Unsupervised online learning of complex sequences in spiking neuronal networks

Younes Bouhadjar*,†
Peter Grünberg Institute (PGI-15)
Jülich Research Centre
Jülich, Germany
y.bouhadjar@fz-juelich.de
ORCID: 0000-0003-4367-8236

Melissa Lober*
Institute for Advanced Simulation (IAS-6)
Jülich Research Centre
Jülich, Germany
Department of Physics, Faculty 1
RWTH Aachen University
Aachen, Germany
m.lober@fz-juelich.de
ORCID: 0000-0002-1833-5170

Emre Neftci
Peter Grünberg Institute (PGI-15)
Jülich Research Centre
Jülich, Germany
e.neftci@fz-juelich.de
ORCID: 0000-0002-0332-3273

Markus Diesmann
Institute for Advanced Simulation (IAS-6) &
JARA Institute Brain Structure-Function Relationships (INM-10)
Jülich Research Centre
Jülich, Germany
Department of Psychiatry, Psychotherapy and Psychosomatics
Medical Faculty &
Department of Physics, Faculty 1
RWTH Aachen University
Aachen, Germany
m.diesmann@fz-juelich.de
ORCID: 0000-0002-2308-5727

Tom Tetzlaff
Institute for Advanced Simulation (IAS-6)
Jülich Research Centre
Jülich, Germany
t.tetzlaff@fz-juelich.de
ORCID: 0000-0001-5794-5425

* equal contribution
† corresponding author

***Abstract*—Learning and processing sequential data constitutes a universal form of computation performed by the brain. Understanding the underlying principles does not only shed light on brain function, but also guides the development of energy efficient neuromorphic computing architectures. In a previous study, we devised a spiking recurrent neural network, the spiking temporal memory (spiking TM) model, implementing this type of computation. It learns sequences in an online, unsupervised manner by means of a local Hebbian synaptic plasticity mechanism. Context specific predictions of upcoming sequence elements are represented by dendritic action potentials. Upon successful learning, the network activity is characterized by a highly sparse and hence energy efficient code. To date, the sequence learning capabilities of the spiking TM model have only been demonstrated for small sequence sets. Here, we systematically investigate the sequence learning capacity of the model by gradually increasing the sequence length and optimizing the plasticity (hyper-) parameters. We show that the spiking TM model at the scale of a few thousand neurons can successfully learn random sequences composed of several tens of elements, with the maximum sequence length exceeding the vocabulary size. After optimizing the plasticity parameters for a given sequence length, the model exhibits high prediction performance for a range of sequence lengths, without additional fine tuning. The learning duration (time to solution) scales supralinearly with the sequence length. Learning longer sequences is hence computationally demanding, and requires accelerated computing architectures.**

***Index Terms*—spiking neuronal network, sequence learning capacity, online learning, plasticity, dendritic action potentials**

I. Introduction

One of the brain's most important abilities is to learn and process sequential data, whether in sensory perception, movement generation, or language comprehension and production. It relies on the capability to form context-dependent predictions, detect anomalies in streams of input data, and recall learned sequences of discrete items in response to a cue [1]. Developing models that implement these types of computation is an important step towards an understanding of brain function and towards more intelligent and energy efficient machines [2].

Several biologically plausible sequence learning models have been introduced, which bring us closer to this goal [3]–[6]. In particular, a recent model developed in [7] offers a mechanistic explanation of how biological systems may learn sequences. It builds on the Temporal Memory algorithm of

979-8-3315-8975-2/25/$31.00 ©2025 IEEE
DOI 10.1109/ICONS69015.2025.00038

the Hierarchical Temporal Memory (HTM) model [8], [9], and implements it using spiking neurons with continuous-time dynamics, without loss in performance. The model includes several biologically inspired mechanisms, such as nonlinear dendritic integration, Hebbian structural synaptic plasticity, and homeostatic control. Upon repeated exposure to sequences of stimuli, it forms sparse, context specific neuronal pathways that encode these sequences, predict upcoming sequence elements, signal anomalies, and replay learned sequences in response to a cue signal [10].

The capacity of a sequence learning model can be understood in terms of the complexity of the sequences it is able to represent. A key marker of sequence complexity is the recurrence of the same element in multiple, distinct temporal contexts. Thus, a meaningful measure of capacity is the network's ability to assign different internal representations to the same input depending on its preceding history. In a previous work in [9] on the Hierarchical Temporal Memory (HTM) model, capacity is defined as the number of distinct transitions between subsequent elements in a sequence the network can represent. This depends primarily on the network's sparsity and the number of input patterns each neuron can recognize through its dendritic segments [11]. For a network comprising approximately 65000 neurons and assuming a potential all-to-all connectivity, they estimate a theoretical capacity of around 320000 transitions. The capacity limit derived in this study is a theoretical limit based on combinatorical considerations, i.e., on the number of possible connectivity patterns that fulfill the requirements of the TM algorithm. This derivation does not account for the plasticity dynamics, i.e., whether all theoretically possible connectivity patterns can actually be formed by the plasticity dynamics.

In this work, we build on the spiking TM model [7] and explore its sequence learning capacity. To achieve successful learning of complex sequences, we revise the model's plasticity mechanisms. By systematically increasing the sequence length, we increase the likelihood of repeated elements, and hence, sequence complexity. We evaluate to what extent and how quickly sequences of different length can be learned. Together, these analyses help clarify the trade-offs between plasticity parameters, sequence structure, and learning efficiency in spiking networks.

II. Methods

A. Network model

In the following, we provide an overview of the network model. A comprehensive mathematical description of the individual model components, their dynamics, and parameters can be found at https://github.com/YounesBouhadjar/SpikingTemporalMemory.

The network is organized into $M = 26$ subpopulations (minicolumns), each consisting of $n_{\mathrm{E}} = 240$ excitatory neurons ($\mathcal{E}_k$, $k = 1, \ldots, M$) and a single inhibitory neuron ($\mathcal{I}_k$; Fig. 1). Within each subpopulation, all excitatory neurons share the same stimulus preference and reliably spike in response to

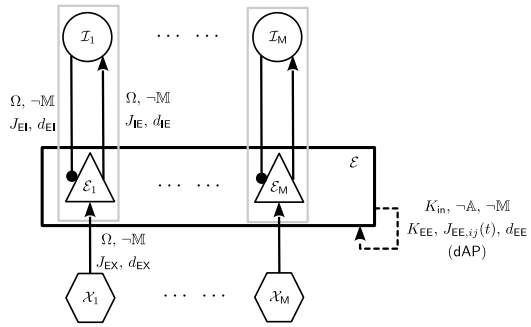

Fig. 1. **Sketch of the network architecture.** *Populations:* $\mathcal{E}_k$: excitatory populations. $\mathcal{I}_k$: inhibitory populations. $\mathcal{X}_k$: external inputs ($k = 1, \ldots, M$). *Connectivity rules:* Triangular arrow heads: excitatory connections. Circular arrow heads: inhibitory connections. Solid lines: static connections. Dashed lines: plastic connections. K_{in}: random connectivity with fixed indegree. Ω: all-to-all connectivity. $\neg\mathbb{M}$: multiple connections between two neurons not permitted. $\neg\mathbb{A}$: self-connections not permitted. Recurrent connections between excitatory neurons (dashed) target distal dendritic branches and may trigger dendritic action potentials (dAPs) signaling predictions. *Connectivity parameters:* synaptic weights J_{xy}, spike transmission delays d_{xy} ($x, y \in \{\mathrm{E}, \mathrm{I}, \mathrm{X}\}$). For details on the graphical notation, see [12]. A detailed mathematical description of the model is provided at https://github.com/YounesBouhadjar/SpikingTemporalMemory.

the presentation of the corresponding (suprathreshold) stimulus ($\mathcal{X}_k$). Excitatory neurons across the network are sparsely and randomly connected, forming recurrent excitatory circuits. Each inhibitory neuron receives inputs from its associated excitatory neurons and projects back to them, thereby implementing a local winner-take-all (WTA) competition that promotes sparse activation of subpopulations (see below).

All neurons are modeled as point neurons with leaky integrate-and-fire (LIF) dynamics. Connections between excitatory neurons target distal dendritic branches, where synaptic input currents are integrated in a nonlinear manner and generate dendritic action potentials (dAPs) if the total synaptic input current on the dendritic branch exceeds a critical value. These dAPs lead to a subthreshold plateau depolarization of the soma lasting for several tens of milliseconds, and thereby bring neurons into a more excitable state. In the TM model, these plateau depolizations serve as "predictions". Upon arrival of the corresponding stimulus, predictive neurons elicit a response spike with a shorter latency, as compared to non-predictive neurons. If the number of predictive neurons within a population exceeds the predefined pattern size ρ, the WTA mechanism is triggered, resulting in the selective activation of only the predicted neurons at the presentation of the respective stimulus. The resulting sparse subset of active neurons represents a specific sequence element in its current temporal context. Note that the "distal dendritic branches" are modeled as distinct nonlinear input channels in a point neuron model; we do not employ classical multicompartments models here.

Synapses between excitatory neurons are plastic, undergoing structural spike-timing-dependent changes alongside a

continuous synaptic leak. This form of structural plasticity modulates the maturity of a synapse in a Hebbian manner. The synaptic weight is zero, unless the maturity exeeds a critical value. Through repeated presentation of a sequence, the network effectively sculpts a sparse sequence-specific pathway across the excitatory subnetwork. In this study, we introduce several modifications to the plasticity mechanisms of the original model. First, we remove the homeostatic mechanism that controls the extent of synaptic potentiation based on the dendritic activity of the postsynaptic neuron. This homeostatic control was originally introduced to implement a competition between sequences with overlapping elements. The success of the mechanism, however, relied heavily on a tuning of the homeostatic set point (the dendritic target activity) to the specific task, reducing generality and biological plausibility. Second, we introduce a slow, activity independent decay of the synaptic maturity [13], [14]. This mechanism allows the network to prune connections that were formed during early learning but are no longer used, thereby freeing up synaptic resources for acquiring new sequences. In addition to supporting continual learning, this synaptic decay is consistent with experimental observations of synaptic turnover in biological neurons [15], [16]. Finally, we introduce a depression that depends on the relative timing of pre- and post-synaptic firing. In the original spiking TM model, the synaptic maturity was reduced by a fixed amount at each presynaptic spike. In the original model, outgoing connections of frequently active neurons, such as those representing common elements in a sequence, experienced disproportionately strong depression, thereby weakening their ability to predict subsequent sequence elements. The revised mechanism removes this imbalance by making synaptic depression context sensitive.

B. Training data and procedure

An input sequence of length C is constructed by randomly, independently and uniformly drawing C elements from a vocabulary of $A = 26$ unique tokens. As a result, the average token multiplicity C/A, i.e., the average number of repetitions of some sequence element, increases with sequence length C and decreases with vocabulary size A. Recurring tokens in complex sequences must be disambiguated based on the temporal context alone. The network must learn to represent identical input elements with distinct, sparse activation patterns within the same subpopulation, depending on the history of previously shown sequence elements. For a given vocabulary, longer sequences therefore correspond to a greater sequence complexity. Once generated, the sequence is presented to the network one element at a time with a fixed inter-stimulus interval ΔT. Each training epoch consists of a single sequence presentation, with a pause of ΔT_{seq} between successive epochs. Throughout training, the spiking activity of all neurons in the network is continuously recorded. To assess learning progress, the training process is periodically paused for performance evaluation. During this phase, the network is stimulated with the target sequence, and both spiking activity and dendritic currents are recorded. The latter serve as indicators of prediction within the network and are used for performance measurement (see Sec. II-C). Details on the sequence model, the stimulation procedure and related parameter values are provided at https://github.com/YounesBouhadjar/SpikingTemporalMemory.

C. Network performance estimation

To assess how well the network learns and predicts sequences, we follow a performance evaluation approach similar to that described in [7]. The prediction performance is evaluated based on dendritic action potentials (dAPs) recorded during sequence presentation. For each element in the input sequence, we monitor the dAP activity across all excitatory subpopulations $\mathcal{E}_k$ within the time interval between the presentation of the current and the previous sequence element. A subpopulation is considered predictive if at least $\rho/2$ of its neurons elicit a dAP, with ρ denoting the pattern size. The predictive state of the network for each presented element of the sequence is encoded as a binary vector $\boldsymbol{o} \in \{0,1\}^M$, where $o_k = 1$ if subpopulation k is predictive, and $o_k = 0$ otherwise. This vector is compared to a binary target vector $\boldsymbol{v}$, which encodes the identity of the subpopulation receiving the current input. The prediction error

$$\epsilon = \frac{1}{C}\sqrt{\sum_{k=1}^{M}(o_k - v_k)^2} \tag{1}$$

is defined as the Euclidean distance between $\boldsymbol{o}$ and $\boldsymbol{v}$, normalized by the sequence length C. To further analyze performance, we also monitor the false positive rate,

$$\text{fp} = \frac{1}{C}\sum_{k=1}^{M}\Theta\left(o_k - v_k\right), \tag{2}$$

and the false negative rate,

$$\text{fn} = \frac{1}{C}\sum_{k=1}^{M}\Theta\left(v_k - o_k\right). \tag{3}$$

Here, $\Theta(\cdot)$ denotes the Heaviside function. Predictive activity in subpopulations not targeted by the current input contributes to fp, while failure to predict the target subpopulation contributes to fn.

In addition to the prediction performance, we also evaluate the time to solution, which quantifies how quickly the network is able to learn a given sequence. A model is considered to have successfully learned the sequence once the prediction error rate ϵ drops below a predefined success threshold $\epsilon^* = 0.1$. The time to solution is reported as the number of required learning episodes n^*, as well as in form of the corresponding biological time τ^*, which accounts for the actual simulated duration of the stimulus presentation and is given by

$$\tau^* = n^*(C\Delta T + \Delta T_{\text{seq}}), \tag{4}$$

with inter-stimulus interval ΔT and inter-sequence interval ΔT_{seq} defining the pause between successive epochs.

In the predictive mode, the dAP triggered plateau potentials are subthreshold events and not directly visible to other neurons. In nature, they can therefore hardly serve as a readout of

the network's predictions. However, correct sequence element predictions can be decoded based on the context-specific sparse patterns of somatic action potentials. In this study, we extract the prediction performance directly from the dAPs to simplify the performane evaluation and thereby speed up the hyperparameter tuning.

D. Parameter optimization

To optimize the learning performance of the model for sequences of a given length C_{opt}, we adjust the plasticity parameters of the model, specifically the potentiation rate λ_+, the depression rate λ_-, and the synaptic decay time constant τ_{P}. To this end, we perform a grid search across the space spanned by these parameters and identify the parameter combination that minimizes the trial averaged prediction error ϵ. The rates λ_+ and λ_- are varied between 0.01 and 0.8, while τ_{P} is sampled from a range spanning $0.5\,\text{s}$ to $80\,\text{s}$. This results in a total of 1352 unique parameter combinations for each optimization run. For each parameter combination, simulations are repeated with three different random-number-generator seeds to account for the variability in network initialization and connectivity. Performances for the parameter optimization are then assessed by averaging the error over these three trials.

E. Simulation details

The network simulations are performed in NEST [17] version 3.6 [18]. The neuron and synapse models are formalized with the help of NESTML [19], [20] version 8.0.0 [21]. Network states are synchronously updated using exact integration of the system dynamics on a discrete time grid with step size $\Delta t = 0.1\,\text{ms}$ [22].

III. Results

A. Sequence learning dynamics

Fig. 2 illustrates the evolution of the prediction error, the false negative rate (fn), and the false positive rate (fp) over the course of training for two sequences of different length. At the beginning of the training, synapses between excitatory neurons of sequentially activated subpopulations start to form due to the structural Hebbian potentiation. This leads to a rapid decrease in the fn rate, indicating that the network begins to correctly predict upcoming elements. However, this improvement comes at the cost of an increased fp rate, as subpopulations not targeted by the current input also begin to exhibit predictive activity. This effect arises because the network has not yet developed the capacity to disambiguate recurring elements appearing in different temporal contexts. Consequently, fn and fp rates often display alternating behavior, reflecting an ongoing competition. The learning curves, particularly the fp rate, are therefore not strictly monotonic. This non-trivial evolution highlights the complexity of the learning process, where forming accurate, context-sensitive representations requires multiple phases of refinement.

The examples shown in Fig. 2 already indicate that the time to solution increases more than proportionally with sequence length. While the fn rate typically drops rapidly, the fp rate decays more slowly. A substantial portion of the training, especially for longer sequences, is spent to remove inappropriate dendritic activity across the network. As the training progresses, the model gradually carves out sparse and context-specific pathways, eventually minimizing both fn and fp.

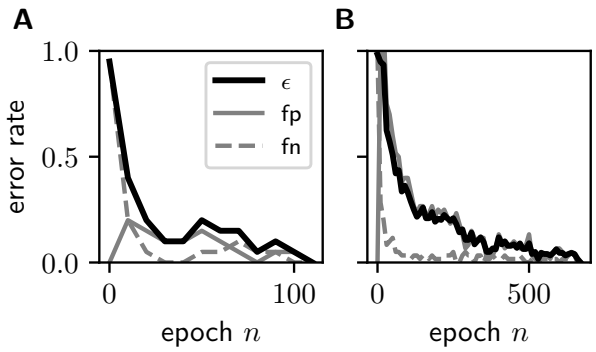

Fig. 2. **Sequence learning progress.** Dependence of the prediction error ϵ (solid black), the false positive rate fp (solid gray), and the false negative rate fn (dashed gray) on the number of training epochs n for two different sequence lengths $C = 20$ (A) and $C = 60$ (B). Plasticity parameters $\lambda_+ = 0.6$, $\lambda_- = 0.1$, $\tau_{\text{P}} = 80\,\text{s}$ are optimized for a sequence length $C_{\text{opt}} = 40$.

B. Sequence learning capacity

The results presented in Fig. 3 demonstrate that spiking TM networks comprising few thousand neurons can learn random sequences composed of several tens of elements. The network successfully learns sequences whose lengths exceed the vocabulary size, and can therefore cope with recurring tokens within the input sequences.

After tuning the plasticity parameters for a given sequence length C_{opt} ($C_{\text{opt}} = 20$ in Fig. 3A,C and $C_{\text{opt}} = 40$ in Fig. 3B,D), the network exhibits high prediction performance for a broad range of sequence lengths near this reference length. This finding aligns with the concept of "hyperparameter transfer" [23]. Such transferability alleviates the need for continuous fine-tuning when dealing with slightly varying sequence lengths. Sequences that are shorter than the reference sequence length C_{opt} are typically learned more efficiently and accurately, compared to sequences that are longer than C_{opt}. For instance, when the model parameters are optimized for a sequence length C_{opt}=20 (Fig. 3A), the network performs well for sequence lengths ranging approximately between 10 and 40. Beyond this range, particularly for significantly longer sequences, performance deteriorates. Moreover, the results highlight that increasing the reference sequence length C_{opt} shifts the network's optimal learning range towards larger sequence lengths (Fig. 3B).

The time to solution, i.e., the time required to reduce the prediction error below the success threshold $\epsilon^* = 0.1$, increases with the sequence length C in a supralinear manner (Fig. 3C,D). For our choice of network parameters, learning a sequence of length $C = C_{\text{opt}} = 40$, for example, requires $n^* \approx 130$ presentations of the sequence, corresponding to a

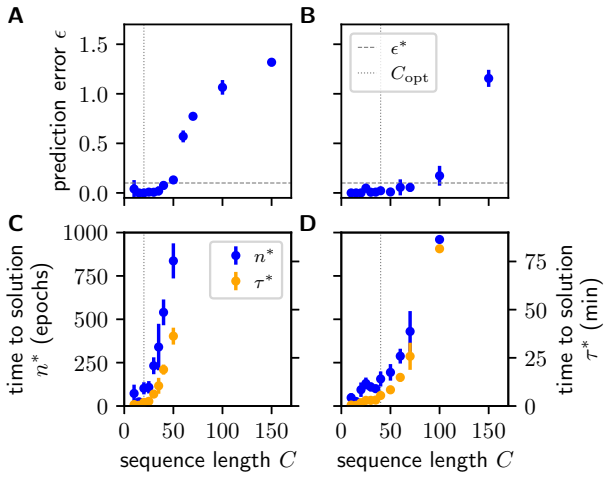

Fig. 3. **Prediction performance and learning speed.** Dependence of the prediction error ϵ (A, B) and the time to solution n^* (number of epochs; blue) and τ^* (total learning duration in seconds; yellow; C, D) on the sequence length C. Plasticity parameters are optimized for sequence lengths $C_{\text{opt}} = 20$ (A, C; $\lambda_+ = 0.6$, $\lambda_- = 0.8$, $\tau_{\text{P}} = 50\,\text{s}$) and $C_{\text{opt}} = 40$ (B, D; $\lambda_+ = 0.6$, $\lambda_- = 0.1$, $\tau_{\text{P}} = 80\,\text{s}$), respectively (vertical dotted lines). Symbols and error bars represent mean and standard deviation across an ensemble of 5 different network realizations. In C and D, only trials with prediction error smaller than $\epsilon^* = 0.1$ (dashed horizontal lines in A, B) are used.

duration of $\tau^* \approx 5\,\text{min}$. Increasing the sequence length to $C = 100$ results in a time to solution of $n^* \approx 940$, i.e., $\tau^* \approx 80\,\text{min}$ (Fig. 3D). From a biological perspective, these numbers are certainly not unrealistically large. However, they illustrate the practical challenge of systematically investigating biological learning by means of simulation, and emphasize the need for accelerated computing architectures [24]–[27].

IV. Discussion

The present study shows that spiking neuronal networks representing local cortical microcircuits at the submillimeter scale can successfully learn complex sequences in an unsupervised, online manner based on local synaptic plasticity mechanisms. The maximum lengths of the sequences that are successfully learned by the proposed model reach behaviorally relevant numbers (such as the number of notes in a melody, or the number of words in a poem). Clearly, these numbers need to be set in relation to network parameters such as the total network size, or here, the total number of excitatory neurons N_{E}, and the number of neurons representing a sequence element, the pattern size ρ. In the present study, these parameters are fixed ($N_{\text{E}} = Mn_{\text{E}} = 6240$, $\rho = 20$). With larger networks or smaller patterns, the network capacity certainly increases. We dedicate a systematic investigation of the network capacity and its dependence on network parameters to future studies.

Previous work suggests that the capacity of the TM model should be substantially larger than what is observed here [9], [11]. In these studies, however, the theoretical capacity limit is derived purely based on combinatorial constraints, that is, on the number of appropriate connectivity patterns that could be realized given a set of network parameters without generating errors. These considerations do not take into account whether some learning scheme or plasticity mechanism can actually form such connectivity patterns. Given the results of our study, we conclude that the plasticity dynamics constitutes a much more severe constraint on the sequence learning capacity than the network or the pattern size. This is highlighted by our observation that the time to solution increases with the sequence length in a strong, supralinear manner. With the given model and its specific type of plasticity, learning longer sequences may potentially be possible, but becomes very slow. Both from a Neuroscience and a Machine Learning perspective, it is hence desirable to develop fast, accelerated computing architectures which permit the learning of long sequences [24]–[28]. It remains an open question to what extent different plasticity mechanisms or network architectures could speed up the learning process for longer sequences in spiking neuronal networks.

The plasticity mechanism employed in this study is composed of a spike-timing dependent contribution and a slow, activity independent decay of the synaptic maturity, which can easily be implemented in neuromorphic hardware [27], [29]–[31]. The decay in synaptic maturity prunes connections that are formed during early learning but are no longer needed after learning. This mechanism frees synaptic resources and is instrumental in achieving good learning performance. As an unwanted side effect, however, it also slowly prunes functional connections that have emerged after successful learning, unless the network continues to be exposed to the learned sequences. In future versions of the model, this "forgetting" could be avoided by equipping the plasticity dynamics with a consolidation process that prevents the removal of strong, learned connections, see [32] and [33] and references therein.

This study is based on randomly generated sequences where tokens are independently drawn with identical probability. With this setup, the sequence complexity can systematically be varied by changing the sequence length. Natural sequences exhibit more structure: tokens occur with different probabilities, and consecutive elements are often correlated. This structure in real-world sequences could be exploited by a hierarchical network, where each level comprises a spiking TM model that learns and processes sequences of sequences represented at lower levels. We dedicate a more detailed investigation of the sequence learning capacity of the spiking TM model for real-world data to future studies.

Acknowledgment

This project was supported by the Federal Ministry of Education and Research (NeuroSys grant no. 03ZU1106CB, 03ZU2106CB; NEUROTEC-II grant no. 16ME0398K, 16ME0399), the German Research Foundation grant no. 368482240/GRK2416 and the Joint Lab "Supercomputing and Modeling for the Human Brain" (SMHB).

References

[1] S. Dehaene, F. Meyniel, C. Wacongne, L. Wang, and C. Pallier, "The neural representation of sequences: from transition probabilities to

algebraic patterns and linguistic trees," *Neuron*, vol. 88, no. 1, pp. 2–19, 2015.

[2] J. Hawkins and S. Blakeslee, *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan, 2007.

[3] S. Klampfl and W. Maass, "Emergence of dynamic memory traces in cortical microcircuit models through stdp," *J. Neurosci.*, vol. 33, no. 28, pp. 11 515–11 529, 2013. [Online]. Available: https://doi.org/10.1523/jneurosci.5044-12.2013

[4] C. Klos, D. Miner, and J. Triesch, "Bridging structure and function: A model of sequence learning and prediction in primary visual cortex," *PLOS Comput. Biol.*, vol. 14, no. 6, p. e1006187, 2018. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1006187

[5] T. Asabuki, P. Kokate, and T. Fukai, "Neural circuit mechanisms of hierarchical sequence learning tested on large-scale recording data," *PLOS Comput. Biol.*, vol. 18, no. 6, p. e1010214, 2022.

[6] L. Kriener, K. Völk, B. von Hünerbein, F. Benitez, W. Senn, and M. A. Petrovici, "Elise: Efficient learning of sequences in structured recurrent networks," *arXiv*, p. 2402.16763, 2024.

[7] Y. Bouhadjar, D. J. Wouters, M. Diesmann, and T. Tetzlaff, "Sequence learning, prediction, and replay in networks of spiking neurons," *PLOS Comput. Biol.*, vol. 18, no. 6, p. e1010233, 2022. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1010233

[8] J. Hawkins, S. Ahmad, D. Dubinsky *et al.*, "Cortical learning algorithm and Hierarchical Temporal Memory," *Numenta Whitepaper*, vol. 1, no. 68, p. 2, 2011.

[9] J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Front. Neural Circuits*, vol. 10, p. 23, 2016. [Online]. Available: https://doi.org/10.3389/fncir.2016.00023

[10] Y. Bouhadjar, D. J. Wouters, M. Diesmann, and T. Tetzlaff, "Coherent noise enables probabilistic sequence replay in spiking neuronal networks," *PLOS Comput. Biol.*, vol. 19, no. 5, p. e1010989, 2023. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1010989

[11] S. Ahmad and J. Hawkins, "How do neurons operate on sparse distributed representations? a mathematical theory of sparsity, neurons and active dendrites," *ArXiv*, p. 1601.00720, 2016. [Online]. Available: https://arxiv.org/abs/1601.00720

[12] J. Senk, B. Kriener, M. Djurfeldt, N. Voges, H.-J. Jiang, L. Schüttler, G. Gramelsberger, M. Diesmann, H. E. Plesser, and S. J. van Albada, "Connectivity concepts in neuronal network modeling," *PLOS Comput. Biol.*, vol. 18, no. 9, p. e1010086, 2022.

[13] S. Akaho, "Capacity and error correction ability of sparsely encoded associative memory with forgetting process," in *ICANN'93: Proceedings of the International Conference on Artificial Neural Networks Amsterdam, The Netherlands 13–16 September 1993 3*. Springer, 1993, pp. 707–710.

[14] D. G. Clark and L. Abbott, "Theory of coupled neuronal-synaptic dynamics," *Phys. Rev. X*, vol. 14, no. 2, p. 021001, 2024.

[15] A. Holtmaat, L. Wilbrecht, G. W. Knott, E. Welker, and K. Svoboda, "Experience-dependent and cell-type-specific spine growth in the neocortex," *Nature*, vol. 441, pp. 979–983, June 2006, doi:10.1038/nature04783.

[16] N. Malakasis, S. Chavlis, and P. Poirazi, "Synaptic turnover promotes efficient learning in bio-realistic spiking neural networks," in *2023 57th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2023, pp. 942–949.

[17] M.-O. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia J.*, vol. 2, no. 4, p. 1430, 2007. [Online]. Available: https://doi.org/10.4249/scholarpedia.1430

[18] J. Villamar, J. Vogelsang, C. Linssen, S. Kunkel, A. Kurth, C. M. Schöfmann, M. A. Benelhedi, P. Nagendra Babu, J. M. Eppler, R. de Schepper, J. Mitchell, A. Morrison, N. Haug, S. Diaz, J. Acimovic, S. Graber, H.-J. Jiang, D. Terhorst, S. Spreizer, J.-E. Welle Skaar, J. Stapmanns, T. Manninen, M. Krüger, M. Lehtimäki, S. Ito, A. Y. Lee, M. Lindahl, and H. E. Plesser, "NEST 3.6," 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8344932

[19] D. Plotnikov, I. Blundell, T. Ippen, J. M. Eppler, B. Rumpe, and A. Morrison, "NESTML: a modeling language for spiking neurons," in *Modellierung 2016*, ser. Lecture Notes in Informatics (LNI), A. Oberweis and R. Reussner, Eds., vol. P-254, Modellierung 2016, Karlsruhe (Germany), 17 Mar 2016 - 19 Mar 2016. Gesellschaft für Informatik e.V. (GI), Mar 2016, pp. 93–108. [Online]. Available: http://juser.fz-juelich.de/record/826510

[20] C. Linssen, P. N. Babu, J. M. Eppler, L. Koll, B. Rumpe, and A. Morrison, "NESTML: a generic modeling language and code generation tool for the simulation of spiking neural networks with advanced plasticity rules," *Front. Neuroinform.*, vol. 19, 2025.

[21] C. Linssen, P. N. Babu, Y. Bouhadjar, L. Ewert, W. Wybo, M. Lober, F. Feller, B. Rumpe, and A. Morrison, "NESTML 8.0.0," 2024. [Online]. Available: https://doi.org/10.5281/zenodo.12191059

[22] S. Rotter and M. Diesmann, "Exact digital simulation of time-invariant linear systems with applications to neuronal modeling," *Biol. Cybern.*, vol. 81, no. 5-6, pp. 381–402, 1999. [Online]. Available: https://doi.org/10.1007/s004220050570

[23] G. Yang, E. J. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao, "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer," *ArXiv*, 2022.

[24] K. Kauth, T. Stadtmann, R. Brandhofer, V. Sobhani, and T. Gemmeke, "Communication architecture enabling 100x accelerated simulation of biological neural networks," in *Proceedings of the Workshop on System-Level Interconnect: Problems and Pathfinding Workshop*, 2020, pp. 1–8.

[25] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, p. 795876, 2022.

[26] K. Kauth, T. Stadtmann, V. Sobhani, and T. Gemmeke, "neuroaix-framework: design of future neuroscience simulation systems exhibiting execution of the cortical microcircuit model 20× faster than biological real-time," *Front. Comput. Neurosci.*, vol. 17, Apr. 2023. [Online]. Available: https://doi.org/10.3389/fncom.2023.1144143

[27] R. Dietrich, P. Spilger, E. Müller, J. Schemmel, and A. C. Knoll, *Sequence Learning with Analog Neuromorphic Multi-Compartment Neurons and On-Chip Structural STDP*. Springer Nature Switzerland, 2025, p. 207–230. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-82487-6_15

[28] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018. [Online]. Available: https://doi.org/10.1109/mm.2018.112130359

[29] Y. Bouhadjar, S. Siegel, T. Tetzlaff, M. Diesmann, R. Waser, and D. J. Wouters, "Sequence learning in a spiking neuronal network with memristive synapses," *Neuromorphic Comput. Eng.*, vol. 3, no. 3, p. 034014, 2023.

[30] S. Siegel, Y. Bouhadjar, T. Tetzlaff, R. Waser, R. Dittmann, and D. J. Wouters, "System model of neuromorphic sequence learning on a memristive crossbar array," *Neuromorphic Comput. Eng.*, vol. 3, no. 2, p. 024002, 2023. [Online]. Available: https://dx.doi.org/10.1088/2634-4386/acca45

[31] S. Siegel, T. Ziegler, Y. Bouhadjar, T. Tetzlaff, R. Waser, R. Dittmann, and D. Wouters, "Demonstration of neuromorphic sequence learning on a memristive array," in *Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference*, 2023, pp. 108–114.

[32] F. Zenke, E. J. Agnes, and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks," *Nature communications*, vol. 6, no. 1, pp. 1–13, 2015.

[33] A. Laborieux, M. Ernoult, T. Hirtzlin, and D. Querlioz, "Synaptic metaplasticity in binarized neural networks," *Nature Communications*, vol. 12, no. 1, 2021. [Online]. Available: http://dx.doi.org/10.1038/s41467-021-22768-y