supported by neurophysiological evidence suggesting increasingly larger spatial and temporal receptive fields along the cortical pathways [3, 4].

Given two contiguous network layers $G^{L-1}$ and $G^L$, neural activations from $G^{L-1}$ will be sent to $G^L$ via feed-forward connections. $G^L$ should be able to encode temporal dependencies of the input using neural activation trajectories from the previous layer according to the activation function (Eq. 2). Therefore, we can assume that at each timestep $t$ the layer $G^L$ will predict a set of sequence-selective inputs from $G^{L-1}$. This prediction will be sent to $G^{L-1}$ via a feedback connection to estimate the prediction error, which is then sent forward to $G^L$. In our approach, this is implemented by comparing actual neural activations $\mathbf{r}$ from $G^{L-1}$ to their prediction $\widetilde{\mathbf{r}}$ computed as the recursive retrieval of learned sequences in $G^L$, so that the network is trained until the difference $(\mathbf{r} - \widetilde{\mathbf{r}})$ is smaller than an error threshold. Convenient threshold values should be chosen so that the layers adapt to dynamic input (yielding higher prediction errors) while showing convergence with stationary input.

### *D. A Predictive Architecture for Incremental Learning*

In our 2-layer architecture (Fig.1), the first network layer $G^1$ receives as input a set of visual features and learns a dictionary of prototype neurons encoding spatial (single-frame) properties. The second layer $G^2$ encodes temporal dependencies from consecutive frames in terms of neural activation trajectories from the previous layer. The order $K$ of the memory of $G^2$ expresses the number of neural activations from the previous layer that leak into the activation of the current layer (Eq. 2). Therefore, $G^2$ processes neural activations from $G^1$ for $K+1$ timesteps. If we assume input at 10 frames per second and a network layer $G^2$ with $K = 9$, then a neuron in $G^2$ encodes a sequence snapshot of 1 second.

Each neuron in the Gamma-GWR model stores a set of $K$ context descriptors that we use to efficiently compute neural activation predictions. We assume that a network *predicts* neural activity from a previous layer if the former is able to recursively reconstruct the input of the latter for a given temporal window. More specifically, given a network layer $G^L$ with a $K$-order memory, the recursive reconstruction of the input from $G^{L-1}$ at time $t$ from the last $K+1$ timesteps is given by

$$\widetilde{\mathbf{r}}_t = < \mathbf{w}_b, \mathbf{c}_b^1, ..., \mathbf{c}_b^K >, \quad b = argmin_i d_i(t). \tag{4}$$

Given the sequence of neural activations $\mathbf{r}_t$ from $G^{L-1}$ at time $t$, the prediction error is computed as

$$e_t = \|\mathbf{r}_t - \widetilde{\mathbf{r}}_t\|. \tag{5}$$

This prediction error is sent forward to $G^L$ that keeps learning if $E_t$ is greater than an error threshold $e_T^L$.

### *E. Action Classification*

Action representations emerge hierarchically through the unsupervised training of GWR networks. For classification purposes, action labels from training samples are attached to neurons in the last layer. Therefore, we use an extension of the unsupervised GWR learning with two labelling functions [2]: one for the training phase and one for predicting the label of unseen samples. Given a set $L$ with $j$ action labels, each neuron in $G^2$ will be linked to one of these action labels according to the label of the training sample $\lambda(\mathbf{x}_i) \in L$. For $G^2$ with $K = 9$, an action label will be predicted for each segment of 10 frames with a sliding window scheme.

While the growth of $G^2$ is modulated by its capability to predict neural activation sequences from $G^1$, the performance of the architecture in terms of the correct classification of sequence labels is used to modulate the growth of $G^1$. More

specifically, feedback connectivity from the "Action labels" layer will have a direct influence on the growth rate of $G^1$ so that a sufficient number of prototype neurons are created as a dictionary of time-independent primitives subsequently used to learn spatiotemporal statistics of the input. In each layer, an error measure is used to modulate the amount of learning (update of $a_T$ and additional training epochs). In the case of $G^2$, the activation threshold will be increased if the prediction error (Eq. 5) exceeds an error threshold. A convenient threshold should take into account a reasonable prediction error tolerance for learned temporal dependencies, linking the choice of this error threshold to the activation threshold (since both the error and the activation are a function of input–neuron discrepancies). In the case of $G^1$, the error measure used to update the growth rate is based on the classification performance of the neurons in $G^2$ in terms of correct neuron–label associations.

## III. PLANNED EXPERIMENTS

The aim of the experiments is to explore different parameters that yield 1) a convenient trade-off between learning adaptability and network convergence, and 2) a good classification accuracy. For this purpose, the following aspects should be taken into careful consideration.

The two parameters modulating the growth rate of the networks are the activation threshold and the firing rate threshold (Fig. 2.c), with the former having stronger influence. The activation threshold $a_T$ establishes the maximum discrepancy (distance) between the input and its best-matching neuron in the network. For larger values of $a_T$, the discrepancy expressed by Eq. 1 will be smaller. Intuitively, the average discrepancy between the input and the network should decrease for a larger number of neurons. On the other hand, there is not such a straightforward relation between the number of neurons and the classification performance. This is because the classification process consists of predicting the label of novel samples by retrieving attached labels to the inputs' best-matching neurons, with the actual distance between the novel inputs and the selected neurons being irrelevant for this task. Therefore, a convenient value for $a_T$ should be chosen by taking into account the distribution of the input and, in the case of a classification task, the classification performance.

Additionally to the above-mentioned parameters, also the maximum age of the connections between neurons must be considered. At each iteration, when a neuron is fired (Eq. 1 or 2), the age of the connections from the neuron to its neighbours is set to 0, while the age of the rest of the connections is increased by a value of 1. This mechanism removes old connections and neurons without any connection as the result of neurons that have not been fired for a while, e.g. in the case of dynamically distributed input. On the other hand, removing a neuron from the network means that the information learned by that unit is permanently forgotten. Therefore, a convenient maximum age of connections $\mu_{max}$ must be set so that the network removes neurons that are no longer fired while avoiding *catastrophic forgetting*, i.e. forgetting previously learned representations during the process of learning new ones.

### A. Action Datasets

We plan to evaluate our architecture with two action datasets:

*KT Full-Body Actions Dataset* [12] comprising 10 full-body actions performed by 13 participants with a normal physical condition. Participants were naive as to the purpose of the experiment and were recorded individually in a home-like environment with a Kinect sensor. Depth maps were sampled with a VGA resolution of 640x480 and a constant frame rate of 30 Hz. The dataset has periodic actions (standing, walking, jogging, sitting, lying down, crawling) and goal-oriented actions (pick up object, jump, fall down, stand up).

*Weizmann Dataset* [32] containing 90 sequences with 10 actions (walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, and skip) performed by 9 subjects. Sequences are sampled at 180x144 with static background and are 3 seconds long. For our experiments, we will use aligned foreground body shapes by background subtraction included in the dataset.

In addition to evaluating the performance of our system in terms of classification accuracy for these two datasets, it would also be interesting to investigate how these two different representations of human motion (i.e. 3D skeleton models and 2D segmented body silhouettes) influence the formation of topographic maps and the overall learning dynamics.

## IV. CONCLUSIONS AND FUTURE WORK

We proposed a self-organizing hierarchy of growing neural networks with reciprocal connectivity to develop robust action representations in an open-ended fashion. We plan to conduct a set of experiments with datasets containing human actions and evaluate the performance of our system with respect to batch-learning versions. We presented a simplified neural architecture based on hierarchical predictive processing, while the convenient values of the parameters modulating the adaptation to dynamic input and learning convergence are subject to ongoing investigation. Good results would motivate future work in several directions. For instance, we could apply the proposed reciprocal connectivity to a more complex neural architecture such as a self-organizing model for the integration of pose-motion features from two converging processing pathways [2].

We are planning to use hierarchical self-organizing learning to obtain robust multimodal action representations from low-level visual and auditory cues. In our current approach, we have assumed that the labels of the training samples are available and correct. In order to foster a more natural learning scenario, action labels could be acquired from speech recognition with action words being learned in a hierarchical fashion from low-level auditory cues. An additional research direction is to extend the associative learning scheme between visual representations of action segments

and action labels so that robust action–to–label associations can develop also in the sporadic absence of training labels or a given amount of label noise.

In this paper, we have proposed open-ended learning in terms of prediction-driven neural dynamics with action representations emerging from the interplay of feedforward–feedback connectivity in a self-organizing hierarchy. However, we have not taken into account other important principles that play a role in open-ended learning such as the influence of reward-driven motivational and attentional functions [33], which will be subject of future research.

## ACKNOWLEDGMENT

This work was partially supported by the DAAD German Academic Exchange Service for the Cognitive Assistive Systems project (Kz:A/13/94748), the Transregio TRR169 on Crossmodal Learning, and the Hamburg Landesforschungsfoerderung.

The authors would like to thank Nourhan Elfaramawy, Sven Magg, Cornelius Weber and Jun Tani for valuable comments and discussions that helped improve the quality of this manuscript.

## REFERENCES

[1] M. Jung, J. Hwang, and J. Tani, "Self-organization of spatio-temporal hierarchy via learning of dynamic visual image patterns on action sequences," *PLoS ONE*, vol. 10, no. 7, p. e0131214, 2015.

[2] G. I. Parisi, C. Weber, and S. Wermter, "Self-organizing neural integration of pose-motion features for human action recognition," *Frontiers in Neurorobotics*, vol. 9, no. 3, 2015.

[3] U. Hasson, E. Yang, I. Vallines, D. J. Heeger, and N. Rubin, "A hierarchy of temporal receptive windows in human cortex," *The Journal of Neuroscience*, vol. 28, no. 10, pp. 2539–2550, 2008.

[4] Y. Lerner, C. J. Honey, L. J. Silbert, and U. Hasson, "Topographic mapping of a hierarchy of temporal receptive windows using a narrated story," *The Journal of Neuroscience*, vol. 31, no. 8, pp. 2906–2915, 2011.

[5] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, 1991.

[6] A. Boemio, S. Fromm, A. Braun, and D. Poeppel, "Hierarchical and asymmetric temporal sensitivity in human auditory cortices," *Nat Neurosci*, vol. 8, pp. 389–395, 2005.

[7] P. Taylor, J. N. Hobbs, J. Burroni, and H. T. Siegelmann, "The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions," *Scientific Reports*, vol. 5, no. 18112, 2015.

[8] R. C. O'Reilly, "Six principles for biologically based computational models of cortical cognition," *Trends in Cognitive Sciences*, vol. 2, no. 11, pp. 455–462, 1998.

[9] M. A. Giese and G. Rizzolatti, "Neural and computational mechanisms of action processing: Interaction between visual and motor representations," *Neuron*, vol. 88, no. 1, pp. 167 – 180, 2015.

[10] M. A. Giese and T. Poggio, "Neural mechanisms for the recognition of biological movements." *Nat Rev Neurosci*, vol. 4, no. 3, pp. 179–192, 2003.

[11] G. Layher, M. A. Giese, and H. Neumann, "Learning representations of animated motion sequences – a neural model," in *35th annual meeting of the Cognitive Science Society*, 2013.

[12] G. I. Parisi, C. Weber, and S. Wermter, "Human action recognition with hierarchical growing neural gas learning," in *Artificial Neural Networks and Machine Learning - ICANN'14*, 2014, pp. 89–96.

[13] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 842–849.

[14] H. H. Zhou, "Csm: A computational model of cumulative learning," *Machine Learning*, vol. 5, no. 4, pp. 383–406, 1990.

[15] J. Lee, *Encyclopedia of the Sciences of Learning*. Boston, MA: Springer US, 2012, ch. Cumulative Learning, pp. 887–893.

[16] R. Rao and D. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, pp. 79–87, 1999.

[17] Y. Huang and R. P. N. Rao, "Predictive coding," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 5, pp. 580–593, 2011.

[18] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Networks*, vol. 12, no. 7-8, pp. 1131–1141, 1999.

[19] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes." *Neural networks*, vol. 16, no. 1, pp. 11–23, 2003.

[20] 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.

[21] D. J. Willshaw and C. V. D. Malsburg, "How patterned neural connections can be set up by self-organization," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 194, no. 1117, pp. 431–445, 1976.

[22] C. A. Nelson, "Neural plasticity and human development: the role of early experience in sculpting memory systems," *Developmental Science*, vol. 3, no. 2, pp. 115–136, 2000.

[23] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1988.

[24] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*. MIT Press, 1995, pp. 625–632.

[25] T. M. Martinetz, "Competitive Hebbian learning rule forms perfectly topology preserving maps," in *ICANN'93: International Conference on Artificial Neural Networks*. Amsterdam: Springer, 1993, pp. 427–434.

[26] S. Marsland, U. Nehmzow, and J. Shapiro, "On-line novelty detection for autonomous mobile robots," *Robotics and Autonomous Systems*, vol. 51, no. 2-3, pp. 191–206, May 2005.

[27] G. Parisi, S. Magg, and S. Wermter, "Human motion assessment in real time using recurrent self-organization," in *RO-MAN'16*, in press.

[28] M. Strickert and B. Hammer, "Merge SOM for temporal data," *Neurocomputing*, vol. 64, 2005.

[29] A. Andreakis, N. von Hoyningen-Huene, and M. Beetz, "Incremental unsupervised time series analysis using merge growing neural gas," in *WSOM*, ser. Lecture Notes in Computer Science, vol. 5629. Springer, 2009, pp. 10–18.

[30] B. de Vries and J. C. Príncipe, "The gamma model–a new neural model for temporal processing," *Neural Networks*, vol. 5, no. 4, pp. 565–576, 1992.

[31] P. A. Estévez and J. R. Vergara, "Nonlinear time series analysis by using gamma growing neural gas," in *WSOM*, 2012, pp. 205–214.

[32] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, 2007.

[33] I. Ivanov, X. Liu, S. Clerkin, K. Schulz, K. Friston, J. H. Newcorn, and J. Fan, "Effects of motivation on reward and attentional networks: an fmri study." *Brain and Behavior*, vol. 2, no. 6, pp. 741–753, 2012.